INTERNATIONAL STANDARD

ISO 10303-219

First edition
2007-12-15

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

Part 219: Application protocol: Dimensional inspection information exchange

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

Partie 219: Protocole d'application: Échange d'information par vérification dimensionnelle

Reference number
ISO 10303-219:2007(E)

PDF disclaimer

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

Adobe is a trademark of Adobe Systems Incorporated.

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

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Figures

Tables

Foreword

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

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

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

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

ISO 10303-219 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

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

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

A complete list of parts of ISO 10303 is available from the Internet:
<http://www.tc184-sc4.org/titles/>

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation 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.

This part of ISO 10303 is a member of the application protocol series. This part of ISO 10303 specifies an application protocol (AP) for the exchange of information resulting from the dimensional inspection of solid parts. Industry is faced with numerous proprietary inspection languages and interfaces. This causes overhead problems associated with maintaining multiple systems or locks users to one vendor. In particular, there are no adequate standard systems for linking coordinate measurement machines and other types of automated inspection systems with systems that analyze and track dimensional inspection results. During the course of four workshops held on dimensional inspection information exchange, industry representatives indicated that a standard or a specification for dimensional inspection information exchange is critical to their future operations. The focus of the proposed standard is the analysis and reporting activity for dimensional inspection. The measurement process itself is not within the scope of this part of ISO 10303. The primary benefit will be a link between dimensional inspection programs, provided by ISO 22093 (DMIS 4.0), Web-based analysis and reporting practices, aimed for in the Metrology Interoperability Project (MIP), and standard information models for maufacturing provided for example, by ISO 10303-224 and ISO 10303-238. The information provided by the DMIS and the MIP will be mappable into entities of this part of ISO 10303 and transportable into other ISO 10303 based implementations.

In addition this part of ISO 10303 captures the digital representation of dimensioning and tolerancing information requirements in standards ISO 1101, and ISO 5459 developed by ISO TC 213 on Geometrical Product Specifications and Verification.

This application protocol defines the context, scope, and information requirements for analyzing and reporting the information and results of dimensional inspection and specifies the integrated resources necessary to satisfy these requirements.

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

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

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

Figure 1 contains the data planning model that provides a high level description of the requirements for this application protocol. 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).

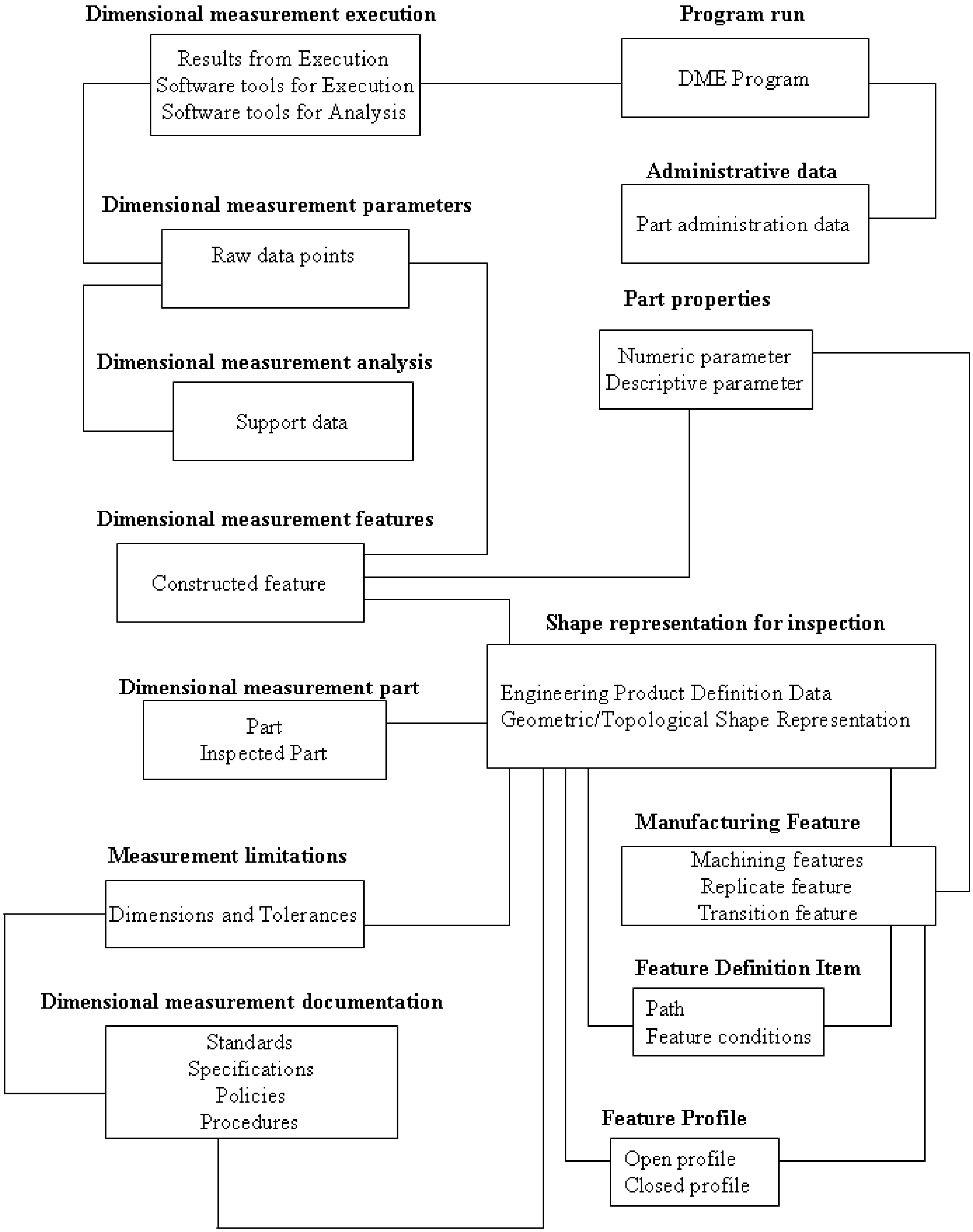


Figure 1 — Data planning model

Industrial automation systems and integration — Product data representation and exchange — Part 219: Application protocol: Dimensional inspection information exchange

1 Scope

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for analyzing the data and reporting the results of dimensional inspections of solid parts or assemblies. Dimensional inspection can occur at any stage of the life cycle of a product where checking for conformance with a design specification is required.

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

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

— data for administering, planning;

— data for executing dimensional inspection;

— data for archiving the results of a dimensional inspection;

— interface for capturing technical data out of the upstream application protocols;

— machining feature classification structure;

— geometric and dimensional tolerances of the parts being manufactured;

— references to standards and specifications declared in the dimensional inspection.

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

— dimensional inspection of liquid surfaces;

— materials properties of parts;

— manufacturing activities;

— mathematical algorithms to perform the dimensional inspection analysis;

— developing or modifying manufacturing process information;

— generating geometry (creating the CAD model);

— generating tolerance requirements;

— inspection of material properties.

2 Normative references

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

— dimension;

— tolerance.

3.2 Terms defined in ISO 5459

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

— datum.

3.3 Terms defined in ISO 10303-1

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

— application;

— application activity model (AAM);

— application interpreted model (AIM);

— application object;

— application protocol (AP);

— application reference model (ARM);

— implementation method;

— integrated resource;

— model;

— product;

— product data;

— protocol implementation conformance statement (PICS);

— unit of functionality (UoF).

3.4 Terms defined in ISO 10303-42

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

— boundary representation (Brep) solid model;

— manifold solid boundary representation.

3.5 Other terms and definitions

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

3.5.1
dimensional measurement equipment
DME
equipment used for dimensional measurement, including measuring machines and sensors, but also software for calculating dimensional features and parameters

3.6 Abbreviations

For the purposes of this document, the following abbreviations apply:

AAM	application activity model
AIC	application interpreted construct
AIM	application interpreted model
AP	application protocol

ARM	application reference model
Brep	boundary representation
DME	dimensional markup equipment
DML	dimensional markup language
ICOM	input, control, output, mechanism
Ngon	polygon with number of sides equal to N
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 dimensional inspection information exchange.

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 specification specified in 5.1 shows how the information requirements are met using the integrated resources and application interpreted constructs of this part of ISO 10303. 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 the dimensional inspection information exchange application protocol. This part of ISO 10303 specifies the following units of functionality:

— administrative_data;

— dimensional_measurement_analysis;

— dimensional_measurement_documentation;

— dimensional_measurement_execution;

— dimensional_measurement_feature;

— dimensional_measurement_part;

— dimensional_measurement_parameters;

— feature_definition_item;

— feature_profile;

— manufacturing_feature;

— functional_limitations;

— part_properties;

— program_run;

— 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 administrative_data

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

The following application objects are used by the administration_data UoF:

— Address;

— Calendar_date;

— Date_time;

— Organization;

— Person_and_organization;

— Time_offset.

4.1.2 dimensional_measurement_analysis

The dimensional_measurement_analysis UoF UoF defines a collection of possible analysis choices for calculating tolerance parameters and feature parameters from measured data.

The following application objects are used by the dimensional_measurement_analysis UoF:

— Dm_analysis_dofs_dml;

— Dof_attribute_dml;

— Dm_feature_analysis_mode_dml;

— Dm_tolerance_analysis_mode_dml;

— Dm_tolerance_analysis_mode_default_dml;

— Dm_feature_analysis_mode_default_dml;

— Dm_parameter_analysis_dml.

4.1.3 dimensional_measurement_documentation

The dimensional_measurement_documentation 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 to 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 dimensional_measurement_documentation UoF:

— Document_assignment;

— Part_dimensioning_standard;

— Specification;

— Specification_usage_constraint.

4.1.4 dimensional_measurement_execution

The dimensional_measurement_execution UoF contains the run information and initial data points resulting from a dimensional inspection operation on a part, perhaps after point correction. Data analysis of these points is then performed to determine parameters and compare the parameters to tolerances.

The following application objects are used by the dimensional_measurement_execution UoF:

— Dm_execution_input;

— Dm_execution_result;

— Dm_execution_result_measurement;

— Dm_data_analysis_software.

4.1.5 dimensional_measurement_features

The dimensional_measurement_feature UoF contains the information necessary to identify shapes, which represent volumes of material that shall be inspected of a part by dimensional inspection.

The following application objects are used by the dimensional_measurement_feature UoF:

— Dm_feature;

— Dmf_arc;

— Dmf_circle;

— Dmf_cone;

— Dmf_cylinder;

— Dmf_edge_point;

— Dmf_ellipse;

— Dmf_generic_feature;

— Dmf_geometric_curve;

— Dmf_geometric_surface;

— Dmf_line_bounded;

— Dmf_line_closed_parallel;

— Dmf_line_unbounded;

— Dmf_pattern;

— Dmf_plane;

— Dmf_plane_closed_parallel;

— Dmf_plane_symmetric;

— Dmf_point;

— Dmf_sphere;

— Dmf_surface_of_revolution_dml;

— Dmf_torus;

— Feature;

— Inspection_feature_relationship.

4.1.6 dimensional_measurement_part

The dimensional_measurement_part UoF contains the information necessary to identify the part that is to be input to the dimensional inspection function and identify the association of properties with that part.

The following application object is used by the Dimensional_measurement_part UoF:

— Part.

4.1.7 dimensional_measurement_parameters

The dimensional_measurement_parameters UoF contains the information necessary to record inspection results.

The following application object is used by the Dimensional_measurement_parameters UoF:

— Dm_dimension_parameter;

— Dm_result_parameter;

— Dm_parameter_value_limits;

— Dm_point;

— Dm_point_parameter;

— Dm_vector_parameter.

4.1.8 feature_definition_items

The feature_definition_items 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_items 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;

— 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;

— Through_bottom_condition;

— Through_pocket_bottom_condition;

— Through_profile_floor;

— Woodruff_slot_end_type.

4.1.9 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.

4.1.10 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.

4.1.11 functional_limitations

The functional_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 functional_limitations UoF:

— Angular_dimension_tolerance;

— Angular_size_dimension_tolerance;

— Angularity_tolerance;

— Circular_runout_tolerance;

— Circularity_tolerance;

— Common_datum;

— Concentricity_tolerance;

— Coordinate_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.

4.1.12 part_properties

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

The following application objects are used by the Part_properties UoF:

— Descriptive_parameter;

— Numeric_parameter;

— Numeric_parameter_with_tolerance;

— Property_parameter.

4.1.13 program_run

The program_run UoF specifies the program used by the measuring machine to produce the dimensional measurements data.

The following application objects are used by the program_run UoF:

— Dm_program_identification;

— Dm_program_run;

— Measurement_location;

— Run_administrator.

4.1.14 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;

— Brep_model;

— Brep_model_element;

— Brep_shape_aspect_representation;

— Brep_shape_representation;

— Cartesian_coordinate_space;

— Cartesian_point;

— Cartesian_vector;

— Cylindrical_base_shape;

— Derived_shape_element

— Direction_element;

— Explicit_base_shape_representation;

— Face_shape_element;

— Face_shape_element_relationship;

— Geometric_model;

— Implicit_base_shape_representation;

— Location_element;

— Ngon_base_shape;

— Offset_shape_element;

— Orientation;

— Path_element;

— Planar_element;

— Shape;

— Shape_aspect;

— Shape_element.

4.2 Application objects

This subclause specifies the application objects for the dimensional inspection information exchange 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 Address

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

— country;

— electronic_mail_address;

— facsimile_number;

— internal_location;

— postal_box;

— postal_code;

— region;

— street_number;

— street;

— telephone_number;

— town.

4.2.1.1 country

The country specifies the name of a country.

4.2.1.2 electronic_mail_address

The electronic_mail_address specifies the electronic address at which electronic mail may be received.

4.2.1.3 facsimile_number

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

4.2.1.4 internal_location

The internal_location specifies an organization-defined address for internal mail delivery.

4.2.1.5 postal_box

The postal_box specifies the number of a postal box.

4.2.1.6 postal_code

Ther postal_code specifies the code that is used by the country's postal service.

4.2.1.7 region

The region specifies the name of a region.

EXAMPLE The counties of Great Britain and the states of the United States of America are examples of regions.

4.2.1.8 street_number

The street_number specifies the number of a location on a street.

4.2.1.9 street

The street specifies the name of a street.

4.2.1.10 telephone_number

The telephone_number specifies the number at which telephone calls may be received.

4.2.1.11 telex_number

The telex_number specifies the number at which telex messages may be received.

4.2.1.12 town

The town specifies the name of a town.

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 1 This application object is harmonized with the same entity from paragraph 4.2.2 in ISO 10303-224:2006.

NOTE 2 Figure 2 illustrates features that have an Angle_taper applied.

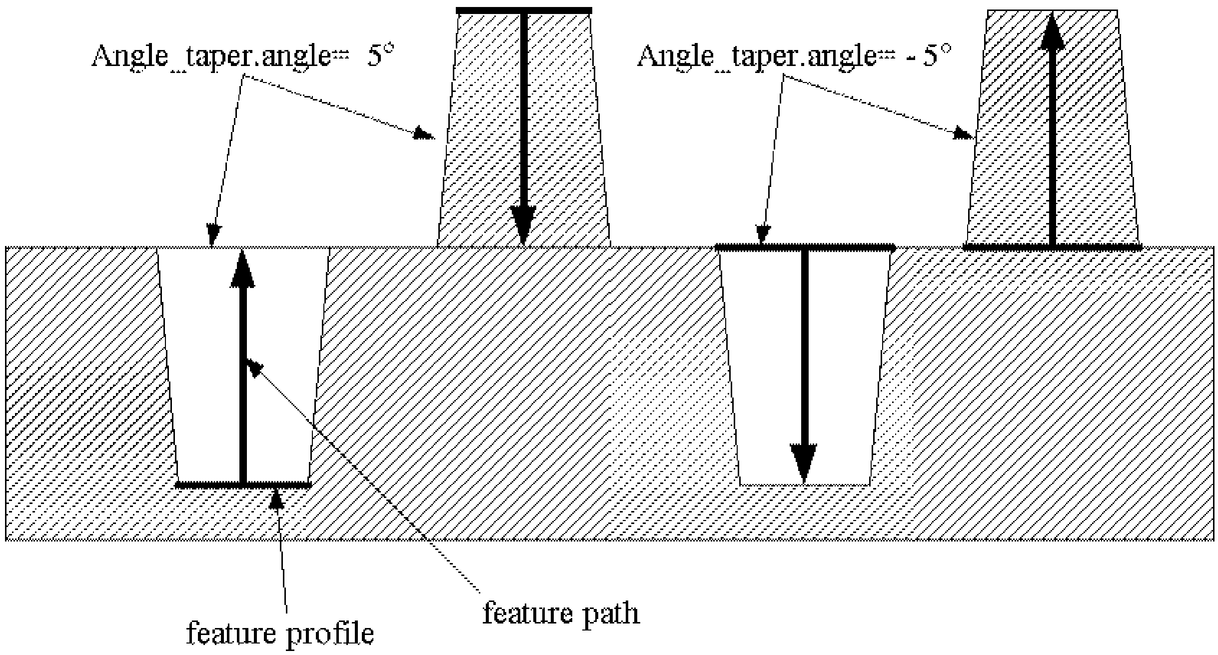


Figure 2 — Angle_taper

The data associated with an Angle_taper are the following:

— angle.

4.2.2.1 angle

The angle specifies the amount of slope from the start of an Angle_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 4.3.1 for the application assertion.

4.2.3 Angular_dimension_tolerance

An Angular_dimension_tolerance is a type of Location_tolerance (see 4.2.160) that defines the allowable variation in the angle between two elements of the shape of a part. Each Angular_dimension_tolerance shall have an origin shape and a termination shape.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.3 in ISO 10303-224:2006.

NOTE 2 Figure 3 illustrates the Angular_dimension_tolerance.

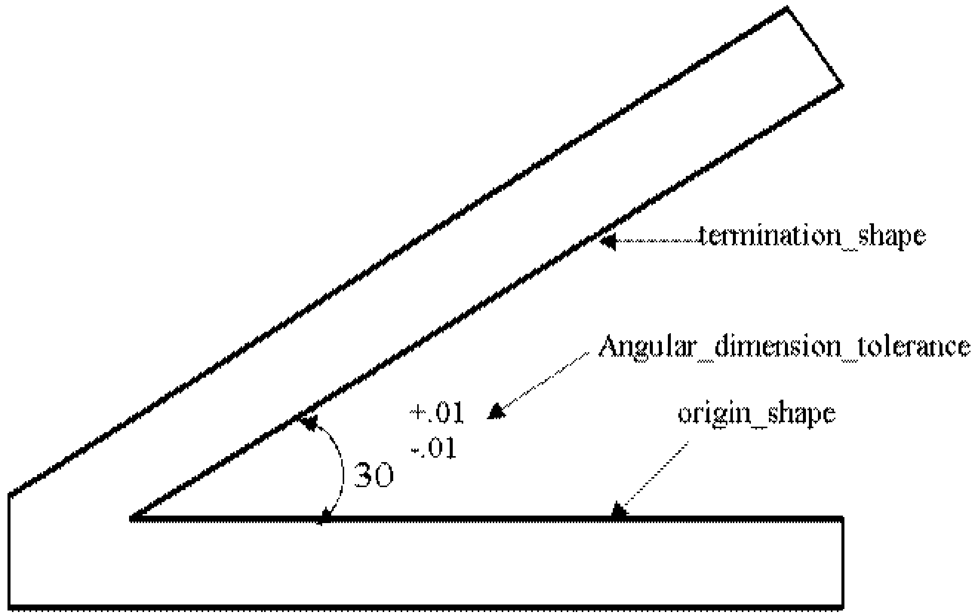


Figure 3 — Angular_dimension_tolerance

The data associated with an Angular_dimension_tolerance are the following:

— feature_type;

— plane_and_direction.

4.2.3.1 feature_type

The feature_type specifies the type of feature to be used as the origin, the feature type shall be either actual, datum, or nominal. The feature_type need not be specified for a particular Angular_dimension_-tolerance.

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

— actual;

— datum;

—nominal.

NOTE See 4.2.3.1.1 to 4.2.3.1.3 for the definition of each allowable value for dimension_note.

4.2.3.1.1 actual: tolerance is for actual value.

4.2.3.1.2 datum: tolerance is for a datum value.

4.2.3.1.3 nominal: tolerance is for a nominal value.

4.2.3.2 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 4.3.2 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 4.2.240) that specifies the allowable variation on the size or gap formed by two angular elements of the shape of a part.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.4 in ISO 10303-224:2006.

NOTE 2 Figure 4 illustrates the Angular_size_dimension_tolerances.

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.

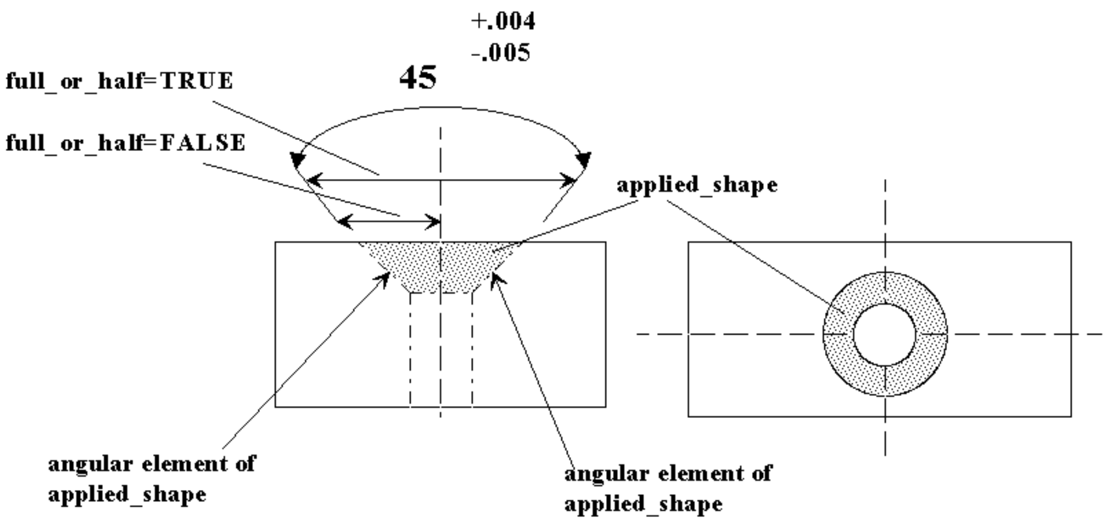


Figure 4 — Angular_size_dimension_tolerance

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 4.2.143) 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 This application object is harmonized with the same entity from paragraph 4.2.5 in ISO 10303-224:2006.

NOTE 2 The Angularity_tolerance definition is derived from clause ISO 1101, 14.9.

NOTE 3 Figure 5 illustrates the Angularity_tolerance for a plane surface.

NOTE 4 Figure 6 illustrates Angularity_tolerance for an axis.

The data associated with an Angularity_tolerance are the following:

— geometric_reference;

— segment_size.

4.2.5.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.3 for the application assertion.

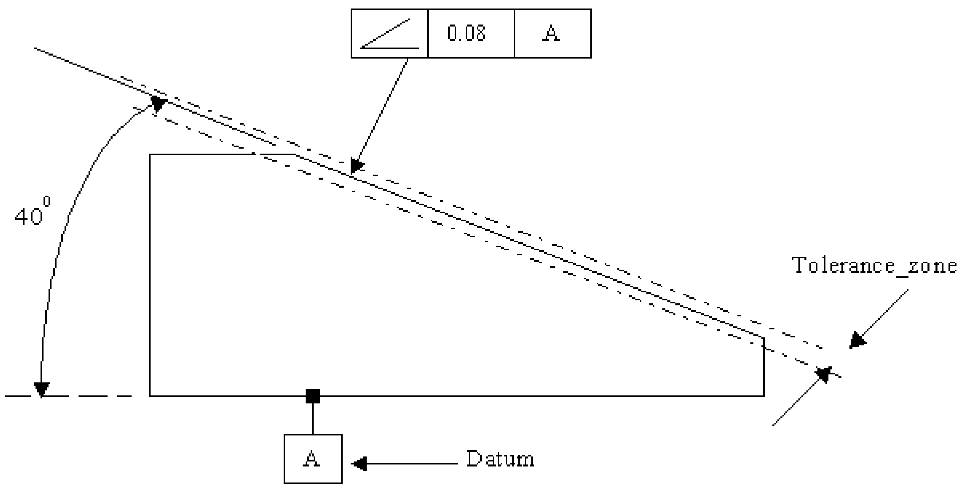


Figure 5 — Angularity_tolerance for a plane surface

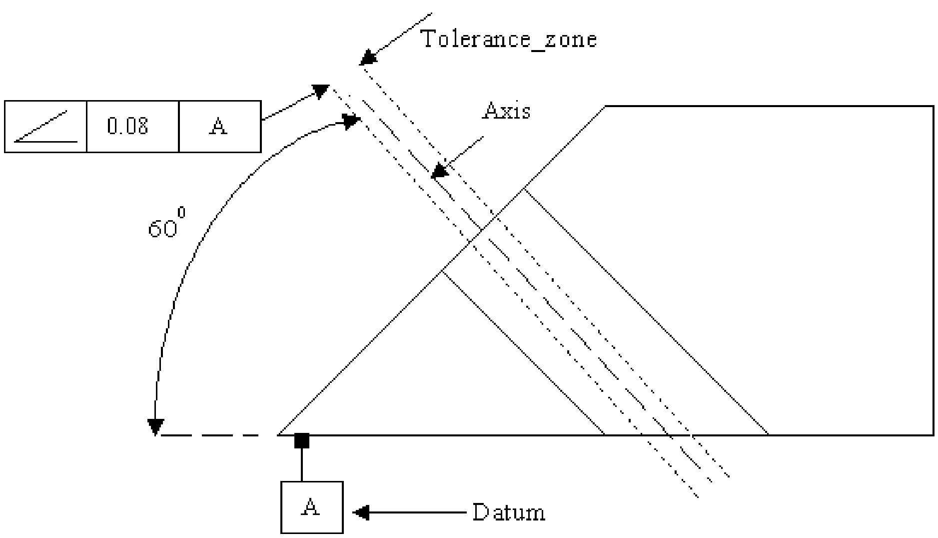


Figure 6 — Angularity_tolerance for an axis

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 Base_shape

A Base_shape is the initial shape of the material before machining of the features. Each Base_shape is either an Explicit_base_shape_representation (see 4.2.114) or an Implicit_base_shape_representation (see 4.2.150).

NOTE This application object is harmonized with the same entity from paragraph 4.2.7 in ISO 10303-224:2006.

4.2.7 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 4.2.147), or a Straight_bevel_gear (see 4.2.250). 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 1 This application object is harmonized with the same entity from paragraph 4.2.8 in ISO 10303-224:2006.

NOTE 2 Figure 7 illustrates the Bevel_gear. Figure 8 and Figure 9 illustrate attributes.

The data associated with a Bevel_gear are the following:

— root_angle;

— tip_angle.

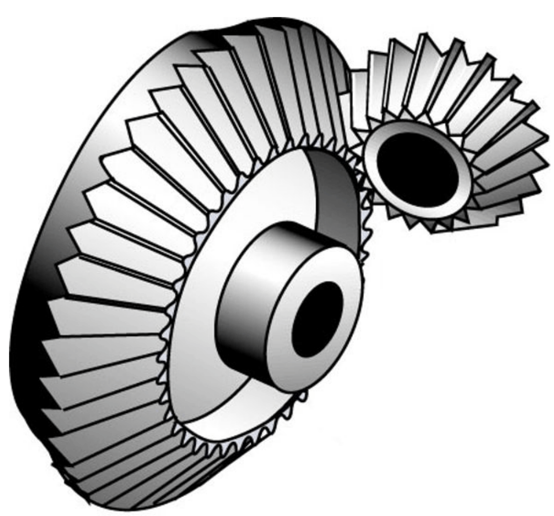


Figure 7 — Bevel_gear

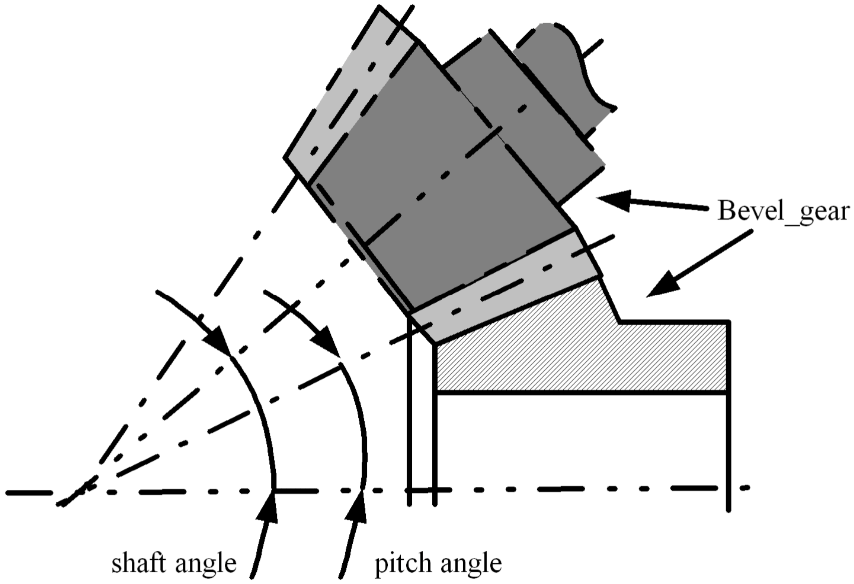


Figure 8 — Bevel_gear with shaft_angle and pitch_angle

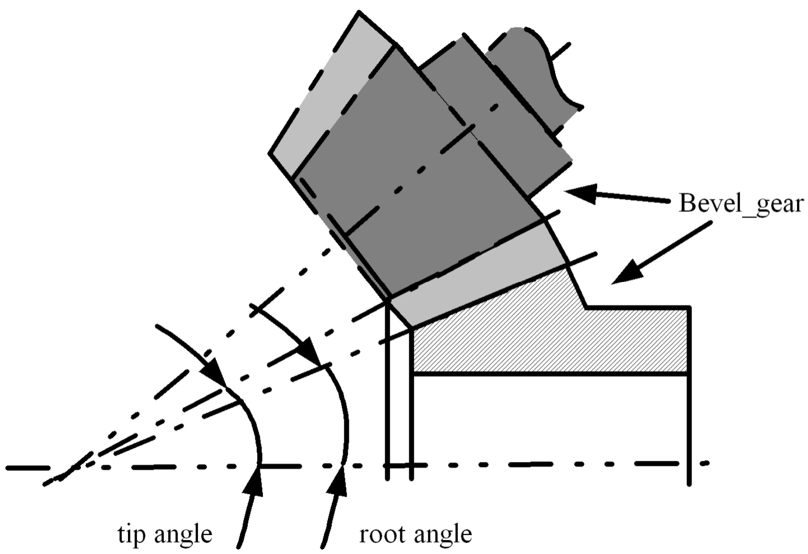


Figure 9 — Bevel_gear with tip_angle and root_angle

4.2.7.1 root_angle

The root_angle specifies the angle between the axis and the root of the teeth. See 4.3.4 for the application assertion.

4.2.7.2 tip_angle

The tip_ the angle specifies between the axis and the tip of the teeth. See 4.3.4 for the application assertion.

4.2.8 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 4.2.44), Flat_hole_bottom (see 4.2.121), Flat_with_radius_hole_bottom (see 4.2.123), Flat_with_taper_hole_bottom (see 4.2.124), or a Spherical_hole_bottom (see 4.2.246).

NOTE This application object is harmonized with the same entity from paragraph 4.2.9 in ISO 10303-224:2006.

The data associated with a Blind_bottom_condition are the following:

— start_or_end.

4.2.8.1 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.

4.2.9 Block_base_shape

A Block_base_shape is a type of Implicit_base_shape_representation (see 4.2.150) that describes the initial shape of the material as a rectangular cross section of some determined length.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.10 in ISO 10303-224:2006.

NOTE 2 Figure 10– illustrates a Block_base_shape.

The data associated with a Block_base_shape are the following:

— height;

— width.

4.2.9.1 height

The height specifies the size of the side of a Block_base_shape along the Y-axis. See 4.3.5 for the application assertion.

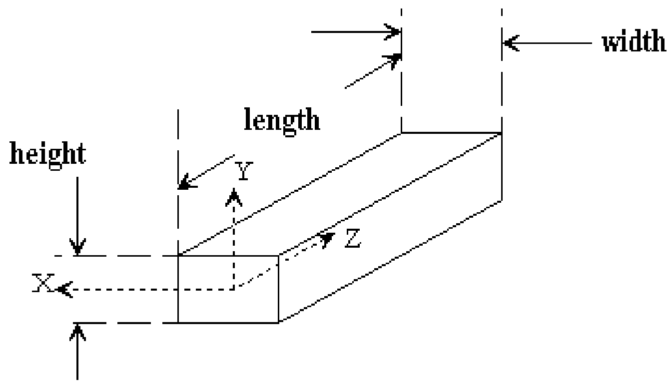


Figure 10 — Block_base_shape

4.2.9.2 width

The width specifies the size of the side of a Block_base_shape along the X-axis. See 4.3.5 for the application assertion.

4.2.10 Boss

A Boss is a type of Multi_axis_feature (see 4.2.169) 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 4.2.26), General_boss (see 4.2.127), or a Rectangular_-boss (see 4.2.214). 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.

NOTE This application object is harmonized with the same entity from paragraph 4.2.11 in ISO 10303-224:2006.

The data associated with a Boss are the following:

— boss_height;

— fillet_radius;

— top_condition.

4.2.10.1 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 4.3.7 for the application assertion.

4.2.10.2 fillet_radius

The fillet_radius specifies a radius shape blend between a Boss and the surrounding surface at the base of the Boss. See 4.3.8 for the application assertion.

4.2.10.3 top_condition

The top_condition specifies the shape of the top of a Boss feature. See 4.3.6 for the application assertion.

4.2.11 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 4.2.141) or a Planar_top_condition (see 4.2.200).

NOTE This application object is harmonized with the same entity from paragraph 4.2.12 in ISO 10303-224:2006.

The data associated with a Boss_top are the following:

— start_or_end;

— top_radius.

4.2.11.1 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.

4.2.11.2 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 4.3.8 for the application assertion.

NOTE It the Boss_top_condition was a General_top_condition that defined a spherical shape, the top_radius may not apply

4.2.12 Brep_model

A Brep_model is a type of Geometric_model (see 4.2.142) that is a solid model containing complete representation of shape using manifold solid boundary representation.

NOTE This application object is harmonized with the same entity from paragraph 4.2.13 in ISO 10303-224:2006.

4.2.13 Brep_model_element

A Brep_model_element is a portion of a boundary representation.

NOTE This application object is harmonized with the same entity from paragraph 4.2.14 in ISO 10303-224:2006.

The data associated with a Brep_-model_element are the following:

— element.

4.2.13.1 element

The element specifies the portion of the Brep_model that defines the Brep_model_element. See 4.3.10 for the application assertion.

4.2.14 Brep_shape_aspect_representation

A Brep_shape_aspect_representation is a grouping of geometric elements with respect to the boundary representation of a part.

NOTE This application object is harmonized with the same entity from paragraph 4.2.15 in ISO 10303-224:2006.

The data associated with a Brep_shape_aspect_representation are the following:

— shape_definition.

4.2.14.1 shape_definition

The shape_definition specifies the shape that is the boundary representation. See 4.3.11 for the application assertion.

4.2.15 Brep_shape_representation

A Brep_shape_representation is a grouping of boundary representations to define a shape.

NOTE This application object is harmonized with the same entity from paragraph 4.2.16 in ISO 10303-224:2006.

The data associated with a Brep_shape_representation are the following:

— shape_definition.

4.2.15.1 shape_definition

The shape_definition specifies the shape that is the boundary representation. See 4.3.12 for the application assertion.

4.2.16 Calendar_date

A calendar_date is a definition of a date as a day in a month of a year. The data associated with a Calendar_date are the following:

— day;

— month;

— year.

4.2.16.1 day

The day specifies the day element of a date

4.2.16.2 month

The month specifies the month element of a date

4.2.16.3 year

The year specifies the year element of a date

4.2.17 Cartesian_coordinate_space

The Cartesian_coordinate_space defines a Cartesian reference system that associates a unique set of parameters (usually x,y,z) with each point in a dimensional space.

4.2.17.1 units

The dimensioality specifies the number of coordinate axies.

NOTE: dimensional axies are 2D or 3D.

4.2.18 Cartesian_point

A cartesian_point is a point defined by its coordinates in a rectangular Cartesian coordinate space, or in a parameter space. The entity is defined in a one, two or three-dimensional space as determined by the number of coordinates in the list.

The data associated with a cartesian_point are the following:

— x;

— y;

— z.

4.2.18.1 x

The x specifies the first coordinate of the point location.

4.2.18.2 y

The y specifies the second coordinate of the point location; this will not exist in the case of a one dimensional point.

4.2.18.3 z

The z specifies the third coordinate of the point location; this will not exist in the case of a one or two-dimensional point.

4.2.19 Cartesian_vector

A Cartesian_vector is a general direction vector in three dimensional space. The actual magnitudes of the components have no effect upon the direction being defined, only the ratio x:y:z is significant. The data associated with a cartesian_vector are the following:

— i;

— j;

— k.

4.2.19.1 i

The i specifies the first coordinate of the vector direction.

4.2.19.2 j

The j specifies the second coordinate of the vector direction.

4.2.19.3 k

The k specifies the third coordinate of the vector direction.

4.2.20 Catalogue_gear

A Catalogue_gear is a is a type of Gear (see 4.2.20) that is a reference to a document containing the information to create a gear on a part.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.18 in ISO 10303-224:2006.

NOTE 2 Figure 11 illustrates a Catalogue_gear.

The data associated with a Catalogue_gear are the following:

— documentation.

4.2.20.1 documentation

The documentation specifies the document that defines information pertaining to a Gear feature. See 4.3.13 for the application assertion.

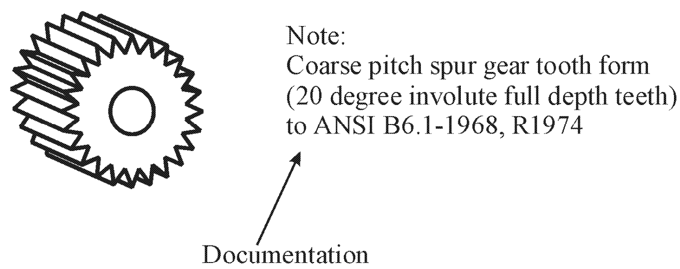


Figure 11 — Catalogue_gear

4.2.21 Catalogue_knurl

A Catalogue_knurl is a type of Knurl (see 4.2.152) that is a reference to a document containing the information to create a knurl on a part.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.19 in ISO 10303-224:2006.

NOTE 2 Figure 12 illustrates a Catalogue_knurl.

The data associated with a Catalogue_knurl are the following:

— documentation.

4.2.21.1 documentation

The documentation specifies the document that defines information pertaining to a Knurl feature. See 4.3.14 for the application assertion.

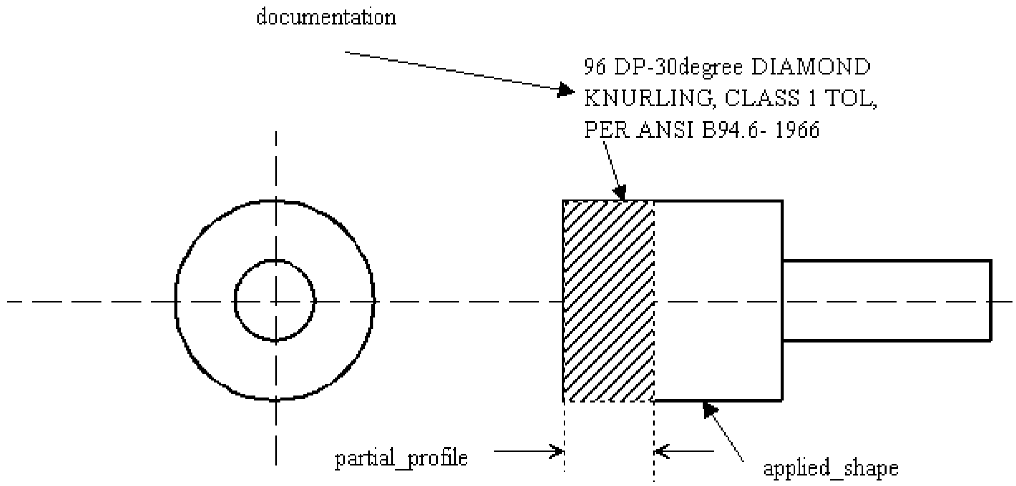


Figure 12 — Catalogue_knurl

4.2.22 Catalogue_marking

A Catalogue_marking is a type of Marking (see 4.2.165) that is a reference to a document containing the information for marking on a surface of a part.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.20 in ISO 10303-224:2006.

NOTE 2 Figure 13 illustrates a Catalogue_marking.

The data associated with a Catalogue_marking are the following:

— documentation.

4.2.22.1 documentation

The documentation specifies the document or specification that defines information pertaining to a Marking feature. See 4.3.15 for the application assertion.

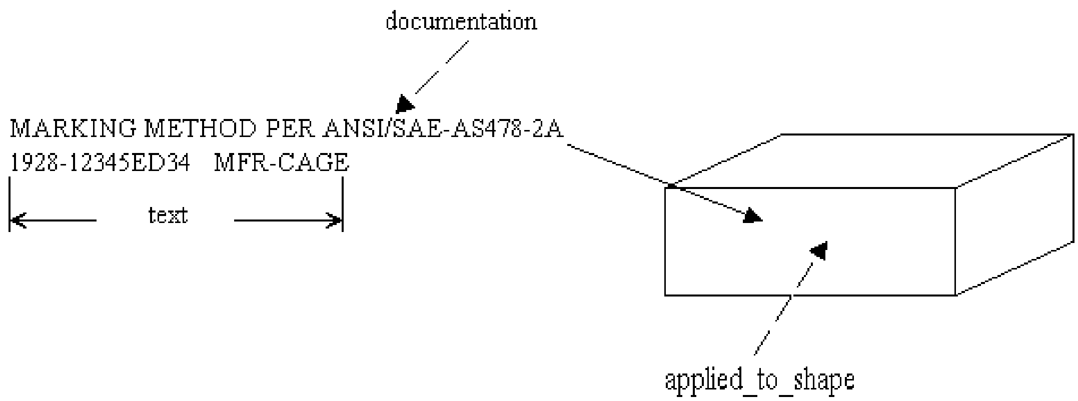


Figure 13 — Catalogue_marking

4.2.23 Catalogue_thread

A Catalogue_thread is a type of Thread (see 4.2.262) that is a reference to a document containing the information to create threads on a part.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.21 in ISO 10303-224:2006.

NOTE 2 Figure 14 illustrates a Catalogue_thread.

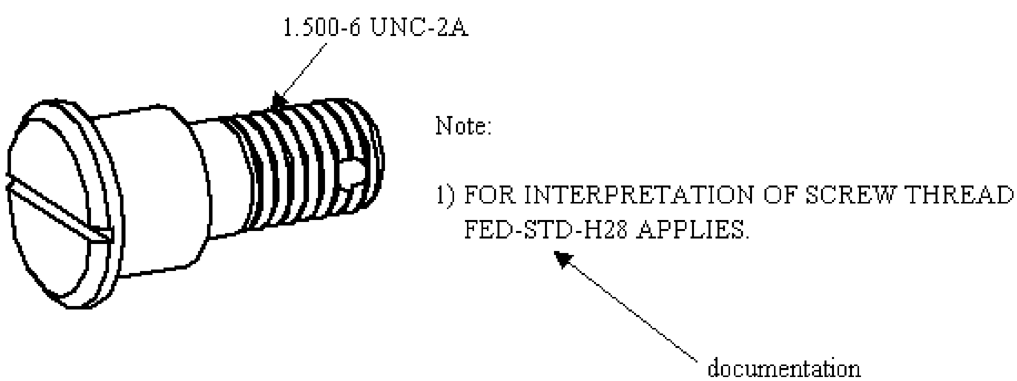


Figure 14 — Catalogue_thread

The data associated with a Catalogue_thread are the following:

— documentation;

— major_diameter.

4.2.23.1 documentation

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

4.2.23.2 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 4.3.17 for the application assertion.

4.2.24 Chamfer

A Chamfer is a type of Transition_feature (see 4.2.274) 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 1 This application object is harmonized with the same entity from paragraph 4.2.22 in ISO 10303-224:2006.

NOTE 2 Figure 15 illustrates a Chamfer.

The data associated with a Chamfer are the following:

— chamfer_face;

— first_face_offset;

— second_face_offset.

4.2.24.1 chamfer_face

The chamfer_face specifies the flat transition face between the two edges of two faces. See 4.3.18 for the application assertion.

4.2.24.2 first_face_offset

The first_face_offset specifies a face and offset amount for one of the edges of the Chamfer. See 4.3.19 for the application assertion.

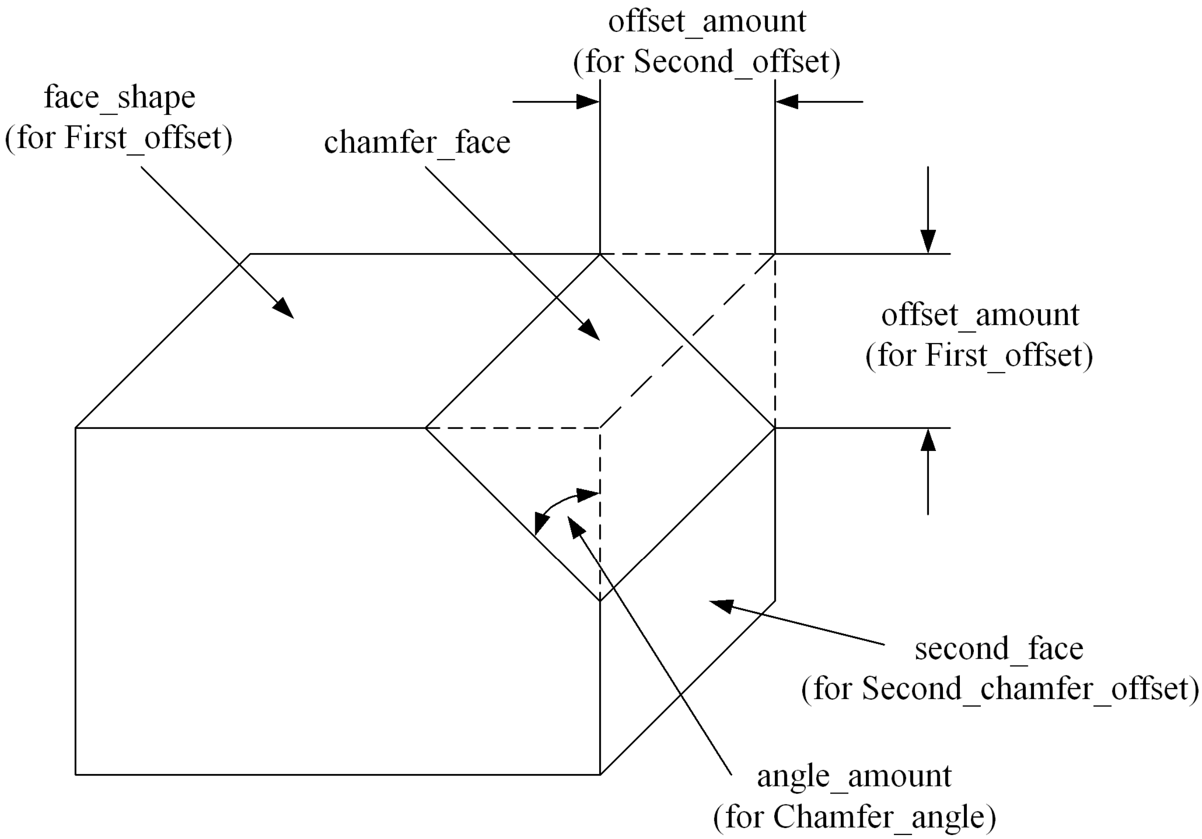


Figure 15 — Chamfer

4.2.24.3 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 4.3.20 for the application assertion.

4.2.25 Chamfer_angle

A Chamfer_angle is a type of Second_chamfer_offset (see 4.2.234) that is the measured angle from the Second_offset face to the Chamfer face for creating a Chamfer feature.

NOTE This application object is harmonized with the same entity from paragraph 4.2.23 in ISO 10303-224:2006.

The data associated with a Chamfer_angle are the following:

— angle_amount.

4.2.25.1 angle_amount

The Angle_amount specifies the angular measurement from a face for creating a Chamfer feature. See 4.3.21 for the application assertion.

4.2.26 Circular_boss

A Circular_boss is a type of Boss (see 4.2.10) that is a cylindrical shape. A Circular_boss may be tapered.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.24 in ISO 10303-224:2006.

NOTE 2 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.

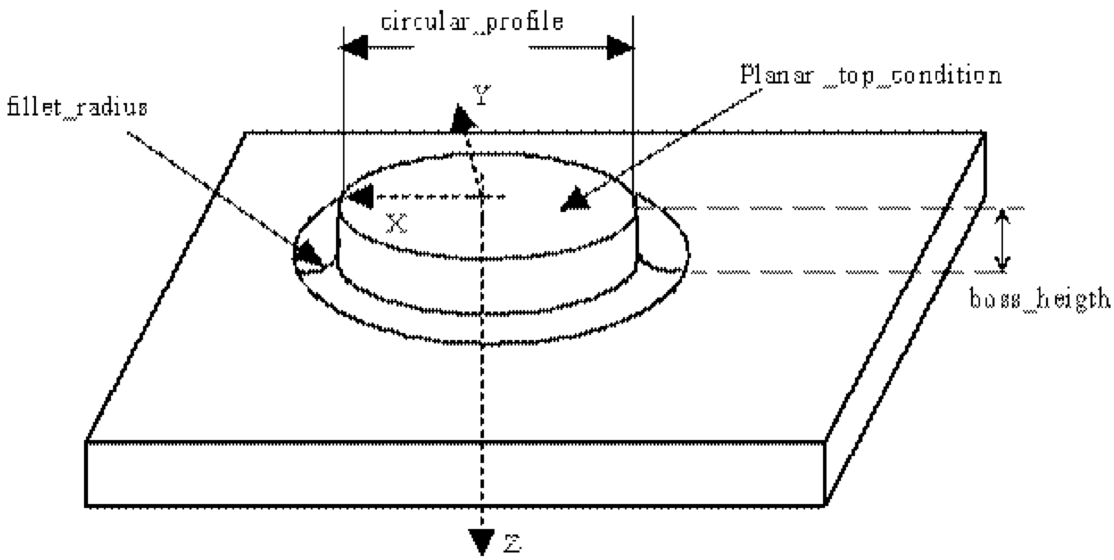


Figure 16 — Circular_boss

4.2.26.1 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 4.3.23, 4.3.24, and 4.3.25 for the application assertion.

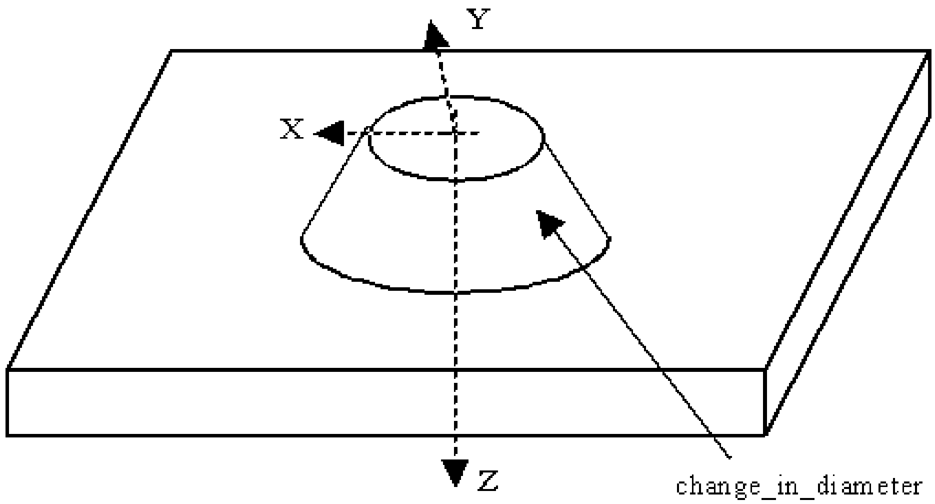


Figure 17 — Circular_boss with taper

4.2.26.2 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 (see 4.2.27) 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 4.3.22 for the application assertion.

4.2.27 Circular_closed_profile

A Circular_closed_profile is a type of Closed_profile (see 4.2.36) that is an enclosed area bounded by a circle. The orientation is at the center of the circle.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.25 in ISO 10303-224:2006.

NOTE 2 Figure 15 illustrates a Circular_closed_profile.

The data associated with a Circular_closed_profile are the following:

— diameter.

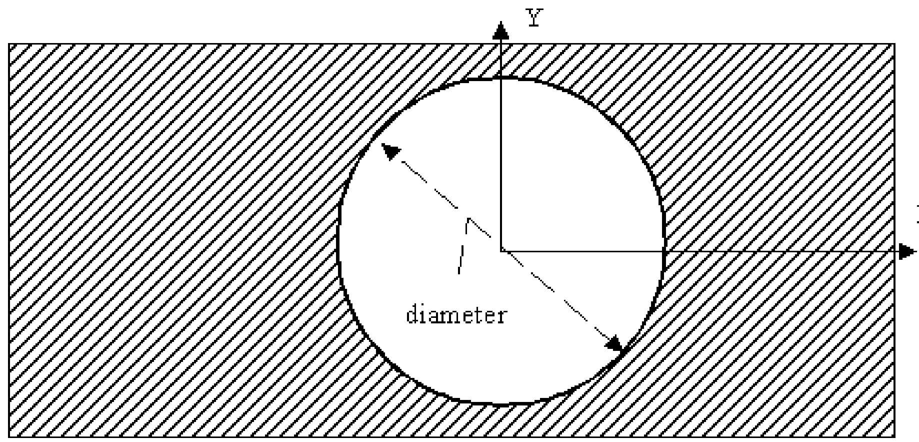


Figure 18 — Circular_closed_profile

4.2.27.1 diameter

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

4.2.28 Circular_closed_shape_profile

A Circular_closed_shape_profile is a type of Shape_profile (see 4.2.239) that defines a completely enclosed volume.

NOTE This application object is harmonized with the same entity from paragraph 4.2.26 in ISO 10303-224:2006.

The data associated with a Circular_closed_shape_profile are the following:

— closed_boundary.

4.2.28.1 closed_boundary

The closed_boundary specifies the outline of the Circular_closed_shape_profil 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 (see 4.2.190), 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 4.3.27 for the application assertion.

4.2.29 Circular_cutout

A Circular_cutout is a type of Cutout (see 4.2.51) that is an enclosed volume of circular shape. A Circular_cutout is similar in definition to a Hole (see 4.2.149), but differs in the type of process required to manufacture.

NOTE This application object is harmonized with the same entity from paragraph 4.2.27 in ISO 10303-224:2006.

The data associated with a Circular_cutout are the following:

— circular_boundary.

4.2.29.1 circular_boundary

The circular_boundary is the distance across the Circular_cutout. The placement and orientation of the Circular_closed_profile (see 4.2.27) shall be the same as the Circular_cutout feature. See 4.3.28 for the application assertion.

4.2.30 Circular_offset_pattern

A Circular_offset_pattern is a type of Circular_pattern (see 4.2.33) with a modification of the placement of a particular occurrence of the base feature relative to its expected placement.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.28 in ISO 10303-224:2006.

NOTE 2 Figure 19 illustrates a Circular_offset_pattern. The rectangular base shape is offset from index number seven.

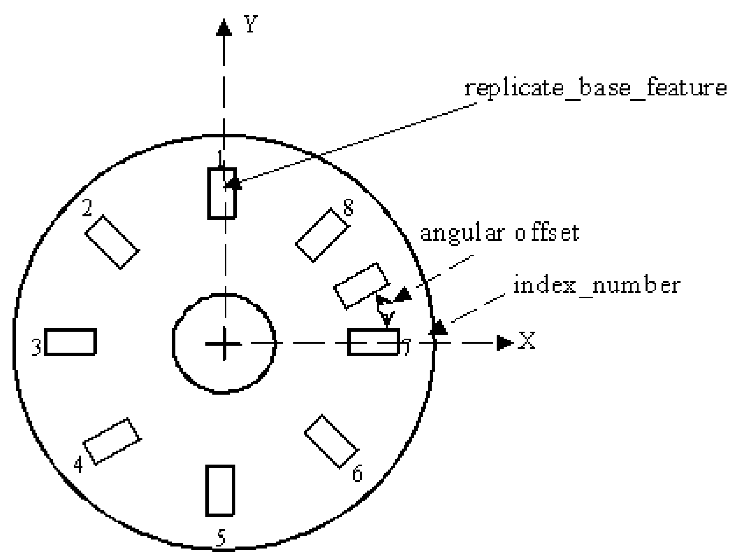


Figure 19 — Circular_offset_pattern

The data associated with a Circular_offset_pattern are the following:

— angular_offset;

— index_number.

4.2.30.1 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 4.3.29 for the application assertion.

4.2.30.2 index_number

The index_number specifies the value for unique identification of a location of a base feature used in a Circular_pattern. See 4.3.29 for the application assertion.

4.2.31 Circular_omit_pattern

A Circular_omit_pattern is a type of Circular_pattern (see 4.2.33) with an omission of a particular occurrence of the base feature.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.29 in ISO 10303-224:2006.

NOTE 2 **Error! Reference source not found.** 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.

4.2.31.1 omit_index

The omit_index specifies the value for unique identification of the base feature to be omitted in a Circular_pattern. See 4.3.30 for the application assertion.

4.2.32 Circular_path

A Circular_path is a type of Path (see 4.2.190) 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 4.2.39) or a Partial_circular_path (see 4.2.187).

NOTE This application object is harmonized with the same entity from paragraph 4.2.30 in ISO 10303-224:2006.

The data associated with a Circular_path are the following:

— radius.

4.2.32.1 radius

The radius specifies the constant distance from an axis for the Circular_path. See 4.3.31 for the application assertion.

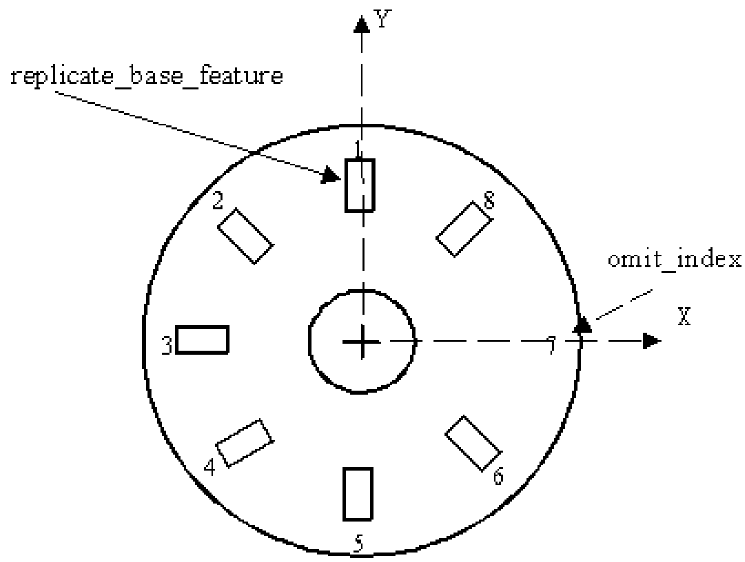


Figure 20 — Circular_omit_pattern

4.2.33 Circular_pattern

A Circular_pattern is a type of Replicate_feature (see 4.2.224) 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 1 This application object is harmonized with the same entity from paragraph 4.2.31 in ISO 10303-224:2006.

NOTE 2 Figure 22 illustrates a Circular_pattern with rotation of a base feature, and Figure 21 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.

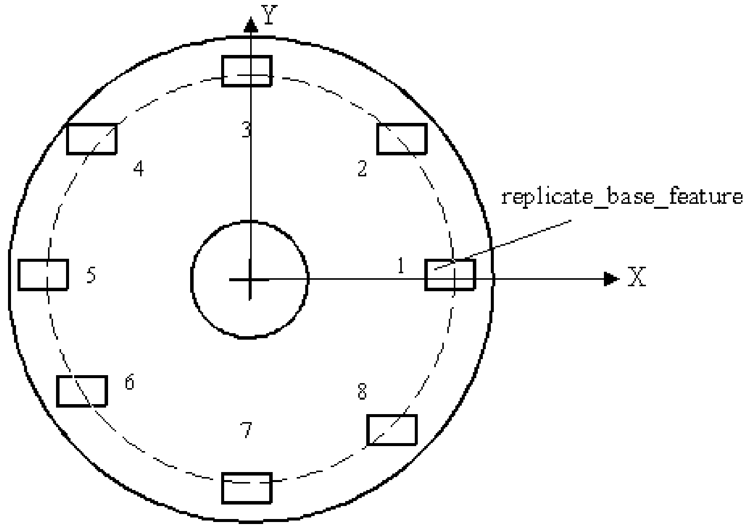


Figure 21 — Circular_pattern without rotation

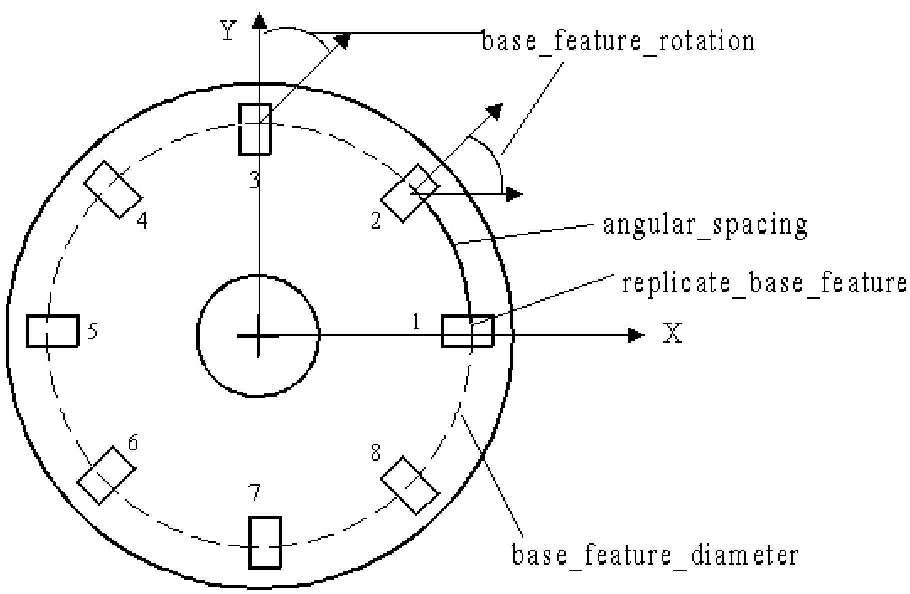


Figure 22 — Circular_pattern with rotation of a base feature

4.2.33.1 angular_spacing

The angular_spacing specifies the angle amount between features in a Circular_pattern. See 4.3.34 for the application assertion.

4.2.33.2 base_feature_diameter

The base_feature_diameter specifies the size of the circle for placement of features in a Circular_pattern. See 4.3.34 for the application assertion.

4.2.33.3 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 4.3.34 for the application assertion.

4.2.33.4 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 4.3.33 for the application assertion.

4.2.33.5 number_of_features

The number_of_features specifies the quantity of a base feature to be used in a Circular_pattern. See 4.3.34 for the application assertion.

4.2.33.6 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 4.3.32 for the application assertion.

4.2.34 Circular_runout_tolerance

A Circular_runout_tolerance is a type of Geometric_tolerance (see 4.2.143) 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 1 This application object is harmonized with the same entity from paragraph 4.2.32 in ISO 10303-224:2006.

NOTE 2 Figure 23 illustrates a Circular_runout_tolerance.

The data associated with a Circular_runout_tolerance are the following:

— geometric_reference;

— runout_angle.

4.2.34.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.35 for the application assertion.

4.2.34.2 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.

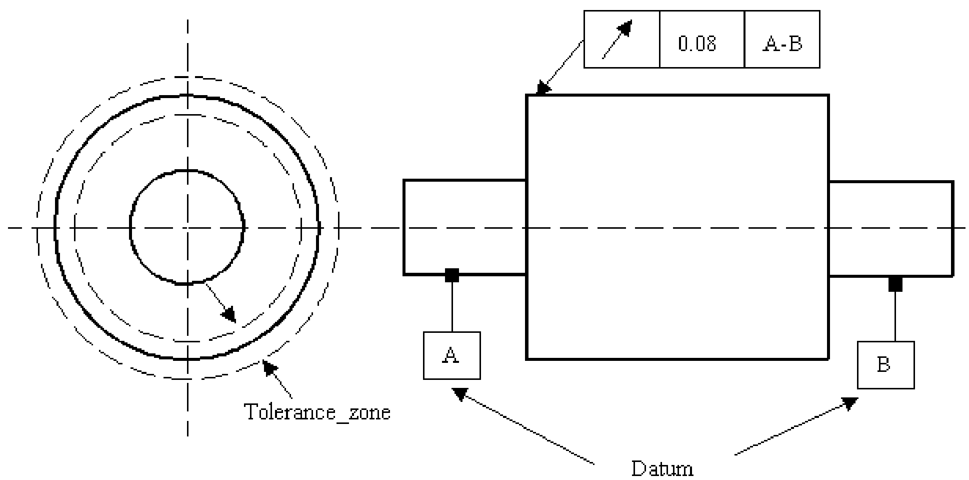


Figure 23 — Circular_runout_tolerance

4.2.35 Circularity_tolerance

A Circularity_tolerance is a type of Geometric_tolerance (see 4.2.143) 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 This application object is harmonized with the same entity from paragraph 4.2.33 in ISO 10303-224:2006.

NOTE 2 The Circularity_tolerance definition is derived from ISO 1101, 14.3.

NOTE 3 Figure 24 illustrates Circularity_tolerance.

4.2.36 Closed_profile

A Closed_profile is a type of Profile (see 4.2.205) that is an outline or shape that bounds an enclosed area with no opening. Each Closed_profile is either a Circular_closed_profile (see 4.2.27), General_closed_-profile (see 4.2.128), Ngon_profile (see 4.2.171), or a Rectangular_closed_profile (see 4.2.216)

NOTE This application object is harmonized with the same entity from paragraph 4.2.35 in ISO 10303-224:2006.

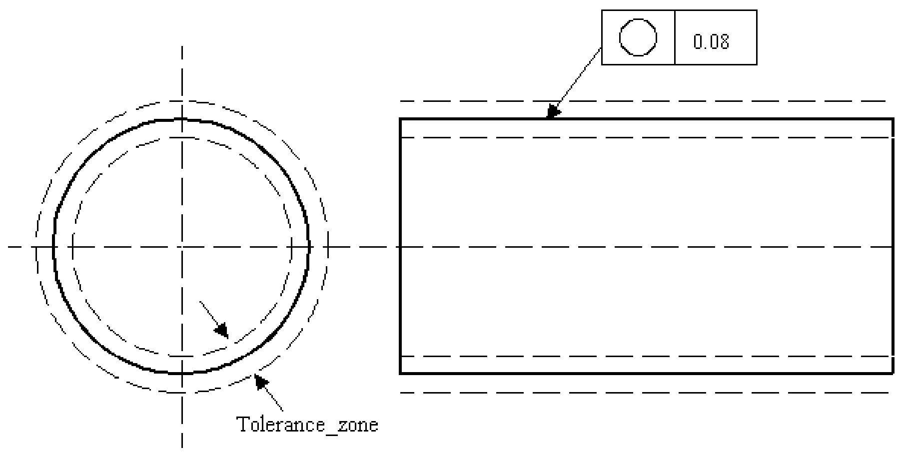


Figure 24 — Circularity_tolerance

4.2.37 Closed_slot

A Closed_slot is a type of Slot (see 4.2.241) that has a course of travel defined by a Path that is closed and has no opening.

NOTE This application object is harmonized with the same entity from paragraph 4.2.36 in ISO 10303-224:2006.

The data associated with a Closed_slot are the following:

— course_of_travel;

— end_condition.

4.2.37.1 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 4.3.36 and 4.3.37 for the application assertion.

4.2.37.2 end_condition

The end_conditions specifies the type of closed shape at the ends of the Slot. See 4.3.38 for the application assertion.

4.2.38 Common_datum

A Common_datum is a type of Datum (see 4.2.55) that is a set of two or more Datum_feature objects which are for establishing a single datum plane or axis.

NOTE This application object is harmonized with the same entity from paragraph 4.2.38 in ISO 10303-224:2006.

The data associated with a Common_datum are the following:

— element.

4.2.38.1 element

The element specifies the list of Datum_feature objects for defining the Common_datum. See 4.3.39 for the application assertion.

4.2.39 Complete_circular_path

A Complete_circular_path is a type of Circular_path (see 4.2.32) that is a direction of travel that begins and ends at the same point on the arc.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.37 in ISO 10303-224:2006.

NOTE 2 Figure 25 illustrates a Slot feature with a Square_u_profile and a Complete_circular_path.

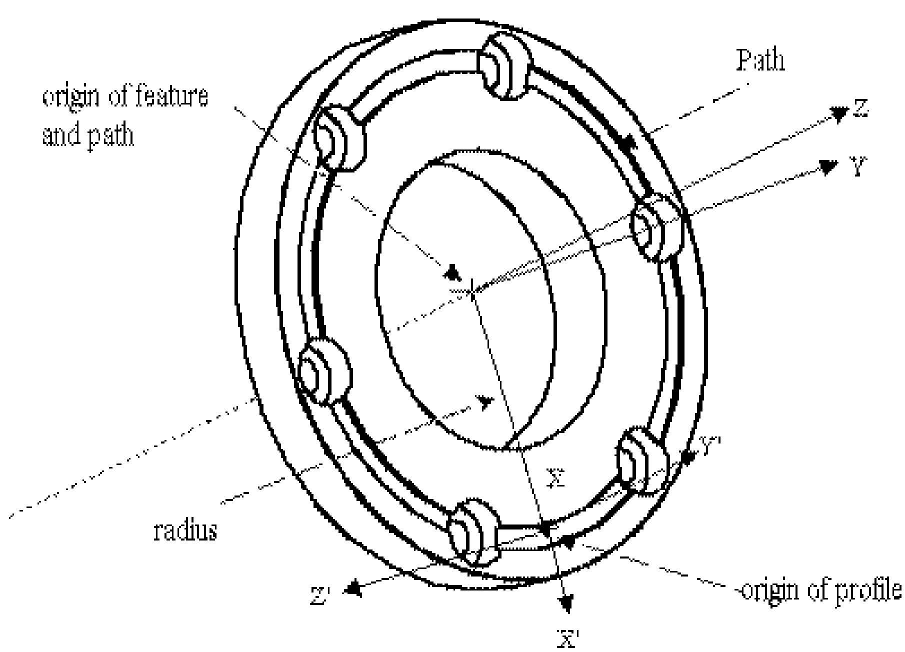


Figure 25 — Complete_circular_path

4.2.40 Compound_feature

A Compound_feature is a type of Machining_feature (see 4.2.162) that is a union of one or more Machining_feature objects to create a more complex feature definition. The placement of a Compound_feature is relative to either the part, another Compound_feature, or a Replicate_feature which uses a Compound_feature as the base feature. Features which are elements of the Compound_feature have placement defined relative to the Compound_feature placement.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.39 in ISO 10303-224:2006..

NOTE 2 Figure 26 illustrates an example of a Compound_feature with one counterbore_hole feature and one countersunk_hole feature.

The data associated with a Compound_feature are the following:

— element;

— feature_description;

— feature_name.

4.2.40.1 element

The element specifies the base features that when combined defines a Compound_feature. There shall be more than one element for a Compound_feature. See 4.3.40 for the application assertion.

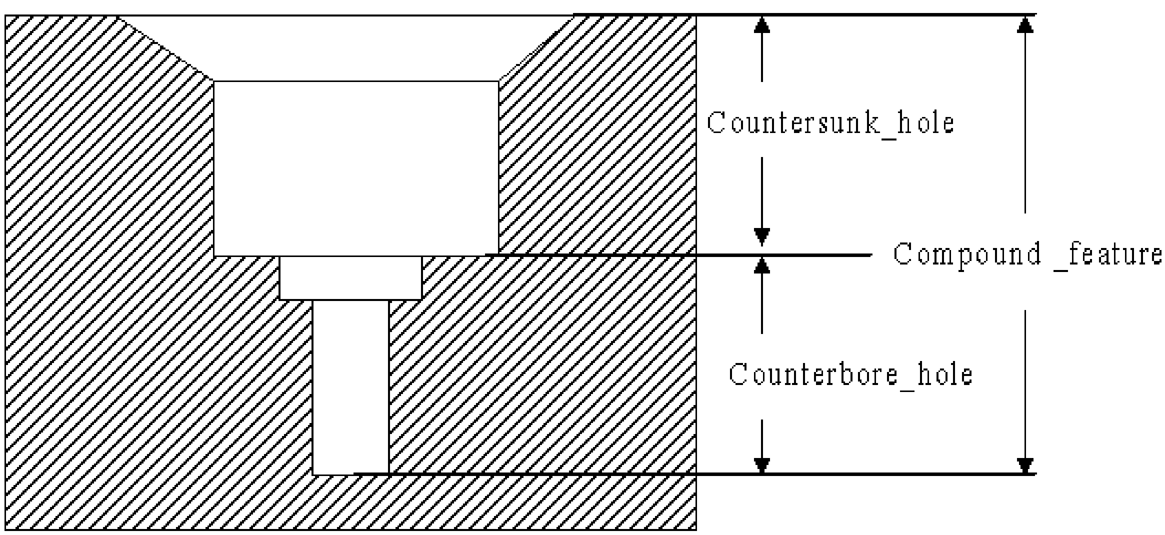


Figure 26 — Compound_feature

4.2.40.2 feature_description

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

4.2.40.3 feature_name

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

4.2.41 Compound_feature_element

A Compound_feature_element is a specification of the type of Machining_feature (see 4.2.162) or Transition_feature (see 4.2.274) to be used for a particular element of a Compound_feature. A Compound_feature consists of two or more Compound_feature_element objects and need not require Compound_feature_element objects to be ordered.

NOTE This application object is harmonized with the same entity from paragraph 4.2.40 in ISO 10303-224:2006.

The data associated with a Compound_feature_element are the following:

— element.

4.2.41.1 element

The element specifies the base feature to be used as one of the components for the Compound_feature. See 4.3.41 and 4.3.42 for the application assertion.

4.2.42 Compound_feature_relationship

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

NOTE This application object is harmonized with the same entity from paragraph 4.2.41 in ISO 10303-224:2006.

The data associated with a Compound_feature_relationship are the following:

— predecessor;

— successor.

4.2.42.1 predecessor

The predecessor specifies the Compound_feature_element with the highest precedence. See 4.3.43 for the application assertion.

4.2.42.2 successor

The successor specifies Compound_feature_element with a lesser precedence. See 4.3.43 for the application assertion.

4.2.43 Concentricity_tolerance

A Concentricity_tolerance is a type of Geometric_tolerance (see 4.2.143) specifying 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 This application object is harmonized with the same entity from paragraph 4.2.42 in ISO 10303-224:2006.

NOTE 2 The Concentricity_tolerance definition is derived from ISO 1101, 14.11..

NOTE 3 Figure 27 illustrates the Concentricity_tolerance.

The data associated with a Concentricity_tolerance are the following:

— geometric_reference;

— value_qualifier.

4.2.43.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.44 for the application assertion.

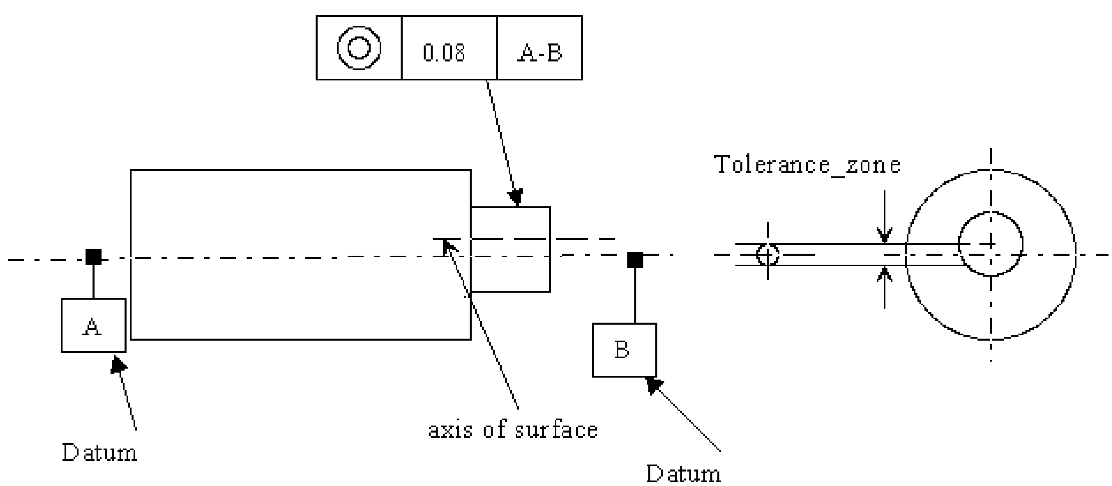


Figure 27 — Concentricity_tolerance

4.2.43.2 value_qualifier

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

4.2.44 Conical_hole_bottom

A Conical_hole_bottom is a type of Blind_bottom_condition (see 4.2.8) 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 1 This application object is harmonized with the same entity from paragraph 4.2.43 in ISO 10303-224:2006.

NOTE 2 Figure 28 illustrates the Conical_hole_bottom for a Round_hole.

The data associated with a Conical_hole_bottom are the following:

— tip_angle;

— tip_radius.

4.2.44.1 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 4.3.45 for the application assertion.

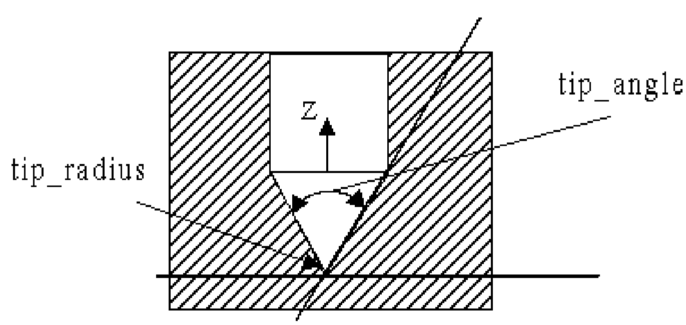


Figure 28 — Conical_hole_bottom

4.2.44.2 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 4.3.45 for the application assertion.

4.2.45 Constant_radius_edge_round

A Constant_radius_edge_round is a type of Edge_round (see4.2.113) that is defined with a constant radius value.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.44 in ISO 10303-224:2006.

NOTE 2 Figure 29 illustrates the Constant_radius_edge_round on a rectangular block.

The data associated with a Constant_radius_edge_round 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 4.2.113.2) to the center of the edge_round_face (see 4.2.113.1). A constant_radius_edge_round may but need not require a first_face_offset. See 4.3.46 for the application assertion.

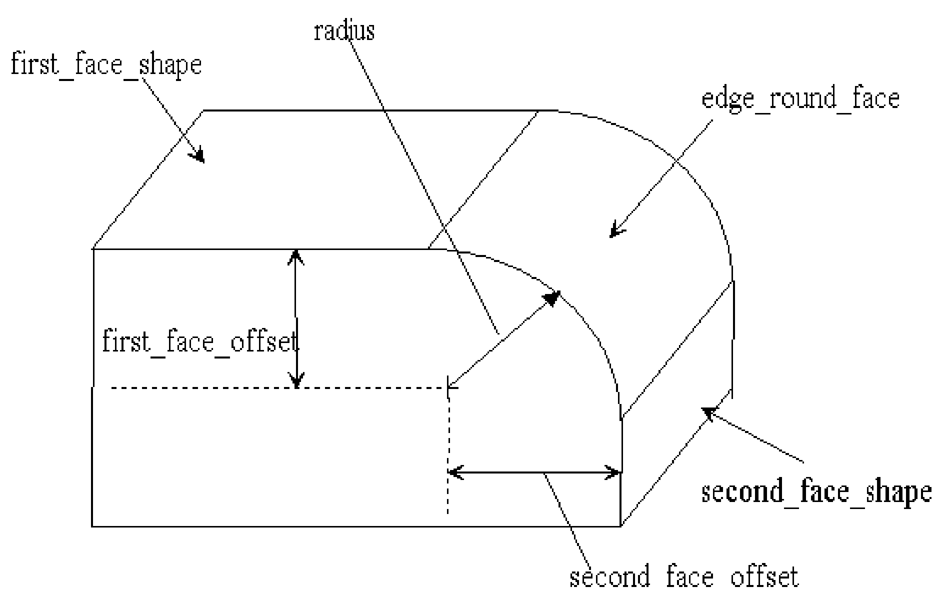


Figure 29 — Constant_radius_edge_round

4.2.45.2 second_face_offset

The second_face_offset specifies the amount of offset from the second_face_shape (see 4.2.113.3) to the center of the edge_round_face (see 4.2.113.1). A constant_radius_edge_round may but need not require a second_face_offset. See 4.3.46 for the application assertion.

4.2.45.3 radius

The radius specifies the amount of curvature for a convex transition between the two faces of a Constant_radius_edge_round. See 4.3.46 for the application assertion.

4.2.46 Constant_radius_fillet

A Constant_radius_fillet is a type of Fillet (see 4.2.119) that is defined with a constant radius value.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.45 in ISO 10303-224:2006.

NOTE 2 Figure 30 illustrates the Constant_radius_fillet. The Fillet surface need not be tangent to the other surfaces because the offsets.

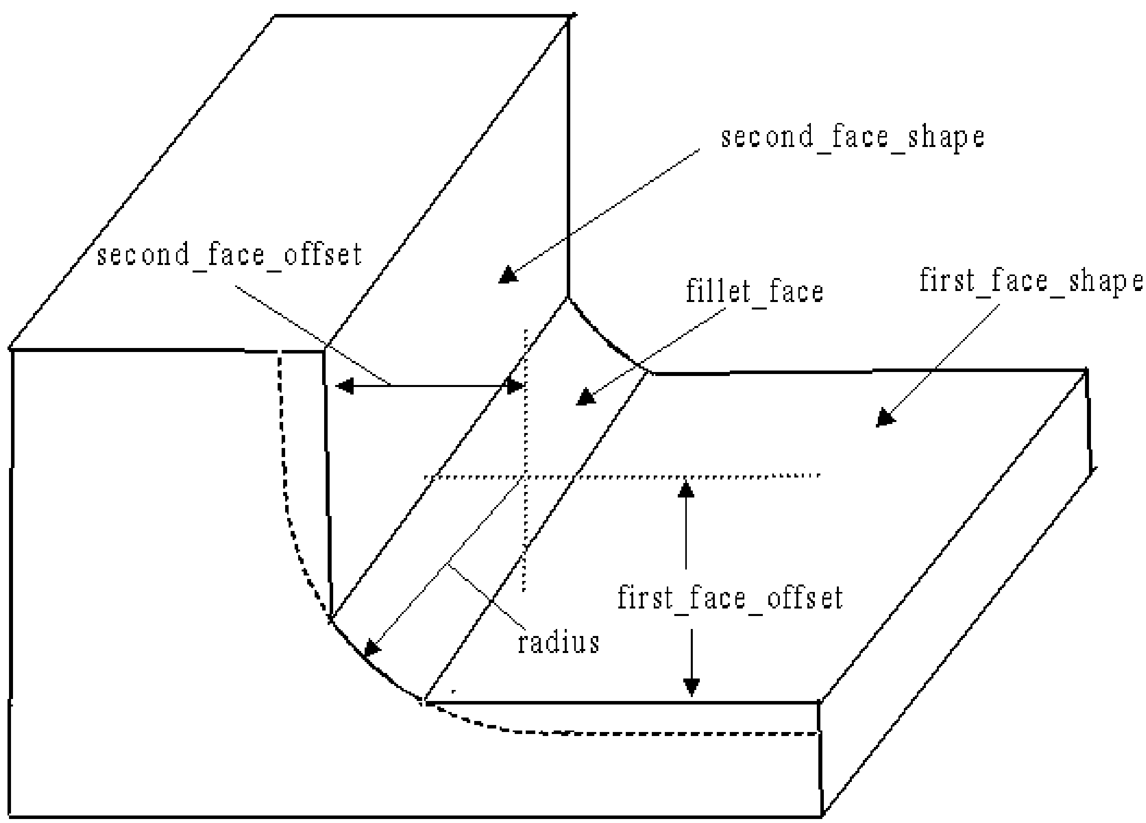


Figure 30 — Constant_radius_fillet

The data associated with a Constant_radius_fillet are the following:

— first_face_offset;

— second_face_offset;

— radius.

4.2.46.1 first_face_offset

The first_face_offset specifies the amount of offset from the first_face_shape (see 4.2.119.2) to the center of the fillet_face (see 4.2.119.1). A constant_radius_fillet may but need not require a first_face_offset. See 4.3.47 for the application assertion.

4.2.46.2 second_face_offset

The second_face_offset specifies the amount of offset from the second_face_shape (see 4.2.119.3) to the center of the fillet_face (see 4.2.119.1). A constant_radius_fillet may but need not require a first_face_offset. See 4.3.47 for the application assertion.

4.2.46.3 radius

The radius specifies the amount of curvature for a concave transition between two surfaces. See 4.3.47 for the application assertion.

4.2.47 Coordinate_tolerance

A Coordinate_tolerance is a type of Location_tolerance (see 4.2.160) that specifies a location tolerance through the origin of the reference frame rather than the origin of a feature.

The data associated with a Coordinate_tolerance are the following:

— axial;

— base_coordinate_system;

— direction.

4.2.47.1 axial

The axial signifies that the feature being toleranced must be capable of being reduced to a line/axial feature. The axial need not be specified for a particular Coordinate_tolerance.

4.2.47.2 base_coordinate_system

The base_coordinate_system is the coordinate system with respect to which the Coordinate_tolerance is specified. See 4.3.48 for the application assertion.

4.2.47.3 direction_type

The direction_type specifies a measurement direction relative to the base_coordinate_system.. The values of the direction_type may be one of the following:

— angle;

— radial;

— x axis;

— y axis;

—z axis.

NOTE See 4.2.47.3.1 - 4.2.47.3.5 for the definition of each allowable value for dimension_note.

4.2.47.3.1 angle: measurement is angular.

4.2.47.3.2 radial: measurement is radial.

4.2.47.3.3 x axis: measurement made along x axis.

4.2.47.3.4 y axis: measurement made along y axis.

4.2.47.3.5 z axis: measurement made along z axis.

4.2.48 Counterbore_hole

A Counterbore_hole is a type of Hole (see4.2.149) 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 1 This application object is harmonized with the same entity from paragraph 4.2.46 in ISO 10303-224:2006.

NOTE 2 Figure 31 illustrates the Counterbore_hole with aCounterbore_hole and a Through_bottom_condition.

The data associated with a Counterbore_hole are the following:

— larger_hole;

— smaller_hole.

4.2.48.1 larger_hole

The larger_hole specifies the Round_hole that will be used as the larger hole for the Counterbore_hole. See 4.3.49 for the application assertion.

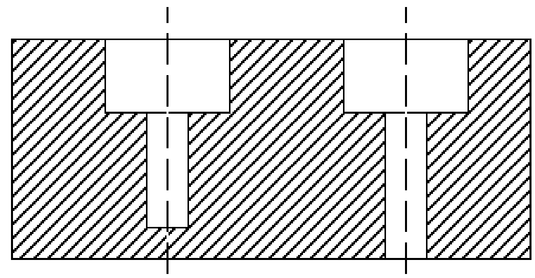


Figure 31 — Counterbore_hole

4.2.48.2 smaller_hole

The smaller_hole specifies the Round_hole that will be used as the smaller hole for the Counterbore_hole. See 4.3.49 for the application assertion.

4.2.49 Countersunk_hole

A Countersunk_hole is a type of Hole (see4.2.149) 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 1 This application object is harmonized with the same entity from paragraph 4.2.47 in ISO 10303-224:2006.

NOTE 2 Figure 32 illustrates the Countersunk_hole with a Blind_bottom_condition and a Through_-bottom_condition.

The data associated with a Countersunk_hole are the following:

— constant_diameter_hole;

— tapered_hole.

4.2.49.1 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 4.3.50 for the application assertion.

4.2.49.2 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 4.3.50 for the application assertion.

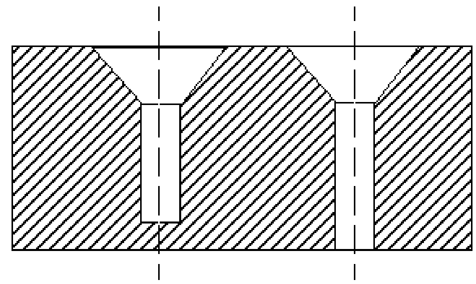


Figure 32 — Countersunk_hole

4.2.50 Curved_dimension_tolerance

A Curved_dimension_tolerance is a type of Size_tolerance (see 4.2.240) that is the tolerance on a dimension for a curve measured along the entire path of the curve.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.48 in ISO 10303-224:2006.

NOTE 2 Figure 33 illustrates the Curved_dimension_tolerance.

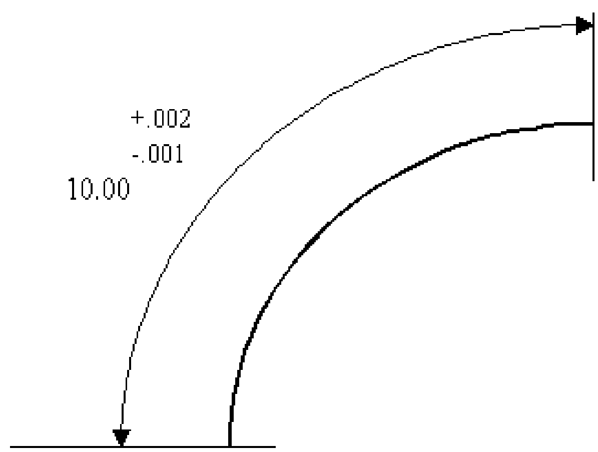


Figure 33 — Curved_dimension_tolerance

4.2.51 Cutout

A Cutout is a type of Pocket (see 4.2.202) 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 4.2.29)or a General_cutout (see 4.2.129).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.50 in ISO 10303-224:2006.

NOTE 2 Figure 34 illustrates the Cutout.

The data associated with a Cutout are the following:

— bottom_condition.

4.2.51.1 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 4.3.51 for the application assertion.

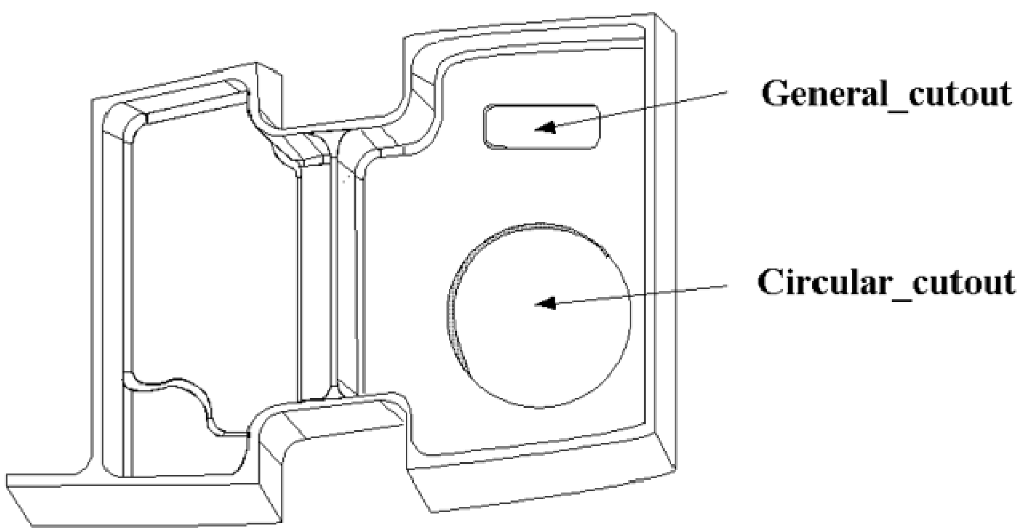


Figure 34 — Cutout

4.2.52 Cylindrical_base_shape

A Cylindrical_base_shape is a type of Implicit_base_shape_representation (see 4.2.150) that is the initial shape of the material which is cylindrical.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.52 in ISO 10303-224:2006.

NOTE 2 Figure 35 illustrates a Cylindrical_base_shape.

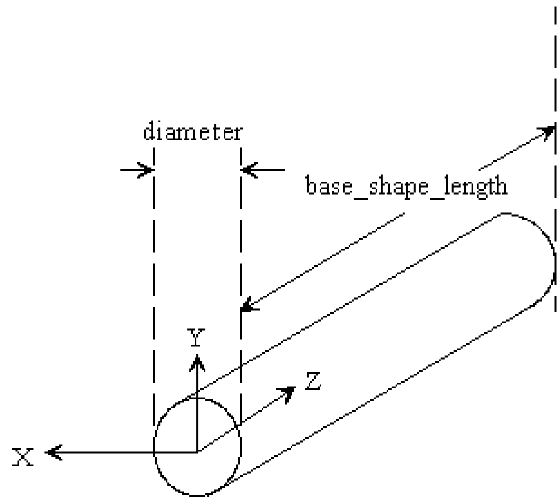


Figure 35 — Cylindrical_base_shape

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 4.3.52 for the application assertion.

4.2.53 Cylindricity_tolerance

A Cylindricity_tolerance is a type of Geometric_tolerance (see 4.2.143) 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 This application object is harmonized with the same entity from paragraph 4.2.53 in ISO 10303-224:2006.

NOTE 2 The Cylindricity_tolerance definition is derived from ISO 1101, 14.4.

NOTE 3 Figure 36 illustrates Cylindricity_tolerance.

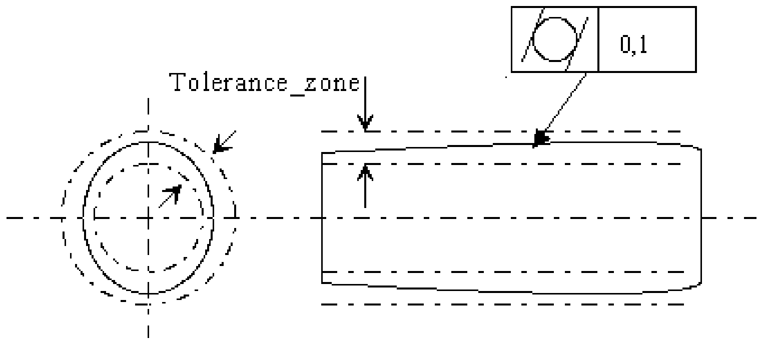


Figure 36 — Cylindricity_tolerance

4.2.54 Date_time

A Date_time is a specification of the time on a date which includes the times offset from universal time.

The data associated with a Date_Time are the following:

— date;

— hour;

— minute;

— offset;

— second.

4.2.54.1 date

The date specifies a calendar_date. See 4.3.53 for the application assertion.

4.2.54.2 hour

The hour specifies the integer specifying the hour represented by the time as described on a twenty four hour clock.

4.2.54.3 minute

The minute specifies a minute within the hour represented by the time.

4.2.54.4 second

The second specifies a second within the minute represented by the time.

4.2.54.5 offset

The offset specifies the offset from universal time. See 4.3.54 for the application assertion.

4.2.55 Datum

A Datum is defined in ISO 5459. A Datum is either a Datum_feature (see 4.2.56) or a Common_datum (see 4.2.38).

NOTE This application object is harmonized with the same entity from paragraph 4.2.54 in ISO 10303-224:2006.

The data associated with a Datum are the following:

— name;

— precedence.

4.2.55.1 name

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

4.2.55.2 precedence

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

4.2.56 Datum_feature

A Datum_feature is a type of Datum (see 4.2.55) that is a feature on a part used to establish a Datum.

NOTE This application object is harmonized with the same entity from paragraph 4.2.55 in ISO 10303-224:2006.

The data associated with a Datum_feature are the following:

— datum_representation;

— modifier.

4.2.56.1 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 4.3.56 and 4.3.57 for the application assertion.

4.2.56.2 modifier

The modifier specifies the material condition applied to the Datum_feature. The modifier need not be specified for a particular Datum_feature. See 4.3.55 for the application assertion.

4.2.57 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 4.2.194) or a Target_area (see 4.2.255).

NOTE This application object is harmonized with the same entity from paragraph 4.2.56 in ISO 10303-224:2006.

The data associated with a Datum_target are the following:

— identifier.

4.2.57.1 identifier

The identifier specifies a unique identification for the Datum_target.

4.2.58 Datum_target_set

A Datum_target_set is a set of Datum_target objects that are used to define a datum reference for a Geometric_tolerance.

NOTE This application object is harmonized with the same entity from paragraph 4.2.57 in ISO 10303-224:2006.

The data associated with a Datum_target_set are the following:

— rule_description;

— target_shape.

4.2.58.1 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.

4.2.58.2 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 4.3.58 for the application assertion.

4.2.59 Defined_gear

A Defined_gear is a type of Gear (see 4.2.126) that is specified explicitly, all attributes for a Gear are declared and defined. Bevel_gear (see 4.2.7), Helical_gear (see 4.2.148) or Spur_gear (see 4.2.247).

NOTE This application object is harmonized with the same entity from paragraph 4.2.59 in ISO 10303-224:2006.

4.2.60 Defined_marking

A Defined_marking is a type of Marking (see4.2.165) that is specified explicitly, all attributes for a Marking are declared and defined.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.60 in ISO 10303-224:2006.

NOTE 2 Figure 37 illustrates the Defined_marking on a rectangular block shape and with text of 'TEXT FOR A MARKING'.

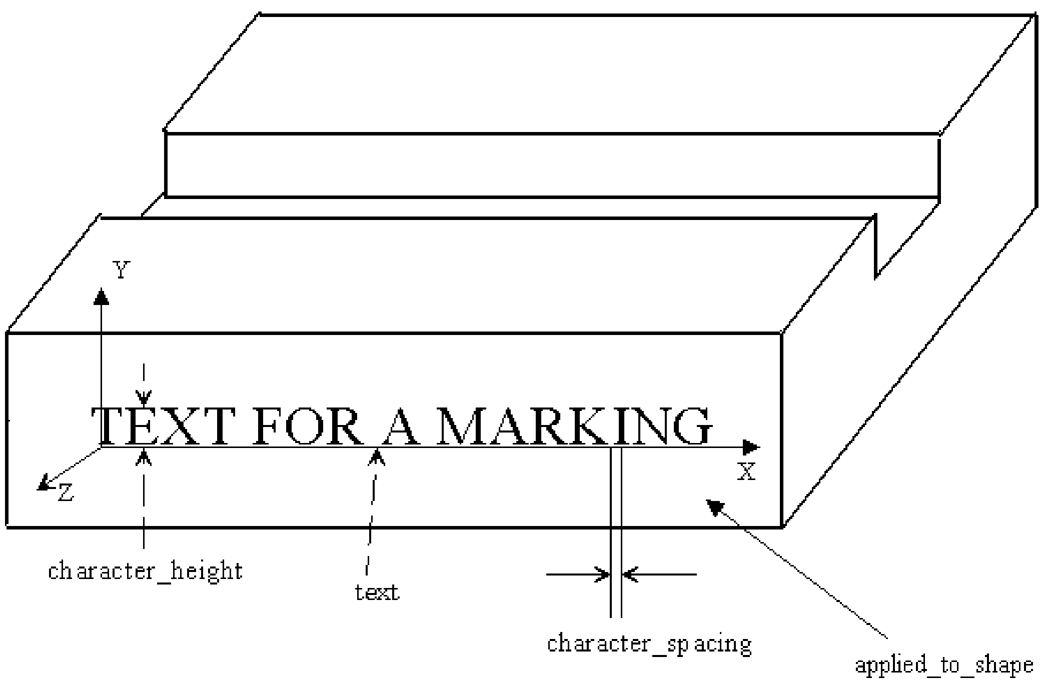


Figure 37 — Defined_marking

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 4.3.60 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 4.3.60 for the application assertion.

4.2.60.3 font_name

The font_name specifies the appearance of the characters. A font consists of typeface, treatment, and size. See 4.3.59 for the application assertion.

EXAMPLE Example of font_name characteristic is type face Times Roman. Example of font_name characteristic is a treatment of Bold or Italic. Example of font_name characteristic is 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 4.3.59 for the application assertion.

4.2.61 Defined_thread

A Defined_thread is a type of Thread (see 4.2.262) that is specified explicitly.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.61 in ISO 10303-224:2006.

NOTE 2 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.

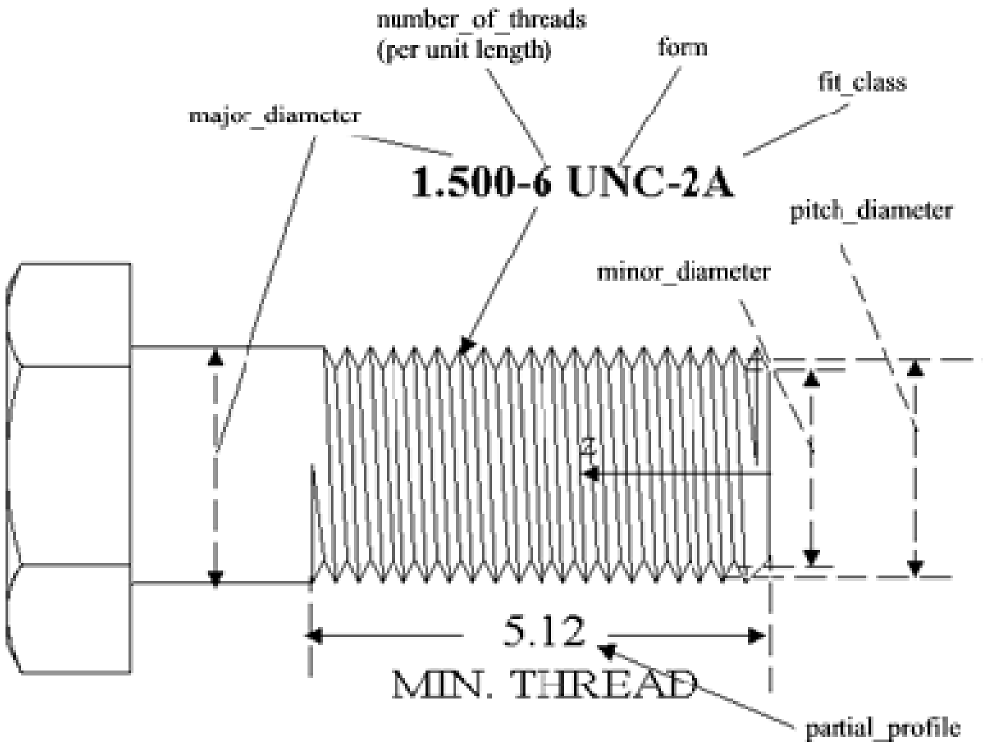


Figure 38 — Defined_thread

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 4.3.61 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 4.3.61 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 4.3.61 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 4.3.61 for the application assertion.

4.2.62 Derived_shape_element

A Derived_shape_element is a specification of that Shape_element (see 4.2.238) that is derived from another Shape_element for dimensioning or tolerancing purposes.

The data associated with a Derived_shape_element are the following:

— is_derived_from;

— role.

4.2.62.1 is_derived_from

The is_derived_from specifies the Shape_element from which the Derived_shape_element is derived. See 4.3.62 for the application assertion.

EXAMPLE The Shape_element may be a cylinder, and the derived_shape_element may be the axis of that cylinder.

4.2.62.2 role

The role specifies the function of the Derived_shape_element.

EXAMPLE Asymmetry plane is an example of a Derived_shape_element.

4.2.63 Descriptive_parameter

A Descriptive_parameter is a type of Property_parameter (see 4.2.209) that is an explanation of the property being defined by a specification.

NOTE This application object is harmonized with the same entity from paragraph 4.2.62 in ISO 10303-224:2006.

The data associated with a Descriptive_parameter are the following:

— descriptive_string.

4.2.63.1 descriptive_string

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

4.2.64 Diagonal_knurl

A Diagonal_knurl is a type of Turned_knurl (see 4.2.275) with helical cuts at an angle about the axis of a surface.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.64 in ISO 10303-224:2006.

NOTE 2 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 4.3.64 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 4.3.63 for the application assertion.

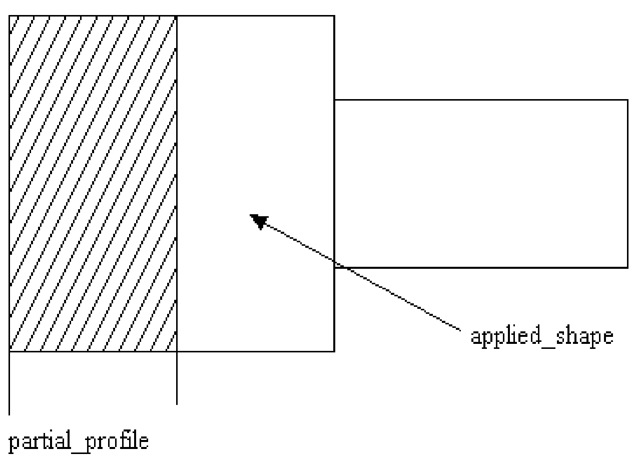


Figure 39 — Diagonal_knurl

4.2.65 Diameter_dimension_tolerance

A Diameter_dimension_tolerance is a type of Size_tolerance (see4.2.240) that is the allowable variation of the size of a hole in a surface.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.65 in ISO 10303-224:2006.

NOTE 2 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 1 This application object is harmonized with the same entity from paragraph 4.2.66 in ISO 10303-224:2006.

NOTE 2 Figure 41 illustrates the Diameter_taper.

The data associated with a Diameter_taper are the following:

— final_diameter.

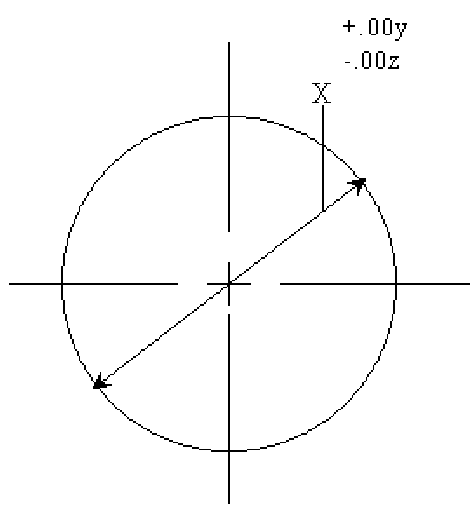


Figure 40 — Diameter_dimension_tolerance

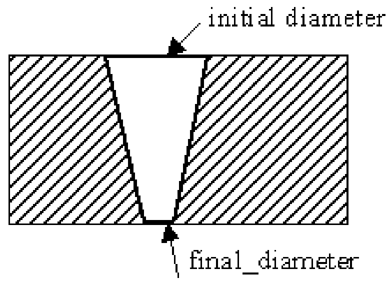


Figure 41 — Diameter_taper

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 4.3.65 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 4.2.275) 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 1 This application object is harmonized with the same entity from paragraph 4.2.67 in ISO 10303-224:2006.

NOTE 2 Figure 42 illustrates a 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 4.3.66 for the application assertion.

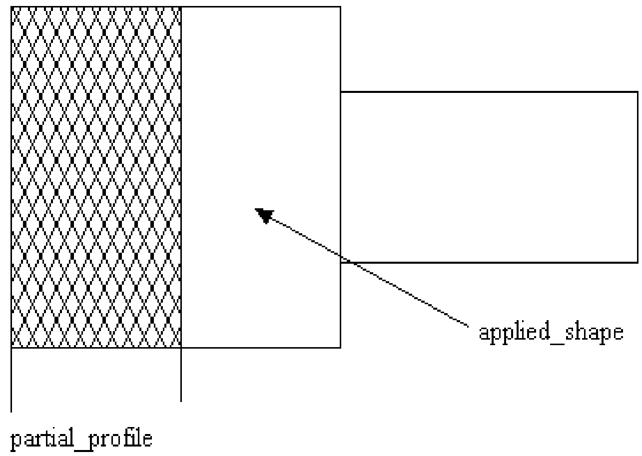


Figure 42 — Diamond_knurl

4.2.68 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 4.2.160) or a Size_tolerance (see 4.2.240)

NOTE This application object is harmonized with the same entity from paragraph 4.2.69 in ISO 10303-224:2006.

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.

4.2.68.1 dimension_description

The dimension_description specifies a textual description of any conditions which may affect the interpretation of the tolerance information that is defined. There may be more than one dimension_description for a Dimensional_tolerance. The dimension_description need not be specified for a particular Dimensional_tolerance.

EXAMPLE A Dimension_tolerance may apply in 2 places.

4.2.68.2 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 4.2.68.2.1 - 4.2.68.2.3 for the definition of each allowable value for dimension_note.

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

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

4.2.68.2.3 user defined: a description specified by the user.

4.2.68.3 dimension_value

The dimension_value specifies the value that has a dimension applied to it.

4.2.68.4 limit

The limit specifies the tolerance value applied to the Dimension_tolerance. The limit need not be specified for a particular Dimensional_tolerance. See 4.3.67 for the application assertion.

4.2.68.5 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.

4.2.68.6 unit_of_measure

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

4.2.69 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.

NOTE This application object is harmonized with the same entity from paragraph 4.2.70 in ISO 10303-224:2006.

The data associated with a Directed_taper are the following:

— angle;

— direction.

4.2.69.1 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 4.3.69 for the application assertion.

4.2.69.2 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 4.3.68 for the application assertion.

4.2.70 Direction_element

A Direction_element is a type of Shape_element (see 4.2.238) that is a Shape_aspect definition for a direction.

NOTE This application object is harmonized with the same entity from paragraph 4.2.71 in ISO 10303-224:2006.

4.2.71 Distance_along_curve_tolerance

A Distance_along_curve_tolerance is a type of Location_tolerance (see 4.2.160) that is the distance calculated between two elements along a path defined by a third element of geometry.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.72 in ISO 10303-224:2006.

NOTE 2 Figure 43 illustrates the Distance_along_curve_tolerance.

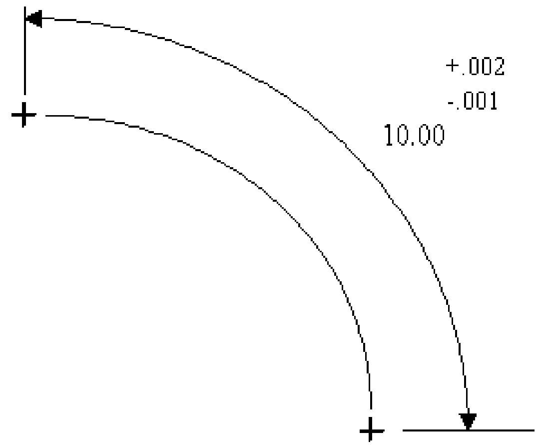


Figure 43 — Distance_along_curve_tolerance

The data associated with a Distance_along_curve_tolerance are the following:

— path;

— with_curve_direction.

4.2.71.1 path

The path specifies the shape that the tolerance applies to. See 4.3.70 for the application assertion.

4.2.71.2 with_curve_direction

The with_curve_direction specifies a boolean value that indicates the direction along the element to apply the tolerance. A value of true indicates the tolerance value is applied from the start point of the curve to the end point of the curve. A value of false indicates that the direction does not matter.

4.2.72 Dm_analysis_dofs_dml

A Dm_analysis_dofs_dml is the possible translational and rotational degrees of freedom available when a dimensional measurement feature is fitted to measured data. The data associated with a Dm_analysis_dofs_dml are the following:

— rotx;

— roty;

— rotz;

— transx;

— transy;

— transz.

4.2.72.1 rotx

The rotx specifies the degree of freedom allowing rotation around the x axis. See 4.3.71 for the application assertion.

4.2.72.2 roty

The roty specifies the degree of freedom allowing rotation around the y axis. See 4.3.71 for the application assertion.

4.2.72.3 roz

The rotz specifies the degree of freedom allowing rotation around the z axis. See 4.3.71 for the application assertion.

4.2.72.4 transx

The transx specifies the degree of freedom allowing translation along the x direction. See 4.3.71 for the application assertion.

4.2.72.5 transy

The transy specifies trhe degree of freedom allowing translation along the y direction. See 4.3.71 for the application assertion.

4.2.72.6 transz

The transz specifies trhe degree of freedom allowing translation along the z direction. See 4.3.71 for the application assertion.

4.2.73 Dm_data_analysis_software

The Dm_data_analysis_software is the software used to acquire and process the measurement points.

The data associated with a Dm_data_analysis_software are the following:

— application_name;

— application_version;

— vendor_or_user_name_name.

4.2.73.1 application_name

The application_name specifies the name of the application software.

4.2.73.2 application_version

The application_version specifies the version of the software used.

4.2.73.3 vendor_or_user_name_name

The vendor_or_user_name name specifies name of the vendor of the software or the user of the software.

4.2.74 Dm_dimension_parameter

A Dm_dimension_parameter is a type of Dm_result_parameter (see 4.2.87), which associates a specified and calculated value of an identified parameter of a dimensional measurement feature.

The data associated with a Dm_Dimension_Parameter are the following:

— calculated_value;

— value_uncertainty.

4.2.74.1 calculated_value

The calculated_value specifies an individual value of a parameter calculated from the measurement points using a specific calculation_method. See 4.3.73 for the application assertion.

4.2.74.2 value_uncertainty

The value_uncertainty specifies the uncertainty of a measurement result is represented by an estimated standard deviation. See 4.3.72 for the application assertion.

4.2.75 Dm_execution_input

A Dm_execution_input is the run information and designed part being inspected.

The data associated with a Dm_execution_input are the following:

— as_designed_part;

— program_run;

— result.

4.2.75.1 as_designed_part

The as_designed_part specifies the designed part being inspected. See 4.3.76 for the application assertion.

4.2.75.2 program_run

The program_run specifies the machine parameters, environmental conditions, and measurement steps used to measure the part. See 4.3.75 for the application assertion.

4.2.75.3 result

The result specifies the result from performing a dimensional inspection. See 4.3.74 for the application assertion.

4.2.76 Dm_execution_result

A Dm_execution_result is the run information and initial data points resulting from a dimensional inspection operation on a part, perhaps after point correction. Data analysis of these points is then performed to determine parameters and compare the parameters to tolerances.

The data associated with a Dm_execution_result are the following:

— dm_measurements;

— program_run.

4.2.76.1 dm_measurements

The dm_measurements specifies the sets of coordinate measurement points that represent the measured surface of a part or of shape_elements of the part. See 4.3.77 for the application assertion.

4.2.76.2 program_run

The program_run specifies the machine parameters, environmental conditions, and measurement steps used to measure the part. See 4.3.78 for the application assertion.

4.2.77 Dm_execution_result_measurement

A Dm_execution_result_measurement is a set of coordinate measurement points that represent the measured surface of a part or of shape_elements of the part.

The data associated with a Dm_execution_result_measurement are the following:

— compensation;

— measurement_points;

— parameter;

— software.

4.2.77.1 compensation

The compensation specifies whether or not coordinate values for the measurement points are compensated.

4.2.77.2 measurement_points

The measurement_points specifies the individual coordinate points describing the surface of a part, each one resulting from a single dimensional inspection surface probing action. In the general case, each measured point is represented by three coordinate values. See 4.3.80 for the application assertion.

4.2.77.3 parameter

The parameter specifies the parameter of a Dm_feature calculated from the set of measurement points. See 4.3.81 for the application assertion.

4.2.77.4 software

The software specifies the software used to acquire and process the measurement data. See 4.3.79 for the application assertion.

4.2.78 Dm_feature

A Dm_feature is a type of Feature (see 4.2.118) that may be usefully described and analyzed by a dimensional measurement procedure. Each Dm_feature is either a Dmf_arc (see 4.2.91), Dmf_-line_unbounded (see 4.2.102), Dmf_circle (see 4.2.92), Dmf_pattern (see 4.2.103), Dmf_cone (see 4.2.93), Dmf_plane (see 4.2.104), Dmf_plane_closed_parallel (see 4.2.105), Dmf_cylinder (see 4.2.94), Dmf_ellipse (see 4.2.96), Dmf_plane_symmetric (see 4.2.106), Dmf_geometric_curve (see 4.2.98), Dmf_point (see 4.2.107), Dmf_geometric_surface (see 4.2.99), Dmf_edge_point (see 4.2.95), Dmf_-sphere (see 4.2.108), Dmf_generic_feature (see 4.2.97), Dmf_line_bounded (see 4.2.100), Dmf_torus (see 4.2.110), Dmf_line_closed_parallel (see 4.2.101), or Dmf_surface_of_revolution_dml (see 4.2.109).

The data associated with a Dm_feature are the following:

— boundary;

4.2.78.1 boundary

The boundary specifies the set of coordinate points that determines the edge limits of a feature. See 4.3.82 for the application assertion.

4.2.79 Dm_feature_analysis_mode_dml

A Dm_feature_analysis_mode_dml is the procedures and algorithms that are to be used by an analysis application for dimensional inspection features. The data associated with a Dm_feature_analysis_mode_-dml are the following:

— axis_method;

— axis_extrapolate;

— aelpr_analysis;

— cccpst_analysis;

— curve_analysis;

— limits_of_size;

— surface_analysis.

4.2.79.1 axis_method

The axis_method specifies a method used to calculate the axis (center) of axial features.

NOTE See 4.2.79.1.1 - 4.2.79.1.3 for the definition of each allowable value for axis_method.

4.2.79.1.1 least_squares: Use a least squares fit on the point data to calculate the feature axis. This is how most CMM analysis packages work.

4.2.79.1.2 min_rad: Minimum Radial Separation. Calculate the maximum cylinder and minimum cylinder which encompasses the measured point data. This method yields a good approximation to physical hard gauges.

NOTE ASME B89.3.1-1972 [3] describes the technique.

4.2.79.1.3 xsect_centers: Calculate the least squares fit for each circular cross section then using the cross section center points calculate a least squares line through them.

4.2.79.2 axis_extrapolate

The axis_extrapolate defines the flag used to indicate if the calculated axis should be extrapolated (projected) so the axis endpoints represent the actual feature endpoints. If the axis_extrapolate is specified as true then the axis endpoints are to be extrapolated to the design length of the axis, if specified as false then the top and bottom cross-section measurements are to be used to calculate the axis endpoints.

NOTE When doing contact measurement of a cylindrical feature (with a CMM) it is not possible to measure the very top of the cylinder. Hence, the possible need for extrapolation of the calculated axis to the endponts.

4.2.79.3 aelpr_analysis

The aelpr_analysis defines the data fitting mode for the features arc, ellipse, line, plane, rectangle

NOTE See 4.2.79.3.1- 4.2.79.3.2 for the definition of each allowable value for aelpr_analysis.

4.2.79.3.1 LSTSQR: Least squares fit

4.2.79.3.2 MINMAX: Minimum maximum two sided fit

4.2.79.4 cccpst_analysis

The cccpst_analysis defines the data fitting mode for the features circle, cone, cylinder, parpln, sphere, torus

NOTE See 4.2.79.4.1 - 4.2.79.4.4 for the definition of each allowable value for cccpst_analysis.

4.2.79.4.1 LSTSQR: Least squares fit

4.2.79.4.2 MINMAX: Minimum maximum two sided fit

4.2.79.4.3 MINCIR: Minimum size circumscribed fit

4.2.79.4.4 MAXINS: Maximum size inscribed fit

4.2.79.5 curve_analysis

The curve_analysis defines feature data fitting mode for curve

NOTE See 4.2.79.5.1 - 4.2.79.5.3 for the definition of each allowable value for curve_analysis.

4.2.79.5.1 BSPLIN: Bi-cubic spline fit

4.2.79.5.2 LSTSQR: Least squares fit

4.2.79.5.3 MINMAX: Minimum maximum two sided fit

4.2.79.6 limits_of_size

The limits_of_size defines the choice of available algorithms when calculating features with diameter, length or width from point data.

EXAMPLE In the case of a cylinder we can calculate a cylinder that is on the outside of all the measured points and one that is on the inside (circumscribed and inscribed).

NOTE See 4.2.79.6.1 - 4.2.79.6.3 for the definition of each allowable value for limits_of_size.

4.2.79.6.1 FUNCTIONAL: Use the functional size. FUNCTIONAL refers to the fact that we use a size that represents the functional use of the feature.

EXAMPLE For a hole (an inner feature) this would be the minimum size calculated. For a pin, it would be the maximum size.

4.2.79.6.2 MRS_AVERAGE: Calculate an average size by adding the minimum and maximum measured sizes and dividing by two.

NOTE This gives results similar to most CMM analysis software.

4.2.79.6.3 TWO_POINT: Indicates both values should be reported in the results. Both values must also fall within the minimum and maximum allowable tolerance for the feature when determining if a tolerance passed or failed.

NOTE This is the most conservative approach.

4.2.79.7 surface_analysis

The surface_analysis defines feature data fitting mode for surface analysis

NOTE See 4.2.79.7.1 - 4.2.79.7.4 for the definition of each allowable value for surface_analysis.

4.2.79.7.1 BEZIER: Bezier spline fit

4.2.79.7.2 LSTSQR: Least squares fit

4.2.79.7.3 MINMAX: Minimum maximum two sided fit

4.2.79.7.4 NURBS: Non-Uniform Rational B-Splines fit

4.2.80 Dm_feature_analysis_mode_default_dml

A Dm_feature_analysis_mode_default_dml is the default parameters to be used if not specifily specified in Dm_feature_analysis_mode_dml. The data associated with a Dm_feature_analysis_mode_default_dml are the following:

— axis_method;

— axis_extrapolate;

— aelpr_analysis;

— cccpst_analysis;

— curve_analysis;

— default_values;

— limits_of_size;

— surface_analysis.

4.2.80.1 axis_method

The axis_method specifies a default method used to calculate the axis (center) of axial features. That method is least squares.

4.2.80.2 axis_extrapolate

The axis_extrapolate defines the default flag used to indicate if the calculated axis should be extrapolated (projected) so the axis endpoints represent the actual feature endpoints. The default is the axis endpoints are to be extrapolated to the design length of the axis.

4.2.80.3 aelpr_analysis

The aelpr_analysis defines thedefault data fitting mode for the features arc, ellipse, line, plane, rectangle. The default is least squares.

4.2.80.4 cccpst_analysis

The cccpst_analysis defines the default data fitting mode for the features circle, cone, cylinder, parpln, sphere, torus. The default is least squares.

4.2.80.5 curve_analysis

The curve_analysis defines feature data fitting mode for curve. The default is least squares.

4.2.80.6 default_values

The default_values specifies the default values for the feature analysis mode parameters. See 4.3.83 for the application assertion.

4.2.80.7 limits_of_size

The limits_of_size defines the choice of available algorithms when calculating features with diameter, length or width from point data. The default is functional.

4.2.80.8 surface_analysis

The surface_analysis defines feature data fitting mode for surface analysis. The default is least squares.

4.2.81 Dm_parameter_analysis_dml

The Dm_parameter_analysis_dml is the collection of possible analysis choices for calculating toleranced parameters and feature parameters from measured data.

The data associated with a Dm_parameter_analysis_dml are the following:

— dofs;

— feature_analysis;

— tolerance_analysis.

4.2.81.1 dofs

The dofs specifies the degrees of freedom available when fitting a feature model to measured data. See 4.3.84 for the application assertion.

4.2.81.2 feature_analysis

The feature_analysis specifies the analysis choices available when calculating parameters of dimensional measurement features from measured data. See 4.3.85 for the application assertion.

4.2.81.3 tolerance_analysis

The tolerance_analysis specifies the analysis choices available when calculating parameters from measured data to compare to tolerances. See 4.3.86 for the application assertion.

4.2.82 Dm_parameter_value_limits

A Dm_parameter_value_limits is the range of possible values calculated for a measured parameter.

The data associated with a Dm_parameter_value_limits are the following:

— calculated_limits;

— limits_method.

4.2.82.1 calculated_limits

The calculated_limits specifies the values of limits on the range of a measured dimensional parameter. See 4.3.87 for the application assertion.

4.2.82.2 limits_method

The limits_method specifies the approach by which the calculated limits are obtained.

4.2.83 Dm_point

A Dm_point is a cartesian point measured or expected on the surface of a part.

The data associated with a Dm_Point are the following:

— expected_point;

— measured_point.

4.2.83.1 expected_point

The expected_point specifies a point calculated beforehand to be an approximate position of a measured point. See 4.3.88 for the application assertion.

4.2.83.2 measured_point

The measured_point specifies a data point acquired by a probing measurement or a simulation. See 4.3.89 for the application assertion.

4.2.84 Dm_point_parameter

A Dm_point_parameter is a type of Dm_result_parameter (see 4.2.87) that definies a point parameter of a Dm_feature calculated from measured or simulated data.

The data associated with a Dm_point_parameter are the following:

— calculated_point;

— value_uncertainty.

4.2.84.1 calculated_point

The calculated_point specifies a Cartesian point having two or three coordinates. See 4.3.89 for the application assertion.

4.2.84.2 value_uncertainty

The value_uncertainty specifies The uncertainty of the result of a measurement reflects the lack of exact knowledge of the value of the measurment and thus the result is complete only when accompanied by a quantitative statement of its uncertainty. See 4.3.90 for the application assertion.

4.2.85 Dm_program_identification

A Dm_program_identification is the protocol or set of measurement process steps used to acquire a set of measured data points.

The data associated with a Dm_program_identification are the following:

— angular_units_dml;

— identifier;

— linear_units_dml;

— program_custodian;

— tolerance_standard_dml;

— version;

4.2.85.1 angular_units_dml

The angular_units_dml specifies the physical units to be used when describing the size of an angle.

EXAMPLE Two examples of angular units are are rad (radians) and degrees.

4.2.85.2 identifier

The identifier specifies the name or number of the protocol used to acquire a set of measured data points.

4.2.85.3 linear_units_dml

The linear_units_dml specifies the physical units to be used when describing a length quantity.

EXAMPLE Examples are mm or m.

4.2.85.4 program_custodian

The program_custodian specifies the person responsible for maintaining and modifying the program. See 4.3.91, and 4.3.92 for the application assertion.

4.2.85.5 tolerance_standard_dml

The tolerance_standard_dml specifies the documentary tolerancing standard that is being used for the specification of part dimensions and form.

EXAMPLE Examples are ISO 1101 or ASME Y14.5.

4.2.85.6 version

The version specifies the version of the protocol, usually represented by a numbered level program.

4.2.86 Dm_program_run

The Dm_program_run is the program used by the measuring machine to produce the dimensional measurements data.

The data associated with a Dm_Program_Run are the following:

— measurement_temperature;

— measurement_humidity;

— part_inspection_status;

— program_id;

— run_administrator;

— run_date;

— run_end;

— run_location;

— run_start;

— workpiece_temperature;

— url.

4.2.86.1 measurement_temperature

The measurement_temperature specifies the temperature measured in the laboratory when the run was performed. The measurement_temperature may be in Celsius or Fahrenheit. See 4.3.97 for the application assertion.

4.2.86.2 measurement_humidity

The measurement_humidity specifies the humidity measured in the laboratory when the run was performed See 4.3.97 for the application assertion.

4.2.86.3 part_inspection_status

The part_inspection_status specifies the status of inspecting the part for this particular program run. The statuses may be pass, fail, rework, error, unknown, or not_calculated.

4.2.86.4 program_id

The program_id specifies a unique identifier for a run within a batch of programs. See 4.3.95 for the application assertion.

4.2.86.5 run_administrator

The run_administrator specifies the person in charge of making the run. See 4.3.98 for the application assertion.

4.2.86.6 run_date

The run_date specifies the date and time that a particular program was run. See 4.3.94, 4.3.93 for the application assertion.

4.2.86.7 run_end

The run_end specifies ending date and time of the inspection program being run. See 4.3.94 and 4.3.93 for the application assertion.

4.2.86.8 run_location

The run_location specifies location where the inspection program was or is to be executed.. See 4.3.96 for the application assertion.

4.2.86.9 run_start

The run_start specifies begining date and time of the inspection program being run. See 4.3.94 and 4.3.93 for the application assertion.

4.2.86.10 workpiece_temperature

The workpiece_temperature specifies the temperature measured on the workpiece when the run was performed. The measurement_temperature may be in Celsius or Fahrenheit See 4.3.97 for the application assertion.

4.2.86.11 url

The url specifies the system and directory and file where the program is stored, expressed in html format.

4.2.87 Dm_result_parameter

A Dm_result_parameter is a specified parameter of a dimensional measurement feature or one that is obtained by measurement and calculation. A Dm_result_parameter is either a Dm_dimension_parameter (see 4.2.74), Dm_point_parameter (see 4.2.84), or a Dm_vector_parameter (see 4.2.90).

The data associated with a Dm_result_parameter are the following:

— analysis_conditions;

— calculation_method;

— limits;

— specification.

4.2.87.1 analysis_conditions

The analysis_conditiions specifies the choices of conditions under which dimensional measurements are analyzed and results are calculated. See 4.3.99 for the application assertion.

4.2.87.2 calculation_method

The calculation_method specifies the algorithm used to derive a particular measurement parameter.

4.2.87.3 limits

The limits specifies the range of values that the calculated parameter can have based on the measurements and calculation algorithm. See 4.3.100 for the application assertion.

4.2.87.4 specification

The specification specifies a dimensional parameter of a part with which the calculated_value may be compared. The specified value may be a geometric dimension with its tolerance or a geometric tolerance. See 4.3.104, 4.3.101, 4.3.103 and 4.3.104 for the application assertion.

4.2.88 Dm_tolerance_analysis_mode_default_dml

A Dm_tolerance_analysis_mode_default_dml is the default parameters to be used if not specifily specified in Dm_tolerance_analysis_mode_dml.

The data associated with a Dm_tolerance_analysis_mode_default_dml are the following:

— method;

— option;

— setting.

4.2.88.1 method

The method specifies how the reference line for a surface profile tolerance is calculated. The default is standard.

4.2.88.2 option

Option specifies the choices of analysis method for calculating a datum plane. The default is best fit.

4.2.88.3 setting

The setting specifies the position of the reference line or curve with respect to the measured surface data points for a surface profile tolerance. The default is standard.

4.2.89 Dm_tolerance_analysis_mode_dml

The Dm_tolerance_analysis_mode_dml is the procedures and algorithms that are to be used by an analysis application for dimensional inspection tolerances.

The data associated with a Dm_tolerance_analysis_mode_dml are the following:

— default_values;

— method;

— option;

— setting.

4.2.89.1 default_values

The default_values specifies the default values for the feature analysis mode parameters. See 4.3.105 for the application assertion.

4.2.89.2 method

The method specifies how the reference line for a surface profile tolerance is calculated.

NOTE See 4.2.89.2.1 - 4.2.89.2.4 for the definition of each allowable value for method.

4.2.89.2.1 STANDARD: This setting minimizes the maximum normal deviation when performing best fit calculations for surface profile tolerances. For each measurement point the normal deviation is the distance between the measured and design points after the measured point is projected onto surface along a surface normal vector. The strength of this setting is that the worst point takes precedence and therefore the best fit is directly affected by the worst normal deviation.

4.2.89.2.2 MINDEV_3D: This setting minimizes the maximum 3D distance between the designed point and the measurement point for all points included in the surface profile tolerance

4.2.89.2.3 LSQ_NRM: This setting minimizes the least squares normal deviation for all measurement points when doing a best fit.

4.2.89.2.4 LSQ_3D: This setting minimizes the least squares 3D distance between the measurement points and the designed points when doing a best fit

4.2.89.3 option

Option specifies the choices of analysis method for calculating a datum plane.

NOTE See 4.2.89.3.1- 4.2.89.3.2 for the definition of each allowable value for method.

4.2.89.3.1 BEST_FIT: The datum plane is calculated using a least squares fit of the points on the datum feature.

4.2.89.3.2 HIGH_POINT: The datum plane is defined by the three highest points on the datum feature because these points determine where the datum surface will rest against a mating part.

4.2.89.4 setting

The setting specifies the position of the reference line or curve with respect to the measured surface data points for a surface profile tolerance.

NOTE See 4.2.89.4.1 - 4.2.89.4.3 for the definition of each allowable value for method.

4.2.89.4.1 STANDARD: The reference line is calculated using least squares fitting

4.2.89.4.2 INNER: The reference line lies just inside the material.

4.2.89.4.3 OUTER: The reference line lies just outside the material

4.2.90 Dm_vector_parameter

A Dm_vector_parameter is a type of Dm_result_parameter (see 4.2.87) that definies a vector parameter of a Dm_feature or a Tolerance calculated from measured or simulated data.

The data associated with a Dm_vector_parameter are the following:

— calculated_vector;

— value_uncertainty.

4.2.90.1 calculated_vector

The calculated_vector specifies a vector in Cartesian space having two or three Cartesian components. See 4.3.106 for the application assertion.

4.2.90.2 value_uncertainty

The value_uncertainty specifies The uncertainty of the result of a measurement reflects the lack of exact knowledge of the value of the measurment and thus the result is complete only when accompanied by a quantitative statement of its uncertainty. See 4.3.107 for the application assertion.

4.2.91 Dmf_arc

A Dmf_arc is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the measurement parameters of an arc. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a arc may be found in ISO 10303-42.

NOTE 2 Figure 44 illustrates the Dmf_arc.

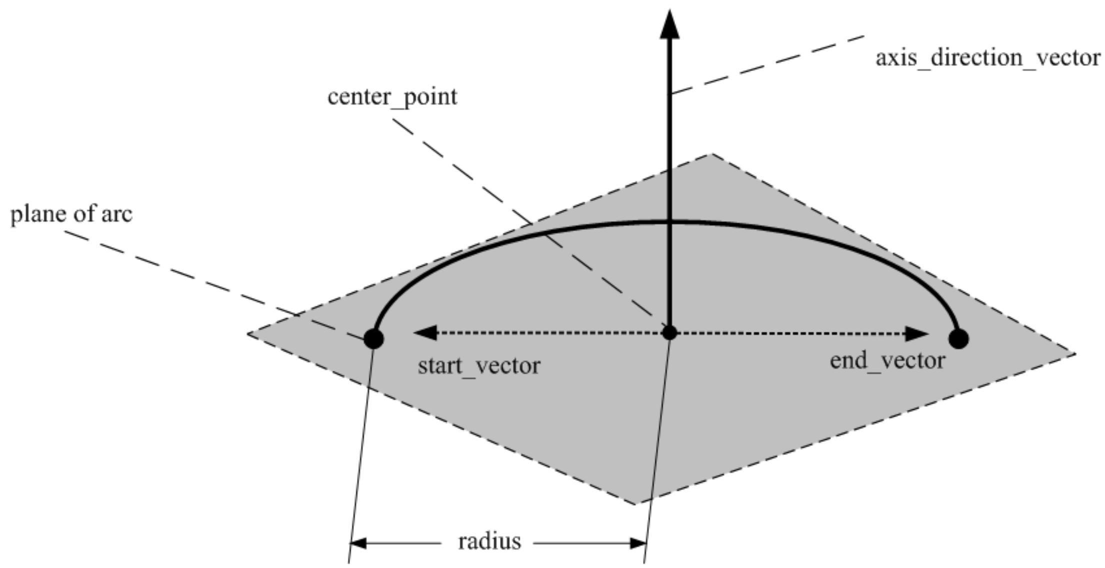


Figure 44 — Dmf_arc

The data associated with a Dmf_arc are the following:

— axis_direction_vector;

— center_point;

— end_vector;

— inner_outer;

— radius;

— start_vector.

4.2.91.1 axis_direction_vector

The axis_direction_vector specifies the vector describing the normal to the plane of the arc and passing through the center point of the arc. See 4.3.110 for the application assertion.

4.2.91.2 center_point

The center_point specifies the point from which the points on the arc periphery are equidistant. See 4.3.109for the application assertion.

4.2.91.3 end_vector

The end_vector specifies the vector directed from the center point of the arc to determine the end point on the arc periphery. See 4.3.110 for the application assertion.

4.2.91.4 inner_outer

The inner_outer specifies whether the part material is outside or inside the arc periphery.

4.2.91.5 radius

The radius specifies the distance from the center point of the arc to the arc periphery. See 4.3.108 for the application assertion.

4.2.91.6 start_vector

The start_vector specifies the vector directed from the center point of the arc to determine the starting point on the arc periphery. See 4.3.110 for the application assertion.

4.2.92 Dmf_circle

A Dmf_circle is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the measurement parameters of a circle. The parameters are derived from measured or simulated coordinate data points.

NOTE 2 Related concept for a circle may be found in ISO 10303-42.

NOTE 2 Figure 45 illustrates the Dmf_circlec.

The data associated with a Dmf_circle are the following:

— axis_direction_vector;

— center_point;

— diameter;

— inner_outer.

4.2.92.1 axis_direction_vector

The axis_direction_vector specifies a vector describing the normal to the plane of the circle and passing through the center point of the circle. See 4.3.113for the application assertion.

4.2.92.2 center_point

The point inside the circle from which all points on the circle are equidistant. See 4.3.112 for the application assertion.

4.2.92.3 diameter

The diameter specifies twice the radius of the circle; the length of a straight line drawn from any point on the circle periphery to the opposite point on the circle periphery such that it passes through the center point. See 4.3.111 for the application assertion.

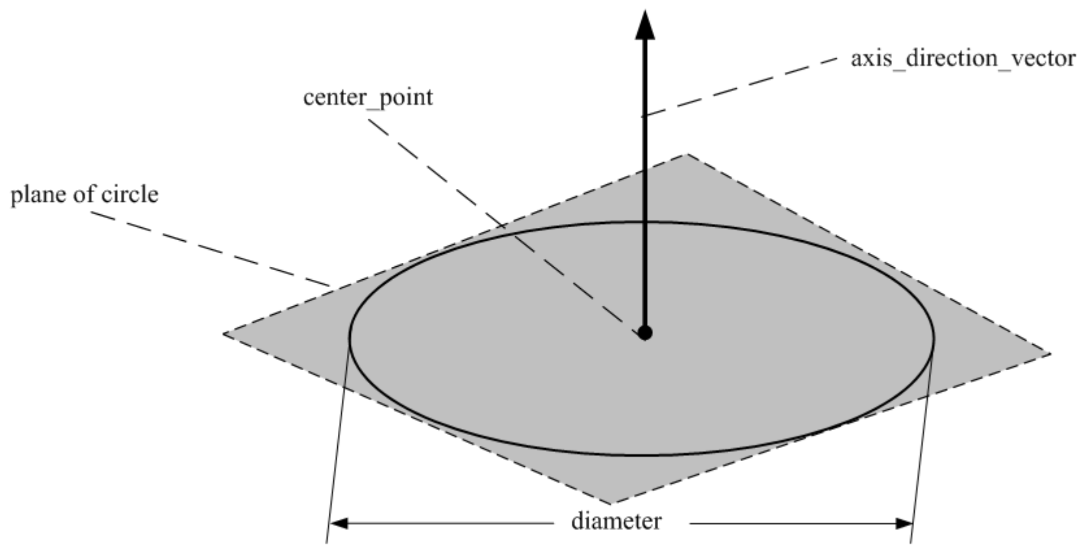


Figure 45 — Dmf_circle

4.2.92.4 inner_outer

The inner_outer specifies whether the part material is inside or outside the circle periphery.

4.2.93 Dmf_cone

A Dmf_cone is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a conical surface. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a cone may be found in ISO 10303-42.

NOTE 2 Figure 46 illustrates the Dmf_cone.

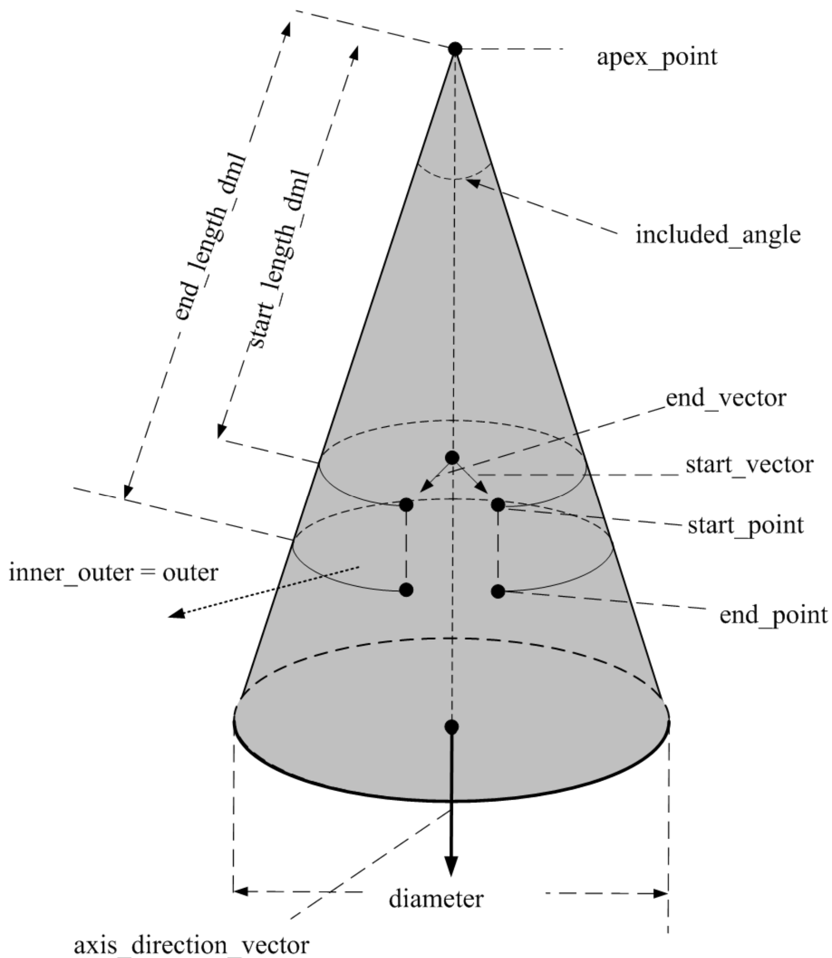


Figure 46 — Dmf_cone

The data associated with a Dmf_cone are the following:

— apex_point;

— axis_direction_vector;

— diameter;

— end_length_dml;

— end_point;

— end_vector;

— included_angle;

— inner_outer;

— start_length_dml;

— start_point;

— start_vector.

4.2.93.1 apex_point

The apex_point specifies the point on the axis of the cone from which the conical surface diverges. See 4.3.115 for the application assertion.

4.2.93.2 axis_direction_vector

The axis_direction_vector specifies the direction of the axis of the cone. See 4.3.116 for the application assertion.

4.2.93.3 diameter

The diameter specifies either the diameter specified at the large end of the cone as derived from a specification or the calculated diameter derived from measured data. See 4.3.114 for the application assertion.

4.2.93.4 end_length_dml

The end_length_dml specifies the length from the cone apex point to the bottom plane See 4.3.114 for the application assertion.

4.2.93.5 end_point

The end_point specifies for cones with bounded length, the point that represents the end point of the conical surface. See 4.3.115for the application assertion.

4.2.93.6 end_vector

The end_vector specifies for cones with bounded periphery, (less than 360°), the vector directed from the axis of the cone to the ending point on the cone periphery. See 4.3.116 for the application assertion.

4.2.93.7 included_angle

The included angle specifies the full apex angle of the cone See 4.3.114 for the application assertion.

4.2.93.8 inner_outer

The inner_outer specifies whether the part material is outside or inside the cone periphery.

4.2.93.9 start_length_dml

The start_length_dml specifies the length from the cone apex_point to the top plane if the cone is truncated. This only needs to be supplied if the cone is truncated. See 4.3.114 for the application assertion.

4.2.93.10 start_point

The start_point specifies for cones with bounded length, the point that represents the start point of the conical surface. See 4.3.115 for the application assertion.

4.2.93.11 start_vector

The start_vector specifies for cones with bounded periphery, (less than 90°) the vector directed from the axis of the cone to the starting point on the cone periphery. See 4.3.116 for the application assertion.

4.2.94 Dmf_cylinder

A Dmf_cylinder is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a cylindrical surface. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a cylindrical surface may be found in ISO 10303-42.

NOTE 2 Figure 47 illustrates the Dmf_cylinder.

The data associated with a Dmf_Cylinder are the following:

— axis_direction_vector;

— diameter;

— center_point;

— end_point;

— inner_outer;

— length_dml;

— start_point.

4.2.94.1 axis_direction_vector

The axis_direction_vector specifies the direction of the axis of the cylinder. See 4.3.119 for the application assertion.

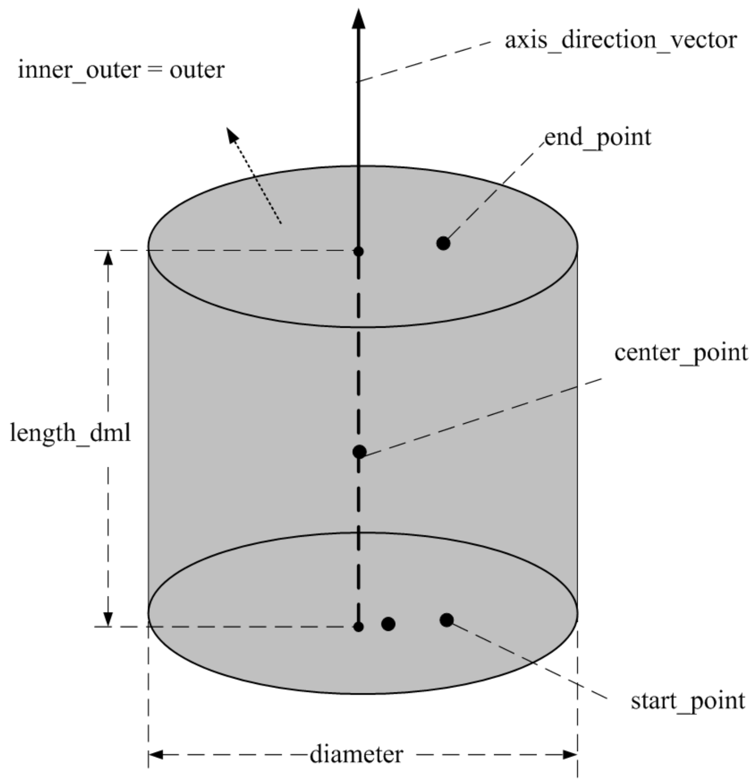


Figure 47 — Dmf_cylinder

4.2.94.2 center_point

The center_point specifies the point on the axis direction vector indicating the location of the center of the cylinder surface. See 4.3.118 for the application assertion.

4.2.94.3 diameter

The diameter specifies twice the radius of the cylinder. See 4.3.117 for the application assertion.

4.2.94.4 end_point

The end_point specifies for cylinders with bounded length, a point that defines the extreme end point of the cylindricals urface. See 4.3.118 for the application assertion.

4.2.94.5 inner_outer

The inner_outer specifies whether the part material is outside or inside the cylinder surface.

4.2.94.6 length_dml

The length_dml specifies the end-to-end length of the cylinder along its axis. If length_dml is not specified, the cylinder is unbounded. See 4.3.117 for the application assertion.

4.2.94.7 start_point

The start_point specifies for cylinders with bounded length, a point that defines the extreme start point of the cylindrical surface. See 4.3.118 for the application assertion.

4.2.95 Dmf_edge_point

A Dmf_edge_point is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a point on the edge of a surface. The parameters are derived from measured or simulated coordinate data points

NOTE 1 This feature has a function equivalent to the DMIS word FEAT/EDGEPT.

NOTE 2 Figure 48 illustrates the Dmf_edge_point.

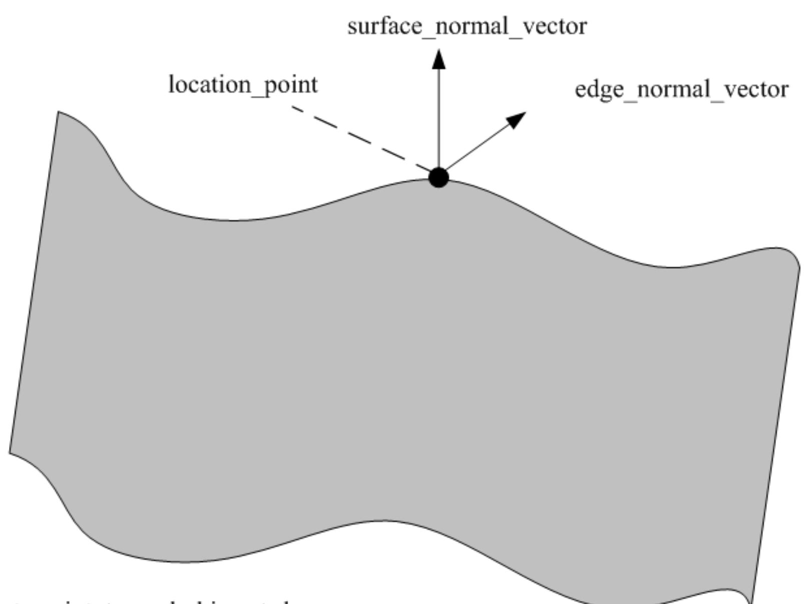
The data associated with a Dmf_edge_point are the following:

— edge_normal_vector;

— location_point;

— point_type_dml.

— surface_normal_vector.



NOTE The attribute point_type_dml is not shown

Figure 48 — Dmf_edge_point

4.2.95.1 edge_normal_vector

The edge_normal_vector specifies the vector normal to and pointing away from the edge in which the edge point lies. See 4.3.120 for the application assertion.

4.2.95.2 location_point

The location_point specifies the coordinates of the edge point itself. See 4.3.121 for the application assertion.

NOTE defined in DMIS

4.2.95.3 point_type_dml

The point_type_dml specifies the type of finish of the edge on which the point is found.

NOTE See 4.2.95.3.1 - 4.2.95.3.3 for the definition of each allowable value for point_type_dml.

4.2.95.3.1 point: No finish specified.

4.2.95.3.2 hedge: Specifies a hemmed edge.

4.2.95.3.3 tedge: Specifies a trimmed edge.

4.2.95.4 surface_normal_vector

The surface_normal_vector specifies the vector normal to and pointing away from the surface adjacent to the edge in which the edge point lies. See 4.3.120 for the application assertion.

4.2.96 Dmf_ellipse

A Dmf_ellipse is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of an ellipse. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a ellipse may be found in ISO 10303-42.

NOTE 2 Figure 49 illustrates the Dmf_ellipse.

The data associated with a Dmf_ellipse are the following:

— center_point;

— foucus_point_one_dml;

— focus_point_two_dml;

— inner_outer;

— major_axis_vector;

— major_diameter;

— minor_diameter;

— normal_direction.

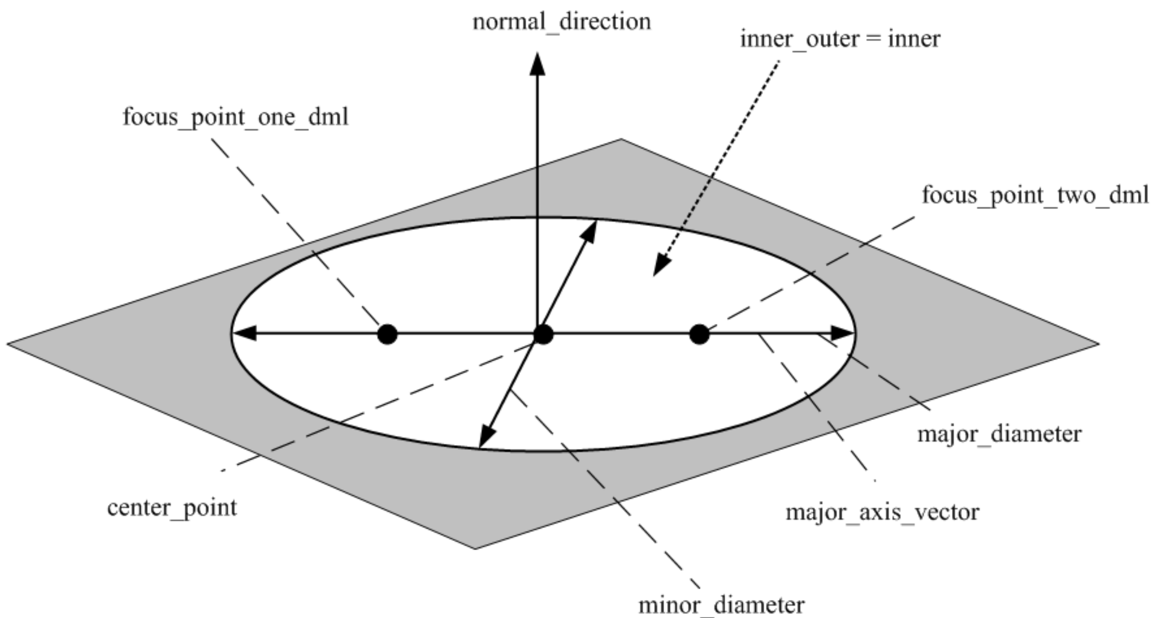


Figure 49 — Dmf_ellipse

4.2.96.1 center_point

The center_point specifies the center of the ellipse. See 4.3.123 for the application assertion.

4.2.96.2 focus_point_one_dml

The focus_point_one_dml specifies the x,y,z coordinates of focus point one of the ellipse. See 4.3.123 for the application assertion.

4.2.96.3 focus_point_two_dml

The focus_point_two_dml specifies the x,y,z coordinates of focus point two of the ellipse. See 4.3.123 for the application assertion.

4.2.96.4 inner_outer

The inner_outer specifies whether the part material is outside or inside the curve of the ellipse.

4.2.96.5 major_axis_vector

The major_axis_vector specifies the vector aligned with the major axis vector of the ellipse. See 4.3.124 for the application assertion.

4.2.96.6 major_diameter

The major_diameter specifies the distance across the ellipse along the major (longer) axis. See 4.3.122 for the application assertion.

4.2.96.7 minor_diameter

The minor_diameter specifies the distance across the ellipse along the minor (shorter) axis. See 4.3.122 for the application assertion.

4.2.96.8 normal_direction

The normal_directionspecifies the vector normal to the plane of the ellipse. See 4.3.124 for the application assertion.

4.2.97 Dmf_generic_feature

A Dmf_generic_feature is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to represent a generic feature of arbitrary shape. The Dmf_generic_feature is constructed from measured or simulated coordinate data points.

NOTE Figure 50 illustrates the Dmf_generic_feature.

The data associated with a Dmf_generic_feature are the following:

— center_point;

— description;

— parameters;

— secondary_vector;

— vector_from_center_of_object.

4.2.97.1 center_point

The center_point specifies the coordinate point that determines the nominal center of the generic feature. See 4.3.127 for the application assertion.

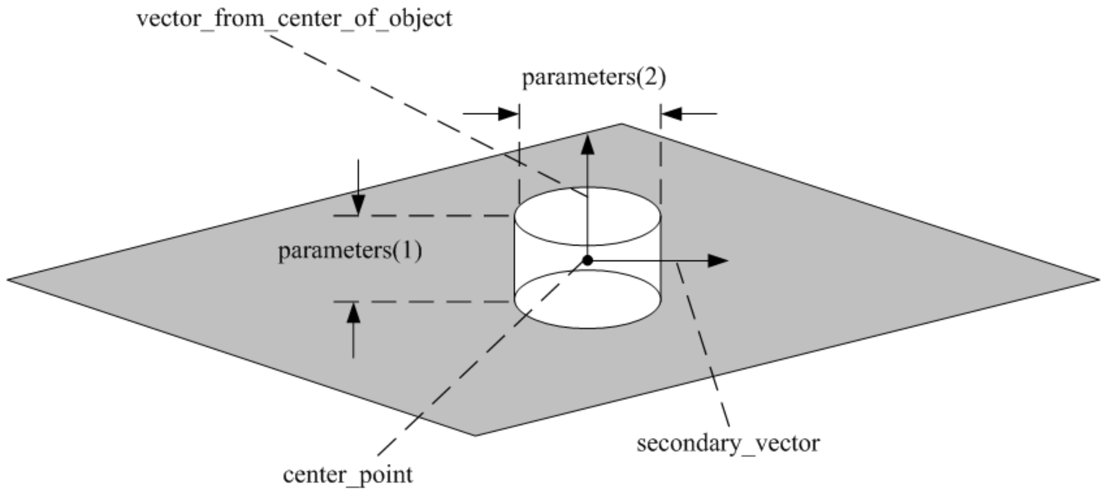


Figure 50 — Dmf_generic_feature

4.2.97.2 description

The description specifies the text that carries information about the generic feature.

4.2.97.3 parameters

The parameters specifies the parameters that determine the shape of the generic feature. See 4.3.125 for the application assertion.

4.2.97.4 secondary_vector

The secondary_vector specifies the vector that determines the azimuthal orientation of the feature with respect to the polar vector, vector_from_center_of_object. See 4.3.126 for the application assertion.

4.2.97.5 vector_from_center_of_object

The vector_from_center_of_object specifies the vector that determines polar orientation of the feature in space. See 4.3.126 for the application assertion.

4.2.98 Dmf_geometric_curve

A Dmf_geometric_curve is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to represent a geometric curve of arbitrary shape but confined to a plane. The Dmf_geometric_curve is constructed from measured or simulated coordinate data points.

NOTE 1 The information associated with this entity is similar to that contained in the DMIS word FEAT/GCURVE.

NOTE 2 Figure 51 illustrates the Dmf_geometric_curve.

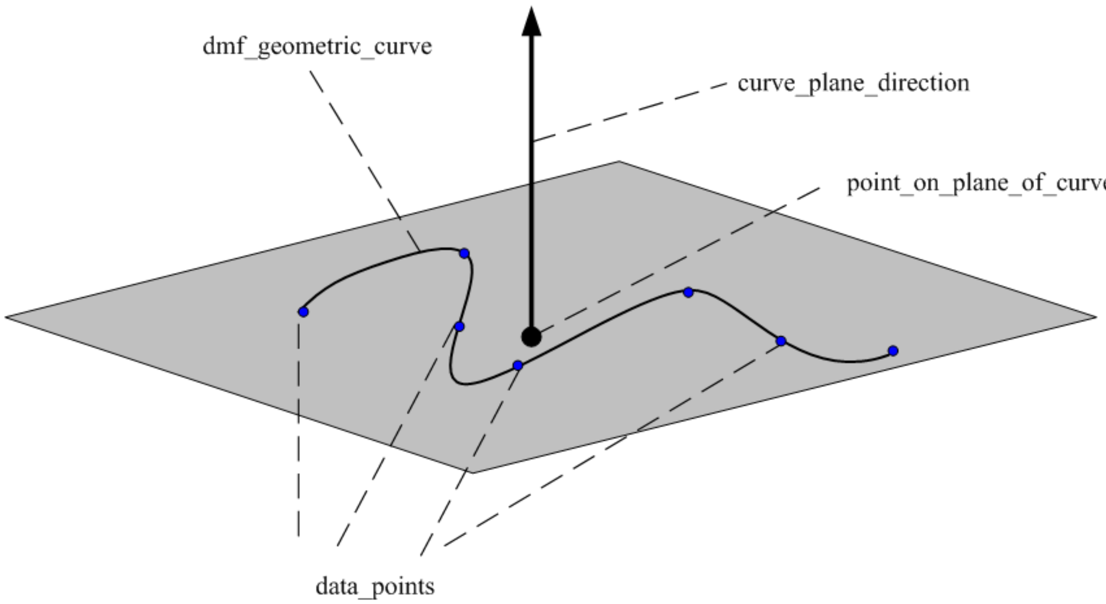


Figure 51 — Dmf_geometric_curve

The data associated with a Dmf_geometric_curve are the following:

— curve_plane_direction;

— data_points;

— point_on_plane_of_curve.

4.2.98.1 curve_plane_direction

The curve_plane_direction specifies the normal direction vector to the plane in which the curve lies. See 4.3.128 for the application assertion.

4.2.98.2 data_points

The data_points specifies the set of coordinate data points that make up the description of the curve. See 4.3.130 for the application assertion.

4.2.98.3 point_on_plane_of_curve

The point_on_plane_of_curve specifies the coordinates of a point in the plane on which the curve lies. See 4.3.129 for the application assertion.

4.2.99 Dmf_geometric_surface

A Dmf_geometric_surface is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to represent a generic surface of arbitrary shape. The Dmf_geometric_surface is constructed from measured or simulated coordinate data points.

NOTE Figure 52 illustrates the Dmf_geometric_surface.

The data associated with a Dmf_geometric_surface are the following:

— data_points;

— local_surface_normal;

— point_on_surface.

4.2.99.1 data_points

The data_points specifies a series of coordinate points that determine the surface. See 4.3.133 for the application assertion.

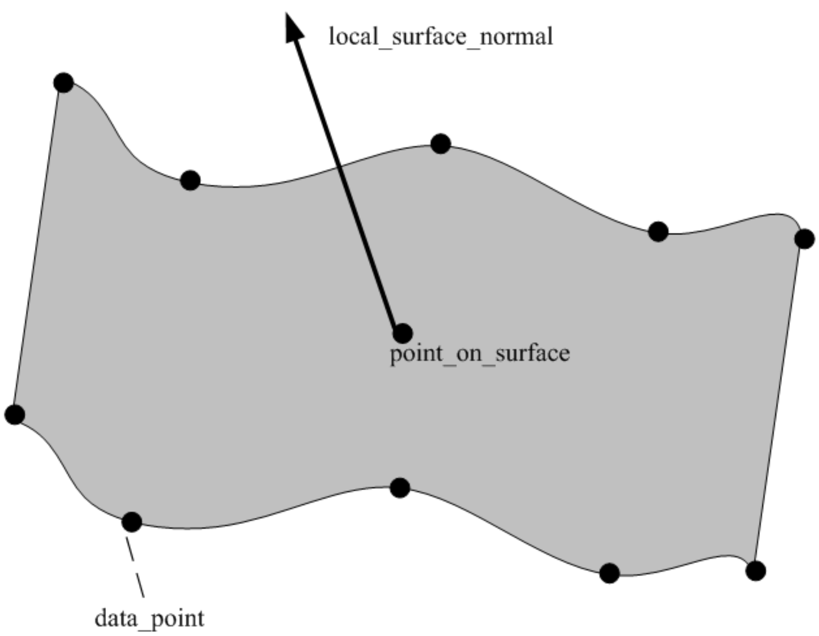


Figure 52 — Dmf_geometric_surface

4.2.99.2 local_surface_normal

The local_surface_normal specifies the normal direction vector to the surface at the point_on_surface See 4.3.132 for the application assertion.

4.2.99.3 point_on_surface

The point_on_surface specifies the coordinates of a particular point lying on the surface associated with a local_surface_normal direction.. See 4.3.131 for the application assertion.

4.2.100 Dmf_line_bounded

A Dmf_line_bounded is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a line bounded by end points. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a bounded line may be found in ISO 10303-42.

NOTE 2 Figure 53 illustrates the Dmf_line_bounded.

The data associated with a Dmf_line_bounded are the following:

— first_end_point;

— legth_dml;

— second_end_point;

— surface_approach_vector;

— vector_dml.

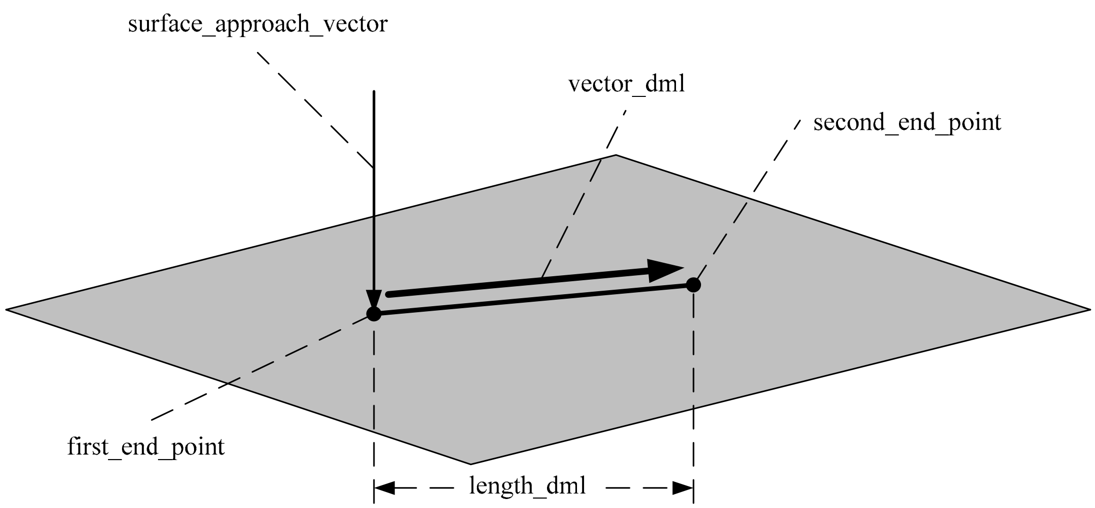


Figure 53 — Dmf_line_bounded

4.2.100.1 first_end_point

The first_end_point specifies the point that determines the beginning of the bounded line. See 4.3.134 for the application assertion.

4.2.100.2 length_dml

The length_dml specifies the distance along the line between the first end point and the second end point. See 4.3.136 for the application assertion.

4.2.100.3 second_end_point

The second_end_point specifies the point that determines the end of the bounded line. See 4.3.134 for the application assertion.

4.2.100.4 surface_approach_vector

The surface_approach_vector specifies for probing of a line on a physical part, specifies the direction from which to approach in order to avoid interference with the part itself or with other interfering components. See 4.3.135 for the application assertion.

4.2.100.5 vector_dml

The vector_dml specifies the direction vector along the line. See 4.3.135 for the application assertion.

4.2.101 Dmf_line_closed_parallel

A Dmf_line_closed_parallel is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a closed ended parallel line feature. The parameters are derived from measured or simulated coordinate data points.

NOTE This feature is defined in DMIS by the FEAT/CPARLN word, primarily to describe straight slots in sheet metal.

NOTE 2 Figure 54 illustrates the Dmf_line_closed_parallel.

The data associated with a Dmf_line_closed_parallel are the following:

— axis_direction_vector;

— center_point;

— end_kind;

— inner_outer;

— feature_length;

— longitude_vector;

— width.

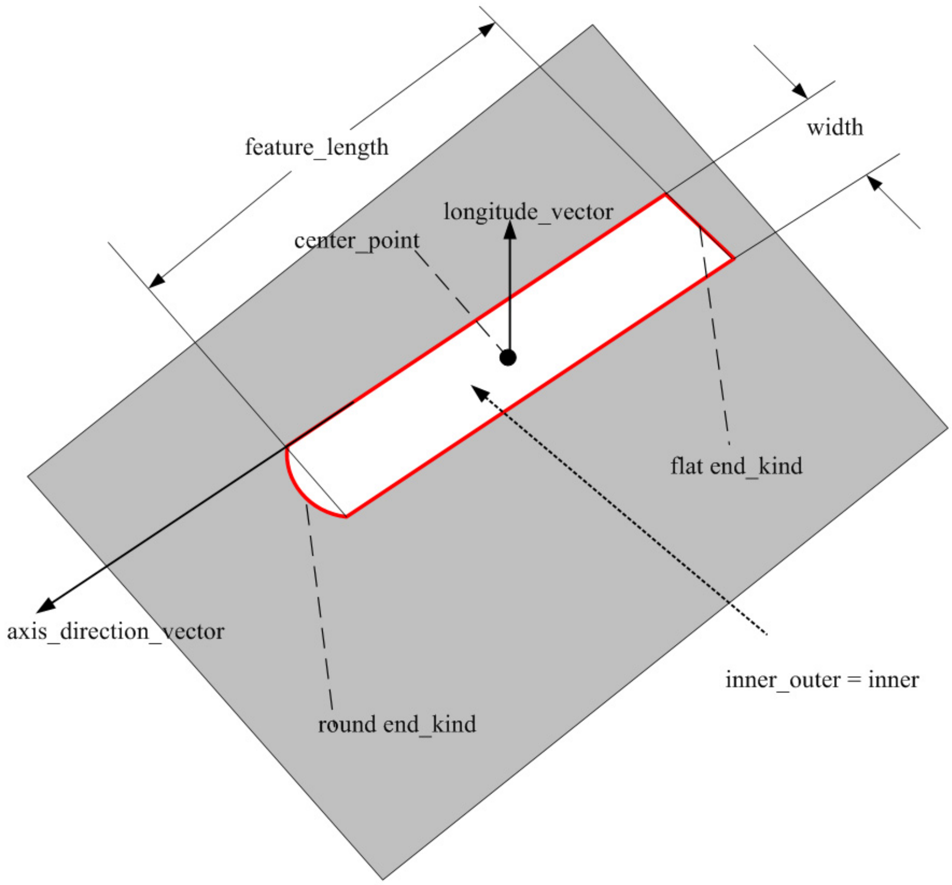


Figure 54 — Dmf_line_closed_parallel

4.2.101.1 axis_direction_vector

The axis_direction_vector specifies the direction vector of the parallel lines. See 4.3.139 for the application assertion.

4.2.101.2 center_point

The center_point specifies the center point of the closed parallel line feature. See 4.3.138 for the application assertion.

4.2.101.3 end_kind

The end_kind specifies whether the closed end is flat or round

4.2.101.4 inner_outer

The inner_outer specifies whether the solid material is outside or inside the Dmf_line_closed_parallel, that is, whether it is a slot or a boss feature.

4.2.101.5 feature_length

The feature_length specifies the length of the parallel lines between the closed ends. See 4.3.137 for the application assertion.

4.2.101.6 longitude_vector

The longitude_vector specifies the vector perpendicular to the plane of the parallel lines. See 4.3.139 for the application assertion.

4.2.101.7 width

The width specifies the width between the parallel lines. See 4.3.137 for the application assertion.

4.2.102 Dmf_line_unbounded

A Dmf_line_unbounded is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a line with no bounding end points. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a unbounded line may be found in ISO 10303-42.

NOTE 2 Figure 55 illustrates the Dmf_line_unbounded.

The data associated with a Dmf_line_unbounded are the following:

— direction_vector;

— point_on_line;

— surface_approach_vector.

4.2.102.1 direction_vector

The direction_vector specifes the vector that determines the direction of the unbounded line. See 4.3.140 for the application assertion.

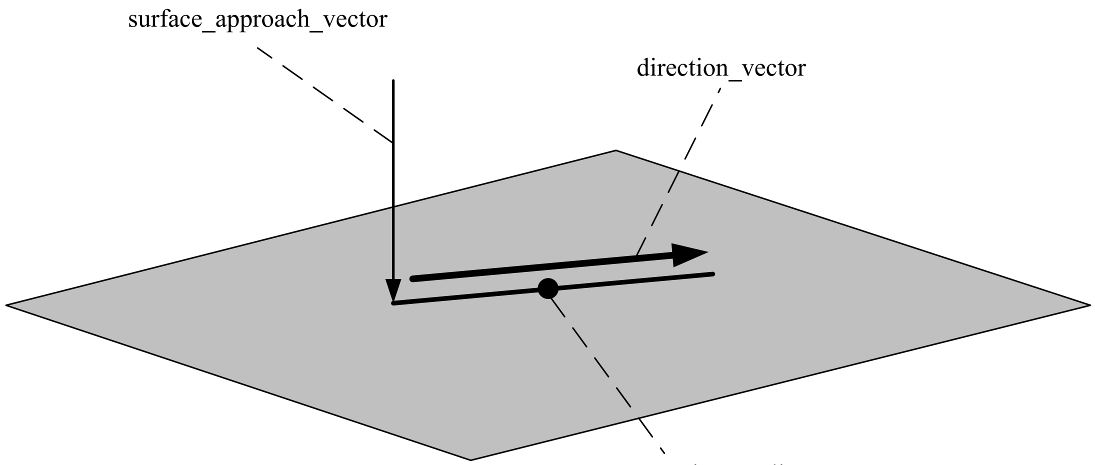


Figure 55 — Dmf_line_unbounded

4.2.102.2 point_on_line

The point_on_line specifies a point on the line that anchors the position of the line. See 4.3.141 for the application assertion.

4.2.102.3 surface_approach_vector

The surface_approach_vector specifies for probing of a line on a physical part, specifies the direction from which to approach in order to avoid interference with the part itself or with other interfering component. See 4.3.140 for the application assertion.

4.2.103 Dmf_pattern

A Dmf_pattern is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a pattern of features. The parameters are derived from measured or simulated coordinate data points.

NOTE Figure 56 illustrates the Dmf_pattern.

The data associated with a Dmf_pattern are the following:

— direction_vector;

— features;

— point.

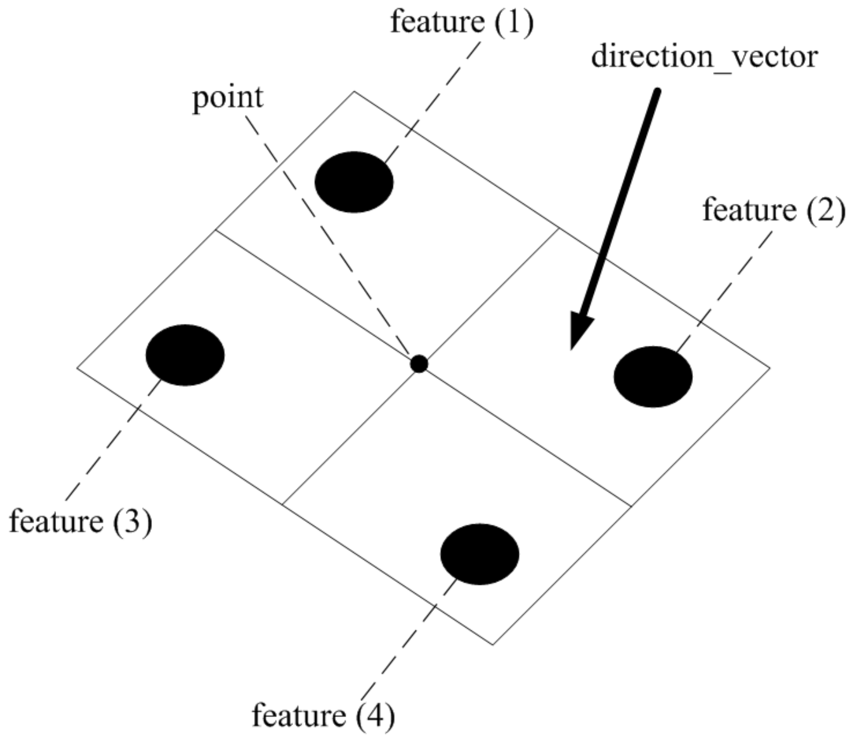


Figure 56 — Dmf_pattern

4.2.103.1 direction_vector

The direction_vector specifies the approach vector for probing the features of the pattern. See 4.3.144 for the application assertion.

4.2.103.2 features

The features specifies the set of features that make up the pattern. See 4.3.142 for the application assertion.

4.2.103.3 point

The point specifies the point determining an origin or center of symmetry for a relative coordinate system in which the pattern may be specified. See 4.3.143 for the application assertion.

4.2.104 Dmf_plane

A Dmf_plane is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a plane. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a plane may be found in ISO 10303-42.

NOTE 2 Figure 57 illustrates the Dmf_plane.

The data associated with a Dmf_plane are the following:

— direction_vector;

— point_on_plane.

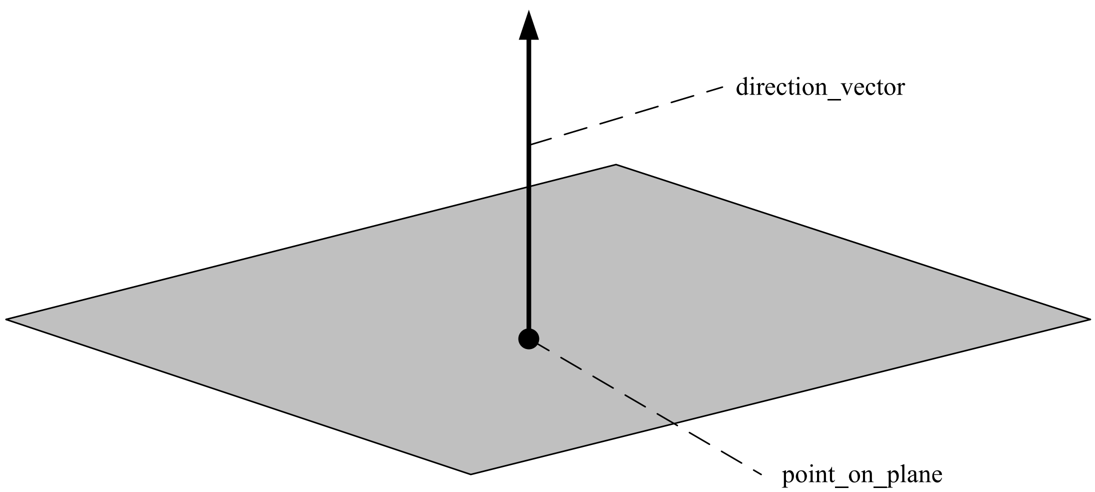


Figure 57 — Dmf_plane

4.2.104.1 direction_vector

The direction_vector specifies the normal vector to the plane. See 4.3.146 for the application assertion.

4.2.104.2 point_on_plane

The point_on_plane specifies a point on the plane that anchors the position of the plane. See 4.3.145 for the application assertion.

4.2.105 Dmf_plane_closed_parallel

A Dmf_plane_closed_parallel is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a closed parallel plane. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 This feature is used primarily to describe machined slots and is an extension of the FEAT/PARPLN statement in DMIS with more attributes.

NOTE 2 Figure 58 illustrates the Dmf_plane_closed_parallel.

The data associated with a Dmf_plane_closed_parallel are the following:

— axis_direction_vector;

— center_point;

— end_kind;

— feature_length;

— height;

— inner_outer;

— longitude_vector;

— normal_dml;

— width.

4.2.105.1 axis_direction_vector

The axis_direction_vector specifies the direction vector of the parallel planes extending from one closed end to the other. See 4.3.149 for the application assertion.

4.2.105.2 center_point

The center_point specifies the center point of the closed parallel plane feature. See 4.3.148 for the application assertion.

4.2.105.3 end_kind

The end_kind specifies whether the closed ends are flat or round.

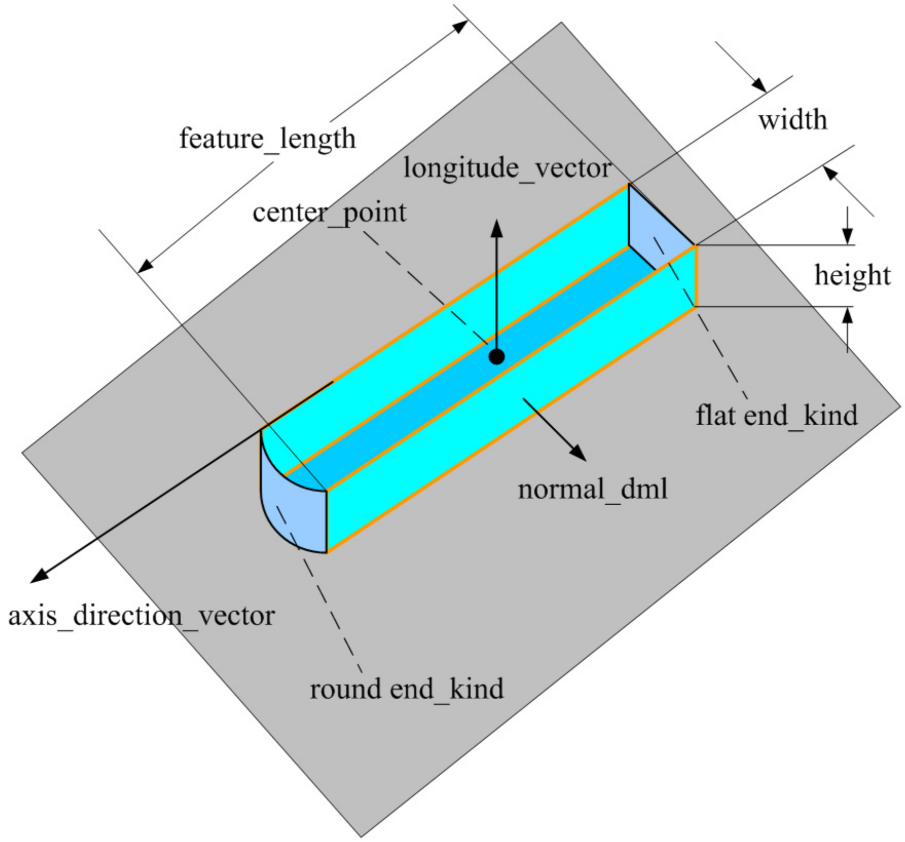


Figure 58 — Dmf_plane_closed_parallel

4.2.105.4 feature_length

The feature_length specifies the length of the parallel planes between the closed ends. See 4.3.147 for the application assertion.

4.2.105.5 height

The height specifies the height of the parallel plane feature along the longitude_vector. See 4.3.147 for the application assertion.

4.2.105.6 inner_outer

The inner_outer specifies whether the solid material is outside or inside the Dmf_plane_closed_parallel, that is, whether it is a slot or a boss feature.

4.2.105.7 longitude_vector

The longitude_vector specifies the second direction vector determining the parallel planes, which is in the same plane as the axis_direction_vector but perpendicular to it and parallel to the parallel planes. See 4.3.149 for the application assertion.

4.2.105.8 normal_dml

The normal_dml specifies the normal vector to the closed plane or plane of the slot. See 4.3.149 for the application assertion.

4.2.105.9 width

The width specifies the width between the parallel planes. See 4.3.147 for the application assertion.

4.2.106 Dmf_plane_symmetric

A Dmf_plane_symmetric is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a pair of planes placed symmetrically around a third plane, called the mid plane. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 This feature is an extension of the FEAT/PARPLN statement in DMIS in order to describe two planes that are symmetrically placed with respect to a mid plane but not necessarily parallel.

NOTE 2 Figure 59 illustrates the Dmf_plane_symmetric.

The data associated with a Dmf_plane_symmetric are the following:

— direction_vector_side_one;

— direction_vector_side_two;

— inner_outer;

— point_on_mid_plane;

— point_on_side_two;

— point_on_one_side;

— width_at_mid_point.

4.2.106.1 direction_vector_side_one

The direction_vector_side_one specifies the vector normal to the plane of side one. See 4.3.152 for the application assertion.

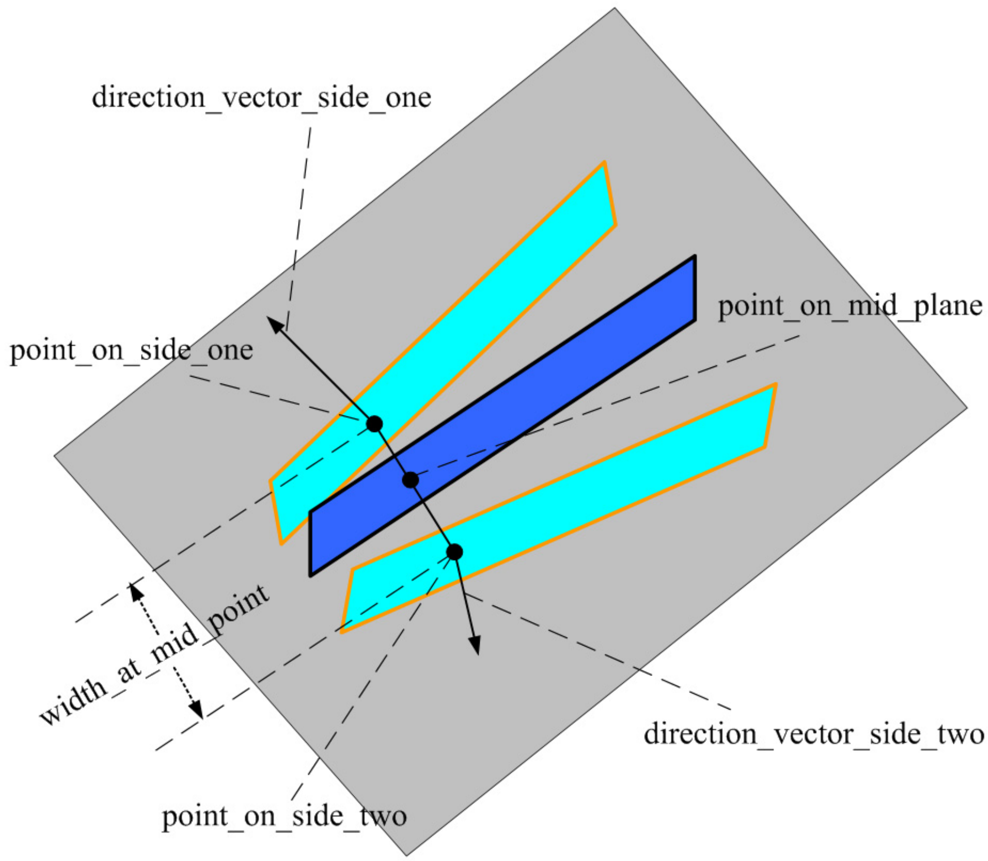


Figure 59 — Dmf_plane_symmetric

4.2.106.2 direction_vector_side_two

The direction_vector_side_two specifies the vector normal to the plane of side two. See 4.3.152 for the application assertion.

4.2.106.3 inner_outer

The inner_outer specifies whether the solid material is outside or inside the Dmf_plane_symmetric feature.

4.2.106.4 point_on_mid_plane

The point_on_mid_plane specifies the coordinates of a point on the middle plane, which is the plane of symmetry. See 4.3.150 for the application assertion.

4.2.106.5 point_on_side_two

The point_on_side_two specifies the coordinates of a point on side two. See 4.3.150 for the application assertion.

4.2.106.6 width_at_mid_point

The width_at_mid_point specifies the shortest distance between the planes passing through the point_on_mid_plane. See 4.3.151for the application assertion.

4.2.106.7 point_on_one_side

The point_on_one_side specifies the coordinates of a point on side one. See 4.3.150 for the application assertion.

4.2.107 Dmf_point

A Dmf_point is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters (coordinates) of a point. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a point may be found in ISO 10303-42.

NOTE 2 Figure 60 illustrates the Dmf_point.

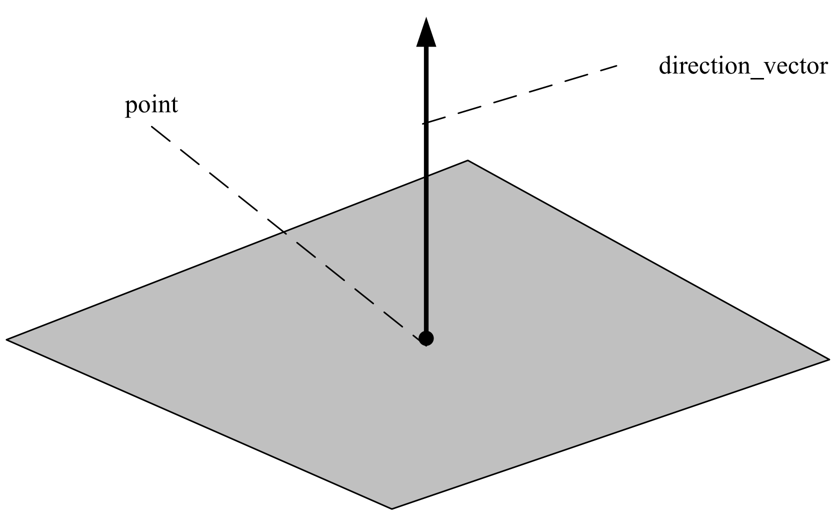


Figure 60 — Dmf_point

The data associated with a Dmf_point are the following:

— direction_vector;

— point.

4.2.107.1 direction_vector

The direction_vector specifies the axis of a probe specified or used to measure a specific point. See 4.3.154 for the application assertion.

4.2.107.2 point

The point specifies a location in some real Cartesian coordinate space, specified by three or sometimes two coordinate values. See 4.3.153 for the application assertion.

4.2.108 Dmf_sphere

A Dmf_sphere is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a spherical surface. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a sphere may be found in ISO 10303-42.

NOTE 2 Figure 61 illustrates the Dmf_sphere.

The data associated with a Dmf_Sphere are the following:

— center_point;

— diameter;

— inner_outer;

— latitude_start_angle;

— latitude_stop_angle;

— longitude_start_angle;

— longitude_stop_angle;

— north_pole_vector;

— prime_meridian_vector.

4.2.108.1 center_point

The center_point specifies the center of the sphere. See 4.3.156 for the application assertion.

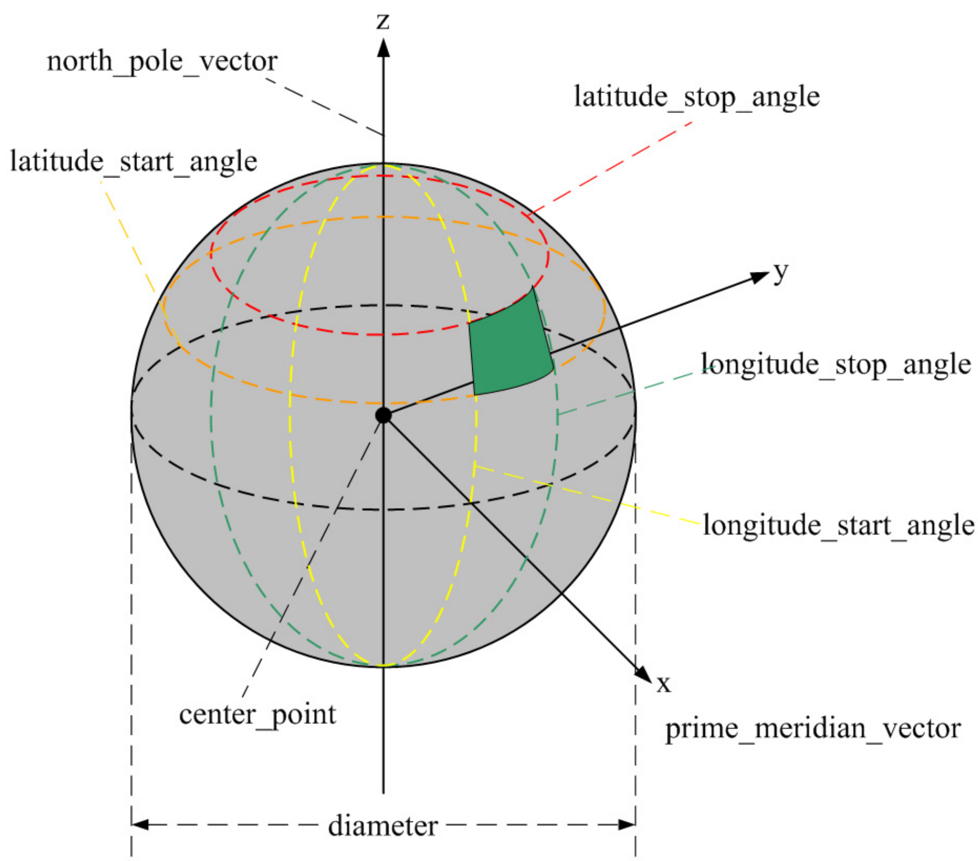


Figure 61 — Dmf_sphere

4.2.108.2 diameter

The diameter specifies twice the radius of the sphere. See 4.3.155 for the application assertion.

4.2.108.3 inner_outer

The inner_outer specifies whether the solid material is outside or inside the Dmf_plane_symmetric feature.

4.2.108.4 latitude_start_angle

The latitude_start_angle specifies for a bounded sphere, the angle specifying the lower limit of latitude of the spherical surface. See 4.3.155 for the application assertion.

4.2.108.5 latitude_stop_angle

The latitude_stop_angle specifies for a bounded sphere, the angle specifying the upper limit of latitude of the spherical surface. See 4.3.155 for the application assertion.

4.2.108.6 longitude_start_angle

The longitude_start_angle specifies for a bounded sphere, the angle specifying the lower limit of longitude of the spherical surface. See 4.3.155 for the application assertion.

4.2.108.7 longitude_stop_angle

The longitude_stop_angle specifies for a bounded sphere, the angle specifying the upper limit of longitude of the spherical surface. See 4.3.155 for the application assertion.

4.2.108.8 north_pole_vector

The north_pole_vector specifies the vector whose direction from the center of the sphere specifies the positive 90° direction for latitude. See 4.3.157 for the application assertion.

4.2.108.9 prime_meridian_vector

The prime_meridian_vector specifies the vector whose direction from the center of the sphere specifies the origin (zero angle) for longitude. See 4.3.157 for the application assertion.

4.2.109 Dmf_surface_of_revolution_dml

The Dmf_Surface_Of_Revolution_Dml is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) that is a feature defined by profile curves and an axis of revolution, the profile curves being represented by point data.

NOTE Figure 62 illustrates the Dmf_surface_of_revolution_dml.

The data associated with a Dmf_surface_of_revolution are the following:

— axis_of_revolution;

— profile_curve.

4.2.109.1 axis_of_revolution

The axis_of_revolution specifies the axis about which the profile_curve is rotated to form the complete surface_of_revolution. See 4.3.159 for the application assertion.

4.2.109.2 profile_curve

The profile_curve specifies the curve of coordinate point data that specifies the generating profile of the revolved surface. See 4.3.158 for the application assertion.

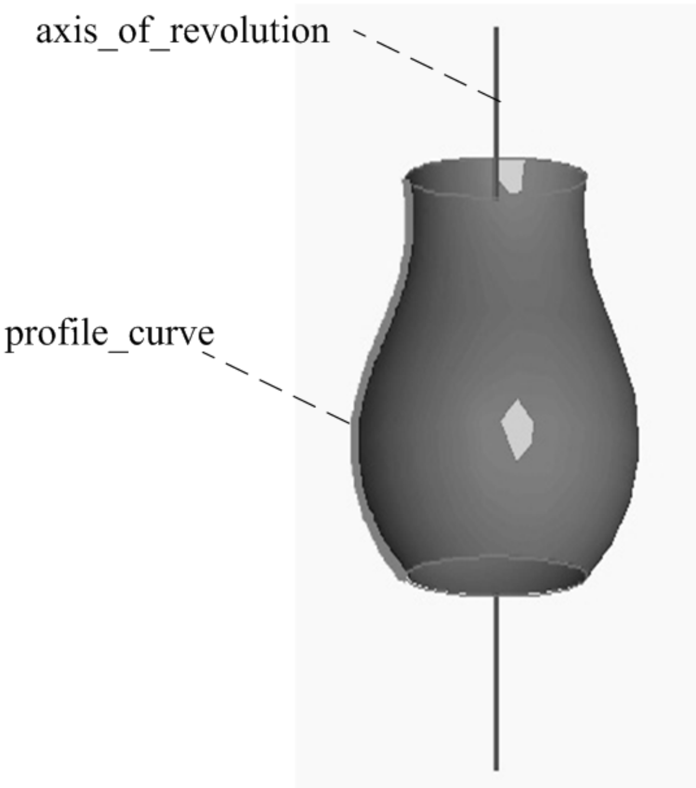


Figure 62 — Dmf_surface_of_revolution_dml

4.2.110 Dmf_torus

A Dmf_torus is a type of Dm_feature (see 4.2.78) that is a Shape_element (see 4.2.238) used in dimensional measurement to evaluate the parameters of a toroidal surface. The parameters are derived from measured or simulated coordinate data points.

NOTE 1 Related concept for a sphere may be found in ISO 10303-42.

NOTE 2 Figure 63 illustrates the Dmf_torus.

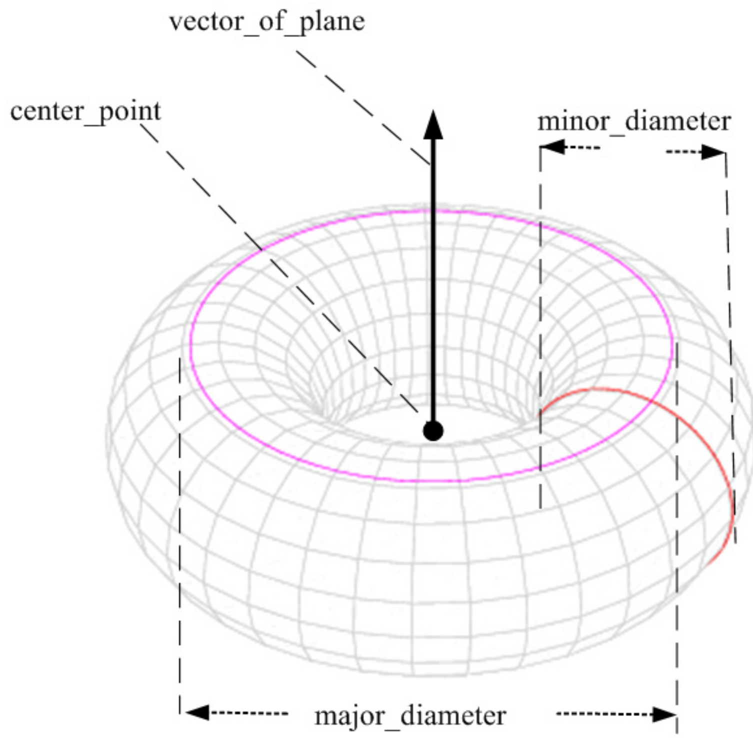


Figure 63 — Dmf_torus

The data associated with a Dmf_torus are the following:

— center_point;

— inner_outer;

— major_diameter;

— minor_diameter;

— vector_of_plane;

4.2.110.1 center_point

The center_point specifies the center of the torus. See 4.3.161 for the application assertion.

4.2.110.2 inner_outer

The inner_outer specifies whether the toroidal surface is inside or outside the part material.

4.2.110.3 major_diameter

The major_diameter specifies the major diameter of the torus. See 4.3.160 for the application assertion.

4.2.110.4 minor_diameter

The minor_diameter specifies the minor diameter of the torus. See 4.3.160 for the application assertion.

4.2.110.5 vector_of_plane

The vector_of_plane specifies the vector specifying the normal to the plane of the torus. See 4.3.162 for the application assertion.

4.2.111 Document_assignment

A Document_assignment is a mechanism to associate a document with an object, where the assigned document provides information about the object it is associated to. Each Document_assignment may be a Part_dimensioning_standard (see 4.2.185). The data associated with a Document_assignment are the following:

— assigned_document;

— role.

4.2.111.1 assigned_document

The assigned_document specifies the Document_file (see 4.2.111) that is used to provide information. See 4.3.163 for the application assertion.

4.2.111.2 role

The role specifies the meaning of the Document_assignment.

4.2.112 Dof_attribute_dml

The Dof_attribute_dml is the specification of the ways each possible coordinate (degree of freedom) is allowed to vary when a dimensional measurement feature is fitted to measured data.

The data associated with a Dof_attribute_dml are the following:

— lower_limit;

— setting;

— upper_limit.

4.2.112.1 lower_limit

The lower_limit specifies the lower limit of allowed values for a coordinate (degree of freedom). See 4.3.164 for the application assertion.

4.2.112.2 setting

The setting specifies the choices for whether a coordinate (degree of freedom) is constrained or is allowed to vary and between what limits when a feature is fitted to measured data.

NOTE See 4.2.112.2.1- 4.2.112.2.3 for the definition of each allowable value for aelpr_analysis.

4.2.112.2.1 ON: the degree of freedom is allowed to take any value to optimize the fit.

4.2.112.2.2 OFF: the degree of freedom is not allowed to vary.

4.2.112.2.3 LIMITS: the coordinate (degree of freedom) is allowed to vary within specified limits. If LIMITS is selected then the upper_limit or lower_limit or both must be selected. The limits are two points along the specified axis or two angles around the specified axis.

4.2.112.3 upper_limit

The upper_limit specifies the upper limit of allowed values for a coordinate (degree of freedom). See 4.3.164 for the application assertion.

4.2.113 Edge_round

An Edge_round is a type of Transition_feature (see 4.2.274) 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 4.2.45).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.73 in ISO 10303-224:2006.

NOTE 2 Figure 64 illustrates the Edge_round

The data associated with an Edge_round are the following:

— edge_round_face;

— first_face_shape;

— second_face_shape.

4.2.113.1 edge_round_face

The edge_round_face specifies the circular transition surface between the two edges of two surfaces. See 4.3.165 for the application assertion.

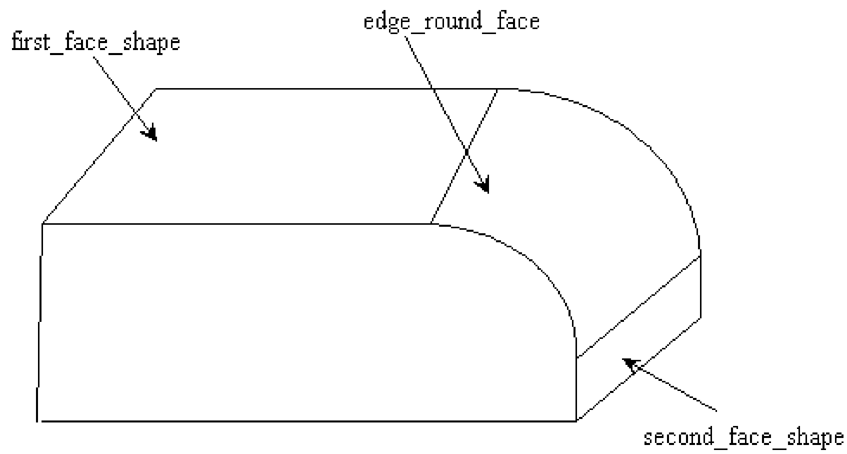


Figure 64 — Edge_round

4.2.113.2 first_face_shape

The first_face shape specifies one of two surfaces the Edge_round feature will transition between. See 4.3.165 for the application assertion.

4.2.113.3 second_face_shape

The second_face shape specifies the other surfaces the Edge_round feature will transition between. See 4.3.165 for the application assertion.

4.2.114 Explicit_base_shape_representation

An Explicit_base_shape_representation is a type of Base_shape (see 4.2.6) that is the geometric representation needed to define the shape of the initial material.

NOTE This application object is harmonized with the same entity from paragraph 4.2.75 in ISO 10303-224:2006.

EXAMPLE A Brep 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:

— brep_form;

— explicit_shape;

— user_defined_description.

4.2.114.1 brep_form

The brep_form specifies the shape that is the representation of the initial material. See 4.3.166 for the application assertion.

4.2.114.2 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 explicit_shape shall be one of the following:

— casting;

— composite shape;

— forging;

— user_defined.

NOTE See 4.2.114.2.1 - 4.2.114.2.4 for the definition of each allowable value for endcut_shape_type.

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

4.2.114.2.2 composite shape: the base shape of the part is defined with non-metallic materials.

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

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

4.2.114.3 user_defined_description

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

4.2.115 Externally_defined_size_dimension

An Externally_defined_size_dimension is a type of Size_tolerance (see 4.2.240) that is used to identify a size dimension whose definition is provided within an external specification or document.

NOTE This application object is harmonized with the same entity from paragraph 4.2.77 in ISO 10303-224:2006.

The data associated with an Externally_defined_size_dimension are the following:

— path;

— tolerance_class;

— tolerance_definition.

4.2.115.1 path

The path specifies the shape that the tolerance applies to. An Externally_defined_size_dimension may but need not require a path.See 4.3.168 for the application assertion.

4.2.115.2 tolerance_class

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

4.2.115.3 tolerance_definition

The tolerance_definition specifies a file being referenced that contains information about the type of tolerance. See 4.3.167 for the application assertion.

4.2.116 Face_shape_element

A Face_shape_element is a type of Shape_element (see 4.2.238) that is a Shape_aspect definition for a Face.

NOTE This application object is harmonized with the same entity from paragraph 4.2.79 in ISO 10303-224:2006.

4.2.117 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.

NOTE This application object is harmonized with the same entity from paragraph 4.2.80 in ISO 10303-224:2006.

The data associated with a Face_shape_element_relationship are the following:

— predecessor;

— successor.

4.2.117.1 predecessor

The predecessor specifies the Face_shape_element (see 4.2.116) with the highest precedence. See 4.3.169 for the application assertion.

4.2.117.2 successor

The successor specifies the Face_shape_element (see 4.2.116) with the lesser precedence. See 4.3.169 for the application assertion.

4.2.118 Feature

The entity Feature is a type of Shape_element (see 4.2.238) and as an abstract supertype serves as an umbrella for both manufacturing shapes and dimensional inspection shapes. A Feature is either a Dm_feature (see 4.2.78) or a Manufacturing_feature (see 4.2.163).

The data associated with a Feature are the following:

— label;

— placement.

4.2.118.1 label

The label specifies the text for identifying the feature

4.2.118.2 placement

The placement secifies the position and orientation of a Feature relative to the base shape for a part. See 4.3.170 for the application assertion.

4.2.119 Fillet

A Fillet is a type of Transition_feature (see 4.2.274) 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 4.2.46).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.81 in ISO 10303-224:2006.

NOTE 2 Figure 65 illustrates the Fillet

The data associated with a Fillet are the following:

— fillet_face;

— first_face_shape;

— second_face_shape.

4.2.119.1 fillet_face

The fillet_face specifies the circular transition surface between the two edges of two surfaces. See 4.3.171 for the application assertion.

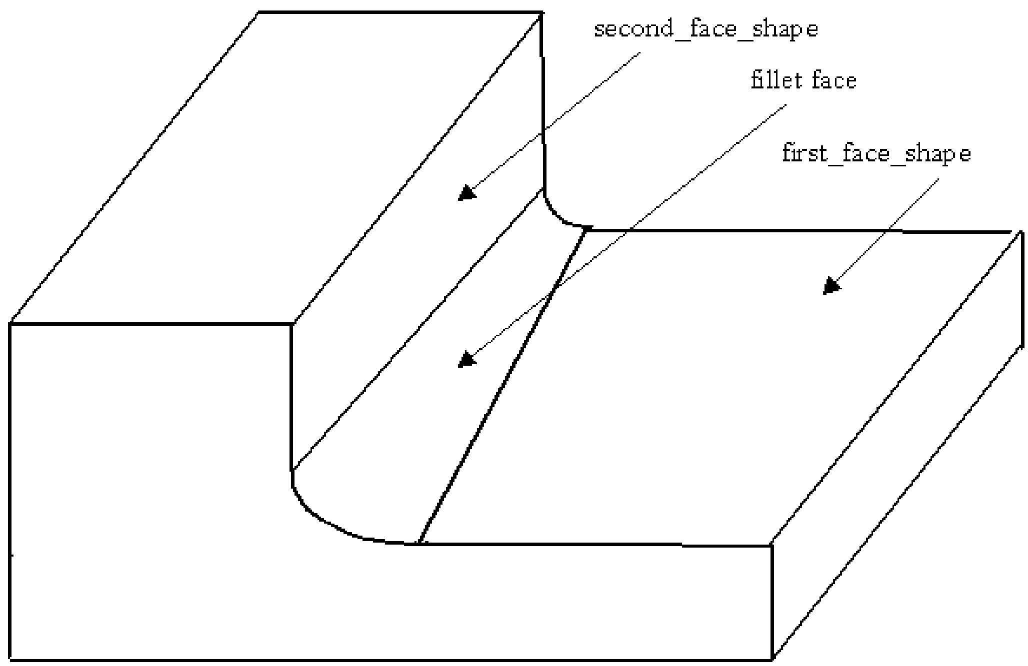


Figure 65 — Fillet

4.2.119.2 first_face_shape

The first_face_shape specifies one of two surfaces the Fillet feature will transition between. See 4.3.171 for the application assertion.

4.2.119.3 second_face_shape

The second_face_shape specifies the second of two surfaces the Fillet feature will transition between. See 4.3.171 for the application assertion.

4.2.120 First_offset

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

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.82 in ISO 10303-224:2006.

NOTE 2 Figure 15 illustrates a First_offset for a Chamfer feature.

The data associated with a First_offset are the following:

— face_shape;

— offset_amount.

4.2.120.1 face_shape

The face specifies a geometric shape for one of two faces the Chamfer feature will transition between. See 4.3.172 for the application assertion.

4.2.120.2 offset_amount

The offset_amount specifies a distance from the edge of a face to the start of the Chamfer. See 4.3.173 for the application assertion.

4.2.121 Flat_hole_bottom

A Flat_hole_bottom is a type of Blind_bottom_condition (see 4.2.8) that is the bottom of a Round_hole that shall be flat and have no corner radius.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.83 in ISO 10303-224:2006.

NOTE 2 Figure 66 illustrates a First_offset for a Flat_hole_bottom.

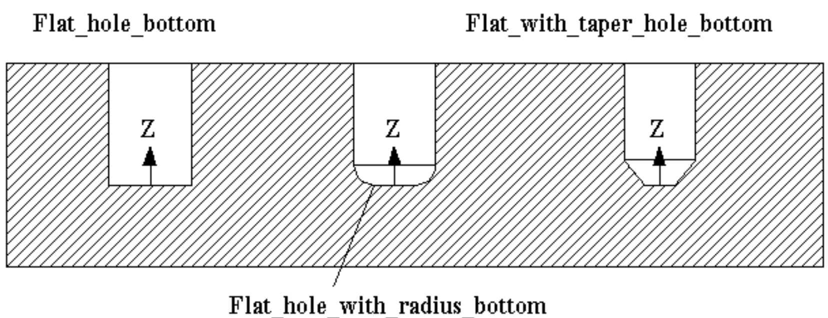


Figure 66 — Flat_hole_bottom types

4.2.122 Flat_slot_end_type

A Flat_slot_end_type is a type of Slot_end_type (see 4.2.242) 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 1 This application object is harmonized with the same entity from paragraph 4.2.84 in ISO 10303-224:2006.

NOTE 2 Figure 67 illustrates a Flat_slot_end_type.

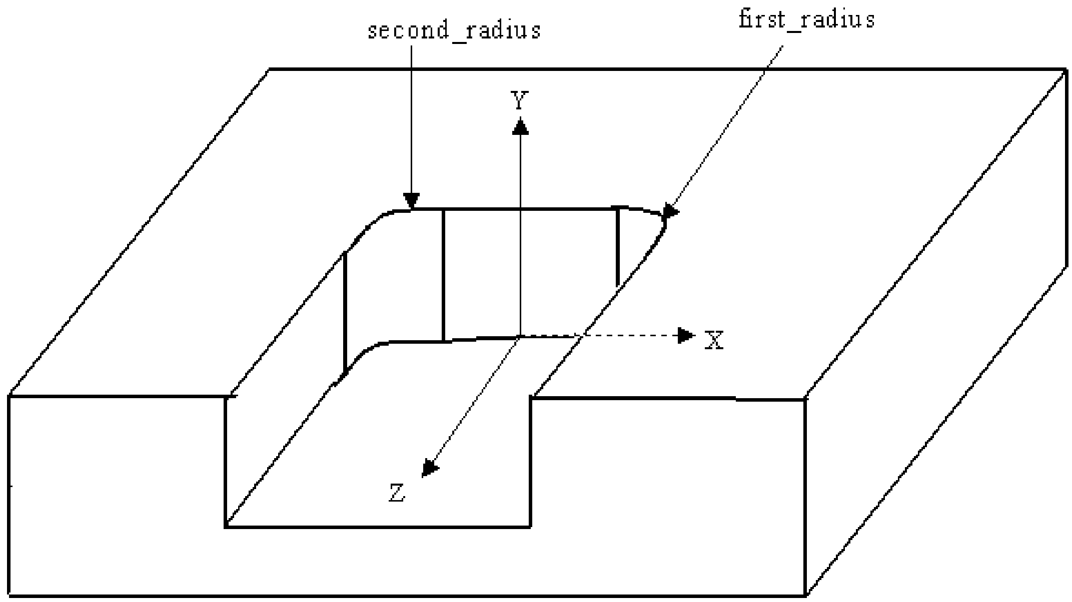


Figure 67 — Flat_slot_end_type

The data associated with a Flat_slot_end_type are the following:

— first_radius;

— second_radius.

4.2.122.1 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 4.3.174 for the application assertion.

4.2.122.2 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 4.3.174 for the application assertion.

4.2.123 Flat_with_radius_hole_bottom

A Flat_with_radius_hole_bottom is a type of Blind_bottom_condition (see 4.2.8) that is the bottom of a Round_hole that shall be flat and have a corner radius that is smaller than the radius of the Round_hole.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.85 in ISO 10303-224:2006.

NOTE 2 Figure 66 illustrates a Flat_with_radius_hole_bottom.

The data associated with a Flat_with_radius_hole_bottom are the following:

— corner_radius.

4.2.123.1 corner_radius

The corner_radius specifies the radius between the side and the floor of a Round_hole (see 4.2.230). See 4.3.175 for the application assertion.

4.2.124 Flat_with_taper_hole_bottom

A Flat_with_taper_hole_bottom is a type of Blind_bottom_condition (see 4.2.8) that is the bottom of a Round_hole that shall be flat and have a planar taper from the sides of the Round_hole.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.86 in ISO 10303-224:2006.

NOTE 2 Figure 66 illustrates a Flat_with_taper_hole_bottom.

The data associated with a Flat_with_taper_hole_bottom are the following:

— final_diameter;

— taper_diameter.

4.2.124.1 final_diameter

The final_diameter specifies the diameter of the Round_hole floor which is a diameter smaller than the initial diameter. See 4.3.176 for the application assertion.

4.2.124.2 taper_diameter

The taper_diameter specifies the angle between the side and the floor of a Round_hole (see 4.2.230), measured with respect to the axis inside the Round_hole. See 4.3.176 for the application assertion.

4.2.125 Flatness_tolerance

A Flatness_tolerance is a type of Geometric_tolerance (see 4.2.143) 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 This application object is harmonized with the same entity from paragraph 4.2.87 in ISO 10303-224:2006.

NOTE 2 The Flatness_tolerance definition is derived from ISO 1101, 14.2.

NOTE 3 Figure 68 illustrates a First_offset for a Flatness_tolerance.

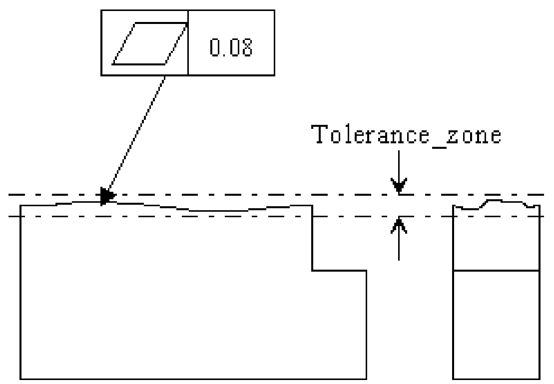


Figure 68 — Flatness_tolerance

The data associated with a Flatness_tolerance are the following:

— segment_size.

4.2.125.1 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.

4.2.126 Gear

A Gear is a type of Machining_feature (see 4.2.162) 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 4.2.20).

NOTE This application object is harmonized with the same entity from paragraph 4.2.88 in ISO 10303-224:2006.

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.

4.2.126.1 applied_shape

The applied_shape specifies a base shape for applying the Gear feature. See 4.3.177 for the application assertion.

4.2.126.2 face_width

The face_width specifies a width over the toothed part of a gear, measured along a generator of the reference cylinder. See 4.3.178 for the application assertion.

NOTE Figure 69 illustrates a face_width.

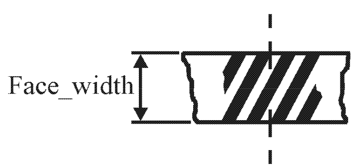


Figure 69 — Gear face_width

4.2.126.3 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.

4.2.126.4 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 4.2.126.6).

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.

4.2.126.5 nominal_tooth_depth

The nominal_tooth_depth specifies the radial distance between the tooth tip and root circles. See 4.3.178 for the application assertion.

NOTE Figure 70 illustrates a nominal_tooth_depth.

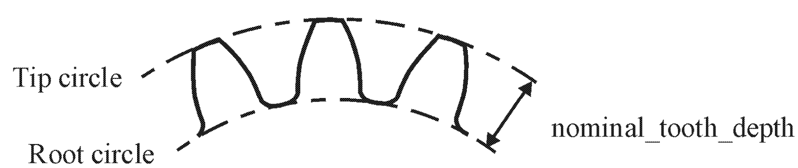


Figure 70 — Gear nominal_tooth_depth

4.2.126.6 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 4.3.178 for the application assertion.

4.2.126.7 number_of_teeth

The number_of_teeth specifies the number of teeth in the circumference produced on the part surface. See 4.3.178 for the application assertion.

4.2.126.8 profile_shift

The profile_shift is the quotient of the rack shift divided by the module. 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 4.3.178 for the application assertion.

NOTE Figure 71 illustrates a profile_shift for a Gear.

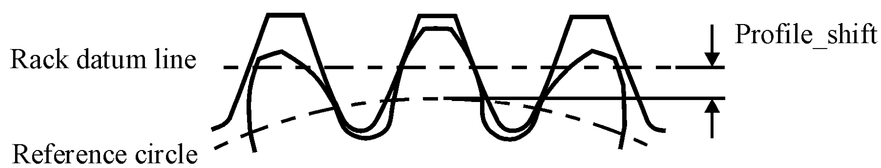


Figure 71 — Gear profile_shift

4.2.126.9 reference_presure_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 4.3.178 for the application assertion.

NOTE Figure 72 illustrates a reference_pressure_angle.

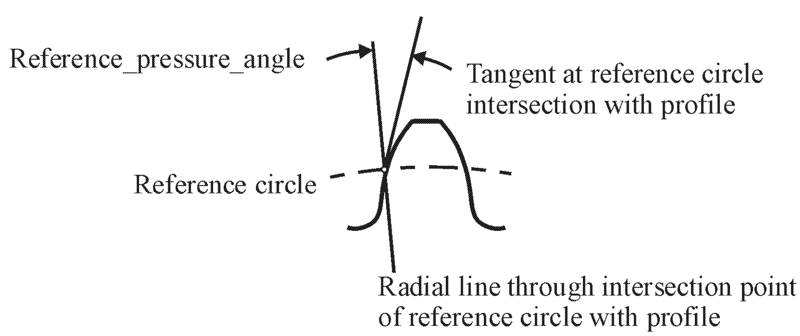


Figure 72 — Gear reference_presure_angle

4.2.126.10 root_fillet_radius

The root_fillet_radius specifies the acceptable radius of the surface between the usable flank and the root surface. See 4.3.178 for the application assertion.

NOTE Figure 73 illustrates a root_fillet_radius for a Gear.

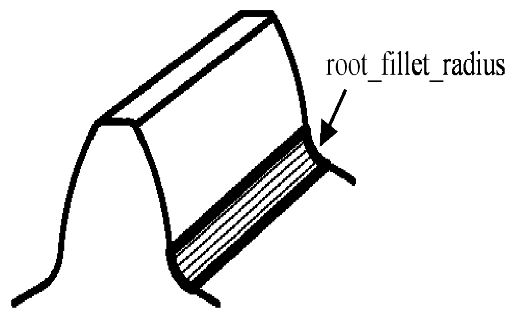


Figure 73 — Gear root_fillet_radius

4.2.126.11 tip_diameter

The tip_diameter specifies the diameter of the tip circle. See 4.3.178 for the application assertion.

NOTE Figure 74 illustrates a tip_diameter.

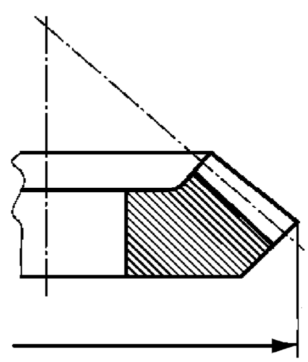


Figure 74 — Gear tip_diameter

4.2.127 General_boss

A General_boss is a type of Boss (see 4.2.10) that is an enclosed volume bounded by an arbitrary shape.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.89 in ISO 10303-224:2006.

NOTE 2 Figure 75 illustrates a General_boss.

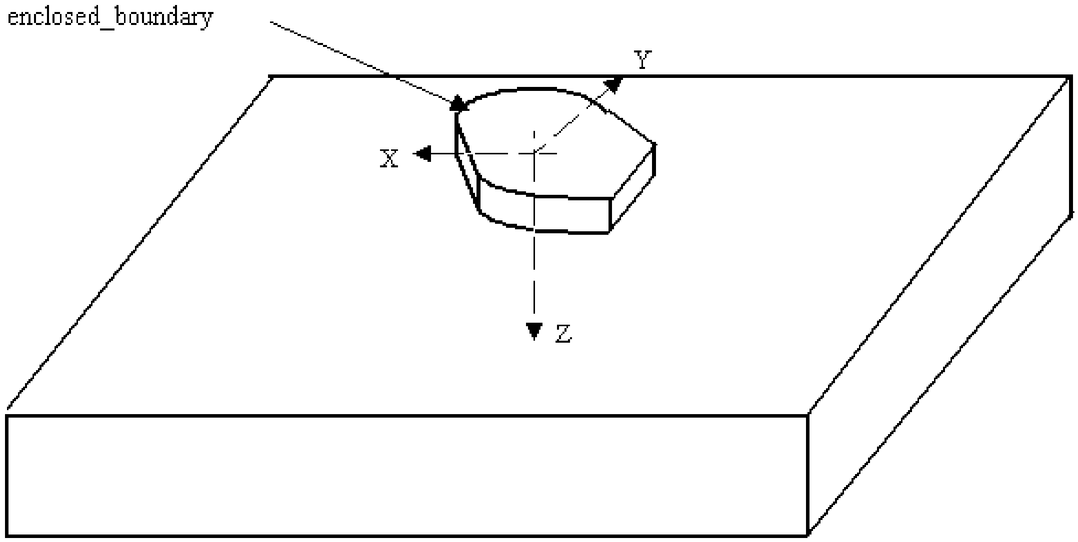


Figure 75 — General_boss

The data associated with a General_boss are the following:

— change_in_boundary;

— enclosed_boundary.

4.2.127.1 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 4.3.180 and 4.3.181 for the application assertion.

4.2.127.2 enclosed_boundary

The enclosed_boundary specifies an outline or shape that bounds an enclosed area with no opening for a General_boss. The enclosed_boundary specifies the area required by a General_boss. The placement of the enclosed_boundary shall be with the origin of the Closed_profile (see 4.2.36) 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 4.3.179 for the application assertion.

4.2.128 General_closed_profile

A General_closed_profile is a type of Closed_profile (see 4.2.36) that is an enclosed area bounded by a arbitrary shape. The orientation is defined by the explicit geometry of the shape.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.90 in ISO 10303-224:2006.

NOTE 2 Figure 76 illustrates a General_closed_profile.

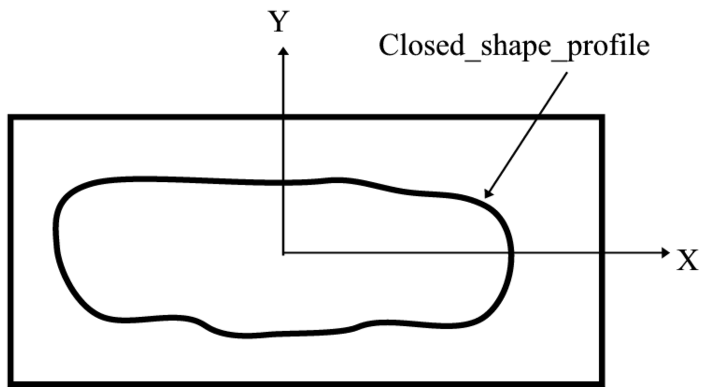


Figure 76 — General_closed_profile

The data associated with a General_closed_profile are the following:

— closed_profile_shape.

4.2.128.1 closed_profile_shape

The closed_profile_shape specifies a closed curve that defines the arbitrary shape of the profile. See 4.3.182 for the application assertion.

4.2.129 General_cutout

A General_cutout is a type of Cutout (see 4.2.51) 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 differs in the type of process required to manufacture.

NOTE This application object is harmonized with the same entity from paragraph 4.2.91 in ISO 10303-224:2006.

The data associated with a General_cutout are the following:

— boundary.

4.2.129.1 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 4.3.183 for the application assertion.

4.2.130 General_open_profile

A General_open_profile is a type of Open_profile (see 4.2.175) that is specified by a shape bounded by an arbitrary planar shape. The orientation is defined by the explicit geometry of the shape.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.92 in ISO 10303-224:2006.

NOTE 2 Figure 77 illustrates a General_open_profile.

The data associated with a General_open_profile are the following:

— enclosed_boundary.

4.2.130.1 enclosed_boundary

The enclosed_boundary specifies a curve with no enclosing bounds that defines the arbitrary planar shape of the profile. See 4.3.184 for the application assertion.

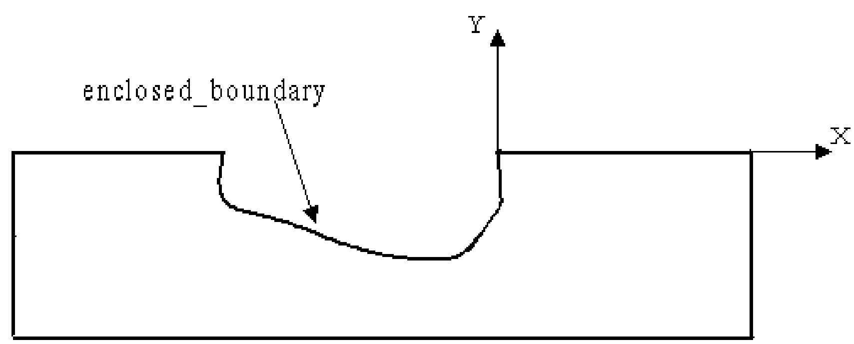


Figure 77 — General_open_profile

4.2.131 General_outside_profile

A General_outside_profile is a type of Profile_feature (see 4.2.206) that specifies 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 This application object is harmonized with the same entity from paragraph 4.2.93 in ISO 10303-224:2006.

NOTE 2 Figure 78 illustrates a General_outside_profile.

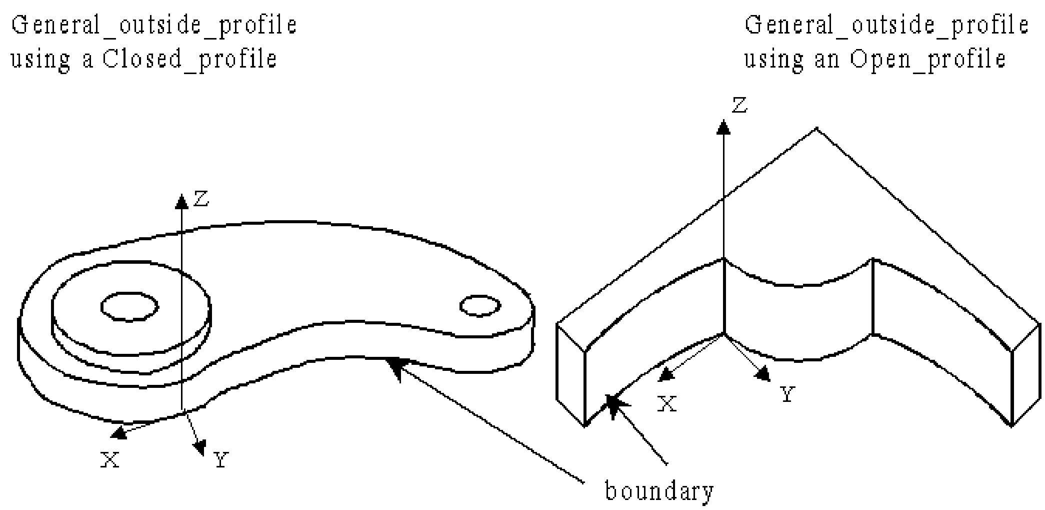


Figure 78 — General_outside_profile

The data associated with a General_outside_profile are the following:

— boundary.

4.2.131.1 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 4.3.185 for the application assertion.

4.2.132 General_path

A General_path is a type of Path (see 4.2.190) that is a direction of travel along an arbitrary curve.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.94 in ISO 10303-224:2006.

NOTE 2 Figure 79 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.

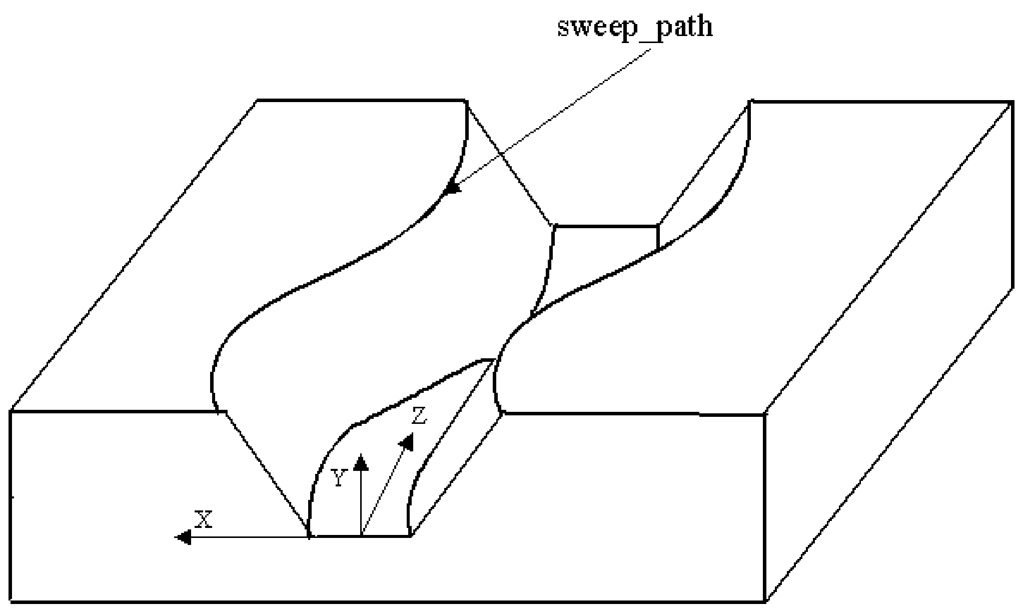


Figure 79 — General_path

4.2.132.1 sweep_path

The sweep_path specifies a continuous set of curves that define an arbitrary direction of travel. See 4.3.186 for the application assertion.

4.2.133 General_pattern

A General_pattern is a type of Replicate_feature (see 4.2.224) 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 1 This application object is harmonized with the same entity from paragraph 4.2.95 in ISO 10303-224:2006.

NOTE 2 Figure 80 illustrates a General_pattern.

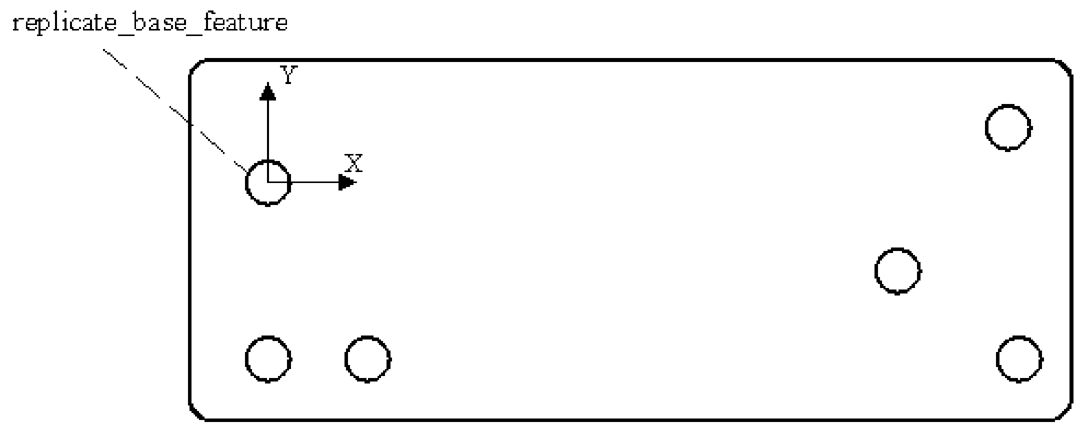


Figure 80 — General_pattern

The data associated with a General_pattern are the following:

— feature_placement.

4.2.133.1 feature_placement

The feature_placement specifies a set of axies 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 4.3.187 for the application assertion.

4.2.134 General_pocket

A General_pocket is a type of Pocket (see 4.2.202) that is a volume of arbitrary shape removed from the part.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.96 in ISO 10303-224:2006.

NOTE 2 Figure 81 illustrates a General_pocket.

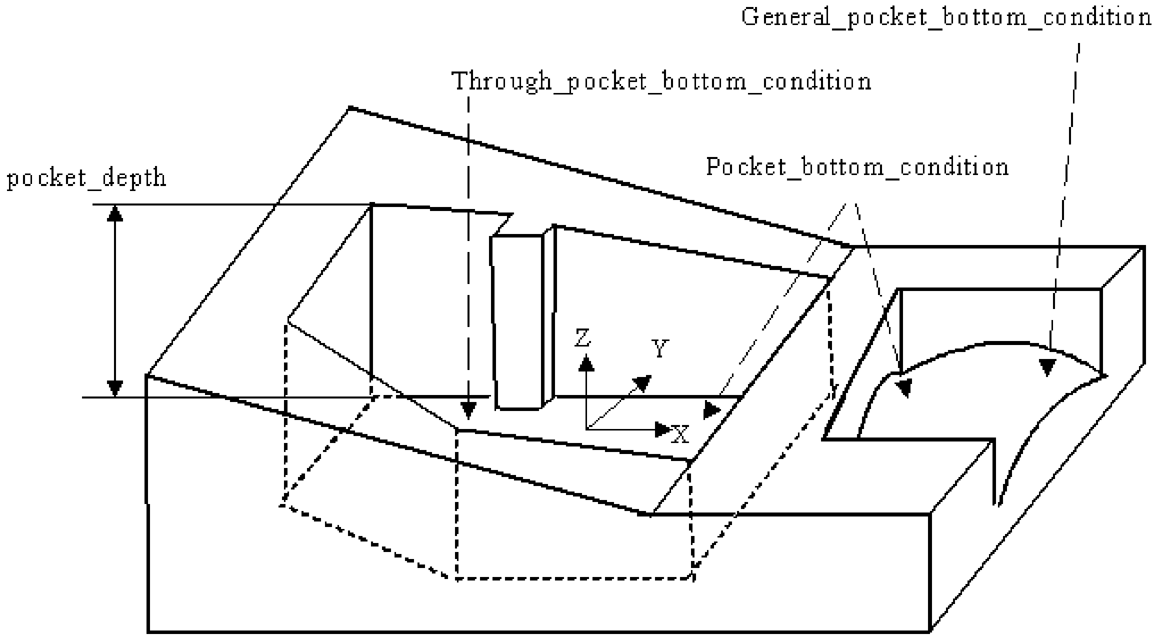


Figure 81 — General_pocket

The data associated with a General_pocket are the following:

— boundary;

— volume_not_removed.

4.2.134.1 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 (See 4.2.205) 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 4.3.188 for the application assertion.

4.2.134.2 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the pocket. The Boss feature (see 4.2.10) defines the shape of the material that is to remain in the pocket. See 4.3.189 or 4.3.190 for the application assertion.

4.2.135 General_pocket_bottom_condition

A General_pocket_bottom_condition is a type of Pocket_bottom_condition (see 4.2.203) that specifies an enclosed area bounded by an arbitrary shape.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.97 in ISO 10303-224:2006.

NOTE 2 Figure 81 illustrates a General_pocket.

The data associated with a General_pocket_bottom_condition are the following:

— floor;

— floor_radius.

4.2.135.1 floor

The floor specifies the face at the bottom of a Pocket feature, adjacent to all the pocket walls. See 4.3.191 for the application assertion.

4.2.135.2 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 4.3.192 for the application assertion.

4.2.136 General_profile_floor

A General_profile_floor is a type of Profile_floor (see 4.2.207) that specifies an enclosed area bounded by an arbitrary shape.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.98 in ISO 10303-224:2006.

NOTE 2 Figure 126 illustrates a Shape_profile with a General_profile_floor.

The data associated with a General_profile_floor are the following:

— floor.

4.2.136.1 floor

The floor specifies the face at the bottom of a General_profile_floor feature, adjacent to all the Shape_profile (see 4.2.239) walls. See 4.3.193 for the application assertion.

4.2.137 General_removal_volume

A General_removal_volume is a type of Multi_axis_feature (see 4.2.169) 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 1 This application object is harmonized with the same entity from paragraph 4.2.99 in ISO 10303-224:2006.

NOTE 2 Figure 82 illustrates a General_removal_volume.

The data associated with a General_removal_volume are the following:

— removal_volume.

4.2.137.1 removal_volume

The removal_volume specifies the arbitrary shape to be removed. See 4.3.194 for the application assertion.

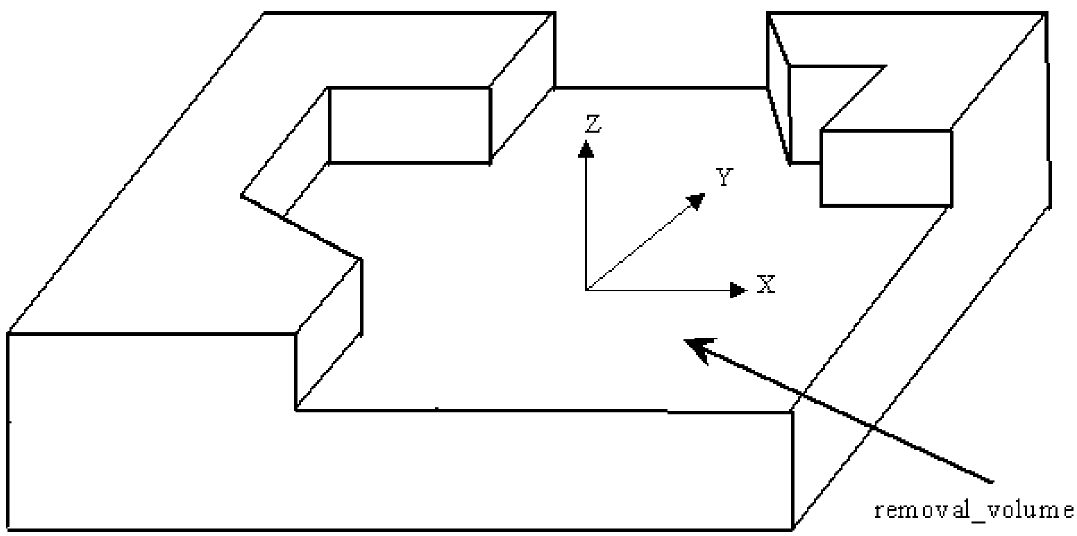


Figure 82 — General_removal_volume

4.2.138 General_revolution

A General_revolution is a type of Revolved_feature (see 4.2.225) 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 1 This application object is harmonized with the same entity from paragraph 4.2.100 in ISO 10303-224:2006.

NOTE 2 Figure 83 illustrates a General_revolution for an outer shape, the material direction points to the material side. Figure 84 illustrates a General_revolution for a volume removal.

The data associated with a General_revolution are the following:

— outer_edge_shape.

4.2.138.1 outer_edge_shape

The outer_edge_shape specifies an outline or shape that shall be revolved about an axis. The General_open_profile (See 4.2.130) 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 Z-axis and Y-axis of the General_revolution. See 4.3.195 for the application assertion.

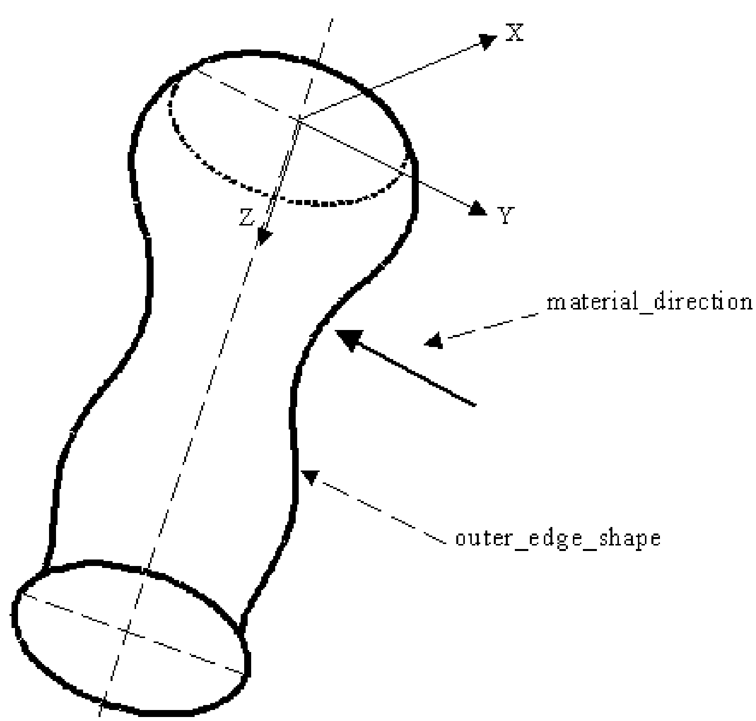


Figure 83 — General_revolution for outer shape

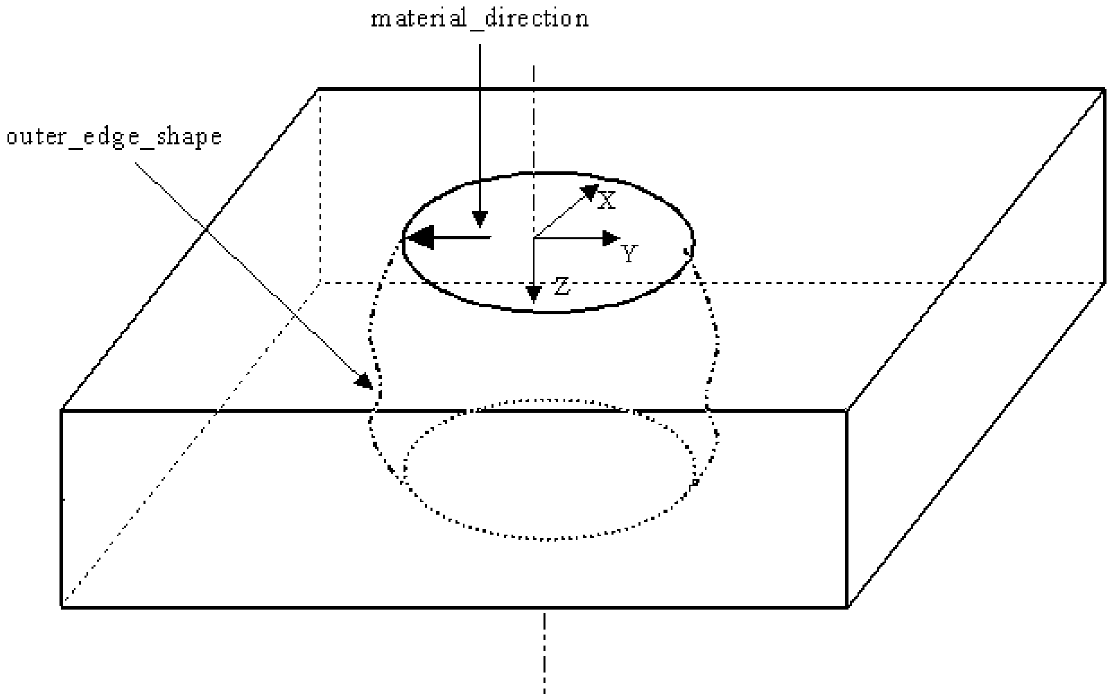


Figure 84 — General_revolution for volume removal

4.2.139 General_rib_top_floor

A General_rib_top_floor is a type of Rib_top_floor (see 4.2.229) that specifies an enclosed area bounded by an arbitrary shape. The floor defining shapes may be defined in a particular sequence.

NOTE This application object is harmonized with the same entity from paragraph 4.2.101 in ISO 10303-224:2006.

The data associated with a General_rib_top_floor are the following:

— rib_top_face.

4.2.139.1 rib_top_face

The rib_top_face specifies a set of faces at the bottom of a Rib_top (see 4.2.228). The order of the faces is achieved using a Face_shape_element_relationship (see 4.2.117). See 4.3.196 for the application assertion.

4.2.140 General_shape_profile

A General_shape_profile is a type of Shape_profile (see 4.2.239) that is a volume of arbitrary shape which defines a portion of the part.

NOTE This application object is harmonized with the same entity from paragraph 4.2.102 in ISO 10303-224:2006.

The data associated with a General_shape_profile are the following:

— profile_boundary.

4.2.140.1 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 4.3.197 for the application assertion.

4.2.141 General_top_condition

A General_top_condition is a type of Boss_top_condition (see 4.2.11) that specifies an enclosed area bounded by an arbitrary shape that defines the top of a Boss feature.

NOTE This application object is harmonized with the same entity from paragraph 4.2.103 in ISO 10303-224:2006.

The data associated with a General_top_condition are the following:

— top_face.

4.2.141.1 top_face

The top_face specifies a face at the top of a Boss feature, adjacent to the Boss sides. See 4.3.198 for the application assertion.

4.2.142 Geometric_model

A Geometric_model is a type of representation dedicated to the description of geometric constructs. The Geometric_model is defined in a Geometric_coordinate_space. A Geometric_model may be a Brep_-model (see 4.2.12).

The data associated with a Geometric_model are the following:

— accuracy;

— description;

— is_defined_in;

— model_extent;

— model_id;

— role;

— version_id;

4.2.142.1 accuracy

The accuracy specifies the CAD model accuracy used in creating the design drawing.

4.2.142.2 description

The description specifies the text to describe the Geometric_model

4.2.142.3 is_defined_in

The is_defined_in specifies the Geometric_coordinate_space in which the Geometric_model is defined. See 4.3.199 for the application assertion.

4.2.142.4 model_extent

The model_extent specifies the length_measure that specifies the radius of a sphere that encloses all elements of the geometric model. The centre of the sphere is located at the origin of the Cartesian_coordinate_space of the Geometric_model. The model_extent need not be specified for a particular Geometric_model.

4.2.142.5 model_id

The model_id specifies a unique identification for the CAD design model.

4.2.142.6 role

The role speicifes the usage of this model for dimensional inspection.

4.2.142.7 version_id

The version_id specifies the text that specifies the version identifier of the Geometric_model.

4.2.143 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: Angularity_tolerance (see 4.2.5), Circular_runout_tolerance (see 4.2.34), Circularity_tolerance (see 4.2.35), Concentricity_tolerance (see 4.2.43), Cylindricity_tolerance (see 4.2.53), Flatness_tolerance (see 4.2.125), Linear_profile_tolerance (see 4.2.157), Parallelism_tolerance (see 4.2.183), Perpendicularity_-tolerance (see 4.2.192), Position_tolerance (see 4.2.204), Straightness_tolerance (see 4.2.252), Surface_-profile_tolerance (see 4.2.253), Symmetry_tolerance (see 4.2.254), or a Total_runout_tolerance (see 4.2.273).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.104 in ISO 10303-224:2006.

NOTE 2 Geometric tolerance definitions are derived from ISO 1101.

The data associated with a Geometric_tolerance are the following:

— applied_shape;

— geometric_tolerance_value;

— applied_to_multiple_datum_frames;

— modifier_control;

— significant_digits;

— unit_of_measure;

— zone_definition.

4.2.143.1 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 4.3.201 for the application assertion.

4.2.143.2 applied_to_multiple_datum_frames

The applied_to_multiple_datum_frames specify the simultaneous or separate application of tolerances for different features tied to the same mobile datum reference frame. The applied_to_multiple_datum_-frames need not be specified for a particular Geometric_tolerance.

4.2.143.3 geometric_tolerance_value

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

4.2.143.4 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 4.3.200 for the application assertion.

4.2.143.5 significant_digits

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

4.2.143.6 unit_of_measure

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

4.2.143.7 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 4.3.202 for the application assertion.

4.2.144 Geometric_tolerance_precedence_relationship

A Geometric_tolerance_precedence_relationship is a composite geometric tolerance. A part may have a shape that has a geometric tolerance and that shape is the basic shape for a pattern which may also have a geometric tolerance. The geometric tolerance for the base shape shall have precedence over the geometric tolerance for the pattern.

NOTE This application object is harmonized with the same entity from paragraph 4.2.105 in ISO 10303-224:2006.

The data associated with a Geometric_tolerance_precedence_relationship are the following:

— base_shape_tolerance;

— pattern_shape_tolerance.

4.2.144.1 base_shape_tolerance

The base_shape_tolerance specifies the Geometric_tolerance which is applied to a shape on a Part. See 4.3.203 for the application assertion.

4.2.144.2 pattern_shape_tolerance

The pattern_shape_tolerance specifies the Geometric_tolerance which is applied to a pattern on a Part. See 4.3.203 for the application assertion.

4.2.145 Groove

A Groove is a type of Revolved_feature (see 4.2.225) that is a narrow channel or depression that is swept through one complete revolution about an axis.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.106 in ISO 10303-224:2006.

NOTE 2 Figure 85 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.

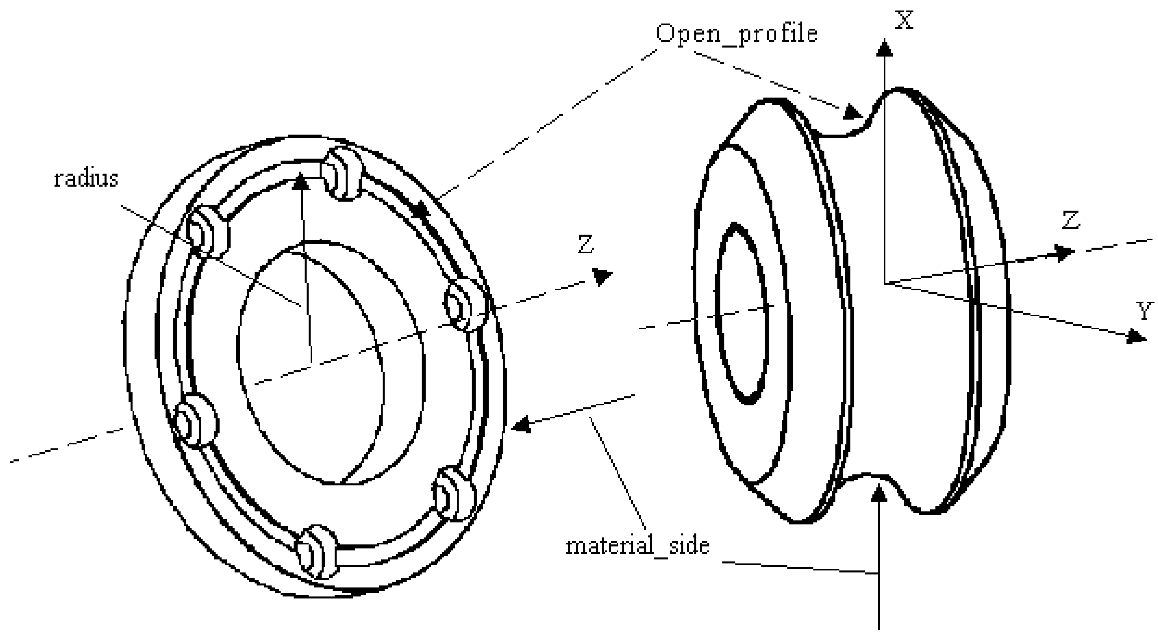


Figure 85 — Groove

4.2.145.1 sweep

The sweep specifies an outline or shape that shall be revolved about an axis. The Open_profile (see 4.2.175) 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 4.3.204 for the application assertion.

4.2.146 Height_dimension

A Height_dimension is a type of Size_tolerance (see 4.2.240) that specifies the size along a straight line that is referred to as height in the referenced shape.

NOTE This application object is harmonized with the same entity from paragraph 4.2.109 in ISO 10303-224:2006.

The data associated with a Height_dimension are the following:

— path.

4.2.146.1 path

The path specifies the shape that the tolerance applies to. A Height_dimension may but need not require a path. See 4.3.205 for the application assertion.

4.2.147 Helical_bevel_gear

The Helical_bevel_gear is a type of Bevel_gear(see 4.2.7) where the tooth traces are non-cylindrical helices, joining the definitions of a s of a Bevel_gear and a Helical_gear.

NOTE This application object is harmonized with the same entity from paragraph 4.2.109 in ISO 10303-224:2006.

The data associated with an Helical_bevel_gear are the following:

— left_or_right_tooth;

— reference_helix_angle.

4.2.147.1 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.

4.2.147.2 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 4.3.206 for the application assertion.

4.2.148 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 1 This application object is harmonized with the same entity from paragraph 4.2.110 in ISO 10303-224:2006.

NOTE 2 Figure 86 illustrates the Helical_gear.

The data associated with an Helical_gear are the following:

— left_or_right_tooth;

— reference_helix_angle.

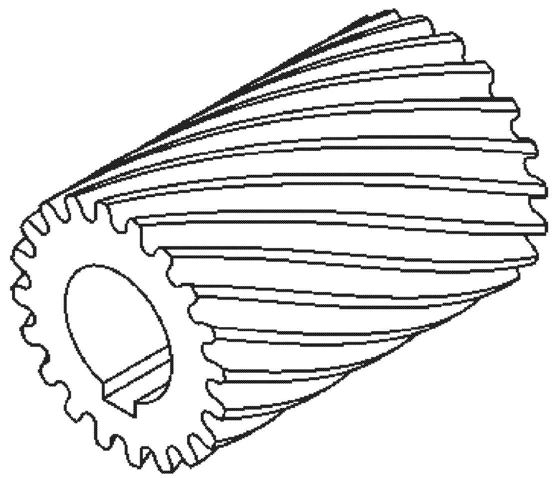


Figure 86 — Helical_gear

4.2.148.1 left_or_right_hand_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 87 illustrates the Helical_gear left_or_right_tooth

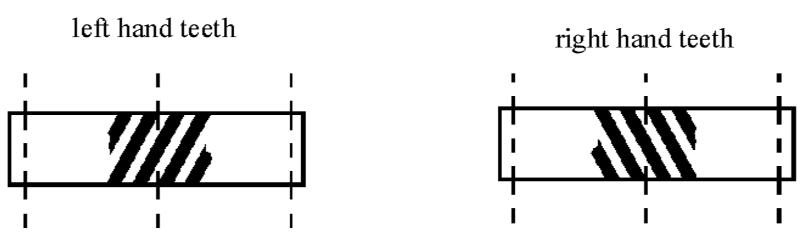


Figure 87 — Helical_gear left_or_right_tooth

4.2.148.2 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 4.3.207 for the application assertion.

NOTE Figure 88 illustrates the Helical_gear reference_helix_angle.

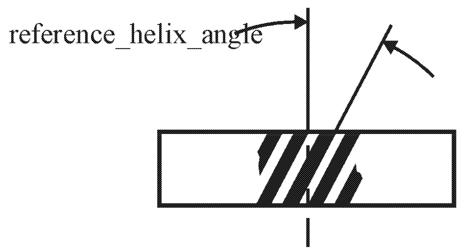


Figure 88 — Helical_gear reference_helix_angle

4.2.149 Hole

A Hole is a type of Multi_axis_feature (see 4.2.169) that is the removal of a cylindrical volume from a part. Each Hole is either a Counterbore_hole (see 4.2.48), Countersunk_hole (see 4.2.49), or a Round_hole (see 4.2.230). The Hole may be positioned at its bottom with the Z-axis in the direction out of the Hole or at a position at its top with the Z-axis in the direction into the part.

NOTE This application object is harmonized with the same entity from paragraph 4.2.111 in ISO 10303-224:2006.

4.2.150 Implicit_base_shape_representation

An Implicit_base_shape_representation is a type of Base_shape (see 4.2.6) that is the type of representation needed to define the shape of the initial material. The shape of the material may be either cylindrical, rectangular, or a polygon of any number of sides. Each Implicit_base_shape_representation is either a Block_base_shape (see 4.2.9), Cylindrical_base_shape (see 4.2.52), or a Ngon_base_shape (see 4.2.170).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.112 in ISO 10303-224:2006.

NOTE 2 Figure 89 illustrates types of Implicit_base_shape_representations.

The data associated with an Implicit_base_shape_representation are the following:

— base_shape_length;

— placement.

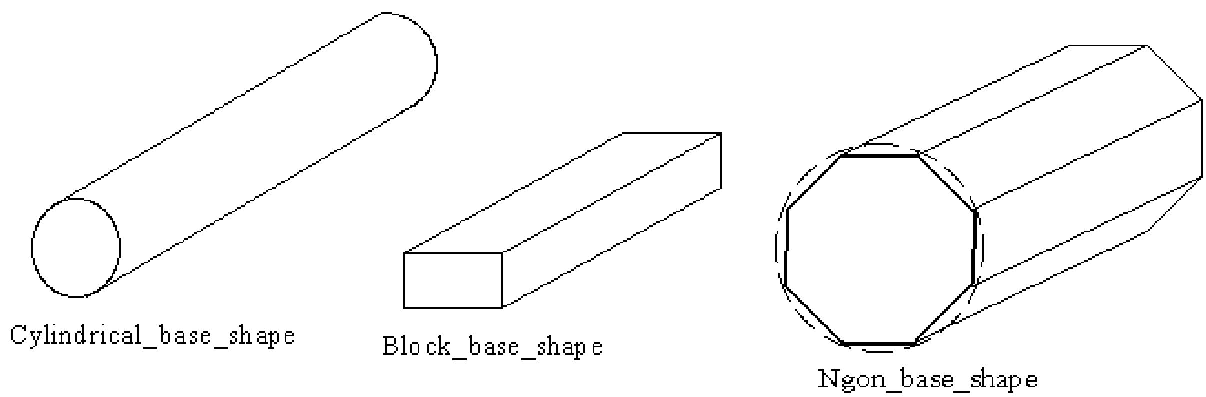


Figure 89 — Implicit_base_shape

4.2.150.1 base_shape_length

The base_shape_length is the size of the length of a Implicit_base_shape_representation. See 4.3.208 for the application assertion.

4.2.150.2 placement

The placement specifies the positioning of the part with respect to basic material stock. The positioning will be different for different types of Implicit_base_shape_representation. See 4.3.209 for the application assertion.

The 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 Z axis shall be positioned in the exact center of the circular profile of the Cylindrical_base_shape.

The 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 Z axis shall be positioned in the exact center of the rectangular profile of the Block_base_shape.

The Ngon_base_shape shall be positioned 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 origin of all axis shall be positioned in the exact center of the ngon profile of the Ngon_base_shape.

4.2.151 Inspection_feature_relationship

An Inspection_feature_relationship definies the relationship between a dimension measurement feature (see 4.2.78) and a manufacturing_feature (see 4.2.163). The Inspection_feature_relationship defines the relationship between designed parameters and calculated parameters from measured points.

The data associated with a Inspection_feature_relationship are the following:

— as_designed_feature;

— inspection_feature.

4.2.151.1 as_designed_feature

The as_designed_feature specifies the manufactured feature that has an associated inspection feature. See 4.3.211 for the application assertion.

4.2.151.2 inspection_feature

The inspection_feature specifes the dimensional measurement feature associated to a manufactured feature. See 4.3.210 for the application assertion.

4.2.152 Knurl

A Knurl is a type of Machining_feature (see 4.2.162) 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 4.2.21) or a Turned_knurl (see 4.2.275).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.114 in ISO 10303-224:2006.

NOTE 2 Figure 90 illustrates types of Turned_knurls.

The data associated with a Knurl are the following:

— applied_shape;

— partial_profile.

4.2.152.1 applied_shape

The applied_shape specifies a base shape for applying the Knurl feature. See 4.3.213 for the application assertion.

4.2.152.2 partial_profile

The partial_profile specifies the placement and length of a surface the Knurl feature is being applied to. The partial_profile need not be specified for a particular Knurl. See 4.3.212 for the application assertion.

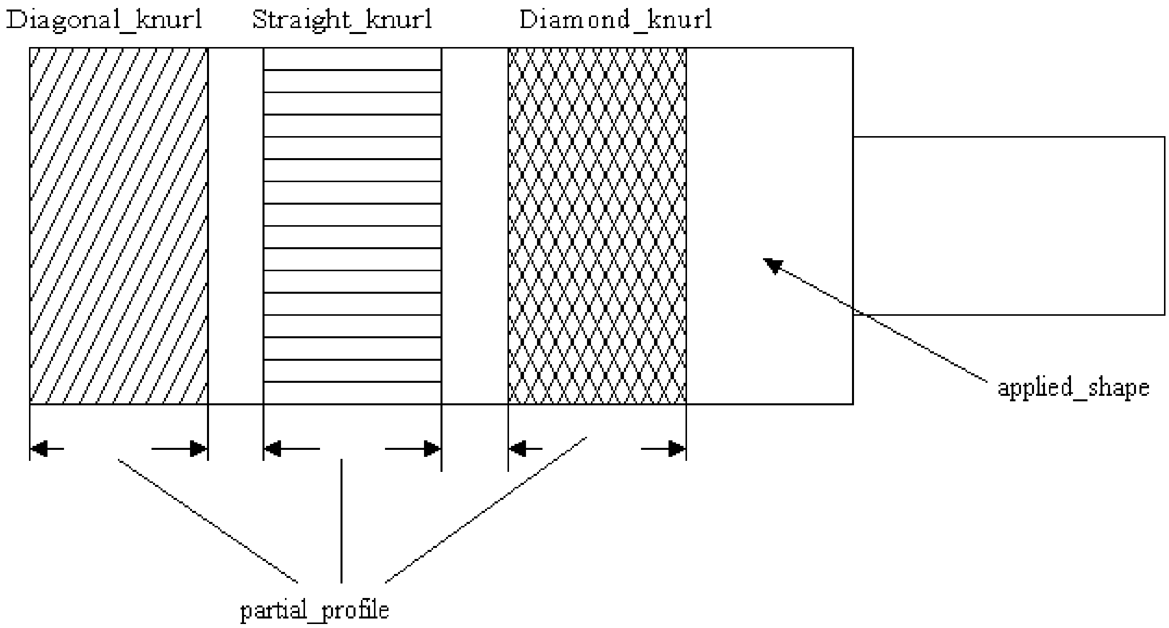


Figure 90 — Knurl and types of Turned_knurl

4.2.153 Length_dimension

A Length_dimension is a type of Size_tolerance (see 4.2.240) that specifies the size along a straight line that is referred to as length in the referenced shape.

NOTE This application object is harmonized with the same entity from paragraph 4.2.115 in ISO 10303-224:2006.

The data associated with a Length_dimension are the following:

— path.

4.2.153.1 path

The path specifies the shape that the tolerance applies to. A Length_dimension may but need not require a path.. See 4.3.214 for the application assertion.

4.2.154 Limits_and_fits

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

NOTE This application object is harmonized with the same entity from paragraph 4.2.117 in ISO 10303-224:2006.

The data associated with a Limits_and_fits are the following:

— deviation;

— fitting type;

— grade.

4.2.154.1 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.

4.2.154.2 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.

4.2.154.3 grade

The grade specifies the quality or the accuracy grade of a tolerance.

NOTE The grade is based on the international standard tolerance grade IT01 to IT18.

4.2.155 Linear_path

A Linear_path is a type of Path (see 4.2.190) that is a direction of travel along a line.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.118 in ISO 10303-224:2006.

NOTE 2 Figure 91 illustrates a Machining feature with a Linear_path. This path has the same orientation as the feature.

The data associated with a Linear_path are the following:

— direction;

— distance.

4.2.155.1 direction

The direction specifies a vector which indicates the the direction of the path starting from the path placement. See 4.3.216 for the application assertion.

4.2.155.2 distance

The distance specifies the length of the path. See 4.3.215 for the application assertion.

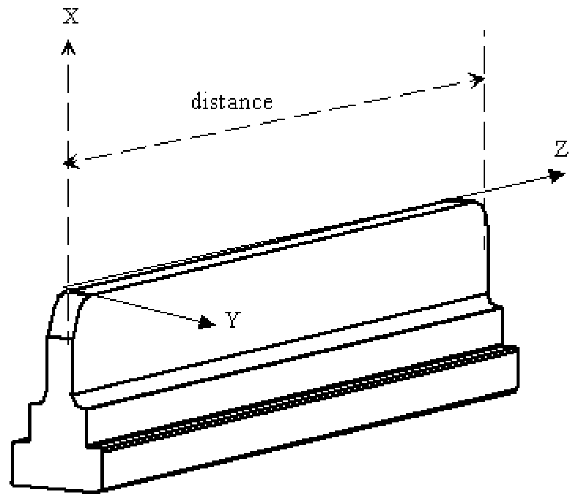


Figure 91 — Linear_path

4.2.156 Linear_profile

A Linear_profile is a type of Open_profile (see 4.2.175) that is a straight line of a specified length. The Linear_profile shall have orientation parallel to the X-axis.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.119 in ISO 10303-224:2006.

NOTE 2 Figure 92 illustrates a Linear_profile that is being applied to the Planar_face feature.

The data associated with a Linear_profile are the following:

— profile_length.

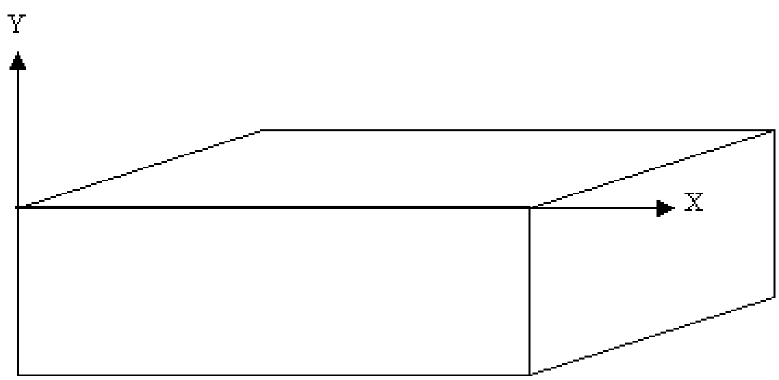


Figure 92 — Linear_profile

4.2.156.1 profile_length

The profile_length specifies the length of the profile. See 4.3.217 for the application assertion.

4.2.157 Linear_profile_tolerance

A Linear_profile_tolerance is a type of Geometric_tolerance (see 4.2.143) that is a uniform boundary or zone along the true profile within which all elements of the surface shall lie.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.120 in ISO 10303-224:2006.

NOTE 2 Figure 93 illustrates the Linear_profile_tolerance

NOTE 3 The Linear_profile_tolerance definition is derived from ISO 1101, 14.5.

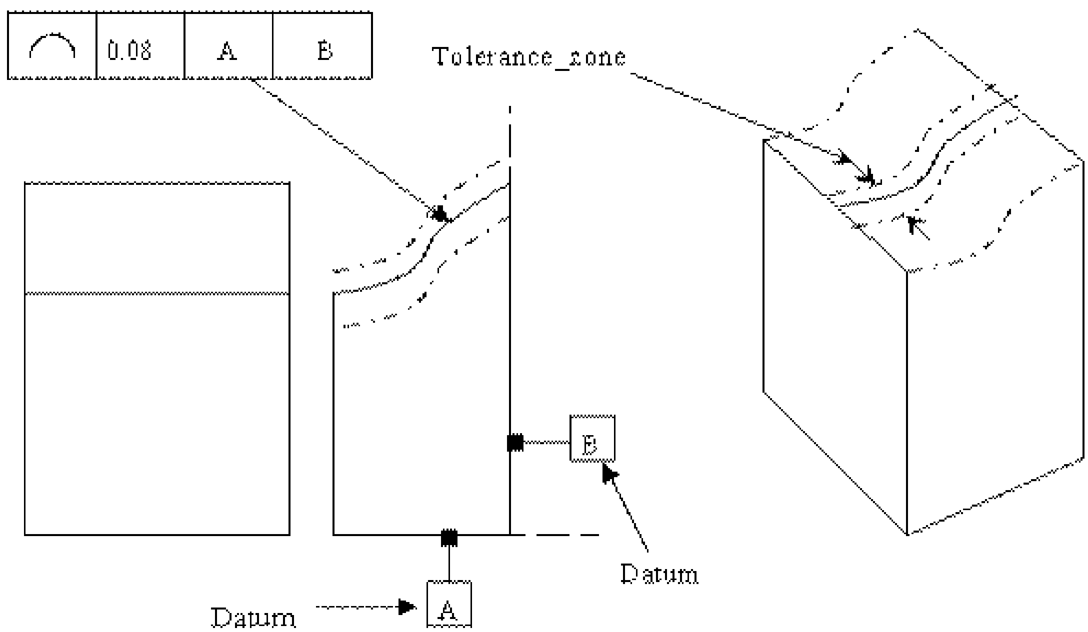


Figure 93 — Linear_profile_tolerance

The data associated with a Linear_profile_tolerance are the following:

— affected_plane;

— geometric_reference.

4.2.157.1 affected_plane

The affected_plane specifies the plane to apply the Linear_profile_tolerance to. The affected_plane need not be specified for a particular Linear_profile_tolerance. See 4.3.219 for the application assertion.

4.2.157.2 geometric_reference

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

4.2.158 Location_dimension_tolerance

A Location_dimension_tolerance is a type of Location_tolerance (see 4.2.160) that is the allowable variation in locating one feature of a part with respect to another.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.121 in ISO 10303-224:2006.

NOTE 2 Figure 94 illustrates the Location_dimension_tolerance

The data associated with a Location_dimension_tolerance are the following:

— feature_type;

— plane_and_direction.

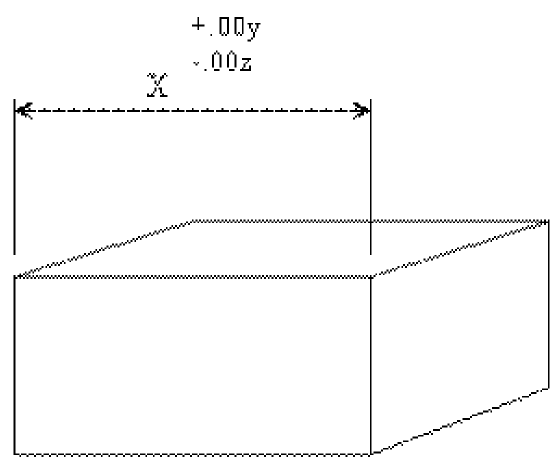


Figure 94 — Location_dimension_tolerance

4.2.158.1 feature_type

The feature_type specifies the type of feature to be used as the origin, the feature type shall be either actual, datum, or nominal. The feature_type need not be specified for a particular Location_dimension_-tolerance.

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

— actual;

— datum;

—nominal.

NOTE See 4.2.158.1.1 to 4.2.158.1.3 for the definition of each allowable value for dimension_note.

4.2.158.1.1 actual: tolerance is for actual value.

4.2.158.1.2 datum: tolerance is for a datum value.

4.2.158.1.3 nominal: tolerance is for a nominal value.

4.2.158.2 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 4.3.220 for the application assertion.

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

4.2.159 Location_element

A Location_element is a type of Shape_element (see 4.2.238) that is a Shape_aspect representation for a reference point.

NOTE This application object is harmonized with the same entity from paragraph 4.2.122 in ISO 10303-224:2006.

4.2.160 Location_tolerance

A Location_tolerance is a type of Dimensional_tolerance (see 4.2.68) 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), Location_dimension-_tolerance (see 4.2.158), Coordinate_tolerance (see 4.2.47) or Distance_along_curve_tolerance (see 4.2.71).

NOTE This application object is harmonized with the same entity from paragraph 4.2.123 in ISO 10303-224:2006.

The data associated with a Location_tolerance are the following:

— directed;

— origin_shape;

— termination_shape.

4.2.160.1 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.

4.2.160.2 origin_shape

The origin_shape specifies the shape on the Part that defines the starting position for a Location _tolerance. See 4.3.221 for the application assertion.

4.2.160.3 termination_shape

The termination_shape specifies the shape on the Part that defines the ending position for a Location_tolerance. See 4.3.221 for the application assertion.

4.2.161 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. The Loop_slot_end_type is an end condition for the Closed_slot (see 4.2.37).

NOTE This application object is harmonized with the same entity from paragraph 4.2.124 in ISO 10303-224:2006.

4.2.162 Machining_feature

A Machining_feature is a type of Manufacturing_feature (see 4.2.163) 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: Knurl (see 4.2.152), Multi_axis_feature (see 4.2.169) Outer_round (see 4.2.182), Revolved_-feature (see 4.2.225), Thread (see 4.2.262), Marking (see 4.2.165), Spherical_cap (see 4.2.245), or a Compound_feature (see 4.2.40).

NOTE This application object is harmonized with the same entity from paragraph 4.2.126 in ISO 10303-224:2006.

The data associated with a Machining_feature are the following:

— placement;

— usage_name.

4.2.162.1 placement

The placement specifies the position and orientation of a Machining_feature relative to the base shape for a part. See 4.3.222 for the application assertion.

4.2.162.2 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.

4.2.163 Manufacturing_feature

A Manufacturing_feature is a type of Shape_element (see 4.2.238) that identifies the types of features necessary to manufacture a machined part. Each Manufacturing_feature is either a Machining_feature (see 4.2.162) a Replicate_feature (see 4.2.224), or a Transition_feature (see 4.2.274)

NOTE This application object is harmonized with the same entity from paragraph 4.2.129 in ISO 10303-224:2006.

4.2.164 Manufacturing_feature_group

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

NOTE This application object is harmonized with the same entity from paragraph 4.2.130 in ISO 10303-224:2006.

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.

4.2.164.1 feature_groups

The feature_group specifies the list of Manufacturing_feature (see 4.2.163) or the Manufacturing_-feature_group (see 4.2.164) to be grouped. See 4.3.223 and 4.3.224 for the application assertion.

4.2.164.2 group_description

The group_description specifes additional text information about the Manufacturing_feature_group.

4.2.164.3 group_name

The group_name specifies identification for the Manufacturing_feature_group.

4.2.165 Marking

A Marking is a type of Machining_feature (see 4.2.162) 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 4.2.22).

NOTE This application object is harmonized with the same entity from paragraph 4.2.131 in ISO 10303-224:2006.

The data associated with a Marking are the following:

— applied_to_shape;

— text.

4.2.165.1 applied_to_shape

The applied_to_shape specifies a base shape for applying the Marking feature. See 4.3.226 for the application assertion.

4.2.165.2 text

The text specifies the characters that will be applied to the part. See 4.3.225 for the application assertion.

4.2.166 Material_condition_modifier

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

NOTE This application object is harmonized with the same entity from paragraph 4.2.133 in ISO 10303-224:2006.

The data associated with a Material_condition_modifier are the following:

— material_type.

4.2.166.1 material_type

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

EXAMPLE maximum material condition (MMC), least material condition (LMC) and regardless of feature size (RFS).

4.2.167 Measurement_Location

A Measurement_location is the location and name of the dimensional measurement device. The data associated with a Measurement_location are the following:

The data associated with a Measurement_Location are the following:

— location;

— machine.

4.2.167.1 location

The location specifies the location of the dimensional measurement device.

4.2.167.2 machine

The machine specifies the name of the dimensional measurement device.

4.2.168 Measurement_uncertainty

A Measurement_uncertainty is the uncertainty qualifier for a measdured value. The data associated with a Measurement_uncertainty are the following:

The data associated with a Measurement_uncertainty are the following:

— uncertainty_name;

— uncertainty_value;

— coverage_factor.

4.2.168.1 uncertainty_name

The uncertainty_name specifies the kind of measure qualifier being definied..

4.2.168.2 uncertainty_value

The uncertainty_value specifies the qualified uncertainty value.

4.2.168.3 coverage_factor

The coverage_factor specifies the multiplier of the uncertainty of the value.

4.2.169 Multi_axis_feature

A Multi_axis_feature is a type of Machining_feature (see 4.2.162) that identifies milling features for a part, and not turned features. Each Multi_axis_feature may be one of the following: Boss (see 4.2.10), General_removal_volume (see 4.2.137), Hole (see 4.2.149), Rounded_end (see 4.2.231), Planar_face (see 4.2.196), Pocket (see 4.2.202), Profile_feature (see 4.2.206), Protrusion (see 4.2.210), Rib_top (see 4.2.228), Slot (see 4.2.241), and Step (see 4.2.249).

NOTE This application object is harmonized with the same entity from paragraph 4.2.140 in ISO 10303-224:2006.

The data associated with Multi_axis_feature are the following:

— maximum_feature_limit.

4.2.169.1 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 4.3.227 for the application assertion.

4.2.170 Ngon_base_shape

An Ngon_base_shape is a type of Implicit_base_shape_representation (see 4.2.150) that specifies the initial shape of the material as a polygon with any number of equal sides.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.141 in ISO 10303-224:2006.

NOTE 2 Figure 95 illustrates the Ngon_base_shape.

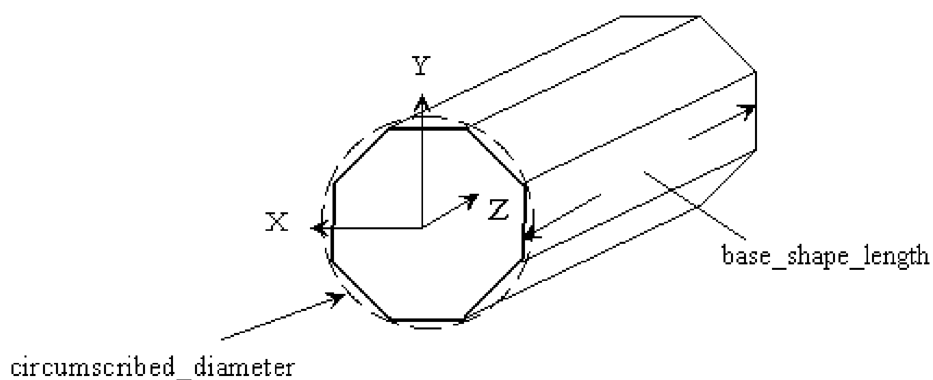


Figure 95 — Ngon_base_shape

The data associated with a Ngon_base_shape are the following:

— circumscribed_or_across_flats;

— corner_radius;

— diameter;

— number_of_sides.

4.2.170.1 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.

4.2.170.2 corner_radius

The corner_radius specifies the size of an arc blend between two sides of the ngon. See 4.3.228 for the application assertion.

4.2.170.3 diameter

The diameter specifies specifies the size of either the circumscribed diameter, or the diameter across the flats. See 4.3.228 for the application assertion.

4.2.170.4 number_of_sides

The number_of_sides specifies how many sides are needed for the Ngon. See 4.3.228 for the application assertion.

4.2.171 Ngon_profile

An Ngon_profile is a type of Closed_profile (see 4.2.36) 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 1 This application object is harmonized with the same entity from paragraph 4.2.142 in ISO 10303-224:2006.

NOTE 2 Figure 96 illustrates the Ngon_profile.

The data associated with a Ngon_profile are the following:

— circumscribed_or_across_flats;

— corner_radius;

— diameter;

— number_of_sides.

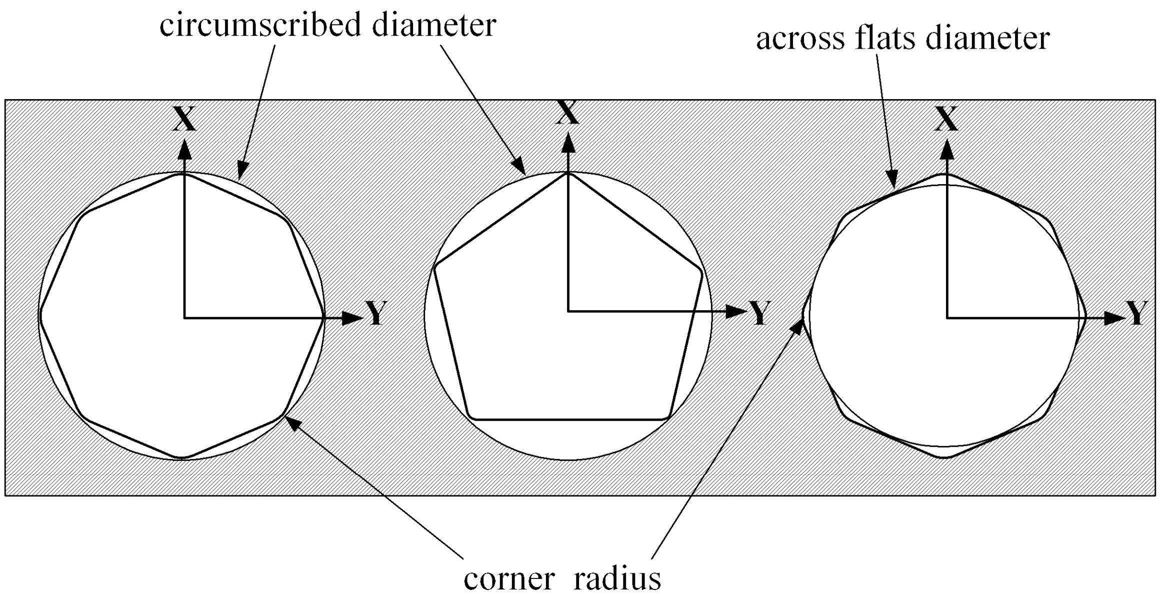


Figure 96 — Ngon_profile

4.2.171.1 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.

4.2.171.2 corner_radius

The corner_radius specifies the size of an arc blend between two sides of the ngon. See 4.3.229 for the application assertion.

4.2.171.3 diameter

The diameter specifies specifies the size of either the circumscribed diameter, or the diameter across the flats. See 4.3.229 for the application assertion.

4.2.171.4 number_of_sides

The number_of_sides specifies how many sides are needed for the Ngon. See 4.3.229 for the application assertion.

4.2.172 Numeric_parameter

A Numeric_parameter is a type of Property_parameter (see 4.2.209) that has 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 4.2.173).

NOTE This application object is harmonized with the same entity from paragraph 4.2.143 in ISO 10303-224:2006.

The data associated with a Numeric_parameter are the following:

— parameter_units;

— parameter_value.

4.2.172.1 parameter_units

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

EXAMPLE watt, meters, degrees, etc.

4.2.172.2 parameter_value

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

4.2.173 Numeric_parameter_with_tolerance

A Numeric_parameter_with_tolerance is a type of Numeric_parameter (see 4.2.172) with an implied tolerance value.

NOTE 1 A thread has an 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.

NOTE 2 This application object is harmonized with the same entity from paragraph 4.2.144 in ISO 10303-224:2006.

The data associated with a Numeric_parameter_with_tolerance are the following:

— implicit_tolerance.

4.2.173.1 implicit_tolerance

The implicit_tolerance specifies the type of tolerance to apply to a numeric parameter value. See 4.3.230 and 4.3.232 for the application assertion.

4.2.174 Offset_shape_element

An Offset_shape_element is a Shape_element (see 4.2.238) that is offset by a constant distance from another Shape_element, for dimensioning or tolerancing purposes. The data associated with an Offset_-shape_element are the following:

— offset;

— offset_direction

4.2.174.1 offset

The offset specifies the offset distance. See 4.3.234 for the application assertion.

4.2.174.2 offset_direction

The offset_direction specifies the direction an offset is being applied to a Shape_element(see 4.2.238). See 4.3.234 for the application assertion.

4.2.175 Open_profile

An Open_profile is a type of Profile (see 4.2.205) 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 (see4.2.130), Linear_profile (see 4.2.156), Partial_circular_profile (see 4.2.188), Rounded_U_profile (see 4.2.232), Square_U_profile (see 4.2.248), Tee_profile (see 4.2.260), or a Vee_profile (see 4.2.276).

NOTE This application object is harmonized with the same entity from paragraph 4.2.145 in ISO 10303-224:2006.

The data associated with an Open_profile are the following:

— profile_limit.

4.2.175.1 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 4.3.236 for the application assertion.

4.2.176 Open_slot

An Open_slot is a type of Slot (see 4.2.241) that has a course of travel defined by a Path that is open and has openings.

NOTE This application object is harmonized with the same entity from paragraph 4.2.146 in ISO 10303-224:2006.

The data associated with an Open_slot are the following:

— course_of_travel;

— end_conditions.

4.2.176.1 course_of_travel

The course_of_travel specifies a 3D space curve, that when combined with a Profile, creates the shape of the Slot. See 4.3.237 for the application assertion.

4.2.176.2 end_conditions

The end_conditions specifies the type of implicit shape at the ends of the Slot. See 4.3.238 for the application assertion.

4.2.177 Open_slot_end_type

An Open_slot_end_type is a type of Slot_end_type (see 4.2.242) that is an end condition of a slot that shall pass through the end of the part.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.147 in ISO 10303-224:2006.

NOTE 2 Figure 97 illustrates the Open_slot_end_type.

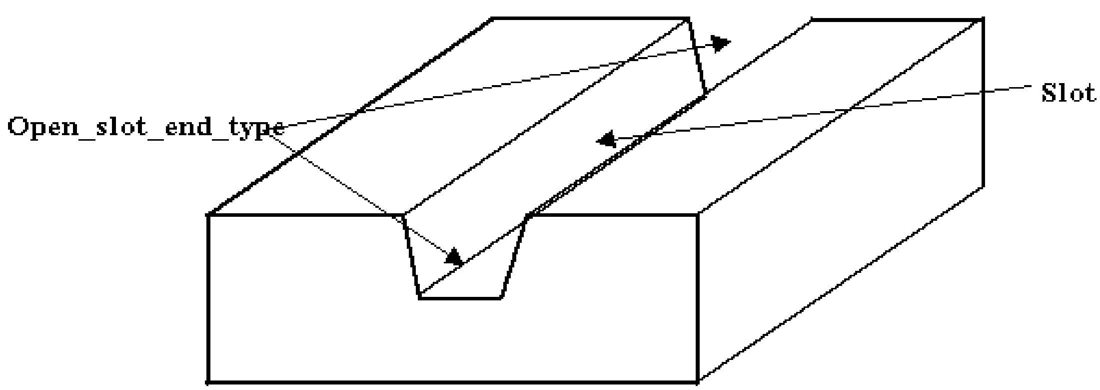


Figure 97 — Open_slot_end_type

4.2.178 Organization

An Organization is a group of people who are involved in a particular process or who share a common interest. The data associated with an Organization are the following:

— address;

— id;

— long_name;

— short_name

4.2.178.1 address

The address specifies the unique identification of the postal delivery or geographic location of an Organization. See 4.3.239 for the application assertion.

4.2.178.2 id

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

4.2.178.3 short_name

The short_name specifies a short, abbreviated name of the organization.

4.2.178.4 long_name

The long_name specifies the full formal name of the organization.

4.2.179 Orientation

An Orientation is the direction and location of the basic shape of a part, a feature on the part, components of a feature (which are Profile objects and Path objects), or a coordinate system of the measuring machine.

NOTE This application object is harmonized with the same entity from paragraph 4.2.150 in ISO 10303-224:2006.

The data associated with an Orientation are the following:

— axis;

— location.

4.2.179.1 axis

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

4.2.179.2 location

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

4.2.180 Outer_diameter

An Outer_diameter is a type of Outer_round (see 4.2.182) that is a sweeping of an outline specified by a line segment one complete revolution about an axis. The line is finite in length and coplanar with the axis. An Outer_diameter may have a constant diameter around the axis of rotation, or it may be tapered.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.151 in ISO 10303-224:2006.

NOTE 2 Figure 98 illustrates the Outer_diameter with and without a taper.

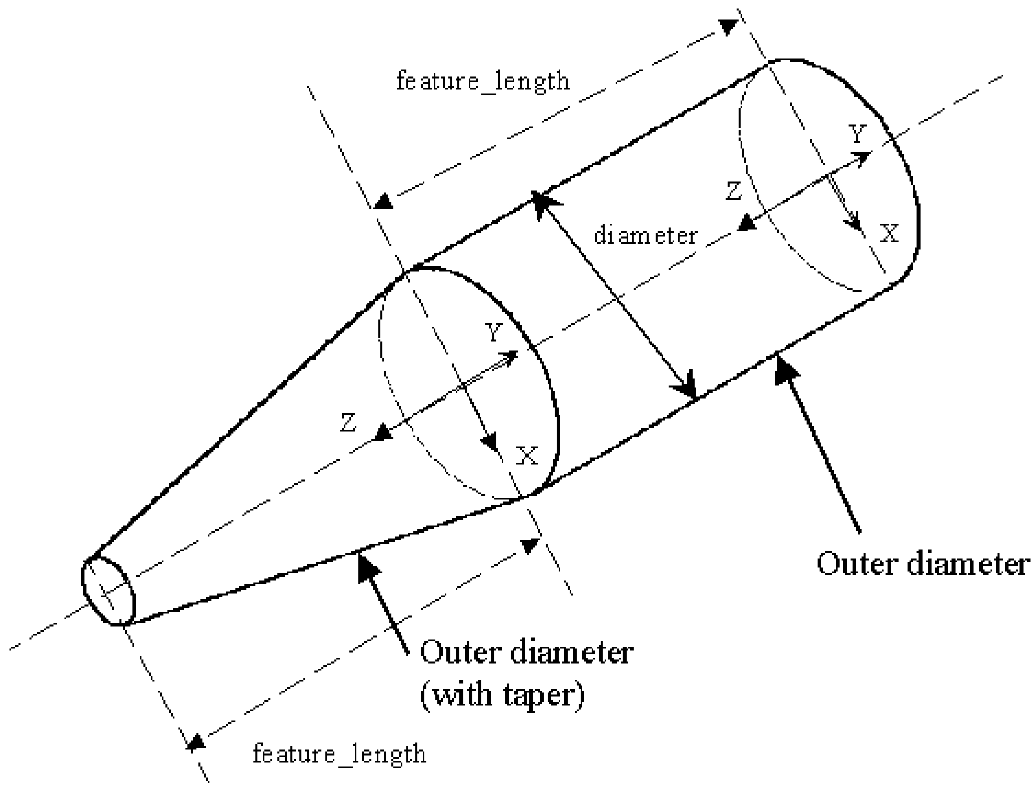


Figure 98 — Outer_diameter

The data associated with an Outer_diameter are the following:

— diameter;

— feature_length;

— reduced_size.

4.2.180.1 diameter

The diameter specifies the maximum diametric size of an Outer_diameter feature. See 4.3.240 for the application assertion.

4.2.180.2 feature_length

The length specifies the size of an Outer_diameter feature, measured along the feature's axis. See 4.3.240 for the application assertion.

4.2.180.3 reduced_size

The reduced_size specifies the constant change in the Outer_diameter along the feature length. See 4.3.241, 4.3.242 and 4.3.243 for the application assertion.

4.2.181 Outer_diameter_to_shoulder

An Outer_diameter_to_shoulder is a type of Outer_round (see 4.2.182) 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 1 This application object is harmonized with the same entity from paragraph 4.2.152 in ISO 10303-224:2006.

NOTE 2 Figure 99 illustrates the Outer_diameter_to_shoulder.

The data associated with an Outer_diameter_to_shoulder are the following:

— diameter;

— feature_length;

— v_shape_boundary.

4.2.181.1 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 4.3.244 for the application assertion.

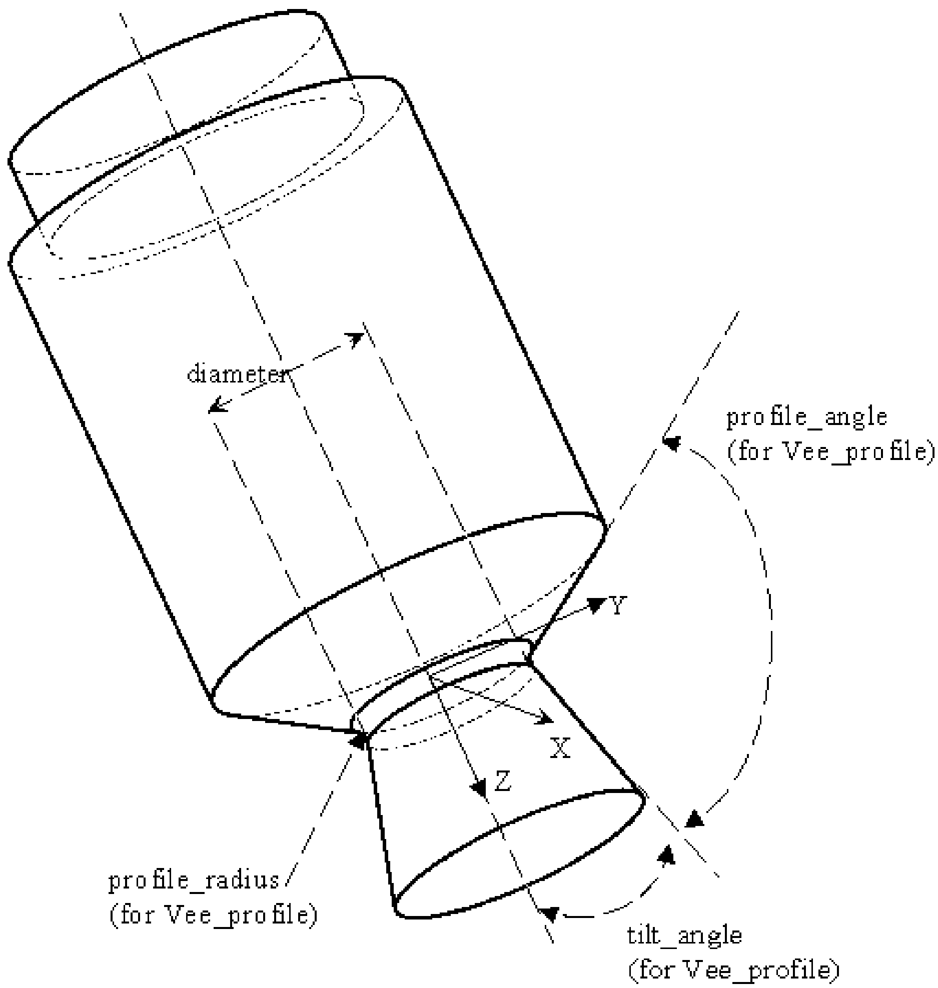


Figure 99 — Outer_diameter_to_shoulder

4.2.181.2 feature_length

The length specifies the size of an Outer_diameter feature_to_shoulder, measured along the feature's axis. See 4.3.244 for the application assertion.

4.2.181.3 v_shape_boundary

The v_shape_boundary specifies an outline or shape that shall be revolved about an axis. The Vee_profile (see 4.2.276) 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 4.3.245 for the application assertion.

4.2.182 Outer_round

An Outer_round is a type of Machining_feature (see 4.2.162) 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 4.2.180) or an Outer_diameter_to_shoulder (see 4.2.181). The axis of revolution shall be the same as the Z-axis of the feature.

NOTE This application object is harmonized with the same entity from paragraph 4.2.153 in ISO 10303-224:2006.

4.2.183 Parallelism_tolerance

A Parallelism_tolerance is a type of Geometric_tolerance (see 4.2.143) 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 100 illustrates Parallelism_tolerance for a plane.

NOTE 2 Figure 101 illustrates Parallelism_tolerance for an axis.

NOTE 3 The Parallelism_tolerance definition is derived from ISO 1101, 14.7.

NOTE 4 This application object is harmonized with the same entity from paragraph 4.2.154 in ISO 10303-224:2006.

The data associated with a Parallelism_tolerance are the following:

— affected_plane;

— geometric_reference;

— segment_size.

4.2.183.1 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 4.3.247 for the application assertion.

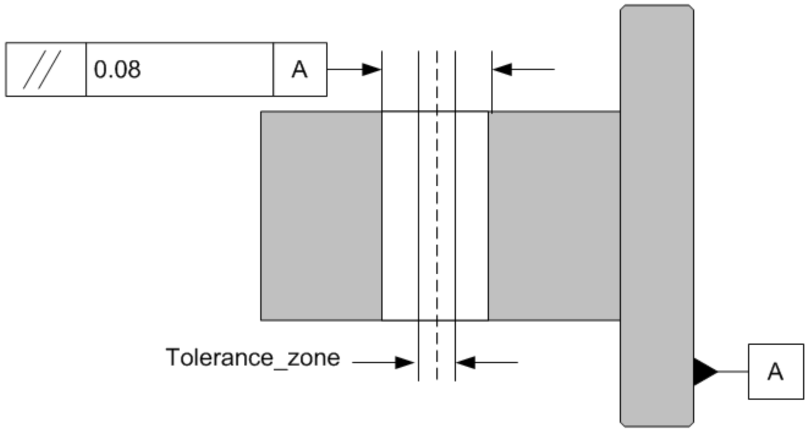


Figure 100 — Parallelism_tolerance for an axis

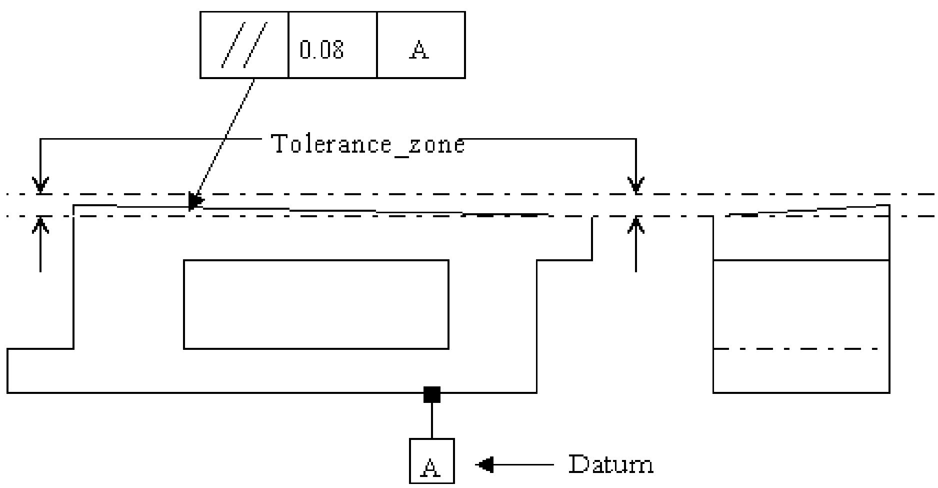


Figure 101 — Parallelism_tolerance for a plane

4.2.183.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.246 for the application assertion.

4.2.183.3 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.

4.2.184 Part

A Part is the physical item, which is intended to be produced through the manufacturing process.

The data associated with a Part are the following:

— lot_number_dml;

— part_description;

— part_id;

— part_name;

— part_revision_id;

— physical_form;

— security_classification;

— serial_number_dml;

— vendor_dml;

4.2.184.1 part_description

The part_description specifies a human interpretable summary of a 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.

4.2.184.2 part_id

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

4.2.184.3 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.

4.2.184.4 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 need not specify the latest version of a Part's design to be manufactured.

4.2.184.5 physical_form

The physical_form specifies the shape of the Part. See 4.3.248 for the application assertion.

4.2.184.6 lot_number_dml

The lot_number_dml specifies the lot number of the batch in which the part was fabricated

4.2.184.7 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.

4.2.184.8 serial_number_dml

The serial_number_dml specifies the serial number of the part, often engraved on it

4.2.184.9 vendor_dml

The vendor_dml specifies the vendor or manufacturer of a part.

4.2.185 Part_dimensioning_standard

A Part_dimensioning_standard is a type of Document_assignment (see 4.2.111) that is the reference to a document that defines the standard used to define the dimension tolerance used.

NOTE This application object is harmonized with the same entity from paragraph 4.2.156 in ISO 10303-224:2006.

The data associated with a Part_dimensioning_standard are the following:

— applied_part.

4.2.185.1 applied_part

The applied_part specifies the part that uses the dimensioning standard.See 4.3.249 for the application assertion.

4.2.186 Partial_area_definition

A Partial_area_definition is the limitations of a surface for applying a Machining_feature. Thread (see 4.2.262) and Knurl (see 4.2.152) 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 1 This application object is harmonized with the same entity from paragraph 4.2.159 in ISO 10303-224:2006.

NOTE 2 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.

4.2.186.1 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 4.3.250 for the application assertion.

4.2.186.2 maximum_length

The maximum_length specifies the dimension along a surface to apply a feature. A Thread is a type of Machining_feature 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 4.3.250 for the application assertion.

4.2.186.3 placement

The placement specifies where to locate the Partial_area_definition. See 4.3.251 for the application assertion.

4.2.187 Partial_circular_path

A Partial_circular_path is a type of Circular_path (see 4.2.32) 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 1 This application object is harmonized with the same entity from paragraph 4.2.160 in ISO 10303-224:2006.

NOTE 2 Figure 102 illustrates two Slot feature with a Square_U_profile and a Partial_circular_paths.

The data associated with a Partial_circular_path are the following:

— sweep_angle.

4.2.187.1 sweep_angle

The sweep_angle specifies the size of the angle to define an arc shaped path. See 4.3.252 for the application assertion.

4.2.188 Partial_circular_profile

A Partial_circular_profile is a type of Open_profile (see 4.2.175) 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 1 This application object is harmonized with the same entity from paragraph 4.2.161 in ISO 10303-224:2006.

NOTE 2 Figure 103 illustrates a Partial_circular_profile.

The data associated with a Partial_circular_profile are the following:

— radius;

— sweep_angle.

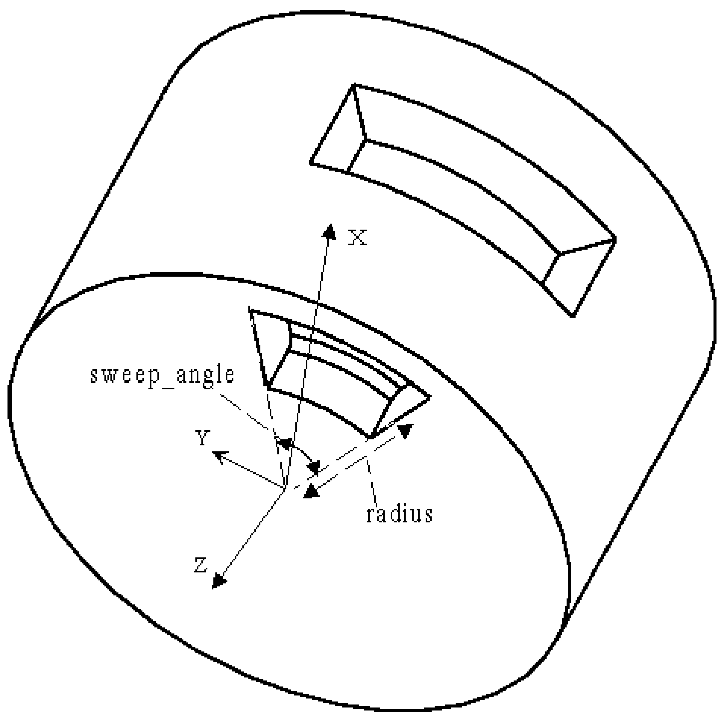


Figure 102 — Partial_circular_path

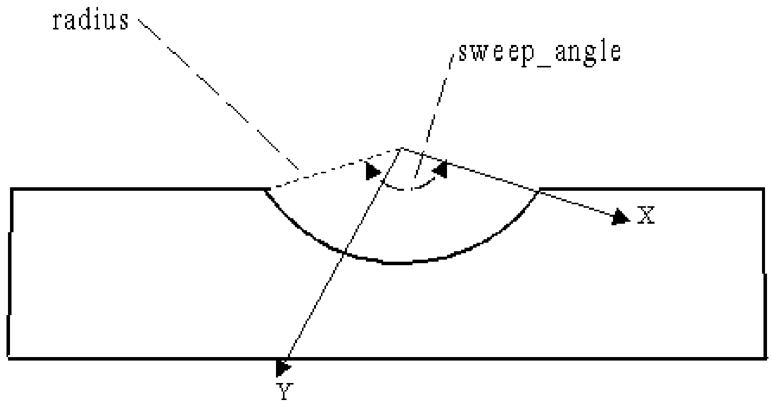


Figure 103 — Partial_circular_profile

4.2.188.1 radius

The radius specifies radius of the arc for a Partial_circular_profile. See 4.3.253 for the application assertion.

4.2.188.2 sweep_angle

The sweep_angle specifies the size of the angle to for a Partial_circular_profile. See4.3.253 for the application assertion.

4.2.189 Partial_circular_shape_profile

A Partial_circular_shape_profile is a type of Shape_profile (see 4.2.239) that defines a volume that is not enclosed on all sides.

NOTE This application object is harmonized with the same entity from paragraph 4.2.162 in ISO 10303-224:2006.

The data associated with a Partial_circular_shape_profile are the following:

— open_boundary.

4.2.189.1 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 4.3.254 for the application assertion.

4.2.190 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 4.2.32), General_path (see 4.2.132), or a Linear_path (see 4.2.155).

NOTE This application object is harmonized with the same entity from paragraph 4.2.163 in ISO 10303-224:2006.

The data associated with a Path are the following:

— placement.

4.2.190.1 placement

The placement specifies where to locate the Path. See 4.3.255 for the application assertion.

4.2.191 Path_element

A Path_element is a type of Shape_element (see 4.2.238) that is a continous set of geometric curves that represent the path for a particular Machining_feature.

NOTE This application object is harmonized with the same entity from paragraph 4.2.164 in ISO 10303-224:2006.

4.2.192 Perpendicularity_tolerance

A Perpendicularity_tolerance is a type of Geometric_tolerance (see 4.2.143) that specifies a requirement to control the perendicularity of a surface or a curve with respect to a datum plane.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.166 in ISO 10303-224:2006.

NOTE 2 Figure 104 and Figure 105 illustrates a Perpendicularity_tolerance.

NOTE 3 The Perpendicularity_tolerance definition is derived from ISO 1101, 14.8.

The data associated with a Perpendicularity_tolerance are the following:

— affected_plane;

— geometric_reference;

— segment_size.

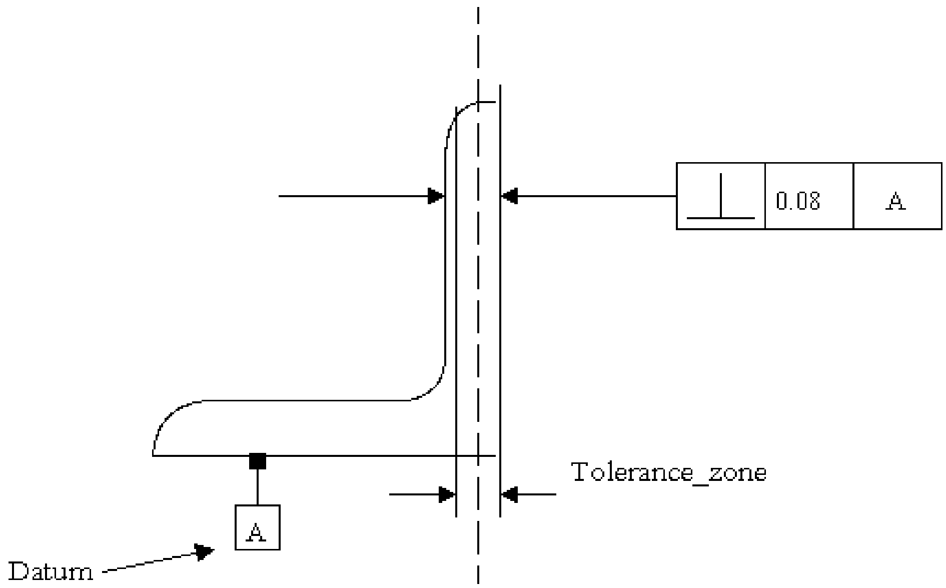


Figure 104 — Perpendicularity_tolerance

4.2.192.1 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 4.3.257 for the application assertion.

4.2.192.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.256 for the application assertion.

4.2.192.3 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.

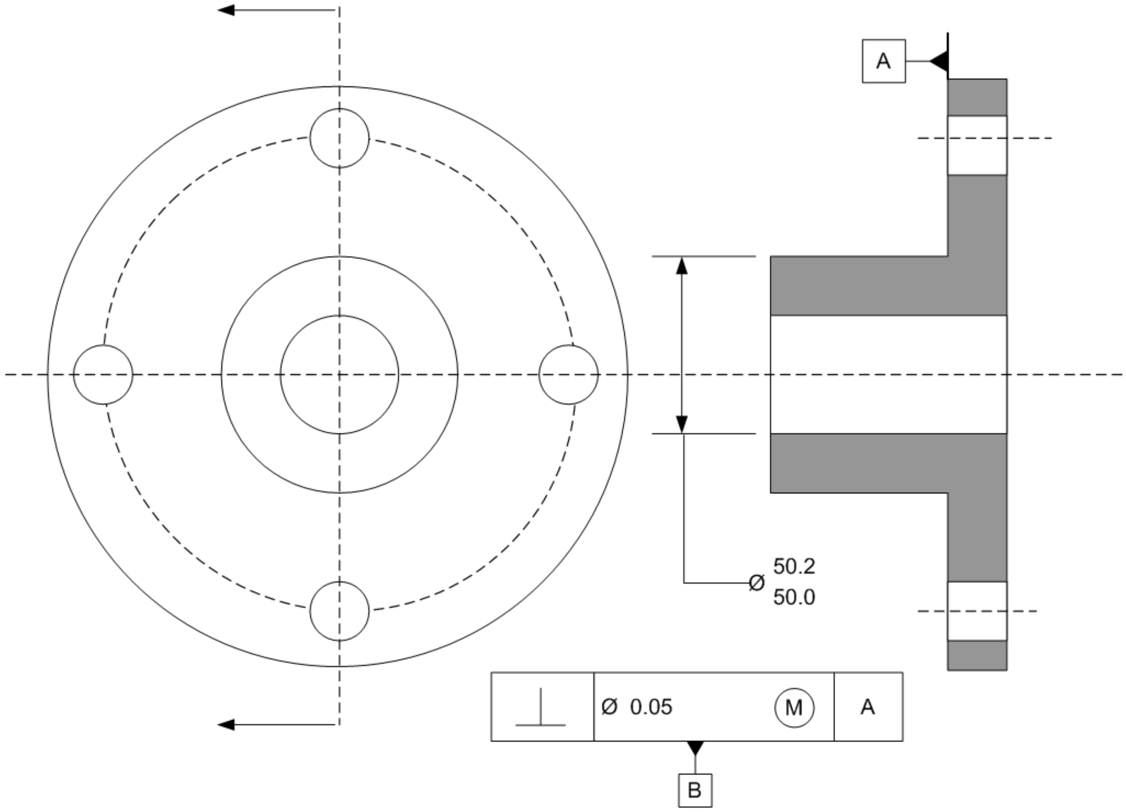


Figure 105 — Perpendicularity_tolerance for an axis

4.2.193 Person_in_organization

A Person_in_organization is the identification of an individual human being in an organisation.

The data associated with a Person_in_orgaization are the following:

— address;

— first_name;

— id;

— last_name;

— middle_names;

— prefix_titles;

— suffix_titles;

4.2.193.1 address

The address specifies the personal address of the person. See 4.3.258 for the application assertion.

4.2.193.2 first_name

The first_name specifies the person's given name.

4.2.193.3 id

The id specifies the unique identifier of the person.

4.2.193.4 last_name

The last_name specifies the person's surname.

4.2.193.5 middle_names

The middle_name specifies the person's middle name.

4.2.193.6 prefix_titles

The prefix_titles specifies the text that specifies the person's social or professional standing and appear before his or her names.

4.2.193.7 suffix_titles

The suffix_titles specifies the text that specifies the person's social or professional standing and appear after his or her names.

4.2.194 Placed_target

A Placed_target is a type of Datum_target (see 4.2.57) that is the implicit definition of a Datum_target. A Placed_target is either a Target_point (see 4.2.258), Target_line (see 4.2.257), Target_rectangle (see 4.2.259), or a Target_circle (see 4.2.256).

NOTE This application object is harmonized with the same entity from paragraph 4.2.169 in ISO 10303-224:2006.

The data associated with Placed_target are the following:

— is_defined_in;

— placement.

4.2.194.1 is_defined_in

The is_defined_in specifies the Geometric_coordinate_space in which the Geometric_model is defined. See 4.3.259 for the application assertion.

4.2.194.2 placement

The placement specifies location and orientation for the implicit definitions of the types of Placed_target. See 4.3.260 for the application assertion.

4.2.195 Planar_element

A Planar_element is a type of Shape_element (see 4.2.238) that is a flat surface.

NOTE This application object is harmonized with the same entity from paragraph 4.2.170 in ISO 10303-224:2006.

The data associated with a Planar_element are the following:

— location;

— normal.

4.2.195.1 location

The location specifies the position of the planar surface. See 4.3.262 for the application assertion.

4.2.195.2 normal

The normal specifies the vector that indicates the normal of a plane being defined for the planar surface. See 4.3.261 for the application assertion.

4.2.196 Planar_face

A Planar_face is a type of Multi_axis_feature (see 4.2.169) 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 1 This application object is harmonized with the same entity from paragraph 4.2.171 in ISO 10303-224:2006.

NOTE 2 Figure 106 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.

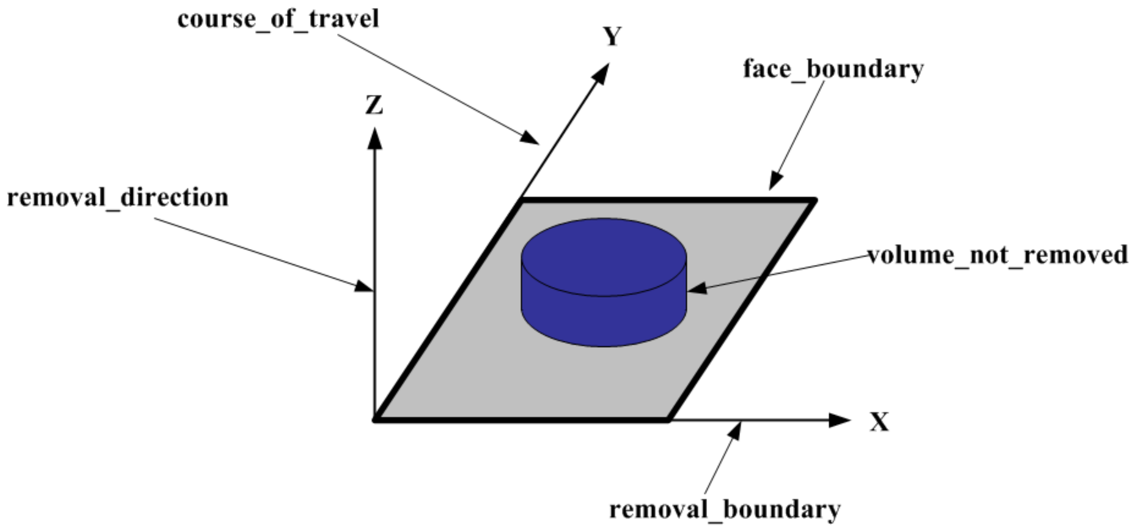


Figure 106 — Planar_face

4.2.196.1 course_of_travel

The course_of_travel specifies a straight line with magnitude and direction. The placement and orientation of the Linear_path(see 4.2.155) shall be the same as the Planar_face feature. See 4.3.265 for the application assertion.

4.2.196.2 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 4.3.263 for the application assertion.

4.2.196.3 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 4.3.263 for the application assertion.

4.2.196.4 removal_depth

The removal_depth specifies a measured distance from the bottom of a Planar_face to a point that is below the Planar_face feature measured along the removal_direction. See 4.3.267 for the application assertion.

4.2.196.5 removal_direction

The removal_direction specifies the direction of material removal from the Planar_face feature. See 4.3.266 for the application assertion.

4.2.196.6 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the pocket. The Boss (see 4.2.10) or Protrusion (see 4.2.210) feature defines the shape of the material that is to remain in the pocket. See 4.3.268 or 4.3.269 for the application assertion.

4.2.197 Planar_pocket_bottom_condition

A Planar_pocket_bottom_condition is a type of Pocket_bottom_condition (see 4.2.203) that characterizes the bottom of a pocket which is flat.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.172 in ISO 10303-224:2006.

NOTE 2 Figure 118 illustrates the Rectangular_open_pocket with a Planar_pocket_bottom_condition. Figure 114 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.

4.2.197.1 floor_location

The floor_location specifies the position of the bottom of a pocket feature for a planar floor. See 4.3.271 for the application assertion.

NOTE Figure 118 illustrates the Rectangular_open_pocket with a Planar_pocket_bottom_condition and a floor_location.

4.2.197.2 floor_normal

The floor_normal specifies the vector which indicates the normal of a plane being defined for the bottom of a pocket. See 4.3.270 for the application assertion.

NOTE Figure 118 illustrates the Rectangular_open_pocket with a Planar_pocket_bottom_condition and a floor_normal.

4.2.197.3 floor_radius

The floor_radius specifies the amount of curvature for an arc between the bottom and the sides of a Pocket feature. See 4.3.272 for the application assertion.

4.2.198 Planar_profile_floor

A Planar_profile_floor is a type of Profile_floor (see 4.2.207) that characterizes the bottom of a Shape_profile feature which is flat.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.173 in ISO 10303-224:2006.

NOTE 2 Figure 126 illustrates a Shape_profile with a Planar_profile_floor.

The data associated with a Planar_profile_floor are the following:

— floor.

4.2.198.1 floor

The floor specifies a planar floor for the Shape_profile (see 4.2.239) feature. The normal to the plane shall be in the direction away from the Planar_profile_floor feature. See 4.3.273 for the application assertion.

4.2.199 Planar_rib_top_floor

A Planar_rib_top_floor is a type of Rib_top_floor (see 4.2.229) that is the bottom of a Rib_top (see 4.2.228) which is flat.

NOTE This application object is harmonized with the same entity from paragraph 4.2.174 in ISO 10303-224:2006.

The data associated with a Planar_rib_top_floor are the following:

— boundary;

— floor_face.

4.2.199.1 boundary

The boundary specifies the complete or partial outside final shape of the Rib_top feature. See 4.3.274 for the application assertion.

4.2.199.2 floor_face

The floor_face specifies a planar floor for the Rib_top (see 4.2.228) feature. The normal to the plane shall be in the direction away from the Rib_top feature. See 4.3.275 for the application assertion.

4.2.200 Planar_top_condition

A Planar_top_condition is a type of Boss_top_condition (see 4.2.11) for a Boss that is flat.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.175 in ISO 10303-224:2006.

NOTE 2 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.

4.2.200.1 top_location

The top_location specifies the position of the top of a boss feature for a planar top. See 4.3.277 for the application assertion.

4.2.200.2 top_normal

The top_normal specifies the vector that indicates the normal of a plane being defined for the top of a boss. See 4.3.276 for the application assertion.

4.2.201 Plus_minus_value

The Plus_minus_value is the upper and lower limits or tolerance value applied directly to a dimension. When applied to a Dimensional_tolerance (see 4.2.68), the dimensional_value is the tolerance value. When applied to a Numeric_parameter_with_tolerance (see 4.2.173), the parameter_value is the tolerance value.

NOTE This application object is harmonized with the same entity from paragraph 4.2.176 in ISO 10303-224:2006.

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.

4.2.201.1 lower_limit

The lower_limit specifies the low limit value.

4.2.201.2 significant_digits

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

4.2.201.3 upper_limit

The upper_limit specifies the high limit value.

4.2.202 Pocket

A Pocket is a type of Multi_axis_feature (see 4.2.169) 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 4.2.51), General_pocket (see 4.2.134), Recess (see 4.2.213), Rectangular_closed_pocket (see 4.2.215), or a Rectangular_open_pocket (see 4.2.220).

NOTE This application object is harmonized with the same entity from paragraph 4.2.177 in ISO 10303-224:2006.

The data associated with a Pocket are the following:

— base_radius;

— bottom_condition;

— change_in_boundary;

— pocket_depth.

4.2.202.1 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 4.3.281 for the application assertion.

4.2.202.2 bottom_condition

The bottom_condition specifies the shape of the bottom of a Pocket feature. See 4.3.282 and 4.3.283 for the application assertion.

4.2.202.3 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 4.3.278 and 4.3.279 for the application assertion.

4.2.202.4 pocket_depth

The pocket_depth specifies a 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 as 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 4.3.280 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.

4.2.203 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 General_pocket_bottom_condition (see 4.2.135), Planar_pocket_bottom_condition (see 4.2.197), or a Through_pocket_bottom_condition (see 4.2.265).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.178 in ISO 10303-224:2006.

NOTE 2 Figure 107 illustrates the start_or_end attribute for a Rectangular_open_pocket.

The data associated with a Pocket_bottom_condition are the following:

— start_or_end.

4.2.203.1 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 4.2.155) for the Pocket, and a value of FALSE if it is at the end of the of the defining Linear_path for the Pocket.

4.2.204 Position_tolerance

A Position_tolerance is a type of Geometric_tolerance (see 4.2.143) that denotes a tolerance zone for a theoretically exact position_tolerance of a surface, and is established with respect to a datum.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.179 in ISO 10303-224:2006.

NOTE 2 Figure 108 illustrates the Position_tolerance.

NOTE 3 The Position_tolerance definition is derived from ISO 1101, 14.10

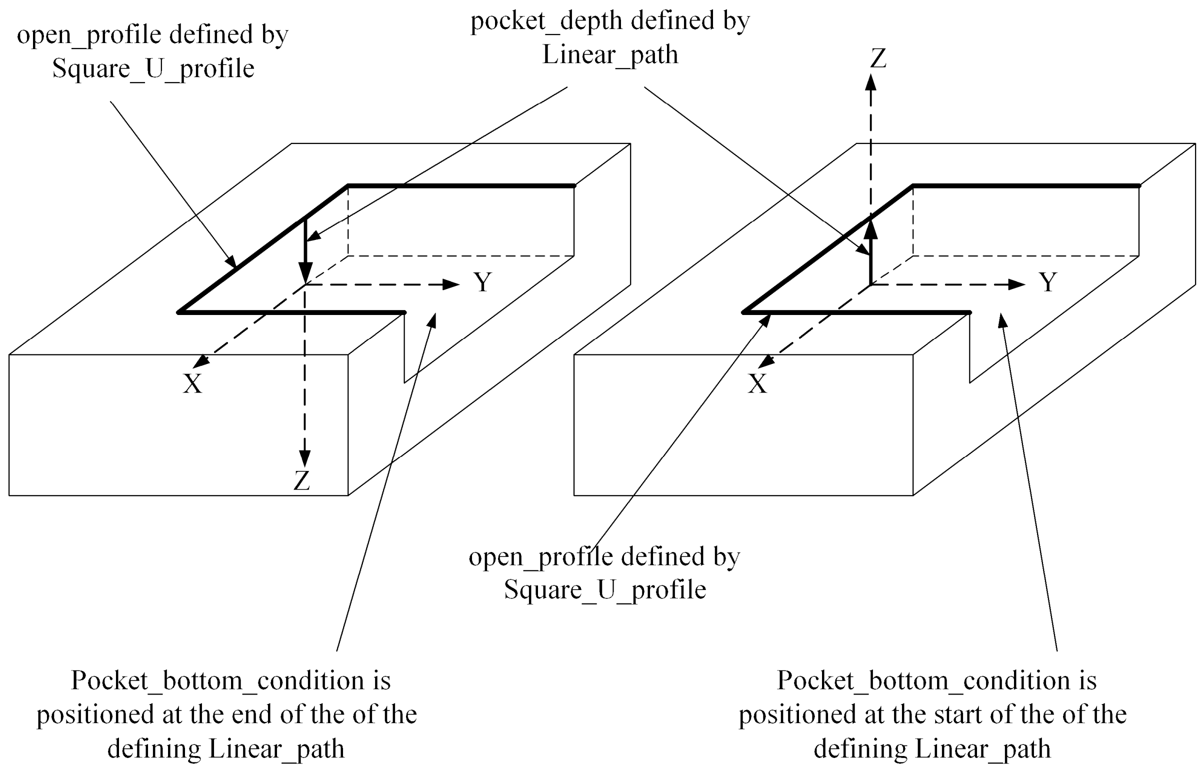


Figure 107 — Pocket_bottom_condition

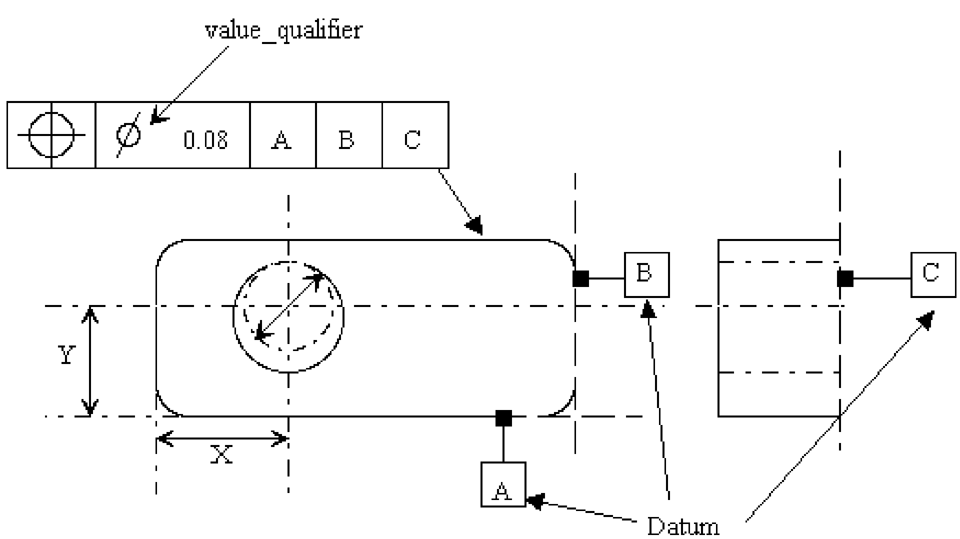


Figure 108 — Position_tolerance

The data associated with a Position_tolerance are the following:

— affected_plane;

— geometric_reference;

— value_qualifier.

4.2.204.1 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 4.3.285 for the application assertion.

4.2.204.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.284 for the application assertion.

4.2.204.3 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.

4.2.205 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 4.2.36) or an Open_profile (see 4.2.175)

NOTE This application object is harmonized with the same entity from paragraph 4.2.181 in ISO 10303-224:2006.

The data associated with a Profile are the following:

— placement.

4.2.205.1 placement

The placement specifies where to locate the Profile in reference to the Part orientation. See 4.3.286 for the application assertion.

4.2.206 Profile_feature

A Profile_feature is a type of Multi_axis_feature (see 4.2.169) that is the removal of excess material from the boundary shape of a part. Each Profile_feature is either a General_outside_profile (see 4.2.131), or a Shape_profile (see 4.2.239).

NOTE This application object is harmonized with the same entity from paragraph 4.2.182 in ISO 10303-224:2006.

The data associated with a Profile_feature are the following:

— profile_swept_shape.

4.2.206.1 profile_swept_shape

The profile_swept_shape specifies an implicit 2D line (see 4.2.156) 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 as 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 4.3.287 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.

4.2.207 Profile_floor

A Profile_floor is the bottom condition for a Shape_profile (see 4.2.239) The Profile_floor may be flat, or any arbitrary shape. Each Profile_floor is either a General_profile_floor (see 4.2.136) or a Planar_profile_floor (see 4.2.198.

NOTE This application object is harmonized with the same entity from paragraph 4.2.183 in ISO 10303-224:2006.

The data associated with a Profile_floor are the following:

— floor_radius;

— start_or_end.

4.2.207.1 floor_radius

The floor_radius specifies the radius of curvature for an arc between the bottom and the sides of a Pocket feature. See 4.3.288 for the application assertion.

4.2.207.2 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.

4.2.208 Projection

A Projection is an extension of a feature from the part so that a tolerance zone can be created. A feature is extended from one end of a feature for a specified length.

NOTE This application object is harmonized with the same entity from paragraph 4.2.185 in ISO 10303-224:2006.

The data associated with a Projection are the following:

— projection_end;

— projection_length.

4.2.208.1 projection_end

The projection_end specifies the physical shape that is the extension of a feature. See 4.3.289 for the application assertion.

4.2.208.2 projection_length

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

4.2.209 Property_parameter

A Property_parameter is an element of information that describes a characteristic that comprises the property. Each Property_parameter may be one of the following: Descriptive_parameter (see 4.2.63) or a Numeric_parameter (see 4.2.172).

NOTE This application object is harmonized with the same entity from paragraph 4.2.188 in ISO 10303-224:2006.

The data associated with a Property_parameter are the following:

— parameter_name.

4.2.209.1 parameter_name

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

4.2.210 Protrusion

A Protrusion is a type of Multi_axis_feature (see 4.2.169) that is an arbitrary shape that extends out from a surrounding surface.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.190 in ISO 10303-224:2006.

NOTE 2 Figure 109 illustrates the Protrusion feature.

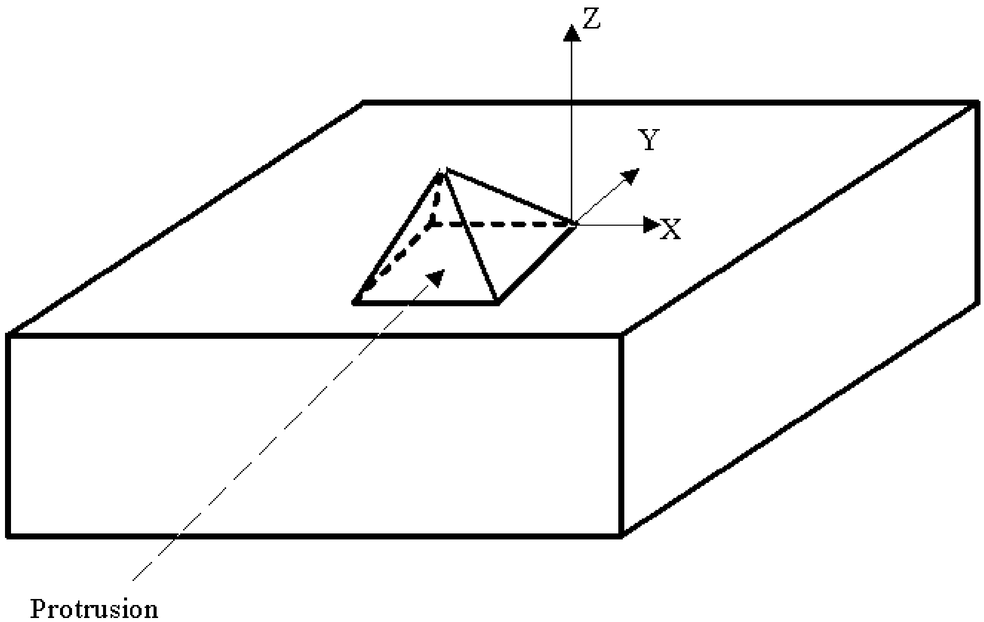


Figure 109 — Protrusion

The data associated with a Protrusion are the following:

— shape_volume.

4.2.210.1 shape_volume

The shape_volume specifies the arbitrary shape that defines the shape of a Protrusion. See 4.3.290 for the application assertion.

4.2.211 Radial_dimension_tolerance

A Radial_dimension_tolerance is a type of Size_tolerance (see 4.2.240) that is the allowable variation for the radial distance from the center of a circular curve to a point on the curve.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.191 in ISO 10303-224:2006.

NOTE 2 Figure 110 illustrates the Radial_dimension_tolerance.

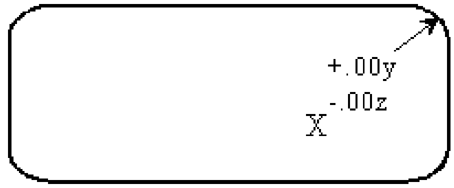


Figure 110 — Radial_dimension_tolerance

4.2.212 Radiused_slot_end_type

A Radiused_slot_end_type is a type of Slot_end_type (see 4.2.242) that is an end condition of a slot that shall be a cylindrical shape tangent to both of the adjacent Slot wall surfaces.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.192 in ISO 10303-224:2006.

NOTE 2 Figure 111 illustrates the Radiused_slot_end_type.

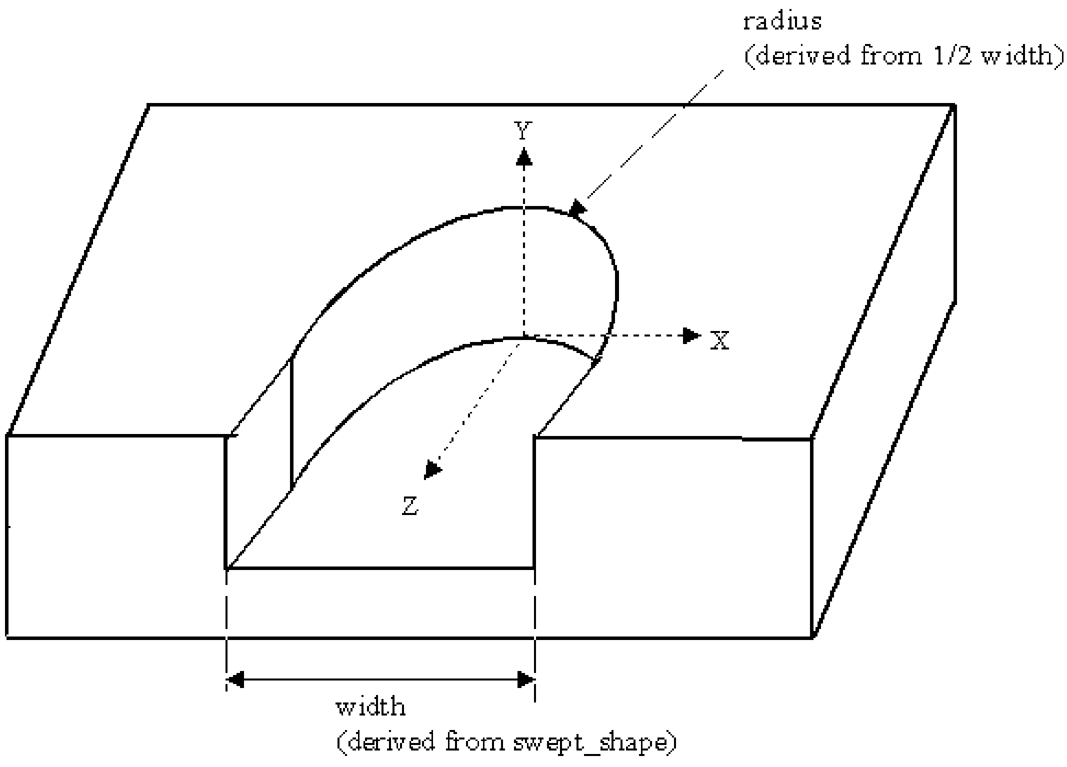


Figure 111 — Radiused_slot_end_type

4.2.213 Recess

A Recess is a type of a Pocket (see 4.2.202) that has a bottom, and a floor radius (see 4.2.213.2), but the sides of the recess 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 4.2.203).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.193 in ISO 10303-224:2006.

NOTE 2 Figure 112 illustrates the Recess feature.

The data associated with a Recess are the following:

— bottom_condition;

— fillet_boundary;

— volume_not_removed.

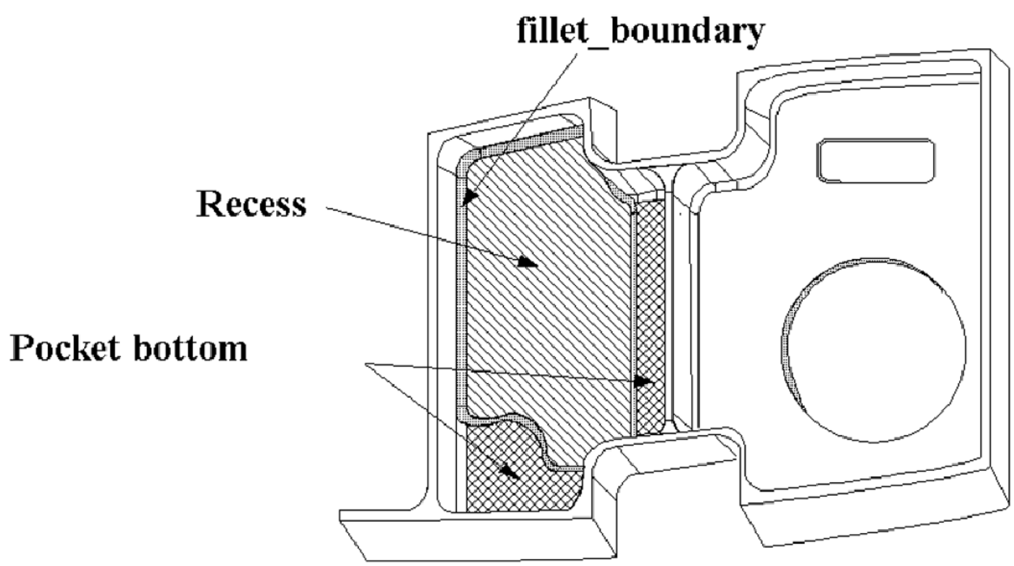


Figure 112 — Recess

4.2.213.1 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 4.3.292 for the application assertion.

4.2.213.2 fillet_boundary

The fillet_boundary specifies an outline or shape that is an enclosed area that shall be a closed 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 4.3.291 for the application assertion.

4.2.213.3 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the pocket. The Boss (see 4.2.10) or Protrusion (see 4.2.210) feature defines the shape of the material that is to remain in the recess 4.3.293 or 4.3.294 for the application assertion.

4.2.214 Rectangular_boss

A Rectangular_boss is a type of Boss (see 4.2.10) that is an enclosed volume with opposite sides that are equal in length.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.194 in ISO 10303-224:2006.

NOTE 2 Figure 113 illustrates the Rectangular_boss. The lengths are determined in the profile view of the Boss.

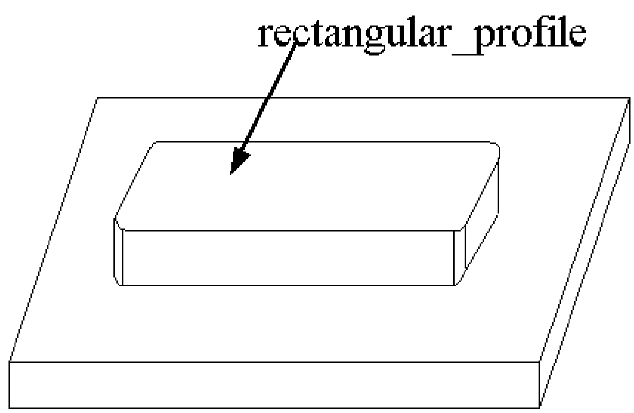


Figure 113 — Rectangular_boss

The data associated with a Rectangular_boss are the following:

— change_in_boundary;

— rectangular_profile.

4.2.214.1 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 4.3.295 and 4.3.296 for the application assertion.

4.2.214.2 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 4.3.297 for the application assertion.

4.2.215 Rectangular_closed_pocket

A Rectangular_closed_pocket is a type of Pocket (see 4.2.202) that is an enclosed volume with opposite sides that are equal in length.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.195 in ISO 10303-224:2006.

NOTE 2 Figure 113 illustrates the Rectangular_closed_pocket

The data associated with a Rectangular_closed_pocket are the following:

— closed_boundary;

— volume_not_removed.

4.2.215.1 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 4.3.298 for the application assertion.

4.2.215.2 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the pocket. The Boss (see 4.2.10) or Protrusion (see 4.2.210) feature defines the shape of the material that is to remain in the pocket. See 4.3.299 or 4.3.300 for the application assertion.

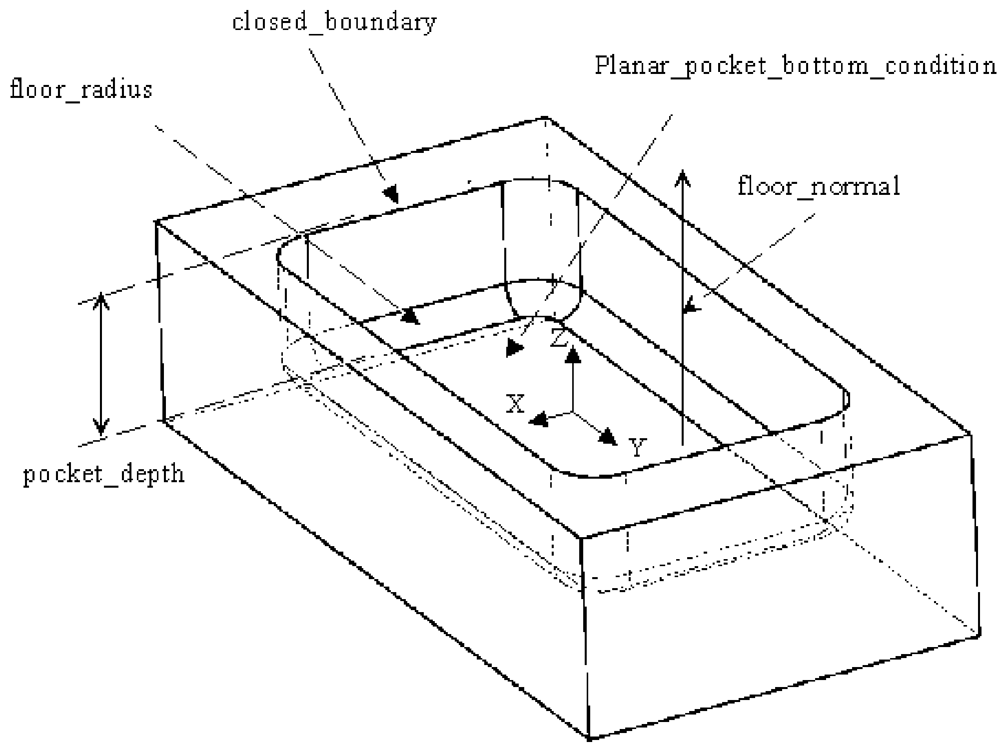


Figure 114 — Rectangular_closed_pocket

4.2.216 Rectangular_closed_profile

A Rectangular_closed_profile is a type of Closed_profile (see 4.2.36) 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 1 This application object is harmonized with the same entity from paragraph 4.2.196 in ISO 10303-224:2006.

NOTE 2 Figure 115 illustrates the Rectangular_closed_profile.

The data associated with a Rectangular_closed_profile are the following:

— corner_radius;

— profile_length;

— profile_width.

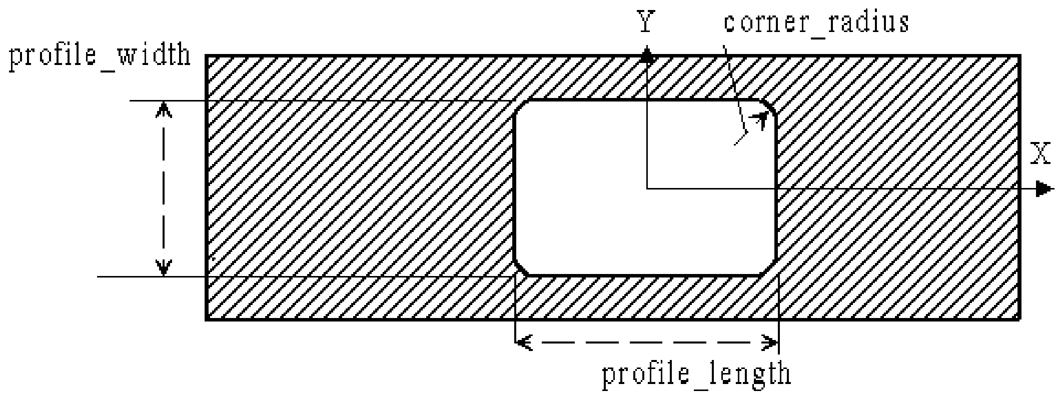


Figure 115 — Rectangular_closed_profile

4.2.216.1 corner_radius

The corner_radius specifies the size of the arc in all four corners of the rectangular profile. See 4.3.301 for the application assertion.

4.2.216.2 profile_length

The profile_length specifies the length of the side, along the X-axis, of the rectangular profile. See 4.3.301 for the application assertion.

4.2.216.3 profile_width

The profile_width specifies the length of the side, along the Y-axis, of the rectangular profile. See 4.3.301 for the application assertion.

4.2.217 Rectangular_closed_shape_profile

The Rectangular_closed_shape_profile is a type of Shape_profile (see 4.2.239) that is an enclosed volume with opposite sides that are equal in length.

NOTE This application object is harmonized with the same entity from paragraph 4.2.197 in ISO 10303-224:2006.

The data associated with a Rectangular_closed_shape_profile are the following:

— closed_boundary.

4.2.217.1 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 4.3.302 for the application assertion.

4.2.218 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 1 This application object is harmonized with the same entity from paragraph 4.2.198 in ISO 10303-224:2006.

NOTE 2 Figure 116 illustrates the Rectangular_offset_pattern.

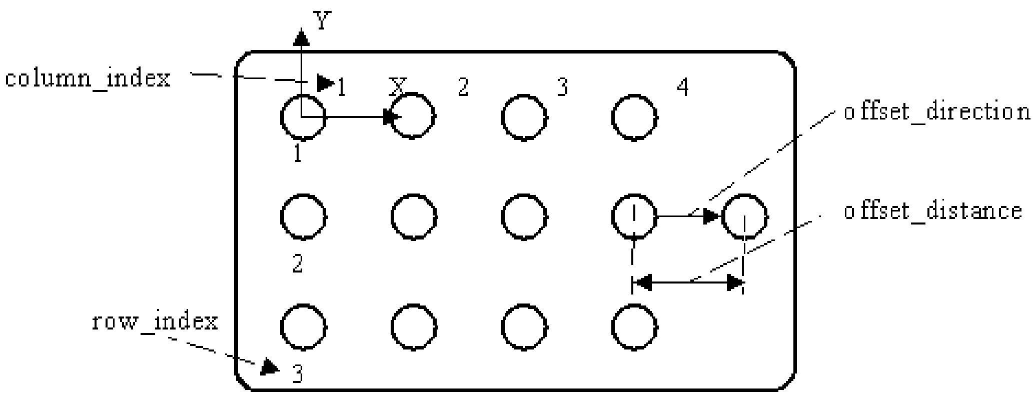


Figure 116 — Rectangular_offset_pattern

The data associated with a Rectangular_offset_pattern are the following:

— column_index;

— offset_direction;

— offset_distance;

— row_index.

4.2.218.1 column_index

The column_index specifies the unique identification for a feature in a column of multiple features. See 4.3.304 for the application assertion.

4.2.218.2 offset_direction

The offset_direction specifies the direction to offset a base feature from it's original position in the rectangular pattern. See 4.3.303 for the application assertion.

4.2.218.3 offset_distance

The offset_distance specifies the amount of offset from a feature location in a Rectangular_pattern for placing another feature. See 4.3.304 for the application assertion.

4.2.218.4 row_index

The row_index specifies the unique identification for a feature in a row of multiple features. See 4.3.304 for the application assertion.

4.2.219 Rectangular_omit_pattern

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

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.199 in ISO 10303-224:2006.

NOTE 2 Figure 117 illustrates the Rectangular_offset_pattern.

The data associated with a Rectangular_omit_pattern are the following:

— column_index;

— row_index.

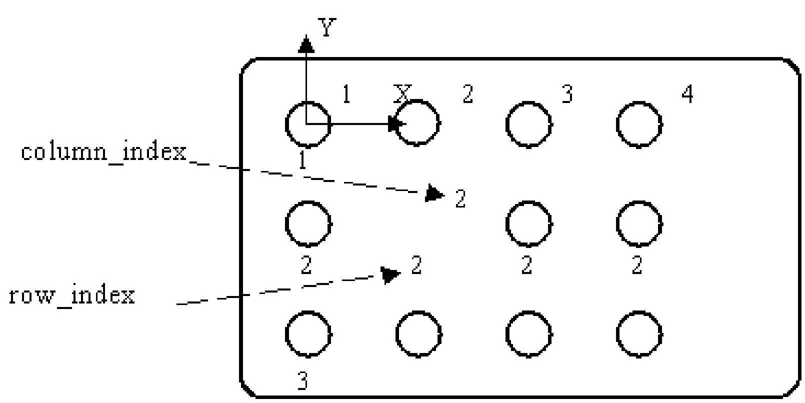


Figure 117 — Rectangular_omit_pattern

4.2.219.1 column_index

The column_index specifies the unique identification for a feature in a column of multiple features. See 4.3.305 for the application assertion.

4.2.219.2 row_index

The row_index specifies the unique identification for a feature in a row of multiple features. See 4.3.305 for the application assertion.

4.2.220 Rectangular_open_pocket

A Rectangular_open_pocket is a type of Pocket (see 4.2.202) 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 4.2.248) 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 (see 4.2.248) shall be the same as the Rectangular_open_pocket.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.200 in ISO 10303-224:2006.

NOTE 2 Figure 118 illustrates the Rectangular_open_pocket.

The data associated with a Rectangular_open_pocket are the following:

— open_boundary;

— volume_not_removed.

4.2.220.1 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 4.3.308 for the application assertion.

4.2.220.2 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 4.3.306 or 4.3.307 for the application assertion.

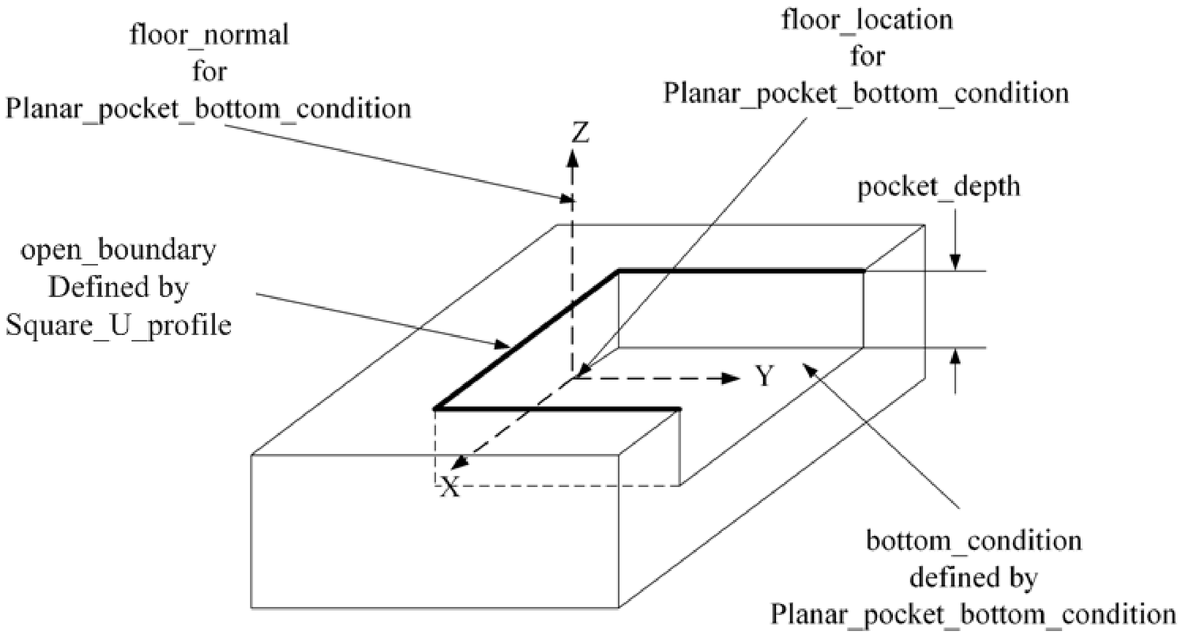


Figure 118 — Rectangular_open_pocket

4.2.221 Rectangular_open_shape_profile

The Rectangular_open_shape_profile is a type of Shape_profile (see 4.2.239) 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.

NOTE This application object is harmonized with the same entity from paragraph 4.2.201 in ISO 10303-224:2006.

The data associated with a Rectangular_open_shape_profile are the following:

— open_boundary.

4.2.221.1 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 4.3.309 for the application assertion.

4.2.222 Rectangular_pattern

A Rectangular_pattern is a type of Replicate_feature (see 4.2.224) that is a shape component arranged in a pattern of rows and columns.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.202 in ISO 10303-224:2006.

NOTE 2 Figure 119 illustrates the 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.

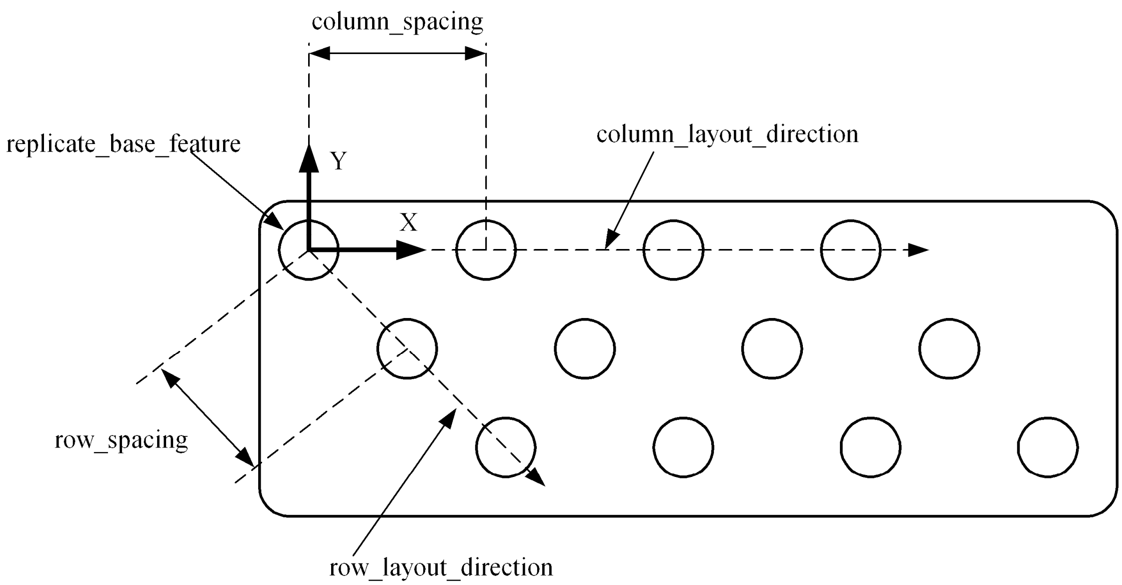


Figure 119 — Rectangular_pattern

4.2.222.1 column_layout_direction

The column_layout_direction specifies the linear direction for defining the columns of the pattern. See 4.3.310 for the application assertion.

4.2.222.2 column_spacing

The column_spacing specifies the amount of space between features in a Rectangular_pattern column. See 4.3.311 for the application assertion.

4.2.222.3 columns

The columns specifies the number of columns for placing features in the Rectangular_pattern. See 4.3.311 for the application assertion.

4.2.222.4 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 4.3.313 for the application assertion.

4.2.222.5 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 4.3.312 for the application assertion.

4.2.222.6 row_layout_direction

The row_layout_direction specifies the linear direction for defining the rows of the pattern. See 4.3.310 for the application assertion.

4.2.222.7 row_spacing

The row_spacing specifies the amount of space between features in a Rectangular_pattern row. See 4.3.311 for the application assertion.

4.2.222.8 rows

The rows specifies the number of rows for placing features in the Rectangular_pattern. See 4.3.311 for the application assertion.

4.2.223 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 4.2.162) or a Replicate_feature (see 4.2.224).

NOTE This application object is harmonized with the same entity from paragraph 4.2.203 in ISO 10303-224:2006.

The data associated with a Replicate_base are the following:

— base_feature.

4.2.223.1 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 4.2.162) or a Replicate_feature (see 4.2.224). See 4.3.314 and 4.3.315 for the application assertion.

4.2.224 Replicate_feature

A Replicate_feature is a type of Machining_feature (see 4.2.162) 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 4.2.33), General_pattern (see 4.2.133), or a Rectangular_pattern (see 4.2.222).

NOTE This application object is harmonized with the same entity from paragraph 4.2.204 in ISO 10303-224:2006.

The data associated with a Replicate_feature are the following:

— placement;

— replicate_base_feature.

4.2.224.1 placement

The placement specifies the position and orientation of a Replicate_feature relative to the base shape for a part. See 4.3.316 for the application assertion.

4.2.224.2 replicate_base_feature

The replicate_base_feature specifies the feature that will be replicated by the Replicate_feature. See 4.3.317 for the application assertion.

4.2.225 Revolved_feature

A Revolved_feature is a type of Machining_feature (see 4.2.162) 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 4.2.138), Groove (see 4.2.145), Revolved_flat (see 4.2.226), or a Revolved_round (see 4.2.227).

NOTE This application object is harmonized with the same entity from paragraph 4.2.207 in ISO 10303-224:2006.

The data associated with a Revolved_feature are the following:

— material_side;

— radius.

4.2.225.1 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 4.3.319 for the application assertion.

4.2.225.2 radius

The radius specifies the distance from the axis of rotation to the placement of the profile that will be swept about the axis. See 4.3.318 for the application assertion.

4.2.226 Revolved_flat

A Revolved_flat is a type of Revolved_feature (see 4.2.225) that is the sweeping of a straight line about an axis.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.208 in ISO 10303-224:2006.

NOTE 2 Figure 120 illustrates the Revolved_flat.

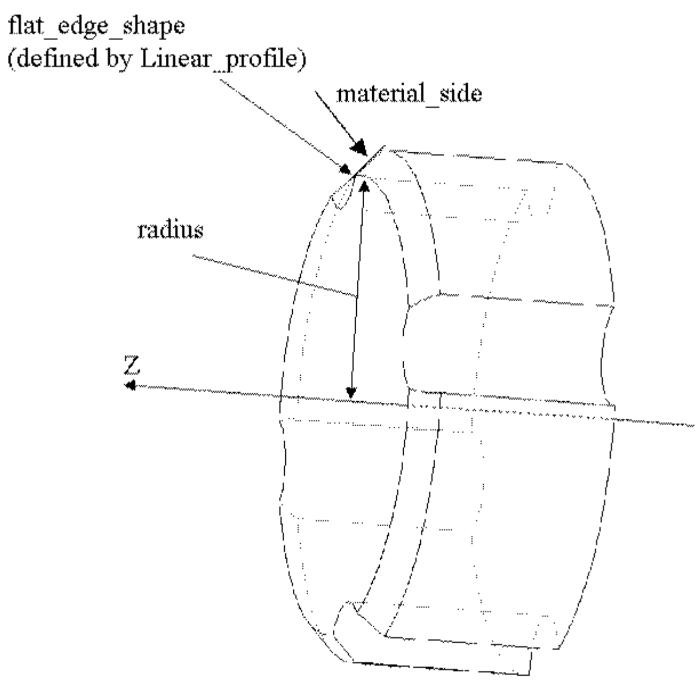


Figure 120 — Revolved_flat

The data associated with a Revolved_flat are the following:

— flat_edge_shape.

4.2.226.1 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 4.3.320 for the application assertion.

4.2.227 Revolved_round

A Revolved_round is a type of Revolved_feature (see 4.2.225) that is the sweeping of an arc about an axis.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.209 in ISO 10303-224:2006.

NOTE 2 Figure 121 illustrates the Revolved_round.

The data associated with a Revolved_round are the following:

— rounded_edge_shape.

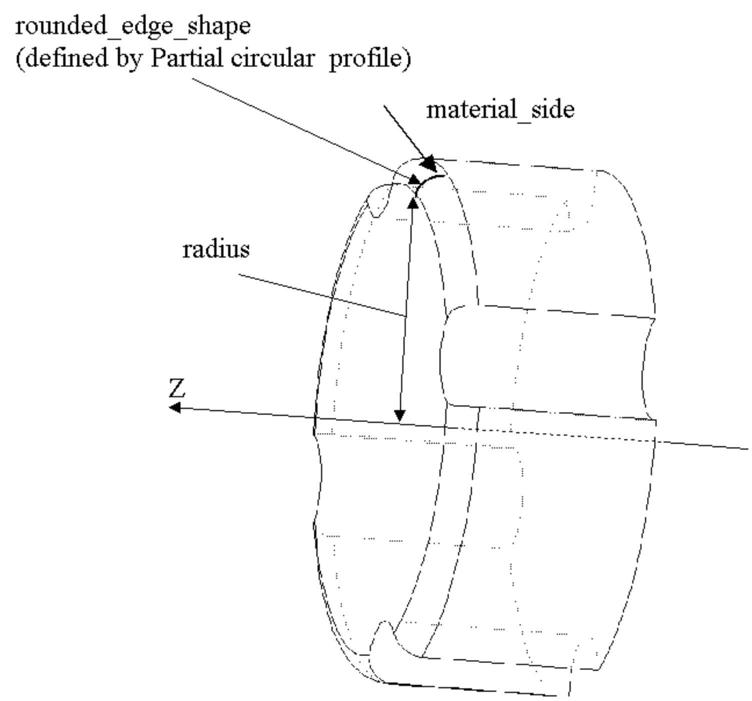


Figure 121 — Revolved_round

4.2.227.1 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 4.3.321 for the application assertion.

4.2.228 Rib_top

The Rib_top is a type of Multi_axis_feature (see 4.2.169) 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.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.210 in ISO 10303-224:2006.

EXAMPLE The material that seperates 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 2 Figure 122 illustrates the Rib_top feature.

The data associated with a Rib_top are the following:

— floor_condition;

— removal_direction.

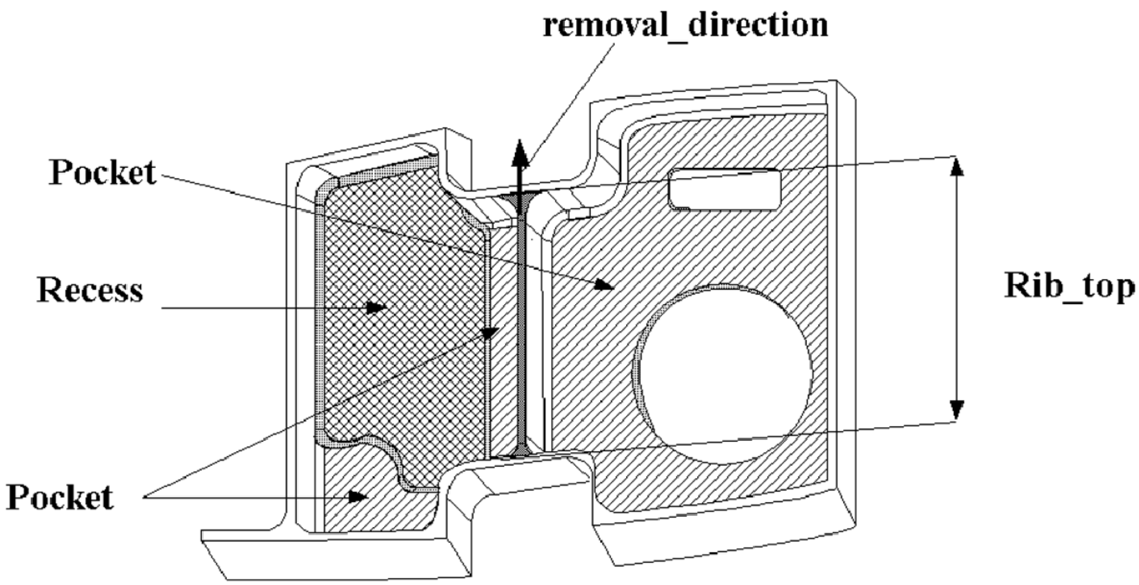


Figure 122 — Rib_top

4.2.228.1 floor_condition

The floor_condition specifies the bottom state of a Rib_top. The floor may be flat or any arbitrary shape. See 4.3.323 for the application assertion.

4.2.228.2 removal_direction

The removal_direction specifies a vector that points in the general direction away from the material for a Rib_top. See 4.3.322 for the application assertion.

4.2.229 Rib_top_floor

A Rib_top_floor is the bottom restriction for a Rib_top (see 4.2.228). The bottom may be flat, or any arbitrary shape. Each Rib_top_floor is either a General_rib_top_floor (see 4.2.139), or a Planar_rib_top_floor (see 4.2.199).

NOTE This application object is harmonized with the same entity from paragraph 4.2.211 in ISO 10303-224:2006.

4.2.230 Round_hole

A Round_hole is a type of Hole (see 4.2.149) 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 1 This application object is harmonized with the same entity from paragraph 4.2.212 in ISO 10303-224:2006.

NOTE 2 Figure 123 illustrates the Round_hole.

The data associated with a Round_hole are the following:

— bottom_condition;

— change_in_diameter;

— diameter;

— hole_depth.

4.2.230.1 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 Blind_bottom_condition (see 4.2.8),or Through_bottom_condition (see 4.2.264). See 4.3.325 and 4.3.324 for the application assertion.

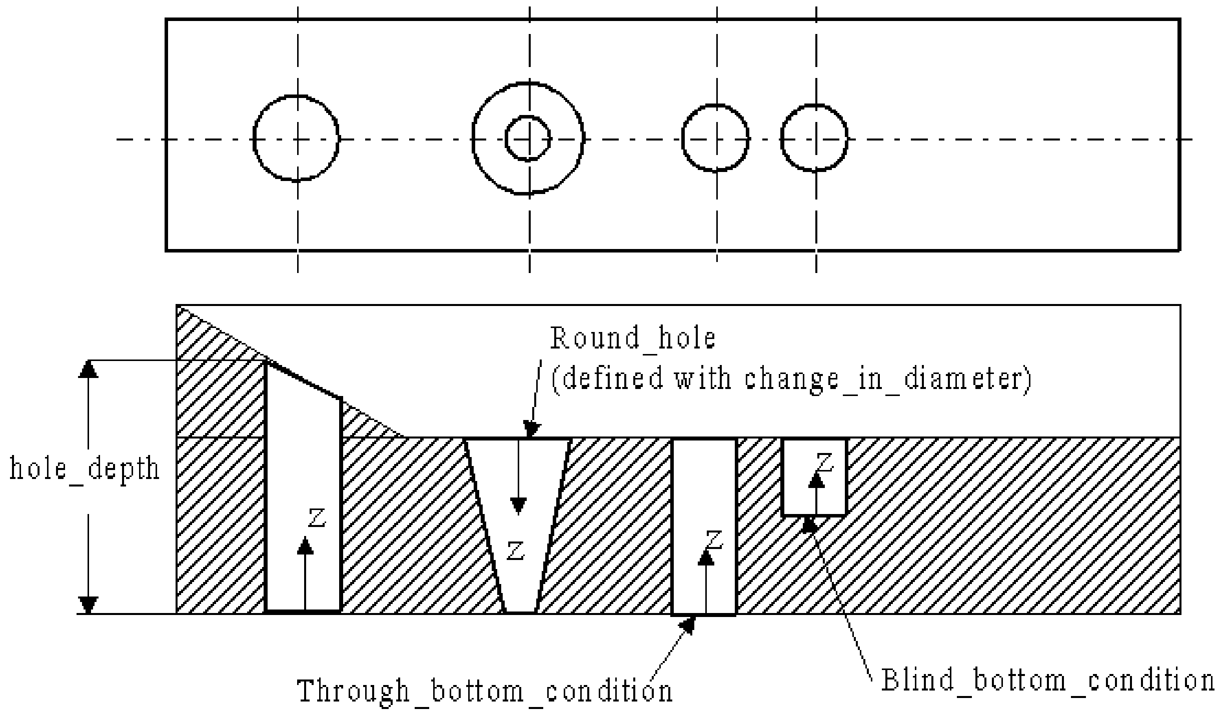


Figure 123 — Round_hole

4.2.230.2 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 4.3.328, 4.3.329 and 4.3.330 for the application assertion.

4.2.230.3 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 4.3.326 for the application assertion.

4.2.230.4 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 that will stop 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 4.3.327 for the application assertion.

4.2.231 Rounded_end

A Rounded_end is a type of Multi_axis_feature (see 4.2.169) that is a partially circular shape passed along a linear path.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.213 in ISO 10303-224:2006.

NOTE 2 Figure 124 illustrates the Rounded_end.

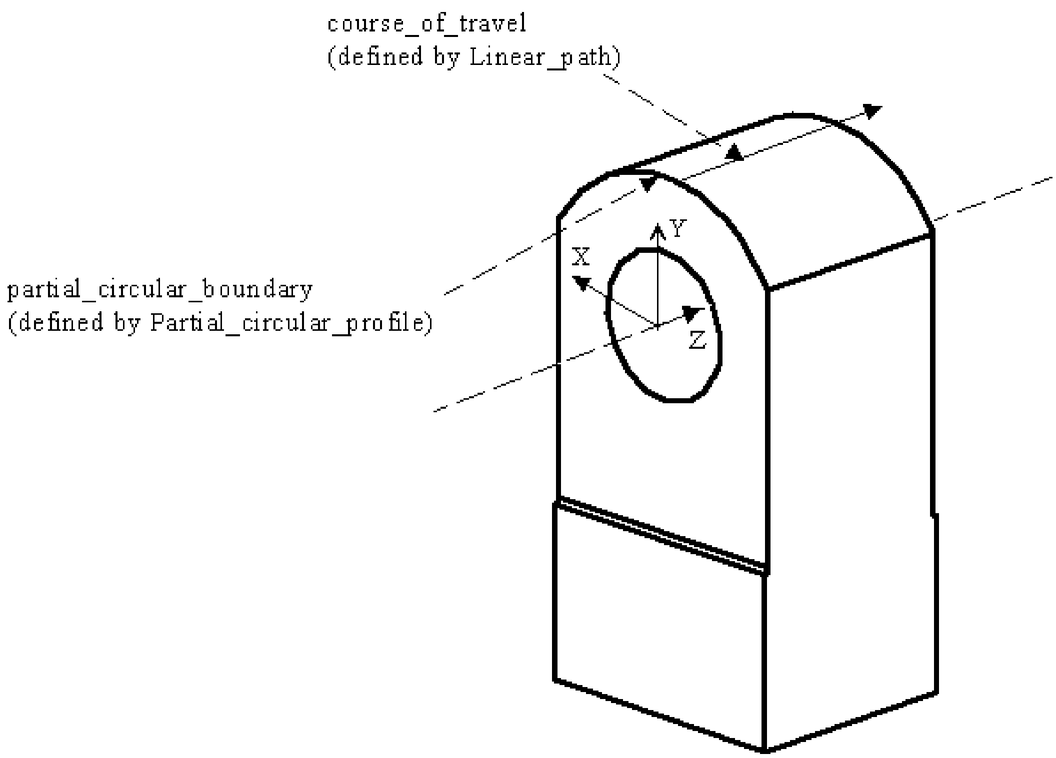


Figure 124 — Rounded_end

The data associated with a Rounded_end are the following:

— course_of_travel;

— partial_circular_boundary.

4.2.231.1 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 4.3.331 for the application assertion.

4.2.231.2 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 4.3.332 for the application assertion.

4.2.232 Rounded_U_profile

A Rounded_U_profile is a type of Open_profile (see 4.2.175) 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 1 This application object is harmonized with the same entity from paragraph 4.2.214 in ISO 10303-224:2006.

NOTE 2 Figure 125 illustrates the Rounded_U_profile.

The data associated with a Rounded_U_profile are the following:

— width.

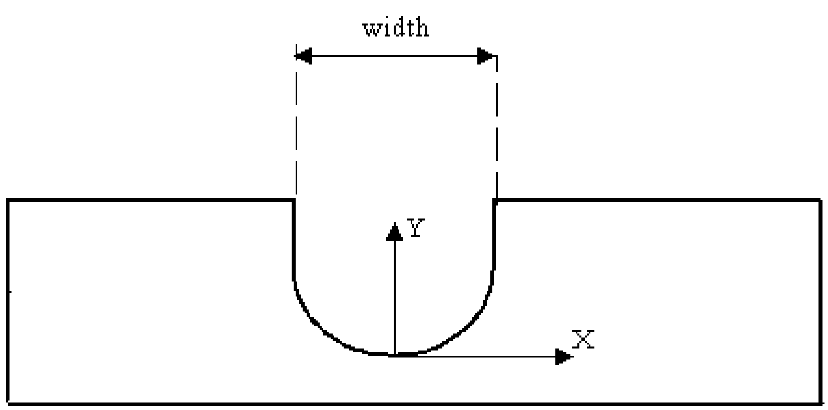


Figure 125 — Rounded_U_profile

4.2.232.1 width

The width specifies the distance across a Rounded_U_profile. See 4.3.333 for the application assertion.

4.2.233 Run_administrator

The Run_administrator is the person in charge of making the run.

The data associated with a Run_administrator are the following:

— administrator;

— shift.

4.2.233.1 administrator

The administrator specifies a person or a company assigned as the administaror of the run. See 4.3.334 and 4.3.335 for the application assertion.

4.2.233.2 shift

The shift specifies the period of time when the administrator is performing the dimensional inspection program.

EXAMPLE An example of shift may be first shift, or night shift

4.2.234 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 4.2.25) or a Second_offset (see 4.2.235).

NOTE This application object is harmonized with the same entity from paragraph 4.2.215 in ISO 10303-224:2006.

The data associated with a Second_chamfer_offset are the following:

— second_face.

4.2.234.1 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 4.2.120). See 4.3.336 for the application assertion.

4.2.235 Second_offset

A Second_offset is a type of Second_chamfer_offset (see 4.2.234) that is the amount of length offset from a face for creating a Chamfer feature.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.216 in ISO 10303-224:2006.

NOTE 2 Figure 15 illustrates a Second_offset for a Chamfer feature.

The data associated with a Second_offset are the following:

— offset_amount.

4.2.235.1 offset_amount

The offset_amount specifies the offset value from the edge of a face to the Chamfer face. See 4.3.337 for the application assertion.

4.2.236 Shape

A Shape is the physical form of the part that is being machined.

NOTE This application object is harmonized with the same entity from paragraph 4.2.217 in ISO 10303-224:2006.

The data associated with a Shape are the following:

— base_shape_definition;

— Brep_form;

— element.

4.2.236.1 base_shape_definition

The base_shape_definition specifies either the implicit or the explicit definition of the Part. See 4.3.338 for the application assertion.

4.2.236.2 Brep_form

The Brep_form specifies the boundary representation shape of the Part. The Brep_form need not be specified for a particular Shape. There may be more than one Brep_form for a Shape. See 4.3.339 for the application assertion.

4.2.236.3 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 4.3.340 for the application assertion.

4.2.237 Shape_aspect

A Shape_aspect is a region of interest with respect to the shape of a part. A Shape_aspect may be an element of the shape of the part or a reference shape that does not lie on the shape of the part, but is used to specify a characteristic of the shape of the part.

NOTE This application object is harmonized with the same entity from paragraph 4.2.218 in ISO 10303-224:2006.

The data associated with a Shape_aspect are the following:

— Brep_form;

— Brep_shape;

— element.

4.2.237.1 Brep_form

The Brep_form specifies aspects of the boundary representation of the Part. There may be more than one Brep_form for a Shape_aspect. See 4.3.342 for the application assertion.

4.2.237.2 Brep_shape

The Brep_shape specifies the boundary representation of the shape of the Part. The Brep_shape need not be specified for a particular Shape_aspect. There may be more than one Brep_shape for a Shape_aspect. See 4.3.341 for the application assertion.

4.2.237.3 element

The element specifies components of the shape of the Part. The element need not be specified for a particular Shape_aspect. See 4.3.343 for the application assertion.

4.2.238 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 4.2.70), Face_shape_element (see 4.2.116), Location_element (see 4.2.159), Feature (see 4.2.118), Path_element (see 4.2.191) or a Planar_element (see 4.2.195).

NOTE This application object is harmonized with the same entity from paragraph 4.2.219 in ISO 10303-224:2006.

4.2.239 Shape_profile

A Shape_profile is a type of Profile_feature (see 4.2.206) 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 4.2.27), Partial_-circular_shape_profile (see 4.2.188), General_shape_profile (see 4.2.140), Rectangular_closed_shape_-profile (see 4.2.217), or a Rectangular_open_shape_profile (see 4.2.221).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.220 in ISO 10303-224:2006.

NOTE 2 Figure 126 illustrates a Shape_profile.

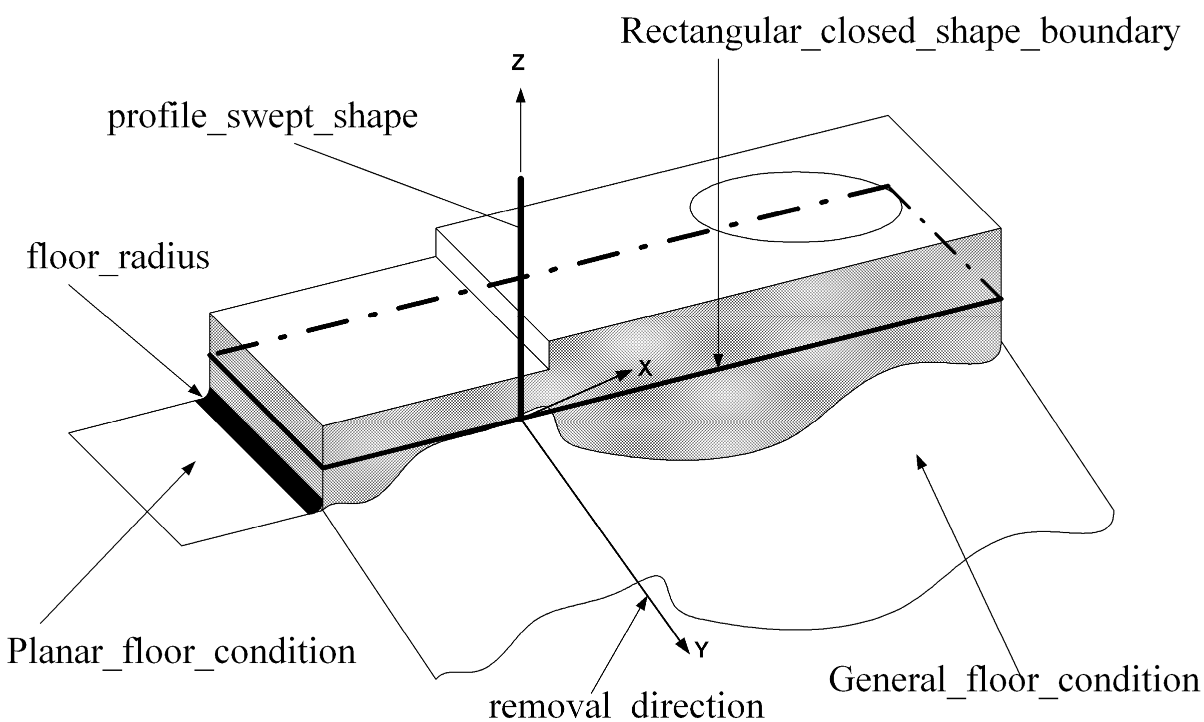


Figure 126 — Shape_profile

The data associated with a Shape_profile are the following:

— floor_condition;

— removal_direction.

4.2.239.1 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 4.2.207) or a Through_profile_floor (see 4.2.266). See 4.3.345 and 4.3.346 for the application assertion.

4.2.239.2 removal_direction

The removal_direction specifies a vector that points in the general direction away from the material for a Shape_profile. See 4.3.344 for the application assertion.

4.2.240 Size_tolerance

A Size_tolerance is a type of Dimensional_tolerance (see 4.2.68) 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), Curved_dimension_tolerance (see 4.2.50), Diameter_dimension_tolerance (see 4.2.65), Height_dimension (see 4.2.146), Length_dimension (see 4.2.153), Width_dimension (see 4.2.277), Radial_dimension_tolerance (see 4.2.211), Thickness_tolerance (see 4.2.261), or an Externally_defined_size_dimension (see 4.2.115).

NOTE This application object is harmonized with the same entity from paragraph 4.2.223 in ISO 10303-224:2006.

The data associated with a Size_tolerance are the following:

— applied_shape;

— envelope.

4.2.240.1 applied_shape

The applied_shape specifies the physical shape of the Part that is being toleranced. See 4.3.347 for the application assertion.

4.2.240.2 envelope

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

4.2.241 Slot

A Slot is a type of Multi_axis_feature(see 4.2.169) 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 4.2.37) or an Open_slot (see 4.2.176).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.224 in ISO 10303-224:2006.

NOTE 2 Figure 127 illustrates a Slot.

The data associated with a Slot are the following:

— sweep_shape.

4.2.241.1 sweep_shape

The sweep_shape specifies the implicit 2D profile definition that, when combine with a Path, creates the shape of the Slot. See 4.3.348 for the application assertion.

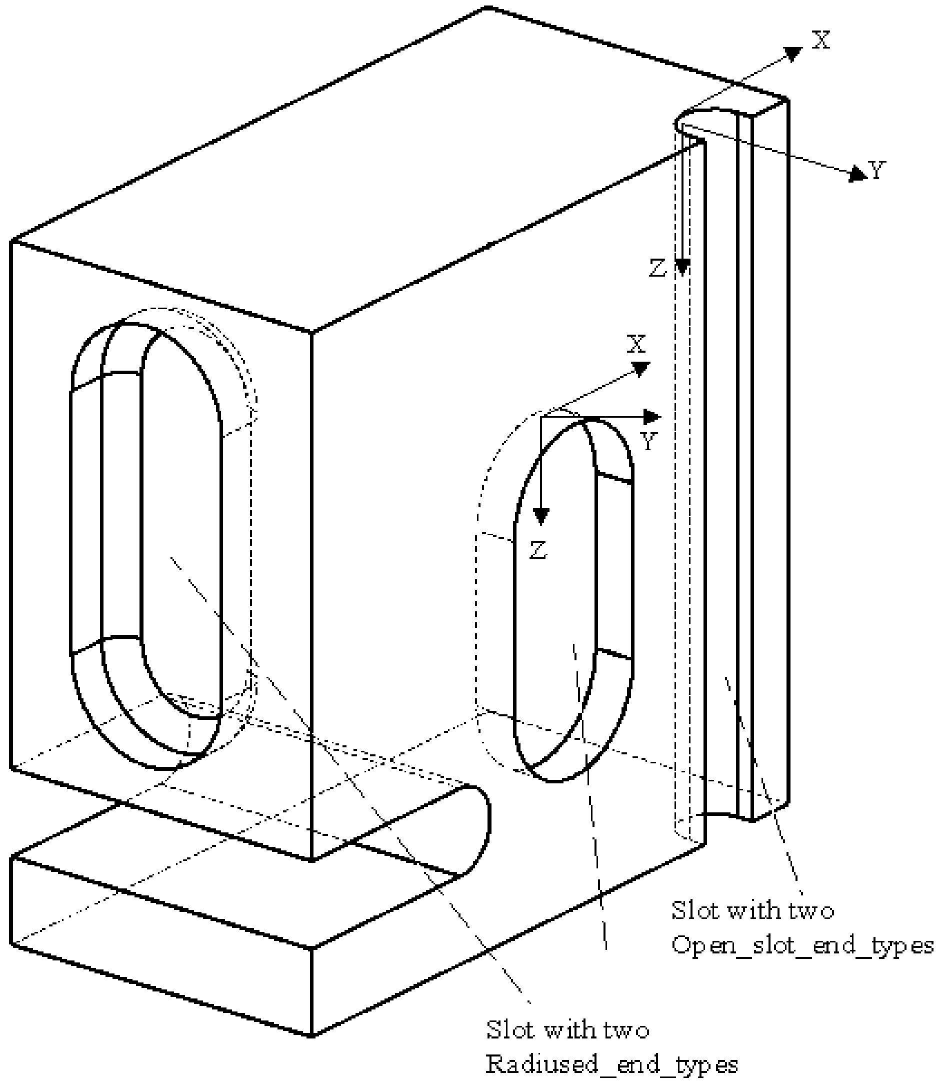


Figure 127 — Slot

4.2.242 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 4.2.122), Open_slot_end_type (see 4.2.177), Radiused_slot_end_type (see 4.2.212), or a Woodruff_slot_end_type (see 4.2.278).

NOTE This application object is harmonized with the same entity from paragraph 4.2.225 in ISO 10303-224:2006.

The data associated with a Slot_end_type are the following:

— first_or_second.

4.2.242.1 first_or_second

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

4.2.243 Specification

A Specification is a document that defines information pertaining to properties or processes for a part or an aspect of a part.

NOTE This application object is harmonized with the same entity from paragraph 4.2.226 in ISO 10303-224:2006.

The data associated with a Specification are the following:

— constraint;

— specification_class;

— specification_description;

— specification_id.

4.2.243.1 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 4.3.349 for the application assertion.

4.2.243.2 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.

4.2.243.3 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.

4.2.243.4 specification_id

The specification_id specifies a unique identifier of the document.

4.2.244 Specification_usage_constraint

A Specification_usage_constraint is a restriction on the application of information defined within a Specification.

NOTE This application object is harmonized with the same entity from paragraph 4.2.227 in ISO 10303-224:2006.

The data associated with a Specification_usage_constraint are the following:

— class_id;

— element.

4.2.244.1 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.

4.2.244.2 element

The element specifies the particular piece or area of information that is being restricted within the Specification.

4.2.245 Spherical_cap

A Spherical_cap is a type of Machining_feature (see 4.2.162) 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 1 This application object is harmonized with the same entity from paragraph 4.2.228 in ISO 10303-224:2006.

NOTE 2 Figure 128 illustrates the Spherical_cap.

The data associated with a Spherical_cap are the following:

— internal_angle;

— radius.

4.2.245.1 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.. See4.3.350 for the application assertion.

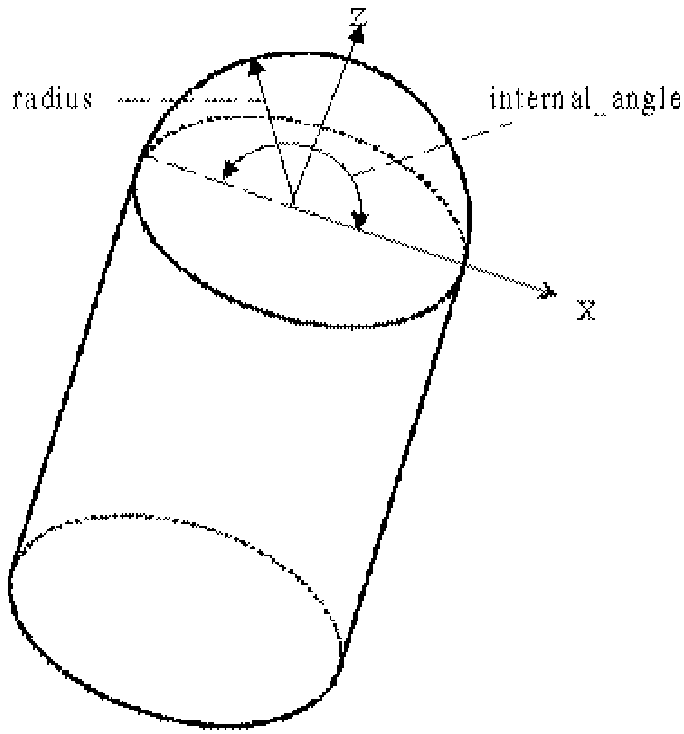


Figure 128 — Spherical_cap

4.2.245.2 radius

The radius specifies the constant distance from a point for defining a sphere. See 4.3.350 for the application assertion.

4.2.246 Spherical_hole_bottom

A Spherical_hole_bottom is a type of Blind_bottom_condition (see 4.2.8) 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 1 This application object is harmonized with the same entity from paragraph 4.2.229 in ISO 10303-224:2006.

NOTE 2 Figure 129 illustrates the Spherical_hole_bottom.

The data associated with Spherical_hole_bottom are the following:

— radius.

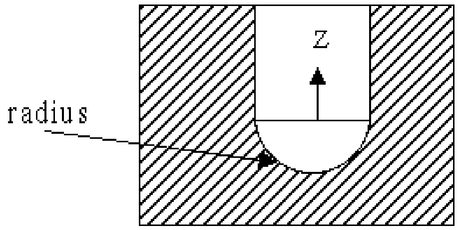


Figure 129 — Spherical_hole_bottom

4.2.246.1 radius

The radius specifies the radius at the bottom of the Round_hole. See 4.3.351 for the application assertion.

4.2.247 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 1 This application object is harmonized with the same entity from paragraph 4.2.230 in ISO 10303-224:2006.

NOTE 2 Figure 130 illustrates the Spur_gear.

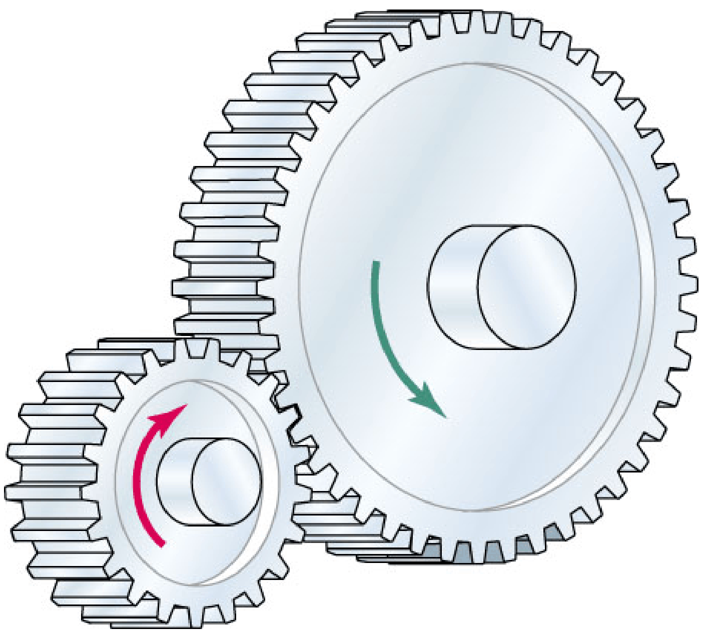


Figure 130 — Spur_gear

4.2.248 Square_U_profile

A Square_U_profile is a type of Open_profile (see 4.2.175) 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 1 This application object is harmonized with the same entity from paragraph 4.2.231 in ISO 10303-224:2006.

NOTE 2 Figure 131 illustrates the Square_U_profile.

The data associated with Square_U_profile are the following:

— depth;

— first_angle;

— first_radius;

— second_angle;

— second_radius;

— width.

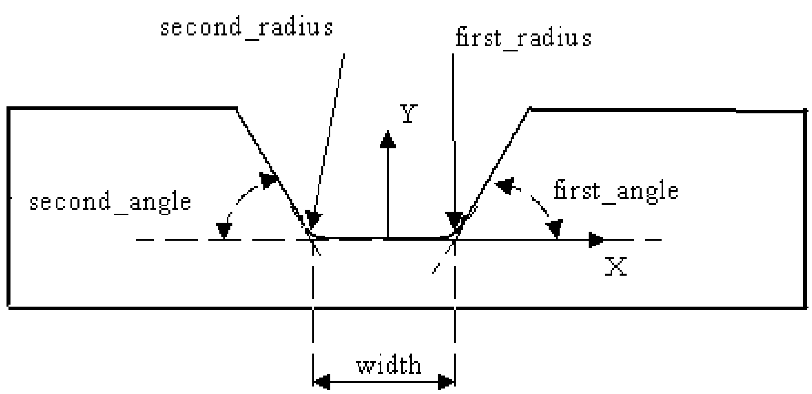


Figure 131 — Square_U_profile

4.2.248.1 depth

The depth specifies the distance, measured along the Y axis, from the profile origin to the top of the profile. See 4.3.352 for the application assertion.

4.2.248.2 first_angle

The first_angle specifies the size of an angle between one side of the profile and the base. See 4.3.352 for the application assertion.

4.2.248.3 first_radius

The first_radius specifies the radius shape blend between one side of the profile and the base. See 4.3.352 for the application assertion.

4.2.248.4 second_angle

The second_angle specifies the size of an angle between the second side of the profile and the base. See 4.3.352 for the application assertion.

4.2.248.5 second_radius

The second_radius specifies the radius shape blend between the second side of the profile and the base. See 4.3.352 for the application assertion.

4.2.248.6 width

The width specifies the size of the base line for a Square_U_profile. See 4.3.352 for the application assertion.

4.2.249 Step

A Step is a type of Multi_axis_feature (see 4.2.169) 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 1 This application object is harmonized with the same entity from paragraph 4.2.232 in ISO 10303-224:2006.

NOTE 2 Figure 132 illustrates the Step.

The data associated with a Step are the following:

— course_of_travel;

— removal_boundary;

— volume_not_removed.

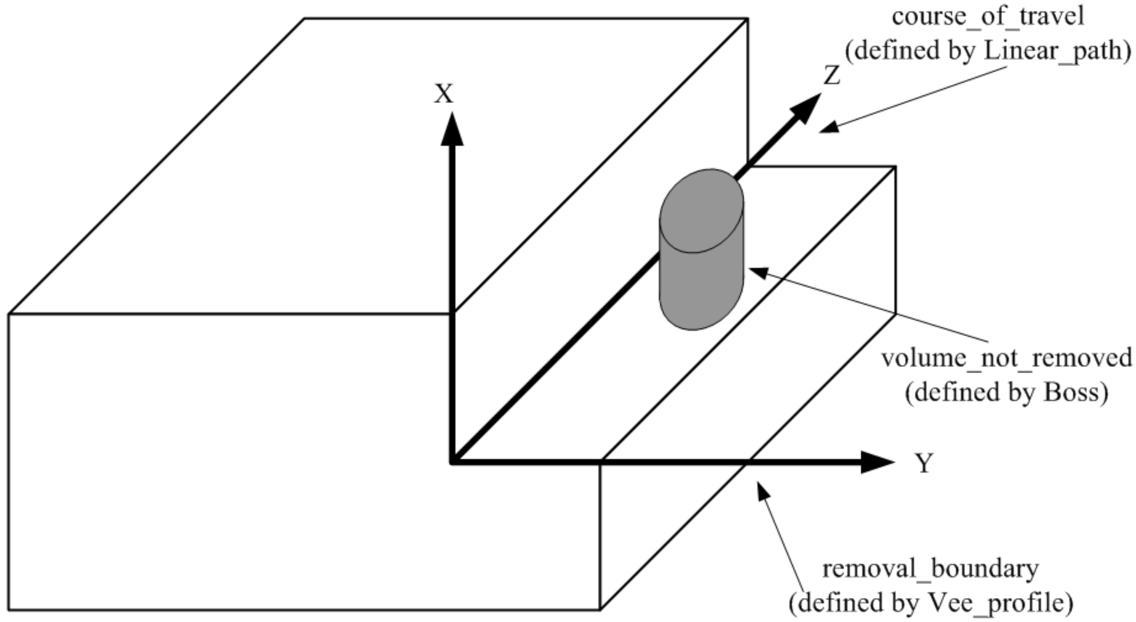


Figure 132 — Step

4.2.249.1 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 4.3.354 for the application assertion.

4.2.249.2 removal_boundary

The removal_boundary specifies the a Vee_profile (see 4.2.276) 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 4.3.353 for the application assertion.

4.2.249.3 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the pocket. The Boss (see 4.2.10) or Protrusion (see 4.2.210) feature defines the shape of the material that is to remain in the pocket. See 4.3.355 or 4.3.356 for the application assertion.

4.2.250 Straight_bevel_gear

The Straight_bevel_gear is a type of Bevel_gear(see 4.2.7) that have straight tooth elements, which if extended, would pass through the point of intersection of their axes.

NOTE This application object is harmonized with the same entity from paragraph 4.2.233 in ISO 10303-224:2006.

4.2.251 Straight_knurl

A Straight_knurl is a type of Turned_knurl (see 4.2.275) that is typified by knurl scoring that is parallel to the axis of the scored surface.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.234 in ISO 10303-224:2006.

NOTE 2 Figure 133 illustrates the Straight_knurl.

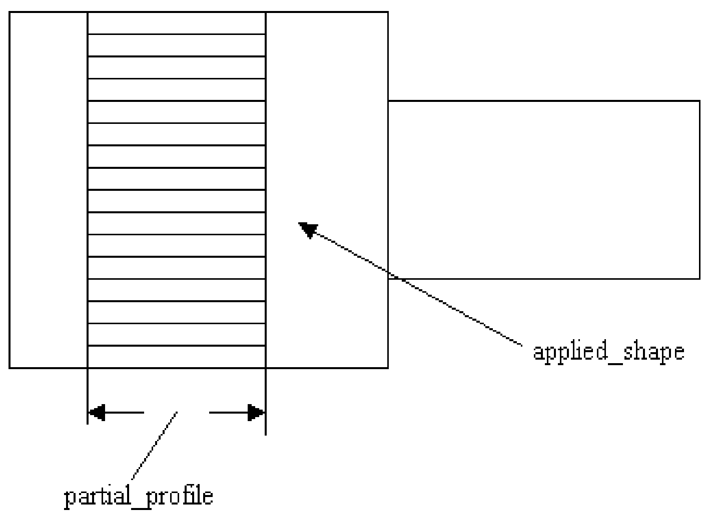


Figure 133 — Straight_knurl

4.2.252 Straightness_tolerance

A Straightness_tolerance is a type of Geometric_tolerance (see 4.2.143) 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 This application object is harmonized with the same entity from paragraph 4.2.235 in ISO 10303-224:2006.

NOTE 2 Figure 134 illustrates the Straightness_tolerance.

NOTE 3 The Straightness_tolerance definition is derived from ISO 1101, 14.1.

The data associated with a Straightness_tolerance are the following:

— affected_plane;

— segment_size.

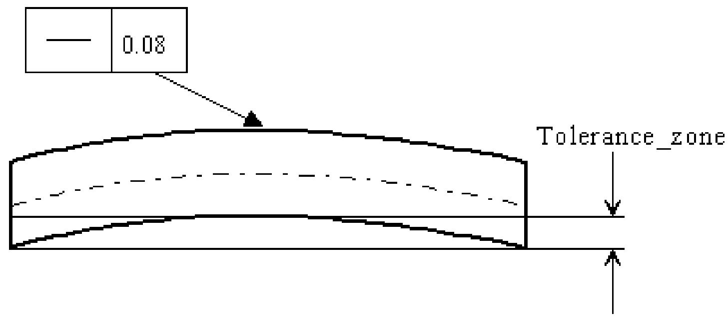


Figure 134 — Straightness_tolerance

4.2.252.1 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 4.3.357 for the application assertion.

4.2.252.2 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.

4.2.253 Surface_profile_tolerance

A Surface_profile_tolerance is a type of Geometric_tolerance (see 4.2.143) that is a uniform boundary or zone along the true profile within which all elements of the surface shall lie.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.236 in ISO 10303-224:2006.

NOTE 2 Figure 135 illustrates Surface_profile_tolerance.

NOTE 3 The Surface_profile_tolerance definition is derived from ISO 1101, 14.6.

The data associated with a Surface_profile_tolerance are the following:

— geometric_reference.

4.2.253.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.358 for the application assertion.

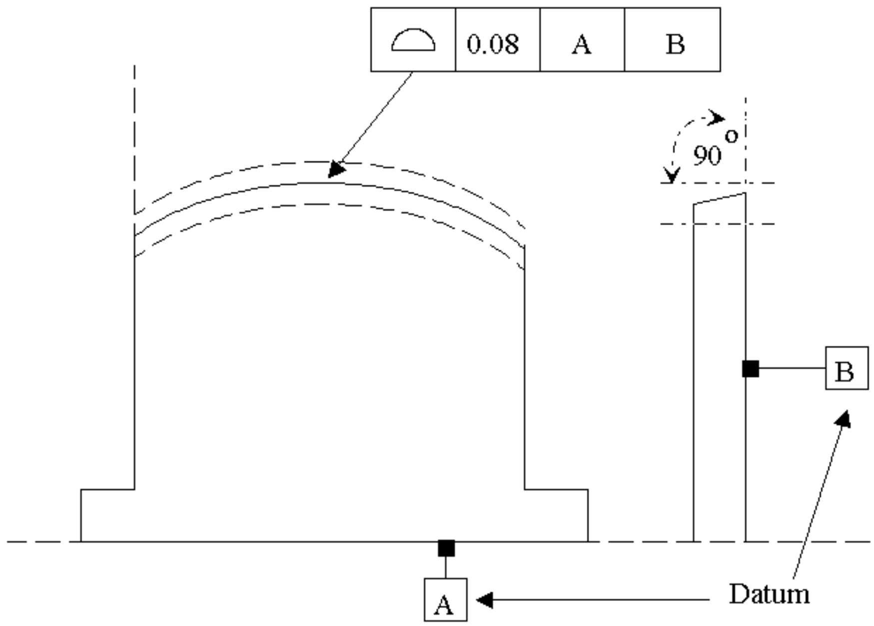


Figure 135 — Surface_profile_tolerance

4.2.254 Symmetry_tolerance

A Symmetry_tolerance is a type of Geometric_tolerance (see 4.2.143) 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 This application object is harmonized with the same entity from paragraph 4.2.239 in ISO 10303-224:2006.

NOTE 2 Figure 136 illustrates the Symmetry_tolerance.

NOTE 3 The Symmetry_tolerance definition is derived from ISO 1101, 14.12.

The data associated with a Symmetry_tolerance are the following:

— affected_plane;

— geometric_reference.

4.2.254.1 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 4.3.360 for the application assertion.

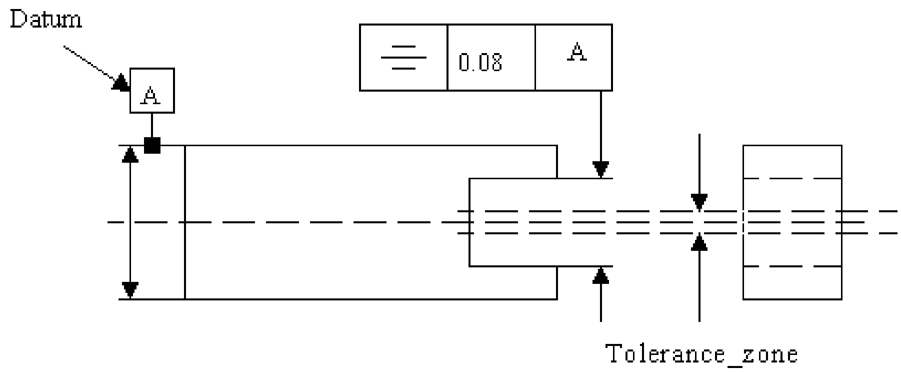


Figure 136 — Symmetry_tolerance

4.2.254.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.359 for the application assertion.

4.2.255 Target_area

A Target_area is a type of Datum_target (see 4.2.57) that is an enclosed area bounded by an arbitrary shape required to define a Datum_target. The shape of the Target_area is described explicitly by a set of curves.

NOTE This application object is harmonized with the same entity from paragraph 4.2.240 in ISO 10303-224:2006.

The data associated with Target_area are the following:

— area_shape.

4.2.255.1 area_shape

The area_shape specifies the physical form of the Target_area shape. See 4.3.361 for the application assertion.

4.2.256 Target_circle

A Target_circle is a type of Placed_target (see 4.2.194) that is an enclosed area bounded by a circle required to define a Datum_target. The origin of the Datum_target is the center of the circle, and the orientation is the x-y plane.

NOTE This application object is harmonized with the same entity from paragraph 4.2.241 in ISO 10303-224:2006.

The data associated with a Target_circle are the following:

— target_diameter.

4.2.256.1 target_diameter

The target_diameter specifies the diameter value of the Target_circle.

4.2.257 Target_line

A Target_line is a type of Placed_target (see 4.2.194) that is a straight curve. The origin shall be the first end point of the Target_line, the second end point shall be located on the Z-axis at a specified length.

NOTE This application object is harmonized with the same entity from paragraph 4.2.242 in ISO 10303-224:2006.

The data associated with a Target_line are the following:

— target_length.

4.2.257.1 target_length

The target_length specifies the length value of the Target_line.

4.2.258 Target_point

A Target_point is a type of Placed_target (see 4.2.194) that is a single point. The origin shall be at the Target_point.

NOTE This application object is harmonized with the same entity from paragraph 4.2.243 in ISO 10303-224:2006.

4.2.259 Target_rectangle

A Target_rectangle is a type of Placed_target (see 4.2.194) 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.

NOTEThis application object is harmonized with the same entity from paragraph 4.2.244 in ISO 10303-224:2006.

The data associated a with Target_rectangle are the following:

— target_length;

— target_width.

4.2.259.1 target_length

The target_length specifies the length value of the Target_rectangle.

4.2.259.2 target_width

The target_width specifies the width value of the Target_rectangle.

4.2.260 Tee_profile

A Tee_profile is a type of Open_profile (see 4.2.175) 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 1 This application object is harmonized with the same entity from paragraph 4.2.245 in ISO 10303-224:2006.

NOTE 2 Figure 137 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.

4.2.260.1 cross_bar_depth

The cross_bar_depth specifies the depth dimension of the Tee cross bar size. See 4.3.362 for the application assertion.

4.2.260.2 cross_bar_width

The cross_bar_width specifies the width dimension of the Tee cross bar size. See 4.3.362 for the application assertion.

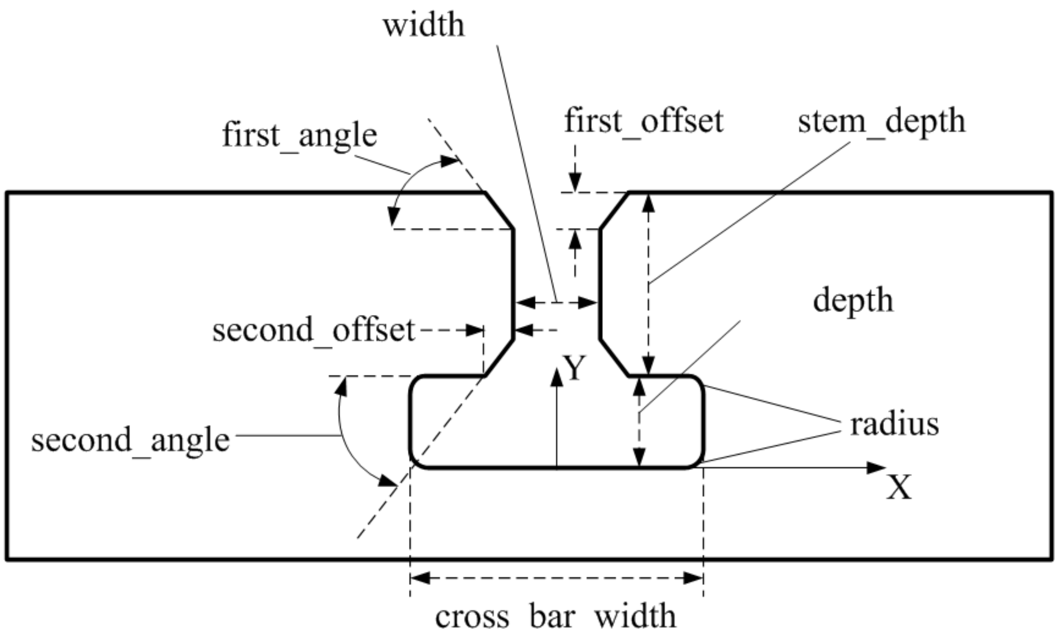


Figure 137 — Tee_profile

4.2.260.3 depth

The depth specifies the depth dimension of the Tee stem. See 4.3.362 for the application assertion.

4.2.260.4 first_angle

The first_angle specifies the angular measurement for creating a chamfer on the open end of a Tee_profile. See 4.3.362 for the application assertion.

4.2.260.5 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 4.3.362 for the application assertion.

4.2.260.6 radius

The radius specifies the arc size for blending the sides of a Tee_profile cross bar. See 4.3.362 for the application assertion.

4.2.260.7 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 4.3.362 for the application assertion.

4.2.260.8 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 4.3.362 for the application assertion.

4.2.260.9 width

The width specifies the width dimension of the Tee stem. See 4.3.362 for the application assertion.

4.2.261 Thickness_tolerance

A Thickness_tolerance is a type of Size_tolerance (see 4.2.240) that represents a thickness.

NOTE This application object is harmonized with the same entity from paragraph 4.2.248 in ISO 10303-224:2006.

The data associated with a Thickness_tolerance are the following:

— path.

4.2.261.1 path

The path specifies the shape that the tolerance applies to. A Thickness_tolerance may but need not require a path. See 4.3.363 for the application assertion.

4.2.262 Thread

A Thread is a type of Machining_feature (see 4.2.162) 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 4.2.23) or a Defined_thread (see 4.2.61).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.246 in ISO 10303-224:2006.

NOTE 2 Figure 138 illustrates the Thread and illustrates Thread attributes.

NOTE 3 Threads may be used to screw parts together.

NOTE 4 An outside thread might be on an 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.

4.2.262.1 applied_shape

The applied_shape specifies the physical shape of the Part that will define where the Thread feature will be applied. See 4.3.367 for the application assertion.

4.2.262.2 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 4.3.364 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.

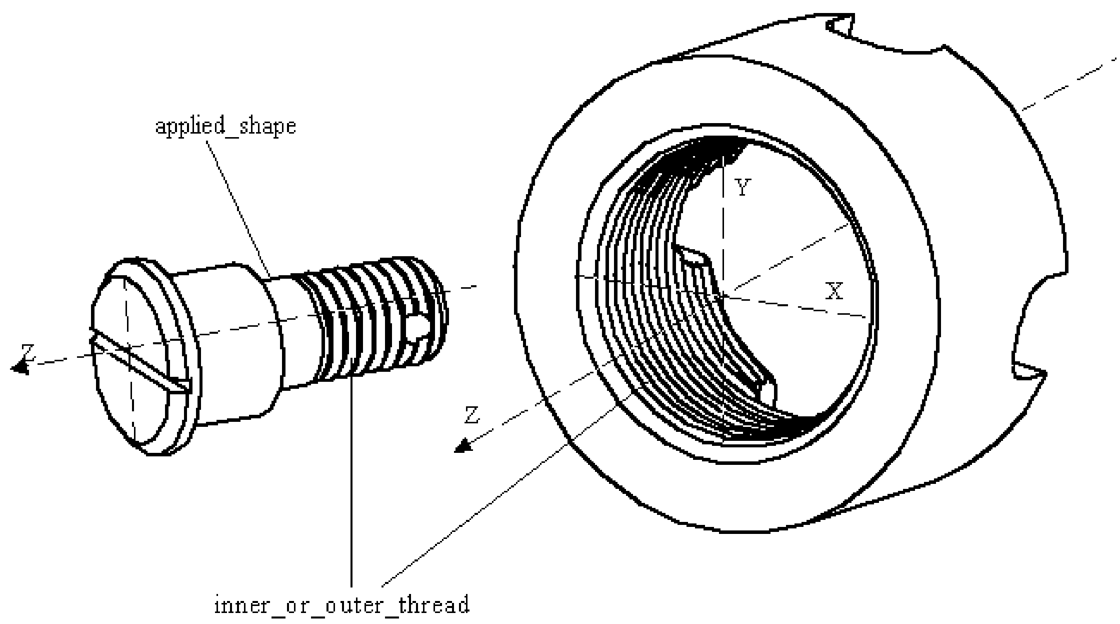


Figure 138 — Thread

4.2.262.3 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 4.3.364 for the application assertion.

EXAMPLE ISO metric screw threads, where the tolerance class for the major diameter differs from that for the pitch diameter, it is necessary to have two fit classes in order to specify the thread.

4.2.262.4 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 4.3.364 for the application assertion.

EXAMPLE Examples of form are: metric, square, unified, sharp V, buttress, standard worm, and knuckle.

4.2.262.5 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.

4.2.262.6 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 4.3.365 for the application assertion.

4.2.262.7 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 4.3.365 for the application assertion.

4.2.262.8 partial_profile

The partial_profile specifies the limitations to be applied on the Thread feature. See 4.3.366 for the application assertion.

4.2.262.9 qualifier

The qualifier specifies additional text information that describes a Thread. The qualifier need not be specified for a particular Thread. See 4.3.364 for the application assertion.

4.2.262.10 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 4.3.368 for the application assertion.

4.2.262.11 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 4.3.364 for the application assertion.

4.2.263 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.

NOTE This application object is harmonized with the same entity from paragraph 4.2.247 in ISO 10303-224:2006.

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.

4.2.263.1 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.

4.2.263.2 length_of_runout

The length_of_runout specifes the length of the runout. See 4.3.369 for the application assertion.

4.2.263.3 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.

4.2.264 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 1 This application object is harmonized with the same entity from paragraph 4.2.249 in ISO 10303-224:2006.

NOTE 2 Figure 139 illustrates the Through_bottom_condition.

4.2.265 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 4.2.202.4) of the Pocket feature.

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.250 in ISO 10303-224:2006.

NOTE 2 Figure 81 illustrates Through_pocket_bottom_condition.

4.2.266 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.

NOTE This application object is harmonized with the same entity from paragraph 4.2.251 in ISO 10303-224:2006.

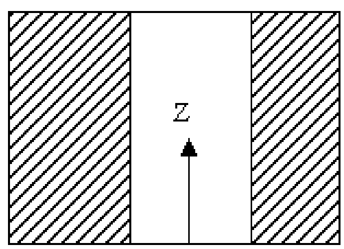


Figure 139 — Through_bottom_condition

4.2.267 Time_offset

A Time_offset is the offset of a time from universal time using a time scale called Coordinated Universal Time, which is the basis for the worldwide system of civil time.

The data associated with a Time_offset are the following:

— direction;

— hour;

— minute;

4.2.267.1 direction

The direction specifies if the offset is ahead, behind or exactly coordinated universal time.

4.2.267.2 hour

The hour specifies an hour offset.

4.2.267.3 minute

The minute specifies the minute offset

4.2.268 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 4.2.68.3) shall be a tolerance value. When applied to a Numeric_parameter_with_tolerance, the parameter_value (see 4.2.172.2) shall be a tolerance value. There shall be a qualifier that describes the tolerance context.

NOTE This application object is harmonized with the same entity from paragraph 4.2.252 in ISO 10303-224:2006

The data associated with a Tolerance_limit are the following:

— limit_qualifier.

4.2.268.1 limit_qualifier

The limit_qualifier specifies a description of the Tolerance_limit context.

EXAMPLE '30.5 MAX' or '5 MIN' are Tolerance_limit examples and the words 'MAX' or 'MIN' are limit_qualifier examples.

4.2.269 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 4.2.68.3) may be a nominal tolerance value. When applied to a Numeric_parameter_with_tolerance, the parameter_value (see 4.2.172.2) may be a nominal tolerance value.

NOTE This application object is harmonized with the same entity from paragraph 4.2.253 in ISO 10303-224:20063.

The data associated with a Tolerance_range are the following:

— lower_range;

— significant_digits;

— upper_range.

4.2.269.1 lower_range

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

4.2.269.2 significant_digits

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

4.2.269.3 upper_range

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

4.2.270 Tolerance_value

A Tolerance_value is the representation of the magnitude of the allowable deviation required for dimensions. These tolerance values may be explicitly defined or may require a specification for definition.

NOTE This application object is harmonized with the same entity from paragraph 4.2.254 in ISO 10303-224:2006.

The data associated with a Tolerance_value are the following:

— defined_value;

4.2.270.1 defined_value

The defined_value specifies the tolerance deviation value. See 4.3.370, 4.3.371, 4.3.372, and 4.3.373 for the application assertion.

4.2.271 Tolerance_zone

A Tolerance_zone is an area where all points of the geometric element that have tolerances shall be contained.

NOTE This application object is harmonized with the same entity from paragraph 4.2.255 in ISO 10303-224:2006.

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.

4.2.271.1 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.

4.2.271.2 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 4.3.375 for the application assertion.

4.2.271.3 form_type

The form_type specifies the shape of the Tolerance_zone.

EXAMPLE 'Cylindrical', 'parallelepiped', 'spherical' are examples for form_type.

4.2.271.4 zone_definition

The zone_definition specifies the defining boundaries for a Tolerance_zone. See 4.3.374 for the application assertion.

4.2.272 Tolerance_zone_definition

A Tolerance_zone_definition is the boundaries of a Tolerance_zone. Each Tolerance_zone_definition shall be defined by at least one shape and may be defined with two shapes.

NOTE This application object is harmonized with the same entity from paragraph 4.2.256 in ISO 10303-224:2006.

The data associated with a Tolerance_zone_definition are the following:

— first_element;

— second_element.

4.2.272.1 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 4.3.376 or the application assertion.

4.2.272.2 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 4.3.376 for the application assertion.

4.2.273 Total_runout_tolerance

A Total_runout_tolerance is a type of Geometric_tolerance (see 4.2.143) 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 This application object is harmonized with the same entity from paragraph 4.2.257 in ISO 10303-224:2006

NOTE 2 Figure 140 illustrates the Total_runout_tolerance.

NOTE 3 The Total_runout_tolerance definition is derived from ISO 1101, 14.13.

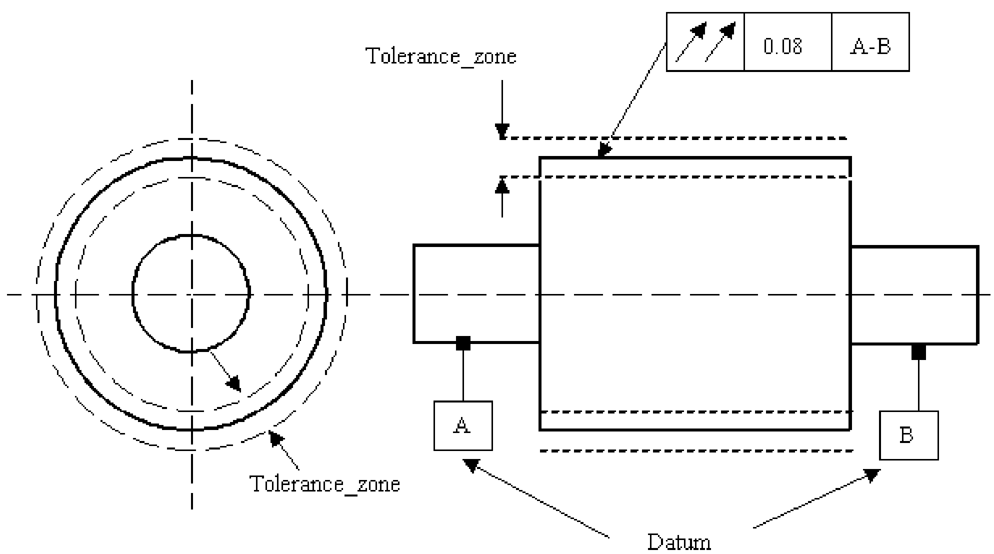


Figure 140 — Total_runout_tolerance

The data associated with a Total_runout_tolerance are the following:

— geometric_reference;

— runout.

4.2.273.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.377 for the application assertion.

4.2.273.2 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.

4.2.274 Transition_feature

A Transition_feature is a type of Manufacturing_feature (see 4.2.163) 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 4.2.24), Edge_round (see 4.2.113), or a Fillet (see 4.2.119).

NOTE This application object is harmonized with the same entity from paragraph 4.2.258 in ISO 10303-224:2006.

4.2.275 Turned_knurl

A Turned_knurl is a type of Knurl (see 4.2.152) 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), Diamond_knurl (see 4.2.67), or a Straight_knurl (see 4.2.251).

NOTE 1 This application object is harmonized with the same entity from paragraph 4.2.259 in ISO 10303-224:2006.

NOTE 2 Figure 141 illustrates the Turned_knurl.

NOTE 3 A knurl may be used to aid in gripping a part.

The data associated with a Turned_knurl are the following:

— diametral_pitch;

— major_diameter;

— nominal_diameter;

— number_of_teeth;

— root_fillet;

— tooth_depth.

4.2.275.1 diametral_pitch

The diametral_pitch specifies the ratio of the number of teeth in the circumference to the nominal diameter. See 4.3.378 for the application assertion.

4.2.275.2 major_diameter

The major_diameter specifies size of the part before a knurl is applied to it. See 4.3.378 for the application assertion.

4.2.275.3 nominal_diameter

The nominal_diameter specifies the size of the part after a knurl has been applied. See 4.3.378 for the application assertion.

4.2.275.4 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 4.3.378 for the application assertion.

4.2.275.5 root_fillet

The root _fillet specifies the dimension of a radius between teeth on a knurling tool. See 4.3.378 for the application assertion.

4.2.275.6 tooth_depth

The tooth_depth specifies the depth from the crest of a tooth to the point where two teeth intersect. See 4.3.378 for the application assertion.

4.2.276 Vee_profile

A Vee_profile is a type of Open_profile (see 4.2.175) 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 1 This application object is harmonized with the same entity from paragraph 4.2.260 in ISO 10303-224:2006.

NOTE 2 Figure 142 illustrates the Vee_profile.

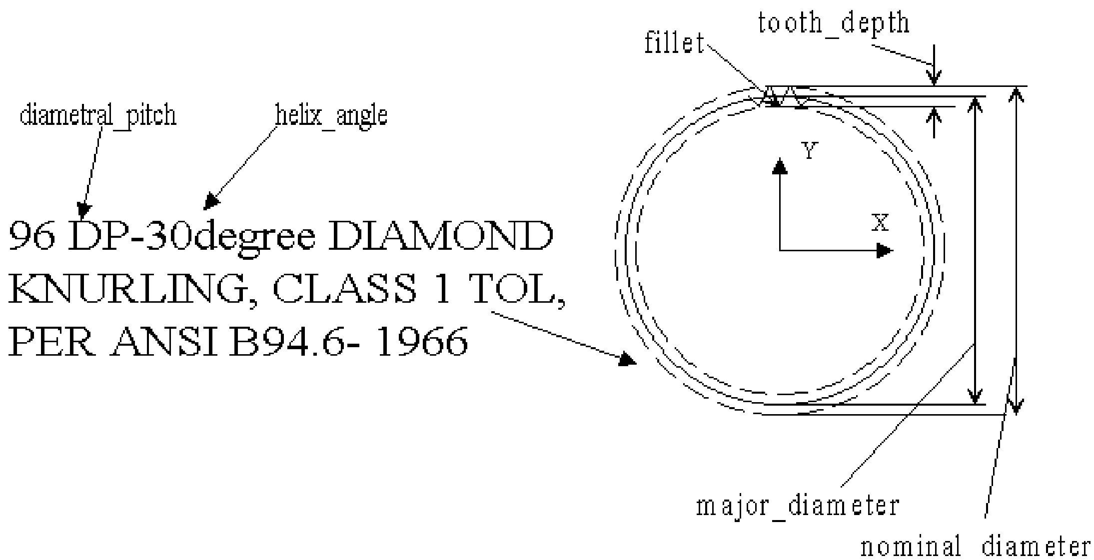


Figure 141 — Turned_knurl

The data associated with a Vee_profile are the following:

— profile_angle;

— profile_radius;

— tilt_angle.

4.2.276.1 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 4.3.379 for the application assertion.

4.2.276.2 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. The profile_radius need not be specified for a particular Vee_profile. See 4.3.379 for the application assertion.

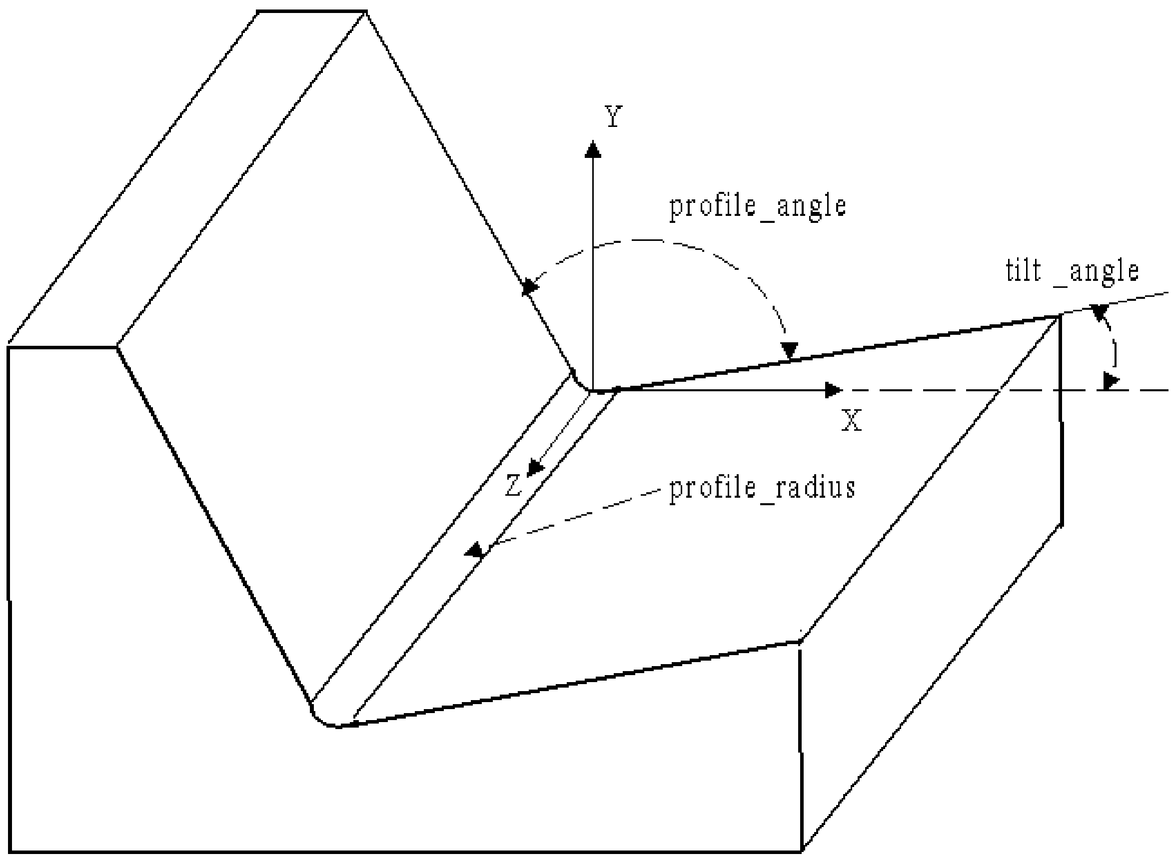


Figure 142 — Vee_profile

4.2.276.3 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 4.3.379 for the application assertion.

4.2.277 Width_dimension

A Width_dimension is a type of Size_tolerance (see 4.2.240) that specifies the size along a straight line that is referred to as width in the referenced shape.

NOTE This application object is harmonized with the same entity from paragraph 4.2.261 in ISO 10303-224:2006.

The data associated with a Width_dimension are the following:

— path.

4.2.277.1 path

The path specifies the shape that the tolerance applies to. A Width_dimension may but need not require a path.. See 4.3.380 for the application assertion.

4.2.278 Woodruff_slot_end_type

A Woodruff_slot_end_type is a type of Slot_end_type (see 4.2.242) 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 1 This application object is harmonized with the same entity from paragraph 4.2.262 in ISO 10303-224:2006.

NOTE 2 Figure 143 illustrates the Woodruff_slot_end_type.

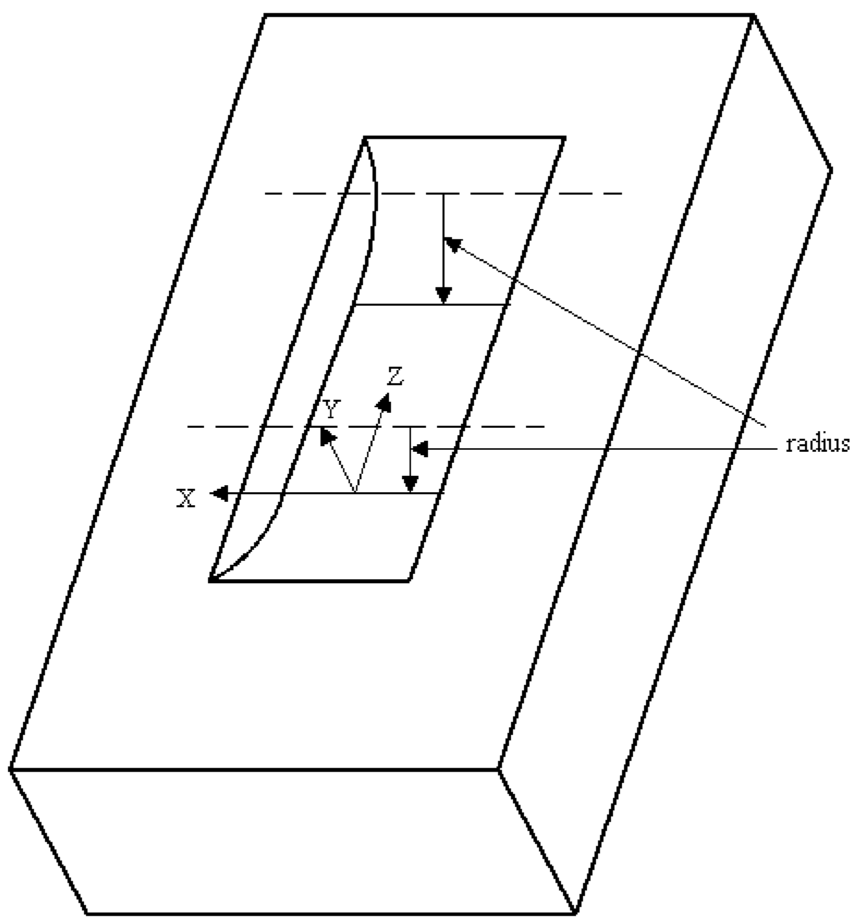


Figure 143 — Woodroff_slot_end_type

The data associated with a Woodruff_slot_end_type are the following:

— radius.

4.2.278.1 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 4.3.381 for the application assertion.

4.3 Application assertions

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.

4.3.1 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.

4.3.2 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.

4.3.3 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.

4.3.4 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.

4.3.5 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.

4.3.6 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.

4.3.7 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.

4.3.8 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.

4.3.9 Boss_top_condition to Numeric_parameter

Each Boss_top_condition has the top_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the top_radius for zero, one, or many Boss_top_condition objects.

4.3.10 Brep_model_element to Brep_model

Each Brep_model_element has the element defined by exactly one Brep_model. Each Brep_model defines the element for zero, one, or many Brep_model_element objects.

4.3.11 Brep_shape_aspect_representation to Brep_model

Each Brep_shape_aspect_representation has the shape_definition defined by exactly one Brep_model. Each Brep_model defines the shape_definition for zero, one, or many Brep_shape_aspect_-representation objects.

4.3.12 Brep_shape_representation to Brep_model

Each Brep_shape_representation has the shape_definition defined by exactly one Brep_model. Each Brep_model defines the shape_definition for zero, one, or many Brep_shape_representation objects.

4.3.13 Catalogue_gear to Document_assignment

Each Catalogue_gear has documentation defined by exactly one Document_assignment Each Document_assignment defines the documentation for zero, one, or many Catalogue_gear objects.

4.3.14 Catalogue_knurl to Document_assignment

Each Catalogue_knurl has documentation defined by exactly one Document_assignment. Each Document_assignment defines the documentation for zero, one, or many Catalogue_knurl objects.

4.3.15 Catalogue_marking to Document_assignment

Each Catalogue_marking has documentation defined by exactly one Document_assignment. Each Document_assignment defines the documentation for zero, one, or many Catalogue_marking objects.

4.3.16 Catalogue_thread to Document_assignment

Each Catalogue_thread has documentation defined by exactly one Document_assignment. Each Document_assignment defines the documentation for zero, one, or many Catalogue_thread objects.

4.3.17 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.

4.3.18 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.

4.3.19 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.

4.3.20 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.

4.3.21 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.

4.3.22 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.

4.3.23 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.

4.3.24 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.

4.3.25 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.

4.3.26 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.

4.3.27 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.

4.3.28 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.

4.3.29 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.

4.3.30 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.

4.3.31 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.

4.3.32 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.

4.3.33 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.

4.3.34 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.

4.3.35 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.

4.3.36 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.

4.3.37 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.

4.3.38 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.

4.3.39 Common_datum to Datum_feature

Each Common_datum has the element defined by two or more Datum_feature objects. Each Datum_-feature defines the element for zero, one, or many Common_datum objects.

4.3.40 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.

4.3.41 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.

4.3.42 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.

4.3.43 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.

4.3.44 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.

4.3.45 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.

4.3.46 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.

4.3.47 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.

4.3.48 Coordinate_tolerance to Orientation

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

4.3.49 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.

4.3.50 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.

4.3.51 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.

4.3.52 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.

4.3.53 Date_time to Calendar_date

Each Date_time has the date defined by exactly one Calendar_date. Each Calendar_date defines the date for zero, one, or many Date_time objects.

4.3.54 Date_time to Time_offset

Each Date_time has the offset defined by zero or one Time_offset. Each Time_offset defines the offset for zero, one, or many Date_time objects.

4.3.55 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.

4.3.56 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.

4.3.57 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.

4.3.58 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.

4.3.59 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.

4.3.60 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.

4.3.61 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.

4.3.62 Derived_shape_element to Shape_element

Each Derived_shape_element has the is_derived_from defined by one or more Shape_element. Each Shape_element defines the is_derived_from for zero, one, or many Derived_shape_element objects.

4.3.63 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.

4.3.64 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.

4.3.65 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.

4.3.66 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.

4.3.67 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.

4.3.68 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.

4.3.69 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.

4.3.70 Distance_along_curve_tolerance to Shape_aspect

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

4.3.71 Dm_analysis_dofs_dml to Dof_attribute_dml

Each Dm_analysis_dofs_dml has a rotx defined by exactly one Dof_attribute_dml. Each Dof_attribute_-dml defines the rotx for zero, one, or many Dm_analysis_dofs_dml objects.

Each Dm_analysis_dofs_dml has a roty defined by exactly one Dof_attribute_dml. Each Dof_attribute_-dml defines the roty for zero, one, or many Dm_analysis_dofs_dml objects.

Each Dm_analysis_dofs_dml has a rotz defined by exactly one Dof_attribute_dml. Each Dof_attribute_-dml defines the rotz for zero, one, or many Dm_analysis_dofs_dml objects.

Each Dm_analysis_dofs_dml has a transx defined by exactly one Dof_attribute_dml. Each Dof_-attribute_dml defines the transx for zero, one, or many Dm_analysis_dofs_dml objects.

Each Dm_analysis_dofs_dml has a transy defined by exactly one Dof_attribute_dml. Each Dof_-attribute_dml defines the transy for zero, one, or many Dm_analysis_dofs_dml objects.

Each Dm_analysis_dofs_dml has a transz defined by exactly one Dof_attribute_dml. Each Dof_-attribute_dml defines the transz for zero, one, or many Dm_analysis_dofs_dml objects.

4.3.72 Dm_dimension_parameter to Measurement_uncertainty

Each Dm_dimension_parameter has the calculated_value defined by zero or one Measurement_-uncertainty. Each Measurement_uncertainty defines the calculated_value for zero, one, or many Dm_dimension_parameter objects.

4.3.73 Dm_dimension_parameter to Numeric_parameter

Each Dm_dimension_parameter has the value_uncertainty defined by zero or one Numeric_parameter. Each Numeric_parameter defines the value_uncertaintyt for zero, one, or many Dm_dimension_-parameter objects.

4.3.74 Dm_execution_input to Dm_executed_result

Each Dm_execution_input has the result defined by exactly one Dm_executed_resul. Each Dm_-executed_resul defines the result for zero, one, or many Dm_execution_inputobjects.

4.3.75 Dm_execution_input to Dm_program_run

Each Dm_execution_input has the program_run defined by exactly one Dm_program_run. Each Dm_-program_run defines the program_run for zero, one, or many Dm_execution_input objects.

4.3.76 Dm_execution_input to Part

Each Dm_execution_input has the as_designed_part defined by exactly one Part. Each Dm_-as_designed_part defines the as_designed_part for zero, one, or many Part objects.

4.3.77 Dm_execution_result to Dm_execution_result_measurement

Each Dm_execution_result has the dm_measurements defined by one or more Dm_execution_result_-measurement. Each Dm_execution_result_measurement defines the dm_measurements for zero, one, or many Dm_execution_result objects.

4.3.78 Dm_execution_result to Dm_program_run

Each Dm_execution_result has the program_run defined by exactly one Dm_program_run. Each Dm_-program_run defines the program_run for zero, one, or many Dm_execution_result objects.

4.3.79 Dm_execution_result_measurement to Dm_data_analysis_software

Each Dm_execution_result_measurement has the software defined by zero or one Dm_data_aquisition_-software. Each Dm_data_analysis_software defines the software for zero, one, or many Dm_-execution_result_measurement objects.

4.3.80 Dm_execution_result_measurement to Dm_point

Each Dm_execution_result_measurement has the measurement_points defined by one or more Dm_point. Each Dm_point defines the measurement_points for zero, one, or many Dm_execution_result_-measurement objects.

4.3.81 Dm_execution_result_measurement to Dm_result_parameter

Each Dm_execution_result_measurement has the parameter defined by exactly one Dm_result_-parameter. Each Dm_result_parameter defines the parameter for zero, one, or many Dm_execution_-result_measurement objects.

4.3.82 Dm_feature to Cartesian_point

Each Dm_feature has the boundarydefined by zero, two or more Cartesian_point. Each Cartesian_point defines the boundaryfor zero, one, or many Dm_feature e objects.

4.3.83 Dm_feature_analysis_mode_dml to Dm_feature_analysis_mode_-default_dml

Each Dm_feature_analysis_mode_dml has the default_values defined by exactly one Dm_feature_-analysis_mode_default_dml. Each Dm_feature_analysis_mode_default_dml defines the default_values for zero, one, or many Dm_feature_analysis_mode_dml objects.

4.3.84 Dm_parameter_analysis_dml to Dm_analysis_dofs_dml

Each Dm_parameter_analysis_dml has the dofs defined by zero or one Dm_analysis_dofs_dml. Each Dm_analysis_dofs_dml defines the dofs for zero, one, or many Dm_parameter_analysis_dml objects.

4.3.85 Dm_parameter_analysis_dml to Dm_feature_analysis_mode_dml

Each Dm_parameter_analysis_dml has the feature_analysis defined by zero or one Dm_feature_-analysis_mode_dml. Each Dm_feature_analysis_mode_dml defines the feature_analysis for zero, one, or many Dm_parameter_analysis_dml objects.

4.3.86 Dm_parameter_analysis_dml to Dm_tolerance_analysis_mode_dml

Each Dm_parameter_analysis_dml has the tolerance_analysis defined by zero or one Dm_tolerance_-analysis_mode_dml. Each Dm_tolerance_analysis_mode_dml defines the tolerance_analysis for zero, one, or many Dm_parameter_analysis_dml objects.

4.3.87 Dm_parameter_value_limits to Numeric_parameter

Each Dm_parameter_value_limits has the calculated_limits defined by zero or two Numeric_parameter. Each Numeric_parameter defines the calculated_limits for zero, one, or many Dm_parameter_value_-limits objects.

4.3.88 Dm_point to Cartesian_point

Each Dm_point has the expected_point defined by zero or one Cartesian_point. Each Cartesian_point defines the expected_point for zero, one, or many Dm_point objects.

Each Dm_point has the measured_point defined by exactly one Cartesian_point. Each Cartesian_point defines the measured_point for zero, one, or many Dm_point objects.

4.3.89 Dm_point_parameter to Cartesian_point

Each Dm_point_parameter has the calculated_point defined by exactly one Cartesian_point. Each Cartesian_point defines the calculated_point for zero, one, or many Dm_point_parameter objects.

4.3.90 Dm_point_parameter to Measurement_uncertainty

Each Dm_point_parameter has the value_uncertainty defined by one to three Measurement_uncertainty. Each Measurement_uncertainty defines the value_uncertainty for zero, one, or many Dm_point_parameter objects.

4.3.91 Dm_program_identification to Organization

Each Dm_program_identification has the program_custodian defined by zero or one Organization. Each Organization defines the program_custodian for zero, one, or many Dm_program_identification objects.

NOTE This assertion is established through Organization_or_person_in_organization_select.

4.3.92 Dm_program_identification to Person_in_organization

Each Dm_program_identification has the program_custodian defined by zero or one Person_in_-organization. Each Person_in_organization defines the program_custodian for zero, one, or many Dm_-program_identification objects.

NOTE This assertion is established through Organization_or_person_in_organization_select.

4.3.93 Dm_program_run to Calendar_date

Each Dm_program_run has the run_start defined by zero or one Calendar_date. Each Calendar_date defines the run_start for zero, one, or many Dm_program_run objects.

Each Dm_program_run has the run_end defined by zero or one Calendar_date. Each Calendar_date defines the run_end for zero, one, or many Dm_program_run objects.

Each Dm_program_run has the run_date defined by exactly one Calendar_date. Each Calendar_date defines the run_date for zero, one, or many Dm_program_run objects.

NOTE This assertion is established through Date_or_date_time_select.

4.3.94 Dm_program_run to Date_time

Each Dm_program_run has the run_start defined by zero or one Date_time. Each Date_time defines the run_start for zero, one, or many Dm_program_run objects.

Each Dm_program_run has the run_end defined by zero or one Date_time. Each Date_time defines the run_end for zero, one, or many Dm_program_run objects.

Each Dm_program_run has the run_date defined by exactly one Date_time. Each Date_time defines the run_date for zero, one, or many Dm_program_run objects.

NOTE This assertion is established through Date_or_date_time_select.

4.3.95 Dm_program_run to Dm_program_identification

Each Dm_program_run has the program_id defined by exactly one Dm_program_identification. Each Dm_program_identification defines the program_id for zero, one, or many Dm_program_run objects.

4.3.96 Dm_program_run to Measurement_location

Each Dm_program_run has the run_location defined by zero or one Measurement_location. Each Measurement_location defines the run_location for zero, one, or many Dm_program_run objects.

4.3.97 Dm_program_run to Numeric_parameter

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

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

Each Dm_program_run has the workpiece_temperature defined by zero or one Numeric_parameter. Each Numeric_parameter defines the workpiece_temperature for zero, one, or many Dm_program_run objects

4.3.98 Dm_program_run to Run_administrator

Each Dm_program_run has the run_administrator defined by exactly one Run_administrator. Each Run_administrator defines the run_administrator for zero, one, or many Dm_program_run objects.

4.3.99 Dm_result_parameter to Dm_parameter_analysis_dml

Each Dm_result_parameter has the analysis_conditions defined by zero or one Dm_parameter_-analysis_dml. Each Dm_parameter_analysis_dml defines the analysis_conditions for zero, one, or many Dm_result_parameter objects.

4.3.100 Dm_result_parameter to Dm_parameter_value_limits

Each Dm_result_parameter has the limits defined by zero or one Dm_parameter_value_limits. Each Dm_parameter_value_limits defines the limits for zero, one, or many Dm_result_parameter objects.

4.3.101 Dm_result_parameter to Dimensional_tolerance

Each Dm_result_parameter has a specification defined by zero or one Dimensional_tolerance. Each Dimensional_tolerance defines the specification for zero, one, or many Dm_result_parameter objects.

NOTE This assertion is established through specification_select.

4.3.102 Dm_result_parameter to Geometric_tolerance

Each Dm_result_parameter has a specification defined by zero or one Geometric_tolerance. Each Geometric_tolerance defines the specification for zero, one, or many Dm_result_parameter objects.

NOTE This assertion is established through specification_select.

4.3.103 Dm_result_parameter to Manufacturing_feature

Each Dm_result_parameter has a specification defined by zero or one Manufacturing_feature. Each Manufacturing_feature defines the specification for zero, one, or many Dm_result_parameter objects.

NOTE This assertion is established through specification_select.

4.3.104 Dm_result_parameter to Shape_element

Each Dm_result_parameter has a specification defined by zero or one Shape_element. Each Shape_-element defines the specification for zero, one, or many Dm_result_parameter objects.

NOTE This assertion is established through specification_select.

4.3.105 Dm_tolerance_analysis_mode_dml to Dm_tolerance_analysis_-mode_default_dml

Each Dm_tolerance_analysis_mode_dml has the default_values defined by exactly one Dm_tolerance_-analysis_mode_default_dml. Each Dm_tolerance_analysis_mode_default_dml defines the default_values for zero, one, or many Dm_tolerance_analysis_mode_dml objects.

4.3.106 Dm_vector_parameter to Cartesian_vector

Each Dm_vector_parameter has the calculated_vector defined by exactly one Cartesian_vector. Each Cartesian_vector defines the calculated_vector for zero, one, or many Dm_vector_parameter objects.

4.3.107 Dm_vector_parameter to Measurement_uncertainty

Each Dm_vector_parameter has the value_uncertainty defined by one to three Measurement_uncertainty. Each Measurement_uncertainty defines the value_uncertainty for zero, one, or many Dm_vecor_parameter objects.

4.3.108 Dmf_arc to Dm_dimension_parameter

Each Dmf_arc has the radius defined by exactly one Dm_dimension_parameter. Each Dm_dimension_-parameter defines the radius for zero, one, or many Dmf_arc objects.

4.3.109 Dmf_arc to Dm_point_parameter

Each Dmf_arc has the center_point defined by exactly one Dm_point_parameter. Each Dm_point_-parameter defines the center_point for zero, one, or many Dmf_arc objects

4.3.110 Dmf_arc to Dm_vector_parameter

Each Dmf_arc has the axis_direction_vector defined by exactly one Dm_vector_parameter. Each Dm_-vector_parameter defines the axis_direction_vector for zero, one, or many Dmf_arc objects.

Each Dmf_arc has the end_vector defined by exactly one Dm_vector_parameter. Each Dm_-vector_parameter defines the end_vector for zero, one, or many Dmf_arc objects.

Each Dmf_arc has the start_vector defined by exactly one Dm_vector_parameter. Each Dm_-vector_parameter defines the start_vector for zero, one, or many Dmf_arc objects.

4.3.111 Dmf_circle to Dm_dimension_parameter

Each Dmf_circle has the diameter defined by exactly one Dm_dimension_parameter. Each Dm_-dimension_parameter defines the diameter for zero, one, or many Dmf_circle objects.

4.3.112 Dmf_circle to Dm_point_parameter

Each Dmf_circle has the center_point defined by exactly one Dm_point_parameter. Each Dm_-point_parameter defines the center_point for zero, one, or many Dmf_circle objects

4.3.113 Dmf_circle to Dm_vector_parameter

Each Dmf_circle has the axis_direction_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the axis_direction_vector for zero, one, or many Dmf_circle objects.

4.3.114 Dmf_cone to Dm_dimension_parameter

Each Dmf_cone has the included_angle defined by exactly one Dm_dimension_parameter. Each Dm_-dimension_parameter defines the included_angle for zero, one, or many Dmf_cone objects.

Each Dmf_cone has the diameter defined by exactly one Dm_dimension_parameter. Each Dm_-dimension_parameter defines the diameter for zero, one, or many Dmf_cone objects.

Each Dmf_cone has the end_length_dml defined by zero or one Dm_dimension_parameter. Each Dm_-dimension_parameter defines the end_length_dml for zero, one, or many Dmf_cone objects.

Each Dmf_cone has the start_length_dmldefined by zero or one Dm_dimension_parameter. Each Dm_-dimension_parameter defines the start_length_dml for zero, one, or many Dmf_cone objects.

4.3.115 Dmf_cone to Dm_point_parameter

Each Dmf_cone has the apex_point defined by exactly one Dm_point_parameter. Each Dm_point_-parameter defines the apex_point for zero, one, or many Dmf_cone objects

Each Dmf_cone has the end_point defined by zero or one Dm_point_parameter. Each Dm_point_-parameter defines the end_point for zero, one, or many Dmf_cone objects

Each Dmf_cone has the start_point defined by zero or one Dm_point_parameter. Each Dm_point_-parameter defines the start_point for zero, one, or many Dmf_cone objects

4.3.116 Dmf_cone to Dm_vector_parameter

Each Dmf_cone has the axis_direction_vector defined by exactly one Dm_vector_parameter. Each Dm_-vector_parameter defines the axis_direction_vector for zero, one, or many Dmf_cone objects.

Each Dmf_cone has the end_vector defined by zero or one Dm_vector_parameter. Each Dm_vector_-parameter defines the end_vector for zero, one, or many Dmf_cone objects.

Each Dmf_cone has the start_vector defined by zero or one Dm_vector_parameter. Each Dm_vector_-parameter defines the start_vector for zero, one, or many Dmf_cone objects.

4.3.117 Dmf_cylinder to Dm_dimension_parameter

Each Dmf_cylinder has the diameter defined by exactly one Dm_dimension_parameter. Each Dm_-dimension_parameter defines the diameter for zero, one, or many Dmf_cylinder objects.

Each Dmf_cylinder has the length_dml defined by zero or one Dm_dimension_parameter. Each Dm_-dimension_parameter defines the length_dml for zero, one, or many Dmf_cylinder objects.

4.3.118 Dmf_cylinder to Dm_point_parameter

Each Dmf_cylinder has the center_point defined by exactly one Dm_point_parameter. Each Dm_point_-parameter defines the center_point for zero, one, or many Dmf_cylinder objects

Each Dmf_cylinder has the end_point defined by exactly one Dm_point_parameter. Each Dm_point_-parameter defines the end_point for zero, one, or many Dmf_cylinder objects

Each Dmf_cylinder has the start_point defined by exactly one Dm_point_parameter. Each Dm_point_-parameter defines the start_point for zero, one, or many Dmf_cylinder objects

4.3.119 Dmf_cylinder to Dm_vector_parameter

Each Dmf_cylinder has the axis_direction_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the axis_direction_vector for zero, one, or many Dmf_cylinder objects.

4.3.120 Dmf_edge_point to Dm_vector_parameter

Each Dmf_edge_point has the edge_normal_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the edge_normal_vector for zero, one, or many Dmf_edge_point objects.

Each Dmf_edge_point has the surface_normal_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the surface_normal_vector for zero, one, or many Dmf_edge_point objects.

4.3.121 Dmf_edge_point to Dm_point_parameter

Each Dmf_edge_point has the location_point defined by exactly one Dm_point_parameter. Each Dm_-point_parameter defines the location_point for zero, one, or many Dmf_edge_point objects

4.3.122 Dmf_ellipse to Dm_dimension_parameter

Each Dmf_ellipse has the major_diameter defined by exactly one Dm_dimension_parameter. Each Dm_dimension_parameter defines the major_diameter for zero, one, or many Dmf_ellipse objects.

Each Dmf_ellipse has the minor_diameter defined by exactly one Dm_dimension_parameter. Each Dm_dimension_parameter defines the minor_diameter for zero, one, or many Dmf_ellipse objects.

4.3.123 Dmf_ellipse to Dm_point_parameter

Each Dmf_ellipse has the center_point defined by exactly one Dm_point_parameter. Each Dm_-point_parameter defines the center_point for zero, one, or many Dmf_ellipse objects

Each Dmf_ellipse has the foucus_point_one_dml defined by zero or one Dm_point_parameter. Each Dm_-point_parameter defines the foucus_point_one_dml for zero, one, or many Dmf_ellipse objects

Each Dmf_ellipse has the focus_point_two_dml defined by zero or one Dm_point_parameter. Each Dm_-point_parameter defines the focus_point_two_dml for zero, one, or many Dmf_ellipse objects

4.3.124 Dmf_ellipse to Dm_vector_parameter

Each Dmf_ellipse has the major_axis_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the major_axis_vector for zero, one, or many Dmf_ellipse objects.

Each Dmf_ellipse has the normal_direction defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the normal_direction for zero, one, or many Dmf_ellipse objects.

4.3.125 Dmf_generic_feature to Dm_dimension_parameter

Each Dmf_generic_feature has the parameters defined by one or more Dm_dimension_parameter. Each Dm_dimension_parameter defines the parameters for zero, one, or many Dmf_generic_feature objects.

4.3.126 Dmf_generic_feature to Dm_vector_parameter

Each Dmf_generic_feature has the secondary_vector defined by zero or one Dm_vector_parameter. Each Dm_vector_parameter defines the secondary_vector for zero, one, or many Dmf_generic_feature objects.

Each Dmf_generic_feature has the vector_from_center_of_object defined by exactly one Dm_-vector_parameter. Each Dm_vector_parameter defines the vector_from_center_of_object for zero, one, or many Dmf_generic_feature objects.

4.3.127 Dmf_generic_feature to Dm_point_parameter

Each Dmf_generic_feature has the center_point defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the center_point for zero, one, or many Dmf_Generic_Feature objects

4.3.128 Dmf_geometric_curve to Dm_point_parameter

Each Dmf_geometric_curve has the point_on_plane_of_curve defined by exactly one Dm_point_-parameter. Each Dm_point_parameter defines the point_on_plane_of_curve for zero, one, or many Dmf_geometric_curve objects.

4.3.129 Dmf_geometric_curve to Dm_vector_parameter

Each Dmf_geometric_curve has the curve_plane_direction defined by exactly one Dm_vector_-parameter. Each Dm_vector_parameter defines the curve_plane_direction for zero, one, or many Dmf_-geometric_curve objects.

4.3.130 Dmf_geometric_curve to Cartesian_point

Each Dmf_geometric_curve has the data_points defined by zero, two or more Cartesian_point. Each Cartesian_point defines the data_points for zero, one, or many Dmf_geometric_curve objects.

4.3.131 Dmf_geometric_surface to Dm_point_parameter

Each Dmf_geometric_surface has the point_on_surface defined by exactly one Dm_point_-parameter. Each Dm_point_parameter defines the point_on_surface for zero, one, or many Dmf_Geometric_Surface objects.

4.3.132 Dmf_geometric_surface to Dm_vector_parameter

Each Dmf_Geometric_Surface has the local_surface_normal defined by exactly one Dm_vector_-parameter. Each Dm_vector_parameter defines the local_surface_normal for zero, one, or many Dmf_-Geometric_Surface objects.

4.3.133 Dmf_geometric_surface to Cartesian_point

Each Dmf_geometric_surface has the data_points defined by zero, two or more Cartesian_point. Each Cartesian_point defines the data_points for zero, one, or many Dmf_geometric_surface objects.

4.3.134 Dmf_line_bounded to Dm_point_parameter

Each Dmf_line_bounded has the first_end_point defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the first_end_point for zero, one, or many Dmf_line_bounded objects.

Each Dmf_line_bounded has the second_end_point defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the second_end_point for zero, one, or many Dmf_line_bounded objects.

4.3.135 Dmf_line_bounded to Dm_vector_parameter

Each Dmf_line_bounded has the surface_approach_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the surface_approach_vector for zero, one, or many Dmf_line_-bounded objects.

Each Dmf_line_bounded has the vector_dml defined by zero or one Dm_vector_parameter. Each Dm_vector_parameter defines the vector_dml for zero, one, or many Dmf_line_bounded objects.

4.3.136 Dmf_line_bounded to Dm_dimension_parameter

Each Dmf_line_bounded has the length_dml defined by zero or one Dm_dimension_parameter. Each Dm_dimension_parameter defines the length_dml for zero, one, or many Dmf_line_bounded objects.

4.3.137 Dmf_line_closed_parallel to Dm_dimension_parameter

Each Dmf_line_closed_parallel has the feature_length defined by exactly one Dm_dimension_parameter. Each Dm_dimension_parameter defines the feature_length for zero, one, or many Dmf_line_closed_-parallel objects.

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

4.3.138 Dmf_line_closed_parallel to Dm_point_parameter

Each Dmf_line_closed_parallel has the center_point defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the center_point for zero, one, or many Dmf_line_closed_parallel objects.

4.3.139 Dmf_line_closed_parallel to Dm_vector_parameter

Each Dmf_line_closed_parallel has the axis_direction_vector defined by exactly one Dm_vector_-parameter. Each Dm_vector_parameter defines the axis_direction_vector for zero, one, or many Dmf_line_closed_parallel objects.

Each Dmf_line_closed_parallel has the longitude_vectordefined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the longitude_vectorfor zero, one, or many Dmf_line_closed_parallel objects.

4.3.140 Dmf_line_unbounded to Dm_vector_parameter

Each Dmf_line_unbounded has the direction_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the direction_vector for zero, one, or many Dmf_line_unbounded objects.

Each Dmf_line_unbounded has the surface_approach_vector defined by exactly one Dm_-vector_parameter. Each Dm_vector_parameter defines the surface_approach_vector for zero, one, or many Dmf_line_unbounded objects.

4.3.141 Dmf_line_unbounded to Dm_point_parameter

Each Dmf_line_unbounded has the point_on_line defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the point_on_line for zero, one, or many Dmf_line_unbounded objects.

4.3.142 Dmf_pattern to Dm_feature

Each Dmf_pattern has the features defined by two or more Dm_feature objects. Each Dm_feature defines the features for zero, one, or many Dmf_pattern objects.

4.3.143 Dmf_pattern to Dm_point_parameter

Each Dmf_pattern has the point defined by exactly one Dm_point_parameter. Each Dm_-point_parameter defines the point for zero, one, or many Dmf_pattern objects.

4.3.144 Dmf_pattern to Dm_vector_parameter

Each Dmf_pattern has the direction_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the direction_vector for zero, one, or many Dmf_pattern objects.

4.3.145 Dmf_plane to Dm_point_parameter

Each Dmf_plane has the point_on_plane defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the point_on_plane for zero, one, or many Dmf_plane objects.

4.3.146 Dmf_plane to Dm_vector_parameter

Each Dmf_plane has the direction_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the direction_vector for zero, one, or many Dmf_plane objects.

4.3.147 Dmf_plane_closed_parallel to Dm_dimension_parameter

Each Dmf_plane_closed_parallel has the feature_length defined by exactly one Dm_dimension_-parameter. Each Dm_dimension_parameter defines the feature_length for zero, one, or many Dmf_plane_closed_parallel objects.

Each Dmf_plane_closed_parallel has the height defined by zero or one Dm_dimension_parameter. Each Dm_dimension_parameter defines the feature_length for zero, one, or many Dmf_plane_closed_parallel objects.

Each Dmf_plane_closed_parallel has the height defined by exactly one Dm_dimension_parameter. Each Dm_dimension_parameter defines the feature_length for zero, one, or many Dmf_plane_closed_parallel objects.

4.3.148 Dmf_plane_closed_parallel to Dm_point_parameter

Each Dmf_plane_closed_parallel has the center_point defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the center_point for zero, one, or many Dmf_plane_closed_parallel objects.

4.3.149 Dmf_plane_closed_parallel to Dm_vector_parameter

Each Dmf_plane_closed_parallel has the axis_direction_vector defined by exactly one Dm_vector_-parameter. Each Dm_vector_parameter defines the axis_direction_vector for zero, one, or many Dmf_-plane_closed_parallel objects.

Each Dmf_plane_closed_parallel has the longitude_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the longitude_vector for zero, one, or many Dmf_plane_-closed_parallel objects.

Each Dmf_plane_closed_parallel has the normal_dml defined by zero or one Dm_vector_parameter. Each Dm_vector_parameter defines the normal_dml for zero, one, or many Dmf_plane_closed_parallel objects.

4.3.150 Dmf_plane_symmetric to Dm_point_parameter

Each Dmf_plane_symmetric has the point_on_mid_plane defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the point_on_mid_plane for zero, one, or many Dmf_plane_symmetric objects.

Each Dmf_plane_symmetric has the point_on_side_two defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines thepoint_on_side_two for zero, one, or many Dmf_plane_symmetric objects.

Each Dmf_plane_symmetric has the point_on_side_one defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the point_on_side_one for zero, one, or many Dmf_plane_symmetric objects.

4.3.151 Dmf_plane_symmetric to Dm_dimension_parameter

Each Dmf_plane_symmetric has the width_at_mid_point defined by exactly one Dm_dimension_-parameter. Each Dm_dimension_parameter defines the width_at_mid_point for zero, one, or many Dmf_plane_symmetric objects.

4.3.152 Dmf_plane_symmetric to Dm_vector_parameter

Each Dmf_plane_symmetric has the direction_vector_side_one defined by exactly one Dm_-vector_parameter. Each Dm_vector_parameter defines the direction_vector_side_one for zero, one, or many Dmf_plane_symmetric objects.

Each Dmf_plane_symmetric has the direction_vector_side_two defined by exactly one Dm_-vector_parameter. Each Dm_vector_parameter defines the direction_vector_side_twofor zero, one, or many Dmf_plane_symmetric objects.

4.3.153 Dmf_point to Dm_point_parameter

Each Dmf_point has the point defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the point for zero, one, or many Dmf_point objects.

4.3.154 Dmf_point to Dm_vector_parameter

Each Dmf_point has the direction_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the direction_vector for zero, one, or many Dmf_point objects.

4.3.155 Dmf_sphere to Dm_dimension_parameter

Each Dmf_sphere has the latitude_start_angle defined by zero or one Dm_dimension_parameter. Each Dm_dimension_parameter defines the latitude_start_angle for zero, one, or many Dmf_sphere objects.

Each Dmf_sphere has the latitude_stop_angle defined by zero or one Dm_dimension_parameter. Each Dm_dimension_parameter defines the latitude_stop_angle for zero, one, or many Dmf_sphere objects.

Each Dmf_sphere has the longitude_start_angle defined by zero or one Dm_dimension_parameter. Each Dm_dimension_parameter defines the longitude_start_angle for zero, one, or many Dmf_sphere objects.

Each Dmf_sphere has the longitude_stop_angle defined by zero or one Dm_dimension_parameter. Each Dm_dimension_parameter defines the longitude_stop_angle for zero, one, or many Dmf_sphere objects.

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

4.3.156 Dmf_sphere to Dm_point_parameter

Each Dmf_sphere has the center_point defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the center_point for zero, one, or many Dmf_sphere objects.

4.3.157 Dmf_sphere to Dm_vector_parameter

Each Dmf_sphere has the prime_meridian_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the prime_meridian_vector for zero, one, or many Dmf_sphere objects.

Each Dmf_sphere has the north_pole_vector defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the north_pole_vector for zero, one, or many Dmf_sphere objects.

4.3.158 Dmf_surface_of_revolution to Dm_point_parameter

Each Dmf_surface_of_revolution has the profile_curve defined by two or more Dm_point_parameter. Each Dm_point_parameter defines the profile_curve for zero, one, or many Dmf_surface_of_revolution objects.

4.3.159 Dmf_surface_of_revolution to Dm_vector_parameter

Each Dmf_surface_of_revolution has the axis_of_revolution defined by exactly one Dm_vector_-parameter. Each Dm_vector_parameter defines the axis_of_revolution for zero, one, or many Dmf_-surface_of_revolution objects.

4.3.160 Dmf_torus to Dm_dimension_parameter

Each Dmf_torus has the major_diameter defined by exactly one Dm_dimension_parameter. Each Dm_dimension_parameter defines the major_diameter for zero, one, or many Dmf_torus objects.

Each Dmf_torus has the minor_diameterdefined by exactly one Dm_dimension_parameter. Each Dm_dimension_parameter defines the minor_diameterfor zero, one, or many Dmf_torus objects.

4.3.161 Dmf_torus to Dm_point_parameter

Each Dmf_torus has the center_point defined by exactly one Dm_point_parameter. Each Dm_point_parameter defines the center_point for zero, one, or many Dmf_torus objects.

4.3.162 Dmf_torus to Dm_vector_parameter

Each Dmf_torus has the vector_of_plane defined by exactly one Dm_vector_parameter. Each Dm_vector_parameter defines the vector_of_plane for zero, one, or many Dmf_torus objects.

4.3.163 Document_assignment to Specification

Each Document_assignment has the assigned_document defined by zero, one, or many Specification objects. Each Specification defines the assigned_document for exactly one Document_assignment.

4.3.164 Dof_attribute_dml to Numeric_parameter

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

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

4.3.165 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.

4.3.166 Explicit_base_shape_representation to Brep_shape_representation

Each Explicit_base_shape_representation has the Brep_form defined by exactly one Brep_-shape_representation. Each Brep_shape_representation defines the Brep_form for zero, one, or many Explicit_base_shape_representation objects.

4.3.167 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.

4.3.168 Externally_defined_size_dimension to Shape_aspect

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

4.3.169 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.

4.3.170 Feature to Orientation

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

4.3.171 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.

4.3.172 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.

4.3.173 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.

4.3.174 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.

4.3.175 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.

4.3.176 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.

4.3.177 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.

4.3.178 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.

4.3.179 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.

4.3.180 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.

4.3.181 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.

4.3.182 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.

4.3.183 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.

4.3.184 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.

4.3.185 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.

4.3.186 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.

4.3.187 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.

4.3.188 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.

4.3.189 General_pocket to Boss

Each General_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 General_pocket objects.

NOTE This assertion is established through Volume_select.

4.3.190 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.

4.3.191 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.

4.3.192 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.

4.3.193 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.

4.3.194 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.

4.3.195 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.

4.3.196 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_face for zero, one, or many General_rib_top_floor objects.

4.3.197 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.

4.3.198 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.

4.3.199 Geometric_model to Cartesian_coordinate_space

Each Geometric_model has the in_defined_by defined by exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space defines the is_defined_in for zero, one, or many Geometric_model objects.

4.3.200 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.

4.3.201 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.

4.3.202 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.

4.3.203 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.

4.3.204 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.

4.3.205 Height_dimension to Shape_aspect

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

4.3.206 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.

4.3.207 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.

4.3.208 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.

4.3.209 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.

4.3.210 Inspection_feature_relationship to Dm_feature

Each Inspection_feature_relationship has inspection_feature defined by exactly one Dm_feature. Each Dm_feature defines the inspection_feature for zero, one, or many Inspection_feature_relationship objects.

4.3.211 Inspection_feature_relationship to Manufacturing_feature

Each Inspection_feature_relationship has as_designed_feature defined by exactly one Manufacturing_-feature. Each Manufacturing_feature defines the as_designed_feature for zero, one, or many Inspection_-feature_relationship objects.

4.3.212 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.

4.3.213 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.

4.3.214 Length_dimension to Shape_aspect

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

4.3.215 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.

4.3.216 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.

4.3.217 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.

4.3.218 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.

4.3.219 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.

4.3.220 Location_dimension_tolerance to Orientation

Each Location_dimension_tolerance has the plane and direction defined by zero or one Orientation. Each Orientation defines the plane and direction for zero, one, or many Location_dimension_tolerance objects.

4.3.221 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_tolerance has the origin_shape defined by exactly one Shape_element. Each Shape_-element defines the origin_shape for zero, one, or many Location_tolerance objects.

4.3.222 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.

4.3.223 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.

4.3.224 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.

4.3.225 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.

4.3.226 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.

4.3.227 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.

4.3.228 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.

4.3.229 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.

4.3.230 Numeric_parameter_with_tolerance to Plus_minus_value

EachNumeric_parameter_with_tolerance has the implicit_tolerance defined by exactly one Plus_-minus_value. Each Plus_minus_value defines the implicit_tolerance for zero, one, or many Numeric_-parameter_with_tolerance objects.

NOTE This assertion is established through the Numeric_parameter_tolerance_select.

4.3.231 Numeric_parameter_with_tolerance to Tolerance_limit

EachNumeric_parameter_with_tolerance has the implicit_tolerance defined by exactly one Tolerance_-limit. Each Tolerance_limit defines the implicit_tolerance for zero, one, or many Numeric_parameter_-with_tolerance objects.

NOTE This assertion is established through the Numeric_parameter_tolerance_select.

4.3.232 Numeric_parameter_with_tolerance to Tolerance_value

EachNumeric_parameter_with_tolerance has the implicit_tolerance defined by exactly one Tolerance_-value. Each Tolerance_value defines the implicit_tolerance for zero, one, or many Numeric_parameter_-with_tolerance objects.

NOTE This assertion is established through the Numeric_parameter_tolerance_select.

4.3.233 Numeric_parameter_with_tolerance to Limit_and_fits

EachNumeric_parameter_with_tolerance has the implicit_tolerance defined by exactly one Limit_-and_fits. Each Limit_and_fits defines the implicit_tolerance for zero, one, or many Numeric_parameter_-with_tolerance objects.

NOTE This assertion is established through the Numeric_parameter_tolerance_select.

4.3.234 Offset_shape_element to Numeric_parameter

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

4.3.235 Offset_shape_element to Direction_element

Each Offset_shape_element has the offset_direction defined by exactly one Direction_element. Each Direction_element defines the offset_direction for zero, one, or many Offset_shape_element objects.

4.3.236 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.

4.3.237 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.

4.3.238 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.

4.3.239 Organization to Address

Each Organization has the address defined by zero or one Address. Each Address defines theaddress for zero, one, or many Organization objects.

4.3.240 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.

4.3.241 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.

4.3.242 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.

4.3.243 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.

4.3.244 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.

4.3.245 Outer_diameter_to_shoulder to Vee_profile

Each Outer_diameter_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.

4.3.246 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.

4.3.247 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.

4.3.248 Part to Shape

Each Part has the physical_form definied by exactly one Shape. Each Shape definies the physical_form for zero, one, or many Part objects.

4.3.249 Part_dimensioning_standard to Part_version

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

4.3.250 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.

4.3.251 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.

4.3.252 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.

4.3.253 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.

4.3.254 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.

4.3.255 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.

4.3.256 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.

4.3.257 Perpendicularity_toleranc 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.

4.3.258 Person_in_organization to Address

Each Person_in_organization has the address defined by zero or one Address. Each Address defines the address for zero, one, or many Person_in_organization objects.

4.3.259 Placed_target to Cartesian_corrdinate_space

Each Placed_target has the is_defined_by defined by exactly one Cartesian_corrdinate_space. Each Cartesian_corrdinate_space defines the is_defined_by for zero, one, or many Placed_target objects.

4.3.260 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.

4.3.261 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.

4.3.262 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.

4.3.263 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.

4.3.264 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.

4.3.265 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.

4.3.266 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.

4.3.267 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.

4.3.268 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.

4.3.269 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.

4.3.270 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.

4.3.271 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.

4.3.272 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.

4.3.273 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.

4.3.274 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.

4.3.275 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.

4.3.276 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.

4.3.277 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.

4.3.278 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.

4.3.279 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.

4.3.280 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.

4.3.281 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.

4.3.282 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.

NOTE This assertion is established through pocket_bottom_condition_select.

4.3.283 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.

NOTE This assertion is established through pocket_bottom_condition_select.

4.3.284 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.

4.3.285 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.

4.3.286 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.

4.3.287 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.

4.3.288 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.

4.3.289 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.

4.3.290 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.

4.3.291 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.

4.3.292 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.

4.3.293 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.

4.3.294 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.

4.3.295 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.

4.3.296 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.

4.3.297 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.

4.3.298 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.

4.3.299 Rectangular_closed_pocket to Boss 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.

4.3.300 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.

4.3.301 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.

4.3.302 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.

4.3.303 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.

4.3.304 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.

4.3.305 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.

4.3.306 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.

4.3.307 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.

4.3.308 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.

4.3.309 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.

4.3.310 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.

4.3.311 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.

4.3.312 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.

4.3.313 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.

4.3.314 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.

4.3.315 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.

4.3.316 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.

4.3.317 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.

4.3.318 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.

4.3.319 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.

4.3.320 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.

4.3.321 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.

4.3.322 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.

4.3.323 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.

4.3.324 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.

4.3.325 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.

4.3.326 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.

4.3.327 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.

4.3.328 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.

4.3.329 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.

4.3.330 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.

4.3.331 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.

4.3.332 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.

4.3.333 Rounded_U_profile to Numeric_parameter

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

4.3.334 Run_administrator to Organization

Each Run_administrator has the administrator defined by exactly one Organization. Each Organization defines the program_custodian for zero, one, or many Run_administrator objects.

NOTE This assertion is established through Organization_or_person_in_organization_select.

4.3.335 Run_administrator to Person_in_organization

Each Run_administrator has the administrator defined by exactly one Person_in_-organization. Each Person_in_organization defines the administrator for zero, one, or many Run_administrator objects.

NOTE This assertion is established through Organization_or_person_in_organization_select.

4.3.336 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.

4.3.337 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.

4.3.338 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.

4.3.339 Shape to Brep_shape_representation

Each Shape has the Brep_form defined by zero, one, or many Brep_shape_representation objects. Each Brep_shape_representation defines the Brep_form for exactly one Shape.

4.3.340 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.

4.3.341 Shape_aspect to Brep_model_element

Each Shape_aspect has the Brep_shape defined by zero, one, or many Brep_model_element objects. Each Brep_model_element defines the Brep_shape for exactly one Shape_aspect.

4.3.342 Shape_aspect to Brep_shape_aspect_representation

Each Shape_aspect has the Brep_form defined by zero, one, or many Brep_shape_aspect_representation objects. Each Brep_shape_aspect_representation defines the Brep_form for exactly one Shape_aspect.

4.3.343 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.

4.3.344 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.

4.3.345 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.

4.3.346 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.

4.3.347 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.

4.3.348 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.

4.3.349 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.

4.3.350 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.

4.3.351 Spherical_hole_bottom to Numeric_parameter

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.

4.3.352 Square_U_profile to Numeric_parameter

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.

4.3.353 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.

4.3.354 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.

4.3.355 Step to Boss

Each Step 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 Stepobjects.

NOTE This assertion is established through Volume_select.

4.3.356 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.

4.3.357 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.

4.3.358 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.

4.3.359 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.

4.3.360 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.

4.3.361 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.

4.3.362 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.

4.3.363 Thickness_tolerance to Shape_aspect

Each Thickness_tolerance has the applied_shape defined zero or one Shape_aspect. Each Shape_aspect defines the applied_shape for zero, one, or many Thickness_tolerance objects.

4.3.364 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.

4.3.365 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.

4.3.366 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.

4.3.367 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.

4.3.368 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.

4.3.369 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.

4.3.370 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.

4.3.371 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.

4.3.372 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.

4.3.373 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.

4.3.374 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.

4.3.375 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.

4.3.376 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.

4.3.377 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.

4.3.378 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.

4.3.379 Vee_profile to Numeric_parameter

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.

4.3.380 Width_dimension to Shape_aspect

Each Width_dimension has the applied_shape defined zero or one Shape_aspect. Each Shape_aspect defines the applied_shape for zero, one, or many Width_dimension objects.

4.3.381 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 adjacent row referenced by the -> or <- symbols;

[n] attribute is an aggregation of which member n is given in the following row;

=> entity is a supertype of the entity given in the following row;

<= entity is a subtype of the entity given in the following row;

= the string, select, or enumeration type is constrained to a choice or value;

\ the reference path expression continues on the next line;

* used in conjunction with braces to indicate that any number of relationship entity data types may be assembled in a relationship tree structure;

-- the text following is a comment (normally a clause reference).

5.1.1 administrative data UoF

5.1.1.1 Address

AIM element: address
Source: ISO 10303-41

5.1.1.1.1 country

AIM element: address.country
Source: ISO 10303-41

5.1.1.1.2 electronic_mail_address

AIM element: address.electronic_mail_address
Source: ISO 10303-41

5.1.1.1.3 facsimile_number

AIM element: address.facsimile_number
Source: ISO 10303-41

5.1.1.1.4 internal_location

AIM element: address.internal_location
Source: ISO 10303-41

5.1.1.1.5 postal_box

AIM element: address.postal_box
Source: ISO 10303-41

5.1.1.1.6 postal_code

AIM element: address.postal_code
Source: ISO 10303-41

5.1.1.1.7 region

AIM element: address.region
Source: ISO 10303-41

5.1.1.1.8 street

AIM element: address.street
Source: ISO 10303-41

5.1.1.1.9 street_number

AIM element: address.street_number
Source: ISO 10303-41

5.1.1.1.10 telephone_number

AIM element: address.telephone_number
Source: ISO 10303-41

5.1.1.1.11 telex_number

AIM element: address.telex_number
Source: ISO 10303-41

5.1.1.1.12 town

AIM element: address.town
Source: ISO 10303-41

5.1.1.2 Calendar_date

AIM element: calendar_date
Source: ISO 10303-41

5.1.1.2.1 day

AIM element: day_in_month_number
Source: ISO 10303-41
Reference path: calendar_date
calendar_date.day_component ->
day_in_month_number

5.1.1.2.2 month

AIM element: month_in_year_number
Source: ISO 10303-41
Reference path: calendar_date
calendar_date.month_component ->
month_in_year_number

5.1.1.2.3 year

AIM element: year_number
Source: ISO 10303-41
Reference path: calendar_date<=
date
date.year_component ->
year_number

5.1.1.3 Date_time

AIM element: date_and_time
Source: ISO 10303-41

5.1.1.3.1 hour

AIM element: hour_in_day
Source: ISO 10303-41
Reference path: date_and_time
date_and_time.time_component->
local_time
local_time.hour_component ->
hour_in_day

5.1.1.3.2 minute

AIM element: minute_in_hour
Source: ISO 10303-41
Reference path: date_and_time
date_and_time.time_component->
local_time
local_time.minute_component ->
minute_in_hour

5.1.1.3.3 second

AIM element: second_in_minute
Source: ISO 10303-41
Reference path: date_and_time
date_and_time.time_component->
local_time
local_time.second_component ->
second_in_minute

5.1.1.3.4 date_time to calendar_date (as date)

AIM element: year_number
Source: ISO 10303-41
Reference path: date_and_time
date_and_time.date_component->
date=>
calendar_date

5.1.1.3.5 date_time to time_offset (as offset)

AIM element: coordinated_universal_time_offset
Source: ISO 10303-41
Reference path: date_and_time
date_and_time.time_component->
local_time
local_time.zone ->
coordinated_universal_time_offset

5.1.1.4 Organization

AIM element: organization
Source: ISO 10303-41

5.1.1.4.1 id

AIM element: organization.id
Source: ISO 10303-41

5.1.1.4.2 short_name

AIM element: organization.name
Source: ISO 10303-41

5.1.1.4.3 long_name

AIM element: organization.description
Source: ISO 10303-41

5.1.1.4.4 organization to address (as address)

AIM element: address
Source: ISO 10303-41
Reference path: organization<-
organizational_address.organizations[i]
organizational_address<=
address

5.1.1.5 Person_in_organization

AIM element: person
Source: ISO 10303-41

5.1.1.5.1 id

AIM element: person.id
Source: ISO 10303-41

5.1.1.5.2 last_name

AIM element: person.last_name
Source: ISO 10303-41

5.1.1.5.3 first_name

AIM element: person.first_name
Source: ISO 10303-41

5.1.1.5.4 middle_names

AIM element: person.middle_names
Source: ISO 10303-41

5.1.1.5.5 prefix_titles

AIM element: person.prefix_titles
Source: ISO 10303-41

5.1.1.5.6 suffix_titles

AIM element: person.suffix_titles
Source: ISO 10303-41

5.1.1.5.7 person_in_organization to address (as address)

AIM element: address
Source: ISO 10303-41
Reference path: person<-
personal_address.people[i]
personal_address<=
address

5.1.1.6 Time_offset

AIM element: coordinated_universal_time_offset
Source: ISO 10303-41

5.1.1.6.1 direction

AIM element: coordinated_universal_time_offset. sense
Source: ISO 10303-41

5.1.1.6.2 hour

AIM element: coordinated_universal_time_offset. hour_offset
Source: ISO 10303-41

5.1.1.6.3 minute

AIM element: coordinated_universal_time_offset. minute_offset
Source: ISO 10303-41

5.1.2 dimensional_measurement_analysis UoF

5.1.2.1 Dm_analysis_dofs_dml

AIM element: dm_analysis_dofs_dml
Source: ISO 10303-219
Reference path: dm_analysis_dofs_dml<=
dimensional_measurement_representation<=
representation

5.1.2.1.1 dm_analysis_dofs_dml to dof_attribute_dml (as rotx)

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: dm_analysis_dofs_dml<=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'rotx upper limit')
(representation_item.name = 'rotx lower limit')
measure_representation_item

5.1.2.1.2 dm_analysis_dofs_dml to dof_attribute_dml (as roty)

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: dm_analysis_dofs_dml<=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'roty upper limit')
(representation_item.name = 'roty lower limit')
measure_representation_item

5.1.2.1.3 dm_analysis_dofs_dml to dof_attribute_dml (as rotz)

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: dm_analysis_dofs_dml<=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'rotz upper limit')
(representation_item.name = 'rotz lower limit')
measure_representation_item

5.1.2.1.4 dm_analysis_dofs_dml to dof_attribute_dml (as transx)

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: dm_analysis_dofs_dml<=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'transx upper limit')
(representation_item.name = 'transx lower limit')
measure_representation_item

5.1.2.1.5 dm_analysis_dofs_dml to dof_attribute_dml (as transy)

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: dm_analysis_dofs_dml<=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'transy upper limit')
(representation_item.name = 'transy lower limit')
measure_representation_item

5.1.2.1.6 dm_analysis_dofs_dml to dof_attribute_dml (as transz)

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: dm_analysis_dofs_dml<=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'transz upper limit')
(representation_item.name = 'transz lower limit')
measure_representation_item

5.1.2.2 Dm_feature_analysis_mode_default_dml

AIM element: dm_feature_analysis_mode_dml
Source: ISO 10303-219
Reference path: dm_feature_analysis_mode_dml<=
dimensional_measurement_representation<=
representation

5.1.2.2.1 axis_method

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode axis method default ')
descriptive_representation_item
descriptive_representation_item.description
descriptive_representation_item.description='least square'

5.1.2.2.2 axis_extrapolate

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode axis extrapolate default')
descriptive_representation_item
descriptive_representation_item.description
descriptive_representation_item.description='true'

5.1.2.2.3 aelpr_analysis

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode aelpr analysis default')
descriptive_representation_item
descriptive_representation_item.description
descriptive_representation_item.description='least square'

5.1.2.2.4 cccpst_analysis

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode ccpst analysis default')
descriptive_representation_item
descriptive_representation_item.description
descriptive_representation_item.description='least square'

5.1.2.2.5 curve_analysis

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode curve analysis default')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='least square'

5.1.2.2.6 limits_of_size

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode limits of size default ')
descriptive_representation_item
descriptive_representation_item.description
descriptive_representation_item.description='functional'

5.1.2.2.7 surface_analysis

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode surface analysis default')
descriptive_representation_item
descriptive_representation_item.description
descriptive_representation_item.description='least square'

5.1.2.3 Dm_feature_analysis_mode_dml

AIM element: dm_feature_analysis_mode_dml
Source: ISO 10303-219
Reference path: dm_feature_analysis_mode_dml<=
dimensional_measurement_representation<=
representation

5.1.2.3.1 axis_method

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode axis method')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='least square')
(descriptive_representation_item.description='minimum radius')
(descriptive_representation_item.description='cross section centers')

5.1.2.3.2 axis_extrapolate

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode axis extrapolate')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='true')
(descriptive_representation_item.description='false')
(descriptive_representation_item.description='unknown')

5.1.2.3.3 aelpr_analysis

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode aelpr analysis')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='least square')
(descriptive_representation_item.description='minimum maximum')

5.1.2.3.4 cccpst_analysis

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode ccpst analysis')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='least square')
(descriptive_representation_item.description='minimum maximum')
(descriptive_representation_item.description='minimum size circumscribed')
(descriptive_representation_item.description='maximum size inscribed')

5.1.2.3.5 curve_analysis

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode curve analysis')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='least squares')
(descriptive_representation_item.description='minimum maximum')
(descriptive_representation_item.description='bspline')

5.1.2.3.6 limits_of_size

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode limits of size')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='functional')
(descriptive_representation_item.description='mrs average')
(descriptive_representation_item.description='two point')

5.1.2.3.7 surface_analysis

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_feature_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature analysis mode surface anaysis')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='least squares')
(descriptive_representation_item.description='minimum maximum')
(descriptive_representation_item.description='nurbs')
(descriptive_representation_item.description='bezier')

5.1.2.3.8 dm_feature_analysis_mode_dml to dm_feature_analysis_mode_-default_dml (as default_values)

AIM element: PATH
Reference path: IDENTICAL MAPPING

5.1.2.4 Dm_parameter_analysis_dml

AIM element: dm_parameter_analysis_dml
Source: ISO 10303-219
Reference path: dm_parameter_analysis_dml<=
action_property

5.1.2.4.1 dm_parameter_analysis_dml to dm_feature_analysis_mode_dml (as feature_analysis)

AIM element: PATH
Source: ISO 10303-219
Reference path: dm_parameter_analysis_dml<=
action_property<-
action_property_representation.property
action_property_representation
action_property_representation.representation->
representation=>
dimensional_measurement_representation=>
dm_feature_analysis_mode_dml

5.1.2.4.2 dm_parameter_analysis_dml to dm_tolerance_analysis_mode_dml (as tolerance_analysis)

AIM element: PATH
Source: ISO 10303-219
Reference path: dm_parameter_analysis_dml<=
action_property<-
action_property_representation.property
action_property_representation
action_property_representation.representation->
representation=>
dimensional_measurement_representation=>
dm_tolerance_analysis_mode_dml

5.1.2.4.3 dm_parameter_analysis_dml to dm_analysis_dofs_dml (as dofs)

AIM element: PATH
Source: ISO 10303-219
Reference path: dm_parameter_analysis_dml<=
action_property<-
action_property_representation.property
action_property_representation
action_property_representation.representation->
representation=>
dimensional_measurement_representation=>
dm_analysis_dofs_dml

5.1.2.5 Dm_tolerance_analysis_mode_default_dml

AIM element: dm_tolerance_analysis_mode_dml
Source: ISO 10303-219
Reference path: dm_feature_tolerance_mode_dml<=
dimensional_measurement_representation<=
representation

5.1.2.5.1 method

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_tolerance_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'tolerance analysis mode method default')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='standard')

5.1.2.5.2 option

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_tolerance_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'tolerance analysis mode option default')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='best fit')

5.1.2.5.3 setting

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_tolerance_analysis_mode_dml dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'tolerance analysis mode setting default')
descriptive_representation_item
descriptive_representation_item.description
descriptive_representation_item.description='standard'

5.1.2.6 Dm_tolerance_analysis_mode_dml

AIM element: dm_tolerance_analysis_mode_dml
Source: ISO 10303-219
Reference path: dm_tolerance_tolerance_mode_dml<=
dimensional_measurement_representation<=
representation

5.1.2.6.1 method

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_tolerance_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature tolerance analysis mode method')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='least square3d')
(descriptive_representation_item.description='least square normal')
(descriptive_representation_item.description='minimum deviation 3d'')
(descriptive_representation_item.description='standard')

5.1.2.6.2 option

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_tolerance_analysis_mode_dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = feature tolerance analysis mode option')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='best fit')
(descriptive_representation_item.description='high point')

5.1.2.6.3 setting

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_tolerance_analysis_mode_dml dml <=
dimensional_measurement_representation<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'feature tolerance analysis mode setting')
descriptive_representation_item
descriptive_representation_item.description
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')
(descriptive_representation_item.description='standard')

5.1.2.6.4 dm_tolerance_analysis_mode_dml to dm_tolerance_analysis_-mode_default_dml (as default_values)

AIM element: PATH
Reference path: IDENTICAL MAPPING

5.1.2.7 Dof_attribute_dml

AIM element: measure_representation_item
Source: ISO 10303-45, ISO 10303-45
Reference path: measure_representation_item <=
representation_item

5.1.2.7.1 setting

AIM element: type_qualifier.name
Source: ISO 10303-45
Reference path: measure_representation_item <=
representation_item}
qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier =type_qualifier
(type_qualifier.name='degree of freedom limited')
(type_qualifier.name='degree of freedom allowed'')
(type_qualifier.name='degree of freedom not allowed')

5.1.2.7.2 dof_attribute_dml to numeric_parameter (as lower_limit)

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: {mesure_representation_item <=
representation_item
representation
representation.items[i]->
representation_item=>
(represerntation_item.name='transx lower limit')
(represerntation_item.name='transy lower limit')
(represerntation_item.name='transz lower limit')
(represerntation_item.name='rotx lower limit')
(represerntation_item.name='roty lower limit')
(represerntation_item.name='rotz lower limit')}
{measure_representation_item
measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.7.3 dof_attribute_dml to numeric_parameter (as upper_limit)

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: {mesure_representation_item <=
representation_item}
representation
representation.items[i]->
representation_item=>
(represerntation_item.name='transx upper limit')
(represerntation_item.name='transy upper limit')
(represerntation_item.name='transz upper limit')
(represerntation_item.name='rotx upper limit')
(represerntation_item.name='roty upper limit')
(represerntation_item.name='rotz upper limit')}
{measure_representation_item
measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3 dimensional_measurement_documentation UoF

5.1.3.1 Document_assignment

AIM element: applied_document_reference
Source: ISO 10303-219
Reference path: applied_document_reference<=
document_reference

5.1.3.1.1 role

AIM element: object_role.name
Source: ISO 10303-41
Reference path: applied_document_reference < =
document_reference
role_select=document_reference
role_select<-
role_association.item_with_role
role_association
role_association.role->
object_role
object_role.name

5.1.3.1.2 document_assignment to specification (as assigned_document)

AIM element: PATH
Source: ISO 10303-41
Reference path: applied_document_reference<=
document_reference
document_reference.assigned_document->
document

5.1.3.2 Part_dimensioning_standard

AIM element: applied_document_reference
Source: ISO 10303-219
Reference path: applied_document_reference<=
document_reference
{document_reference.role->
object_role
object_role.name= 'dimensioning standard'}

5.1.3.2.1 part_dimensioning_standard to part (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

5.1.3.3 Specification

AIM element: document
Source: ISO 10303-41

5.1.3.3.1 specification_class

AIM element: document_with_class.class
Source: ISO 10303-41
Reference path: document=>
document_with_class

5.1.3.3.2 specification_description

AIM element: document.description
Source: ISO 10303-41
Reference path: document
document.description

5.1.3.3.3 specification_id

AIM element: document.id
Source: ISO 10303-41
Reference path: document
document.id

5.1.3.3.4 specification to specification_usage_constraint (as constraint)

AIM element: PATH
Source: ISO 10303-41
Reference path: document <-
document_usage_constraint.source
document_usage_constraint

5.1.3.4 Specification_usage_constraint

AIM element: document_usage_constraint
Source: ISO 10303-41

5.1.3.4.1 class_id

AIM element: document_usage_constraint.subject_element_value
Source: ISO 10303-41

5.1.3.4.2 element

AIM element: document_usage_constraint.subject_element
Source: ISO 10303-41

5.1.4 dimensional_measurement_execution UoF

5.1.4.1 Dm_data_analysis_software

AIM element: representation
Source: ISO 10303-43

5.1.4.1.1 application_version

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'application version')
descriptive_representation_item
descriptive_representation_item.description

5.1.4.1.2 application_name

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'application name')
descriptive_representation_item
descriptive_representation_item.description

5.1.4.1.3 vendor_or_user_name

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'vendor or user name')
descriptive_representation_item
descriptive_representation_item.description

5.1.4.2 Dm_execution_input

AIM element: dm_execution_input
Source: ISO 10303-219
Reference path: dm_execution_input <=
executed_action<=
action

5.1.4.2.1 dm_execution_input to dm_program_run (as program_run)

AIM element: PATH
Source: ISO 10303-219
Reference path: dm_execution_input <=
executed_action<=
action
action.chosen_method
action_method=>
dm_program_run

5.1.4.2.2 dm_execution_input to dm_execution_result (as result)

AIM element: PATH
Source: ISO 10303-219
Reference path: dm_execution_input <=
executed_action<=
action<-
action_relationship.relating_action
action_relationship
action_relationship.related_action->
action=>
executed_action=>
dm_execution_result

5.1.4.2.3 dm_execution_input to part (as as_designed_part)

AIM element: PATH
Source: ISO 10303-219
Reference path: dm_execution_input <=
executed_action<=
action<-
action_assignment.assigned_action
action_assignment=>
applied_action_assignment
applied_action_assignment.items[i]->
action_item
action_item=product_definition
product_definition

5.1.4.3 Dm_execution_result

AIM element: dm_execution_result
Source: ISO 10303-219
Reference path: dm_execution_result <=
executed_action<=
action

5.1.4.3.1 dm_execution_result to dm_execution_result_measurement (as dm_measurements)

AIM element: PATH
Source: ISO 10303-219
Reference path: dm_execution_result <=
executed_action<=
action
characterized_action_definition=action
characterized_action_definition<-
action_property.definition
action_property=>
dm_execution_result_measurement

5.1.4.3.2 dm_execution_result to dm_program_run (as program_run)

AIM element: dm_program_run
Source: ISO 10303-219
Reference path: dm_execution_result <=
executed_action<=
action
action.chosen_method
action_method=>
dm_program_run

5.1.4.4 Dm_execution_result_measurement

AIM element: dm_execution_result_measurement
Source: ISO 10303-219
Reference path: dm_execution_result_measurement <=
action_property

5.1.4.4.1 compensation

AIM element: action_property.name
Source: ISO 10303-41
Reference path: dm_execution_result_measurement <=
action_property
action_property.name
(action_property.name='TRUE')
(action_property.name='FALSE')

5.1.4.4.2 dm_execution_result_measurement to dm_result_parameter (as parameter)

AIM element: PATH
Source: ISO 10303-219
Reference path: dm_execution_result_measurement <=
action_property<-
action_property_representation.property
action_property_representation
action_property_representation.representation->
representation=>
dm_result_parameter

5.1.4.4.3 dm_execution_result_measurement to dm_data_analysis_software (as software)

AIM element: PATH
Reference path: dm_execution_result_measurement <=
action_property<-
action_property_representation.property
action_property_representation
(action_property_representation.name='dm data acquisition software')
action_property_representation.representation->
representation

5.1.4.4.4 dm_execution_result_measurement to dm_point (as measurement_-points)

AIM element: PATH
Reference path: dm_execution_result_measurement <=
action_property<-
action_property_representation.property
action_property_representation
(action_property_representation.name='measurement points')
action_property_representation.representation->
representation=>
dm_point

5.1.5 dimensional_measurement_features UoF

5.1.5.1 Dm_feature

AIM element: dm_instanced_feature
Source: ISO 10303-219
Reference path: dm_instanced_feature <=
[shape_aspect]
[dm_feature_definition <=
characterized_object]

5.1.5.1.1 dm_feature to cartesian_point (as boundary)

AIM element: PATH
Reference path: dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
representation
(representation.name = 'boundary')
representation.items[i] ->
representation_item=>
geometric_representation_item=>
point=>
cartesian_point

5.1.5.2 Dmf_arc

AIM element: dmf_arc
Source: ISO 10303-219
Reference path: dmf_arc <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.2.1 inner_outer

AIM element: PATH
Reference path: dmf_arc <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.2.2 dmf_arc to dm_vector_parameter (as axis_direction_vector)

AIM element: PATH
Reference path: dmf_arc <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='axis direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.2.3 dmf_arc to dm_vector_parameter (as start_vector)

AIM element: PATH
Reference path: dmf_arc <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='start vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.2.4 dmf_arc to dm_vector_parameter (as end_vector)

AIM element: PATH
Reference path: dmf_arc <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='end vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.2.5 dmf_arc to dm_dimension_parameter (as radius)

AIM element: PATH
Reference path: dmf_arc <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='radius'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.2.6 dmf_arc to dm_point_parameter (as center_point)

AIM element: PATH
Reference path: dmf_arc <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.3 Dmf_circle

AIM element: dmf_circle
Source: ISO 10303-219
Reference path: dmf_circle <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.3.1 inner_outer

AIM element: PATH
Reference path: dmf_circle <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.3.2 dmf_circle to dm_point_parameter (as center_point)

AIM element: PATH
Reference path: dmf_circle <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.3.3 dmf_circle to dm_dimension_parameter (as diameter)

AIM element: PATH
Reference path: dmf_circle <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='diameter'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.3.4 dmf_circle to dm_vector_parameter (as axis_direction_vector)

AIM element: PATH
Reference path: dmf_circle <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='axis direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.4 Dmf_cone

AIM element: dmf_cone
Source: ISO 10303-219
Reference path: dmf_cone <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.4.1 inner_outer

AIM element: PATH
Reference path: dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.4.2 dmf_cone to dm_vector_parameter (as axis_direction_vector)

AIM element: PATH

Reference path:

dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='axis direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.4.3 dmf_cone to dm_vector_parameter (as start_vector)

AIM element: PATH
Reference path: dmf_ cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='start vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.4.4 dmf_cone to dm_vector_parameter (as end_vector)

AIM element: PATH
Reference path: dmf_ cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='end vector'}
property_definition_representation.used_representation ->
representation =>
(representation.name='end vector')
dm_result_parameter=>
dm_vector_parameter

5.1.5.4.5 dmf_cone to dm_point_parameter (as apex_point)

AIM element: PATH
Reference path: dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='apex point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.4.6 dmf_cone to dm_point_parameter (as start_point)

AIM element: PATH
Reference path: dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='start point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.4.7 dmf_cone to dm_point_parameter (as end_point)

AIM element: PATH
Reference path: dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='end point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.4.8 dmf_cone to dm_dimension_parameter (as included_angle)

AIM element: PATH

Reference path: dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='included angle'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.4.9 dmf_cone to dm_dimension_parameter (as diameter)

AIM element: PATH

Reference path:

```
dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='diameter'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter
```

5.1.5.4.10 dmf_cone to dm_dimension_parameter (as end_length_dml)

AIM element: PATH
Reference path: dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='end length dml'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.4.11 dmf_cone to dm_dimension_parameter (as start_length_dml)

AIM element: PATH
Reference path: dmf_cone <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='start point dml'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.5 Dmf_cylinder

AIM element: dmf_cylinder
Source: ISO 10303-219
Reference path: dmf_cylinder <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.5.1 inner_outer

AIM element: PATH
Reference path: dmf_cylinder <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.5.2 dmf_cylinder to dm_vector_parameter (as axis_direction_vector)

AIM element: PATH

Reference path:
dmf_ cylinder <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='axis direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.5.3 dmf_cylinder to dm_point_parameter (as center_point)

AIM element: PATH

Reference path:

dmf_ cylinder <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.5.4 dmf_cylinder to dm_point_parameter (as end_point)

AIM element: PATH

Reference path:
dmf_ cylinder <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='end point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.5.5 dmf_cylinder to dm_point_parameter (as start_point)

AIM element: PATH
Reference path: dmf_ cylinder <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='start point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.5.6 dmf_cylinder to dm_dimension_parameter (as diameter)

AIM element: PATH

Reference path:

```
dmf_ cylinder <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='diameter'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter
```


5.1.5.5.7 dmf_cylinder to dm_dimension_parameter (as length_dml)

AIM element: PATH
Reference path: dmf_ cylinder <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='length dml'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.6 Dmf_edge_point

AIM element: dmf_edge_point
Source: ISO 10303-219
Reference path: dmf_edge_point <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.6.1 point_type_dml

AIM element: PATH
Reference path: dmf_edge_point<=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'point type dml')
descriptive_representation_item
(descriptive_representation_item.description='h edge')
(descriptive_representation_item.description='t edge')

5.1.5.6.2 dmf_edge_point to dm_vector_parameter (as surface_normal_-vector)

AIM element: PATH

Reference path:

dmf_edge_point <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='surface normal vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.6.3 dmf_edge_point to dm_vector_parameter (as edge_normal_vector)

AIM element: PATH
Reference path: dmf_edge_point <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='edge normal vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.6.4 dmf_edge_point to dm_point_parameter (as location_point)

AIM element: PATH
Reference path: dmf_edge_point <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='location point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.7 Dmf_ellipse

AIM element: dmf_ellipse
Source: ISO 10303-219
Reference path: dmf_ellipse <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.7.1 inner_outer

AIM element: PATH
Reference path: dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.7.2 dmf_ellipse to dm_vector_parameter (as normal_direction)

AIM element: PATH
Reference path: dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='normal direction'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.7.3 dmf_ellipse to dm_vector_parameter (as major_axis_vector)

AIM element: PATH
Reference path: dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='major axis vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.7.4 dmf_ellipse to dm_point_parameter (as center_point)

AIM element: PATH
Reference path: dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.7.5 dmf_ellipse to dm_point_parameter (as focus_point_one_dml)

AIM element: PATH
Reference path: dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='focus point one dml'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.7.6 dmf_ellipse to dm_point_parameter (as focus_point_two_dml)

AIM element: PATH

Reference path:
dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='focus point two dml'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.7.7 dmf_ellipse to dm_dimension_parameter (as minor_diameter)

AIM element: PATH

Reference path:
dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='minor diametor'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.7.8 dmf_ellipse to dm_dimension_parameter (as major_diameter)

AIM element: PATH
Reference path: dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='major diameter'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.8 Dmf_generic_feature

AIM element: dmf_generic_feature
Source: ISO 10303-219
Reference path: dmf_generic_feature <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.8.1 description

AIM element: characterized_object.description
Source: ISO 10303-41
Reference path: dmf_ ellipse <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'description')
descriptive_representation_item

5.1.5.8.2 dmf_generic_feature to dm_point_parameter (as center_point)

AIM element: PATH
Reference path: dmf_generic_feature <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.8.3 dmf_generic_feature to dm_vector_parameter (as vector_from_-center_of_object)

AIM element: PATH

Reference path:
dmf_generic_feature <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='vector from center of object'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.8.4 dmf_generic_feature to dm_vector_parameter (as secondary-_vector)

AIM element: PATH
Reference path: dmf_generic_feature <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='secondary vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.8.5 dmf_generic_feature to dm_dimension_parameter (as parameters)

AIM element: PATH
Reference path: dmf_generic_feature <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='parameters'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.9 Dmf_geometric_curve

AIM element: dmf_geometric_curve
Source: ISO 10303-219
Reference path: dmf_geometric_curve <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.9.1 dmf_geometric_curve to dm_point_parameter (as point_on_plane_-of_curve)

AIM element: PATH
Reference path: dmf_geometric_curve <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point on plane of curve'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.9.2 dmf_geometric_curve to dm_vector_parameter (as curve_plane_-direction)

AIM element: PATH
Reference path: dmf_geometric_curve <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='curve plane direction'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.9.3 dmf_geometric_curve to cartesian_point (as data_points)

AIM element: PATH
Reference path: dmf_geometric_curve <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='data points'}
property_definition_representation.used_representation ->
representation]
representation.items[i] ->
representation_item=>
geometric_representation_item=>
point=>
cartesian_point

5.1.5.10 Dmf_geometric_surface

AIM element: dmf_geometric_surface
Source: ISO 10303-219
Reference path: dmf_geometric_surface<=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.10.1 dmf_geometric_surface to dm_point_parameter (as point_on_-surface)

AIM element: PATH
Reference path: dmf_geometric_surface <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point on surface'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.10.2 dmf_geometric_surface to dm_vector_parameter (as local_-surface_normal)

AIM element: PATH
Reference path: dmf_geometric_surface <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='local surface normal'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.10.3 dmf_geometric_surface to cartesian_point (as data_points)

AIM element: PATH
Reference path: dmf_geometric_surface <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='data points'}
property_definition_representation.used_representation ->
representation
representation.items[i] ->
representation_item=>
geometric_representation_item=>
point=>
cartesian_point

5.1.5.11 Dmf_line_bounded

AIM element: dmf_line_bounded
Source: ISO 10303-219
Reference path: dmf_line_bounded <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.11.1 dmf_line_bounded to dm_point_parameter (as first_end_point)

AIM element: PATH
Reference path: dmf_line_bounded <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='first end point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.11.2 dmf_line_bounded to dm_point_parameter (as second_end-_point)

AIM element: PATH
Reference path: dmf_line_bounded <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='second end point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.11.3 dmf_line_bounded to dm_vector_parameter (as vector_dml)

AIM element: PATH

Reference path:

dmf_line_bounded <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='vector dml'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.11.4 dmf_line_bounded to dm_vector_parameter (as surface_approach-_vector)

AIM element: PATH
Reference path: dmf_line_bounded <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='surface approach vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.11.5 dmf_line_bounded to dm_dimension_parameter (as length_dml)

AIM element: PATH
Reference path: dmf_line_bounded <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='length dml'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.12 Dmf_line_closed_parallel

AIM element: dmf_line_closed_parallel
Source: ISO 10303-219
Reference path: dmf_line_closed_parallel <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.12.1 inner_outer

AIM element: PATH
Reference path: dmf_line_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.12.2 end_kind

AIM element: PATH

Reference path: dmf_line_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'end kind')
descriptive_representation_item
(descriptive_representation_item.description='round')
(descriptive_representation_item.description='square')

5.1.5.12.3 dmf_line_closed_parallel to dm_point_parameter (as center_point)

AIM element: PATH
Reference path: dmf_line_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.12.4 dmf_line_closed_parallel to dm_vector_parameter (as longitude_-vector)

AIM element: PATH

Reference path: dmf_line_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='longitude vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.12.5 dmf_line_closed_parallel to dm_vector_parameter (as axis-_direction_vector)

AIM element: PATH
Reference path: dmf_line_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='axis direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.12.6 dmf_line_closed_parallel to dm_dimension_parameter (as feature_-length)

AIM element: PATH
Reference path: dmf_line_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='feature length'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.12.7 dmf_line_closed_parallel to dm_dimension_parameter (as width)

AIM element: PATH
Reference path: dmf_line_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='width'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.13 Dmf_line_unbounded

AIM element: dmf_line_unbounded
Source: ISO 10303-219
Reference path: dmf_line_unbounded <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.13.1 dmf_line_unbounded to dm_point_parameter (as point_on_line)

AIM element: PATH
Reference path: dmf_line_unbounded <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point on line'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.13.2 dmf_line_unbounded to dm_vector_parameter (as direction-_vector)

AIM element: PATH
Reference path: dmf_line_unbounded <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.13.3 dmf_line_unbounded to dm_vector_parameter (as surface_-approach_vector)

AIM element: PATH
Reference path: dmf_line_unbounded <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='surface approach vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.14 Dmf_pattern

AIM element: dmf_pattern
Source: ISO 10303-219
Reference path: dmf_pattern <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.14.1 dmf_pattern to dm_point_parameter (as point)

AIM element: PATH
Reference path: dmf_pattern <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.14.2 dmf_pattern to dm_vector_parameter (as direction_vector)

AIM element: PATH
Reference path: dmf_pattern <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.14.3 dmf_pattern to dm_feature (as features)

AIM element: PATH
Reference path: dmf_pattern <=
dm_feature_definition<=
instaqnced_feature<=
shape_aspect<-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
{dm_feature_relationship}
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
dm_instanced_feature<=
dm_feature_definition<=

5.1.5.15 Dmf_plane

AIM element: dmf_plane
Source: ISO 10303-219
Reference path: dmf_plane <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.15.1 dmf_plane to dm_vector_parameter (as direction_vector)

AIM element: PATH
Reference path: dmf_plane <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.15.2 dmf_plane to dm_point_parameter (as point_on_plane)

AIM element: PATH
Reference path: dmf_plane <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point on plane'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.16 Dmf_plane_closed_parallel

AIM element: dmf_plane_closed_parallel
Source: ISO 10303-219
Reference path: dmf_plane_closed_parallel <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.16.1 end_kind

AIM element: PATH
Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'end kind')
descriptive_representation_item
(descriptive_representation_item.description='round')
(descriptive_representation_item.description='square')

5.1.5.16.2 inner_outer

AIM element: PATH

Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.16.3 dmf_plane_closed_parallel to dm_point_parameter (as center-_point)

AIM element: PATH
Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.16.4 dmf_plane_closed_parallel to dm_vector_parameter (as axis_-direction_vector)

AIM element: PATH
Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='axis direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.16.5 dmf_plane_closed_parallel to dm_vector_parameter (as normal_dml)

AIM element: PATH
Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='normal dml'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.16.6 dmf_plane_closed_parallel to dm_vector_parameter (as longitude_-vector)

AIM element: PATH
Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='longitude vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.16.7 dmf_plane_closed_parallel to dm_dimension_parameter (as width)

AIM element: PATH
Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='width'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.16.8 dmf_plane_closed_parallel to dm_dimension_parameter (as height)

AIM element: PATH
Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='height'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.16.9 dmf_plane_closed_parallel to dm_dimension_parameter (as feature_length)

AIM element: PATH
Reference path: dmf_plane_closed_parallel <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='feature length'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.17 Dmf_plane_symmetric

AIM element: dmf_plane_symmetric
Source: ISO 10303-219
Reference path: dmf_plane_symmetric <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.17.1 inner_outer

AIM element: PATH
Reference path: dmf_plane_symmetric <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.17.2 dmf_plane_symmetric to dm_point_parameter (as point_on_side_-one)

AIM element: PATH
Reference path: dmf_plane_symmetric <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point on one side'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.17.3 dmf_plane_symmetric to dm_point_parameter (as point_on_side_-two)

AIM element: PATH
Reference path: dmf_plane_symmetric <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point on side two'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.17.4 dmf_plane_symmetric to dm_point_parameter (as point_on_mid_-plane)

AIM element: PATH
Reference path: dmf_plane_symmetric <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point on mid plane'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.17.5 dmf_plane_symmetric to dm_vector_parameter (as direction_-vector_side_one)

AIM element: PATH
Reference path: dmf_plane_symmetric <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='direction vector side one'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.17.6 dmf_plane_symmetric to dm_vector_parameter (as direction_-vector_side_two)

AIM element: PATH
Reference path: dmf_plane_symmetric <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='direction vector side two'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.17.7 dmf_plane_symmetric to dm_dimension_parameter (as width_at_-mid_point)

AIM element: PATH
Reference path: dmf_plane_symmetric <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='width at mid point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.18 Dmf_point

AIM element: dmf_point
Source: ISO 10303-219
Reference path: dmf_point <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.18.1 dmf_point to dm_point_parameter (as point)

AIM element: PATH
Reference path: dmf_point <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.18.2 dmf_point to dm_vector_parameter (as direction_vector)

AIM element: PATH
Reference path: dmf_point <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='direction vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.19 Dmf_sphere

AIM element: dmf_sphere
Source: ISO 10303-219
Reference path: dmf_sphere <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.19.1 inner_outer

AIM element: PATH
Reference path: dmf_plane_symmetric <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.19.2 dmf_sphere to dm_point_parameter (as center_point)

AIM element: PATH

Reference path: dmf_sphere <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.19.3 dmf_sphere to dm_vector_parameter (as prime_meridian_vector)

AIM element: PATH
Reference path: dmf_sphere <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='prime meridian vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.19.4 dmf_sphere to dm_vector_parameter (as north_pole_vector)

AIM element: PATH
Reference path: dmf_sphere <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='north pole vector'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.19.5 dmf_sphere to dm_dimension_parameter (as latitude_start_angle)

AIM element: PATH
Reference path: dmf_sphere <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='latitude start angle'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.19.6 dmf_sphere to dm_dimension_parameter (as latitude_stop_angle)

AIM element: PATH
Reference path: dmf_sphere <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='latitude stop angle'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.19.7 dmf_sphere to dm_dimension_parameter (as longitude_start_-angle)

AIM element: PATH
Reference path: dmf_sphere <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='longitude start angle'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.19.8 dmf_sphere to dm_dimension_parameter (as longitude_stop_angle)

AIM element: PATH
Reference path: dmf_sphere <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='longitude stop angle'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.19.9 dmf_sphere to dm_dimension_parameter (as diameter)

AIM element: PATH
Reference path: dmf_sphere <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='diameter'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.20 Dmf_surface_of_revolution_dml

AIM element: dmf_surface_of_revolution_dml
Source: ISO 10303-219
Reference path: dmf_surface_of_revolution_dml <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.20.1 dmf_surface_of_revolution to dm_vector_parameter (as axis_of_-revolution)

AIM element: PATH
Reference path: dmf_surface_of_revolution_dml <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='axis of revolution'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.20.2 dmf_surface_of_revolution to dm_point_parameter (as profile_-curve)

AIM element: PATH
Reference path: dmf_surface_of_revolution_dml <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='profile curve'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.21 Dmf_torus

AIM element: dmf_torus
Source: ISO 10303-219
Reference path: dmf_torus <=
{dm_feature_definition =>
dm_instanced_feature}
dm_feature_definition <=
characterized_object

5.1.5.21.1 inner_outer

AIM element: PATH
Reference path: dmf_torus <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'inner or outer')
descriptive_representation_item
(descriptive_representation_item.description='inner')
(descriptive_representation_item.description='outer')

5.1.5.21.2 dmf_torus to dm_point_parameter (as center_point)

AIM element: PATH
Reference path: dmf_torus <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='center point'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_point_parameter

5.1.5.21.3 dmf_torus to dm_vector_parameter (as vector_of_plane)

AIM element: PATH

Reference path:
dmf_torus <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='vector of plane'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_vector_parameter

5.1.5.21.4 dmf_torus to dm_dimension_parameter (as major_diameter)

AIM element: PATH
Reference path: dmf_torus <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='major diameter'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.21.5 dmf_torus to dm_dimension_parameter (as minor_diameter)

AIM element: PATH
Reference path: dmf_torus <=
dm_feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
{property_definition_representation<-
name_attribute_select=property_definition_representation
name_attribute_select<-
name_attribute.named_item
name_attribute
name_attribute.attribute_value='minor diameter'}
property_definition_representation.used_representation ->
representation =>
dm_result_parameter=>
dm_dimension_parameter

5.1.5.22 Feature

AIM element: instanced_feature
Source: ISO 10303-219
Reference path: instanced_feature <=
[shape_aspect]
[feature_definition <=
characterized_object]

5.1.5.22.1 label

AIM element: characterized_object.description
Source: ISO 10303-41
Reference Path: instanced_feature <=
characterized_object
characterized_object.description

5.1.5.22.2 feature to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.5.23 Inspection_feature_relationship

AIM element: dm_feature_relationship
Source: ISO 10303-219
Reference path: dm_feature_relationship <=
shape_aspect_relationship

5.1.5.23.1 inspection_feature_relationship to dm_feature (as inspection_-feature)

AIM element: PATH
Reference path: dm_feature_relationship<=
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
dm_instanced_feature<=
dm_feature_definition

5.1.5.23.2 inspection_feature_relationship to manufacturing_feature (as designed_feature)

AIM element: PATH
Reference path: shape_aspect_relationship=>
dm_feature_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
instanced_feature<=
feature_definition

5.1.6 dimensional_measurement_part UoF

5.1.6.1 Part

AIM element: product_definition_formation
Source: ISO 10303-41
Rules: product_definition_formation_requires_security_classification – (See 5.2.4.16)
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'}
application_context_element.frame_of_reference->
application_context<-
application_protocol_definition
application_protocol_definition.application_interpreted_model_schema_name=
'dimensional inspection schema'

5.1.6.1.1 part_description

AIM element: product.description
Source: ISO 10303-41
Rules: product_requires_version – (See 5.2.4.15)
Reference path: product_definition_formation
product_definition_formation.of_product ->
product
product.description

5.1.6.1.2 part_id

AIM element: product.id
Source: ISO 10303-41
Rules: product_requires_version – (See 5.2.4.15)
Reference path: product_definition_formation
product_definition_formation.of_product ->
product
product.id

5.1.6.1.3 part_name

AIM element: product.name
Source: ISO 10303-41
Rules: product_requires_version – (See 5.2.4.15)
Reference path: product_definition_formation
product_definition_formation.of_product ->
product
product.name

5.1.6.1.4 part_revision_id

AIM element: product_definition_formation.id
Source: ISO 10303-41

5.1.6.1.5 security_classification

AIM element: security_classification
Source: ISO 10303-41
Rules: dependent_instantiable_security_classification_level – (See 5.2.4.10)
product_definition_formation_requires_security_classification(See 5.2.4.16)
restrict_security_classification_level – (See 5.2.4.18)
Reference path: product_definition_formation
security_classification_item = product_definition_formation
security_classification_item <-
applied_security_classification_assignment.items[i]
applied_security_classification_assignment <=
security_classification_assignment
security_classification_assignment.assigned_security_classification ->
security_classification

5.1.6.1.6 serial_number_dml

AIM element: serial_numbered_effectivity.effectivity_start_id
Source: ISO 10303-41
Reference path: product_definition_formation
effectivity_item= product_definition_formation
effectivity_item<-
applied_effectivity_assignment.items[i]
applied_effectivity_assignment<=
effectivity_assignment
effectivity_assignment.assigned_effectivity->
effectivity=>
serial_numbered_effectivity
serial_numbered_effectivity.effectivity_start_id

5.1.6.1.7 lot_number_dml

AIM element: lot_effectivity.effectivity_lot_id
Source: ISO 10303-41
Rule:
Reference path: product_definition_formation
effectivity_item= product_definition_formation
effectivity_item<-
applied_effectivity_assignment.items[i]
applied_effectivity_assignment<=
effectivity_assignment
effectivity_assignment,assigned_effectivity->
effectivity=>
lot_effectivity
lot_effectivity.effectivity_lot_id

5.1.6.1.8 vendor_dml

AIM element: product_definition_formation.description
Source: ISO 10303-41

5.1.6.1.9 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

5.1.7 dimensional_measurement_parameters UoF

5.1.7.1 Dm_dimension_parameter

AIM element: dm_dimension_parameter
Source: ISO 10303-219
Reference path:
dm_dimension_parameter <=
dm_result_parameter<=
representation

5.1.7.1.1 dm_dimension_parameter to numeric_parameter (as calculated_-value)

AIM element: PATH
Reference path:
dm_dimension_parameter<=
dm_result_parameter<=
representation
representation.items[i]->
representation_item=>
(represerntation_item.name='calculated value')
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.7.1.2 dm_dimension_parameter to measurement_uncertainty (as value_-uncertainty)

AIM element: PATH
Reference path: dm_dimension_parameter<=
dm_result_parameter<=
representation
representation.items[i]->
representation_item=>
(represerntation_item.name='value uncertainty')
{measure_representation_item
measure_representation_item<=
measure_with_unit =>
length_measure_with_unit}
qualified_representation_item
qualified_representation_item.qualifiers[i]->
value_qualifier
value_qualifier=uncertainty_qualifier
uncertainty_qualifier=>
(standard_uncertainty)
(expanded_uncertainty)

5.1.7.2 Dm_parameter_value_limits

AIM element: dm_parameter_value_limits
Source: ISO 10303-219
Reference path: dm_parameter_value_limits<=
representation

5.1.7.2.1 limits_method

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: dm_parameter_value_limits<=
representation
representation.items[i] ->
representation_item=>
(representation_item.name = 'limits method')
descriptive_representation_item
descriptive_representation_item.description

5.1.7.2.2 dm_parameter_value_limits to numeric_parameter (as calculated_-limits)

AIM element: PATH
Reference path: dm_parameter_value_limits<=
representation
representation.items[i]->
representation_item=>
(represerntation_item.name='calculated limits')
measure_representation_item=>
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.7.3 Dm_point

AIM element: dm_point
Source: ISO 10303-219
Reference path: dm_point<=
representation

5.1.7.3.1 dm_point to cartesian_point (as measured_point)

AIM element: PATH
Reference path: dm_point <=
representation
representation.items[i] ->
representation_item=>
(representation_item.name='measured point')
geometric_representation_item=>
point=>
cartesian_point

5.1.7.3.2 dm_point to cartesian_point (as expected_point)

AIM element: PATH
Reference path: dm_point <=
representation
representation.items[i] ->
representation_item=>
(representation_item.name='expected point')
geometric_representation_item=>
point=>
cartesian_point

5.1.7.4 Dm_point_parameter

AIM element: dm_point_parameter
Source: ISO 10303-219
Reference path: dm_point_parameter <=
dm_result_parameter<=
representation

5.1.7.4.1 dm_point_parameter to cartesian_point (as calculated_point)

AIM element: PATH
Reference path: dm_point_parameter <=
dm_result_parameter<=
representation
representation.items[i] ->
representation_item=>
geometric_representation_item=>
point=>
cartesian_point

5.1.7.4.2 dm_point_parameter to measurement_uncertainty (as value_-uncertainty)

AIM element: PATH
Reference path: dm_point_parameter<=
dm_result_parameter<=
representation
representation.items[i]->
{representation_item=>
geometric_representation_item=>
point=>
cartesian_point}
representation_item=>
measure_representation_item
measure_representation_item<=
measure_with_unit<-
measure_qualification.qualified_measure
measure_qualification
measure_qualification.qualifiers[i]->
value_qualifier
value_qualifier=uncertainty_qualifier
uncertainty_qualifier
{uncertainty_qualifier.measure_name='value uncertainty'}
(uncertainty_qualifier.description='x uncertainty')
(uncertainty_qualifier.description='y uncertainty')
(uncertainty_qualifier.description='z uncertainty')
(uncertainty_qualifier=>
standard_uncertainty)
(uncertainty_qualifier=>
standard_uncertainty=>
expanded_uncertainty)

5.1.7.5 Dm_result_parameter

AIM element: dm_result_parameter
Source: ISO 10303-219
Reference path: dm_result_parameter<=
representation

5.1.7.5.1 calculation_method

AIM element: descriptive_representation_item
Source: ISO 10303-45
Reference path: dm_result_parameter<=
representation
representation.items[i]->
representation_item=>
representation_item.name='calculation method'
descriptive_representation_item
descriptive_representation_item.description

5.1.7.5.2 dm_result_parameter to dm_parameter_analysis_dml (as analysis_-conditions)

AIM element: PATH
path: dm_result_parameter <=
representation<-
action_property_representation.representation
action_property_representation
action_property_representation.property->
action_property=>
dm_parameter_analysis_dml

5.1.7.5.3 dm_result_parameter to dm_parameter_value_limits (as limits)

AIM element: PATH
path: dm_result_parameter <=
representation<-
representation_relationship.rep_1
representation_relationship
representation_relationship.rep_2->
representation=>
dm_parameter_value_limits

5.1.7.5.4 dm_result_parameter to dimensional_tolerance (as specification)

AIM element: PATH
path: dm_result_parameter <=
representation<-
representation_relationship.rep_1-
representation_relationship
representation_relationship.rep_2->
representation
shape_representation
shape_dimension_representation

5.1.7.5.5 dm_result_parameter to geometric_tolerance (as specification)

AIM element: PATH
path: dm_result_parameter <=
representation<-
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition
represented_definition=property_definition
property_definition
property_definition.definition->
characterized_definition
characterized_definition = shape_definition
shape_definition
shape_definition = shape_aspect
shape_aspect<-
geometric_tolerance.toleranced_shape_aspect
geometric_tolerance

5.1.7.5.6 dm_result_parameter to manufacturing_feature (as specification)

AIM element: PATH
path: dm_result_parameter <=
representation<-
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition
represented_definition=property_definition
property_definition
property_definition.definition->
characterized_definition
characterized_definition = shape_definition
shape_definition
shape_definition = shape_aspect
shape_aspect=>
(instanced_feature)
(transition_feature)

5.1.7.5.7 dm_result_parameter to shape_element (as specification)

AIM element: PATH
path: dm_result_parameter <=
representation<-
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition
represented_definition=property_definition
property_definition
property_definition.definition->
characterized_definition
characterized_definition = shape_definition
shape_definition
shape_definition = shape_aspect
shape_aspect

5.1.7.6 Dm_vector_parameter

AIM element: dm_vector_parameter
Source: ISO 10303-219
Reference path: dm_vector_parameter<=
dm_result_parameter<=
representation

5.1.7.6.1 dm_vector_parameter to cartesian_vector (as calculated_vector)

AIM element: PATH
Reference path: dm_vector_parameter<=
dm_result_parameter<=
representation
representation.items[i] ->
representation_item=>
geometric_representation_item=>
vector

5.1.7.6.2 dm_vector_parameter to measurement_uncertainty (as value_-uncertainty)

AIM element: PATH
Reference path: dm_vector_parameter<=
dm_result_parameter<=
representation
representation.items[i]->
{representation_item=>
geometric_representation_item=>
vector}
representation_item=>
measure_representation_item
measure_representation_item<=
measure_with_unit<-
measure_qualification.qualified_measure
measure_qualification
measure_qualification.qualifiers[i]->
value_qualifier
value_qualifier=uncertainty_qualifier
uncertainty_qualifier
{uncertainty_qualifier.measure_name='value uncertainty'}
(uncertainty_qualifier.description='i uncertainty')
(uncertainty_qualifier.description='j uncertainty')
(uncertainty_qualifier.description='k uncertainty')
(uncertainty_qualifier=>
standard_uncertainty)
(uncertainty_qualifier=>
standard_uncertainty=>
expanded_uncertainty)

5.1.7.7 Measurement_uncertainty

AIM element: uncertainty_qualifier
Source: ISO 10303-45
Reference Path: uncertainty_qualifier=>
standard_uncertainty

5.1.7.7.1 coverage_factor

AIM element: expanded_uncertainty.coverage_factor
Source: ISO 10303-45
Reference Path: uncertainty_qualifier=>
standard_uncertainty=>
expanded_uncertainty
expanded_uncertainty.coverage_factor

5.1.7.7.2 uncertainty_name

AIM element: uncertainty_qualifier.name
Source: ISO 10303-45
Reference Path: uncertainty_qualifier
uncertainty_qualifier.name

5.1.7.7.3 uncertainty_value

AIM element: standard_uncertainty.uncertainty_value
Source: ISO 10303-45
Reference Path: uncertainty_qualifier=>
standard_uncertainty
standard_uncertainty.uncertainty_value

5.1.8 feature_definition_items UoF

5.1.8.1 Angle_taper

AIM element: taper
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness (See 5.2.4.20)
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}

5.1.8.1.1 angle_taper to numeric_parameter (as angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: taper <=
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 =>
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}

5.1.8.2 Blind_bottom_condition

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness (See 5.2.4.20)
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}

5.1.8.2.1 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'

5.1.8.3 Boss_top_condition

AIM element: boss_top
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness (See 5.2.4.20)
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}

5.1.8.3.1 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')}

5.1.8.3.2 boss_top to numeric_parameter (as top_radius)

AIM element: PATH
Source: ISO 10303-41
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
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}

5.1.8.4 Chamfer_angle

AIM element: chamfer_offset
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)5.2.4.20
Reference Path: chamfer_offset <=
shape_aspect
{shape_aspect
shape_aspect.description = 'second offset'}

5.1.8.4.1 chamfer_angle to numeric_parameter (as angle_amount)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: chamfer_offset <=
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 =>
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}

5.1.8.5 Circular_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)5.2.4.20
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
(shape_aspect.description = 'partial circular')
(shape_aspect.description = 'complete circular')}

5.1.8.5.1 circular_path to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: path_feature_component <=
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 =>
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}

5.1.8.6 Complete_circular_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)5.2.4.20
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complete circular'}

5.1.8.7 Conical_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)5.2.4.20
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'conical'}

5.1.8.7.1 conical_hole_bottom to numeric_parameter (as tip_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: hole_bottom <=
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 =>
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}

5.1.8.7.2 conical_hole_bottom to numeric_parameter (as tip_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: hole_bottom <=
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 =>
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}

5.1.8.8 Diameter_taper

AIM element: taper
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.8.1 diameter_taper to numeric_parameter (as final_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: taper <=
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 =>
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}

5.1.8.9 Directed_taper

AIM element: taper
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.9.1 directed_taper to direction_element (as direction)

AIM element: PATH

Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)

Reference Path: taper <=
shape_aspect
shape_definition = shape_aspect
shape_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.name = 'direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.8.9.2 directed_taper to numeric_parameter (as angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: taper <=
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 =>
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}

5.1.8.10 First_offset

AIM element: chamfer_offset
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: chamfer_offset <=
shape_aspect
{shape_aspect
shape_aspect.description = 'first offset'}

5.1.8.10.1 first_offset to face_shape_element (as face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: chamfer_offset <=
shape_aspect
shape_definition = shape_aspect
shape_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.name = 'first face shape'}
representation =>
shape_representation=>
face_shape_representation

5.1.8.10.2 first_offset to numeric_parameter (as offset_amount)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: chamfer_offset <=
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 =>
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}

5.1.8.11 Flat_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'flat'}

5.1.8.12 Flat_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'flat'}

5.1.8.12.1 flat_slot_end_type to numeric_parameter (as first_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: slot_end <=
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 =>
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}

5.1.8.12.2 flat_slot_end_type to numeric_parameter (as second_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: slot_end <=
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 =>
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}

5.1.8.13 Flat_with_radius_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'flat with radius'}

5.1.8.13.1 flat_with_radius_hole_bottom to numeric_parameter (as corner_-radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: hole_bottom <=
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 =>
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}

5.1.8.14 Flat_with_taper_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'flat with taper'}

5.1.8.14.1 flat_with_taper_hole_bottom to numeric_parameter (as final_-diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: hole_bottom <=
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 =>
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}

5.1.8.14.2 flat_with_taper_hole_bottom to numeric_parameter (as taper angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: hole_bottom <=
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 =>
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}

5.1.8.15 General_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.8.15.1 general_path to path_element (as sweep_path)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: path_feature_component <=
shape_aspect
shape_definition = shape_aspect
shape_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.name = 'sweep path'}
representation =>
shape_representation=>
path_shape_representation

5.1.8.16 General_pocket_bottom_condition

AIM element: pocket_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: pocket_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.8.16.1 general_pocket_bottom_condition to face_shape_element (as floor)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: pocket_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_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.name = 'floor face'}
representation =>
shape_representation =>
shape_representation_with_parameters}
representation

5.1.8.16.2 general_pocket_bottom_condition to numeric_parameter (as floor_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: pocket_bottom <=
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 =>
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}

5.1.8.17 General_profile_floor

AIM element: profile_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: profile_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.8.17.1 general_profile_floor to face_shape_element (as floor)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: profile_floor <=
shape_aspect
shape_definition = shape_aspect
shape_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.name = 'floor'}
representation =>
shape_representation =>
shape_representation_with_parameters}
representation

5.1.8.18 General_rib_top_floor

AIM element: rib_top_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: rib_top_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.8.18.1 general_rib_top_floor to ordered_face_element (as rib_top_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: rib_top_floor<=
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 =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'rib top face'}
representation =>
shape_representation =>
face_shape_representation

5.1.8.19 General_top_condition

AIM element: boss_top
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: boss_top <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.8.19.1 general_top_condition to face_shape_element (as top_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: boss_top <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation =>
shape_representation_with_parameters}
representation

5.1.8.20 Linear_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.description = 'linear'}

5.1.8.20.1 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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation =>
shape_representation_with_parameters}
representation

5.1.8.20.2 linear_path to numeric_parameter (as distance)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: path_feature_component <=
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 =>
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}

5.1.8.21 Loop_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'loop'}

5.1.8.22 Open_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'open'}

5.1.8.23 Partial_area_definition

AIM element: applied_area
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: applied_area <=
shape_aspect
{shape_aspect
shape_aspect.of_shape->
product_definition_shape}

5.1.8.23.1 partial_area_definition to numeric_parameter (as effective_length)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: applied_area <=
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 =>
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}

5.1.8.23.2 partial_area_definition to numeric_parameter (as maximum_-length)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: applied_area <=
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 =>
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}

5.1.8.23.3 partial_area_definition to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: applied_area <=
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 =>
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

5.1.8.24 Partial_circular_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.description = 'partial circular'}

5.1.8.24.1 partial_circular_path to numeric_parameter (as sweep_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: path_feature_component <=
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 =>
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}

5.1.8.25 Path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.25.1 path to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: path_feature_component <=
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 =>
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

5.1.8.26 Planar_pocket_bottom_condition

AIM element: pocket_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: pocket_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'planar'}

5.1.8.26.1 planar_pocket_bottom_condition to direction_element (as floor_-normal)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: pocket_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_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.name = 'floor normal'}
representation =>
shape_representation=>
direction_shape_representation

5.1.8.26.2 planar_pocket_bottom_condition to location_element (as floor_-location)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)5.2.4.17
Reference Path: pocket_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_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.name = 'floor location'}
representation =>
shape_representation=>
location_shape_representation

5.1.8.26.3 planar_pocket_bottom_condition to numeric_parameter (as floor_-radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: pocket_bottom <=
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 =>
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}

5.1.8.27 Planar_profile_floor

AIM element: profile_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: profile_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'planar'}

5.1.8.27.1 planar_profile_floor to planar_element (as floor)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: profile_floor <=
shape_aspect
shape_definition = shape_aspect
shape_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.name = 'floor'}
representation =>
planar_shape_representation

5.1.8.28 Planar_rip_top_floor

AIM element: rib_top_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: rib_top_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'planar'}

5.1.8.28.1 planar_rib_top_floor to closed_profile (as boundary)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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)

5.1.8.28.2 planar_rib_top_floor to planar_element (as planar_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)5.2.4.17
Reference Path: rib_top_floor<=
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 =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'rib top face'}
representation =>
shape_representation =>
planar_shape_representation

5.1.8.29 Planar_top_condition

AIM element: boss_top
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: boss_top <=
shape_aspect
{shape_aspect
shape_aspect.description = 'planar'}

5.1.8.29.1 planar_top_condition to direction_element (as top_normal)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: boss_top <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation=>
direction_shape_representation

5.1.8.29.2 planar_top_condition to location_element (as top_location)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: boss_top <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation=>
location_shape_representation

5.1.8.30 Pocket_bottom_condition

AIM element: pocket_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.30.1 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'

5.1.8.31 Profile_floor

AIM element: profile_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.31.1 start_or_end

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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')}

5.1.8.31.2 profile_floor to numeric_parameter (as floor radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.8.32 Radiused_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'radiused'}

5.1.8.33 Rip_top_floor

AIM element: rib_top_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.34 Second_chamfer_offset

AIM element: chamfer_offset
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: chamfer_offset <=
shape_aspect
{shape_aspect
shape_aspect.description = 'second offset'}

5.1.8.34.1 second_chamfer_offset to face_shape_element (as second_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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

5.1.8.35 Second_offset

AIM element: chamfer_offset
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: chamfer_offset <=
shape_aspect
{shape_aspect
shape_aspect.description = 'second offset'}

5.1.8.35.1 second_offset to numeric_parameter (as offset_amount)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.8.36 Slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.36.1 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')}

5.1.8.37 Spherical_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'spherical'}

5.1.8.37.1 spherical_hole_bottom to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.8.38 Through_bottom_condition

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.39 Through_pocket_bottom_condition

AIM element: pocket_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.40 Through_profile_floor

AIM element: profile_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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}

5.1.8.41 Woodruff_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'woodruff'}

5.1.8.41.1 woodruff_slot_end_type to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9 feature_profile UoF

5.1.9.1 Circular_closed_profile

AIM element: circular_closed_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: circular_closed_profile <=
shape_aspect

5.1.9.1.1 circular_closed_profile to numeric_parameter (as diameter)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.2 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 5.2.4.20)
Reference Path: (circular_closed_profile <=)
(closed_path_profile <=)
(ngon_closed_profile <=)
(rectangular_closed_profile <=)
shape_aspect

5.1.9.3 General_closed_profile

AIM element: closed_path_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: closed_path_profile <=
shape_aspect

5.1.9.3.1 general_closed_profile to path_element (as closed_profile_shape)

AIM element: PATH
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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

5.1.9.4 General_open_profile

AIM element: open_path_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: open_path_profile <=
shape_aspect

5.1.9.4.1 general_open_profile to path_element (as enclosed_boundary)

AIM element: PATH
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)5.2.4.17
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

5.1.9.5 Linear_profile

AIM element: linear_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: linear_profile <=
shape_aspect

5.1.9.5.1 linear_profile to numeric_parameter (as profile_length)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.6 Ngon_profile

AIM element: ngon_closed_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: ngon_closed_profile <=
shape_aspect

5.1.9.6.1 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')

5.1.9.6.2 ngon_profile to numeric_parameter (as diameter)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.6.3 ngon_profile to numeric_parameter (as corner_radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.6.4 ngon_profile to numeric_parameter (as number_of_sides)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.7 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 5.2.4.20)
Reference Path: (linear_profile <=)
(open_path_profile <=)
(partial_circular_profile <=)
(rounded_u_profile <=)
(square_u_profile <=)
(tee_profile <=)
(vee_profile <=)
shape_aspect

5.1.9.7.1 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

5.1.9.8 Partial_circular_profile

AIM element: partial_circular_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: partial_circular_profile <=
shape_aspect

5.1.9.8.1 partial_circular_profile to numeric_parameter (as radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.8.2 partial_circular_profile to numeric_parameter (as sweep_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.9 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 5.2.4.20)
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}

5.1.9.9.1 profile to orientation (as placement)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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

5.1.9.10 Rectangular_closed_profile

AIM element: rectangular_closed_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: rectangular_closed_profile <=
shape_aspect

5.1.9.10.1 rectangular_closed_profile to numeric_parameter (as corner_-radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.10.2 rectangular_closed_profile to numeric_parameter (as profile_-length)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.10.3 rectangular_closed_profile to numeric_parameter (as profile_-width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.11 Rounded_u_profile

AIM element: rounded_u_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: rounded_u_profile <=
shape_aspect

5.1.9.11.1 rounded_u_profile to numeric_parameter (as depth)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.11.2 rounded_u_profile to numeric_parameter (as width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.12 Square_u_profile

AIM element: square_u_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: square_u_profile <=
shape_aspect

5.1.9.12.1 square_u_profile to numeric_parameter (as depth)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.12.2 square_u_profile to numeric_parameter (as first_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.12.3 square_u_profile to numeric_parameter (as first_radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.12.4 square_u_profile to numeric_parameter (as second_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.12.5 square_u_profile to numeric_parameter (as second_radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.12.6 square_u_profile to numeric_parameter (as width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13 Tee_profile

AIM element: tee_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: tee_profile <=
shape_aspect

5.1.9.13.1 tee_profile to numeric_parameter (as cross_bar_depth)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13.2 tee_profile to numeric_parameter (as cross_bar_width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13.3 tee_profile to numeric_parameter (as depth)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13.4 tee_profile to numeric_parameter (as first_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13.5 tee_profile to numeric_parameter (as first_offset)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13.6 tee_profile to numeric_parameter (as radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13.7 tee_profile to numeric_parameter (as second_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13.8 tee_profile to numeric_parameter (as second_offset)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.13.9 tee_profile to numeric_parameter (as width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: ee_profile <= t
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}

5.1.9.14 Vee_profile

AIM element: vee_profile
Source: ISO 10303-522
Rules: 5.2.4.20 shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: vee_profile <=
shape_aspect

5.1.9.14.1 vee_profile to numeric_parameter (as first length)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.14.2 vee_profile to numeric_parameter (as second length)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.14.3 vee_profile to numeric_parameter (as profile_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.14.4 vee_profile to numeric_parameter (as profile_radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.9.14.5 vee_profile to numeric_parameter (as tilt_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}

5.1.10 manufacturing_feature UoF

5.1.10.1 Bevel_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: gear <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.1.1 bevel_gear to numeric_parameter (as root_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.1.2 bevel_gear to numeric_parameter (as tip_angle)

AIM element: PATH

Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)

Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.2 Boss

AIM element: boss
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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')}

5.1.10.2.1 boss to boss_top_condition (as top_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.2.2 boss to linear_path (as boss_height)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.2.3 boss to numeric_parameter (as fillet_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.3 Catalogue_gear

AIM element: externally_defined_feature_definition
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}]

5.1.10.3.1 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}

5.1.10.4 Catalogue_knurl

AIM element: externally_defined_feature_definition
Source: ISO 10303-41
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}]

5.1.10.4.1 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}

5.1.10.5 Catalogue_marking

AIM element: externally_defined_feature_definition
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}]

5.1.10.5.1 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}

5.1.10.6 Catalogue_thread

AIM element: externally_defined_feature_definition
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}]

5.1.10.6.1 Catalogue_thread to numeric_parameter (as major_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: externally_defined_feature_definition <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.6.2 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}

5.1.10.7 Chamfer

AIM element: chamfer
Source: ISO 10303-522
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: chamfer <=
transition_feature <=
shape_aspect

5.1.10.7.1 chamfer to face_shape_element (as chamfer_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)5.2.4.17
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
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

5.1.10.7.2 chamfer to first_offset (as first_face_offset)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.7.3 chamfer to second_chamfer_offset (as second_face_offset)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.8 Circular_boss

AIM element: boss
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: boss <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'circular'}

5.1.10.8.1 circular_boss to angle_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.8.2 circular_boss to diameter_taper (as change_in_diameter)

AIM element:	PATH
Rules:	shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21) shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.8.3 circular_boss to directed_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.8.4 circular_boss to circular_closed_profile (as circular_profile)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.9 Circular_closed_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}

5.1.10.9.1 circular_closed_shape_profile to complete_circular_profile (as closed_boundary)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)5.2.4.17
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

5.1.10.10 Circular_cutout

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: pocket <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'circular cutout'}

5.1.10.10.1 circular_cutout to circular_closed_profile (as circular_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.11 Circular_offset_pattern

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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]

5.1.10.11.1 circular_offset_pattern to numeric_parameter (as angular_offset)

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
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}

5.1.10.11.2 circular_offset_pattern to numeric_parameter (as index_number)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
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}

5.1.10.12 Circular_omit_pattern

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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]

5.1.10.12.1 circular_omit_pattern to numeric_parameter (as omit_index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
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}

5.1.10.13 Circular_pattern

AIM element: circular_pattern
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: circular_pattern <=
replicate_feature <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.13.1 circular_pattern to circular_offset_pattern (as relocated_base_-feature)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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]

5.1.10.13.2 circular_pattern to circular_omit_pattern (as missing_base_-feature)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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]

5.1.10.13.3 circular_pattern to numeric_parameter (as angular_spacing)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.13.4 circular_pattern to numeric_parameter (as base_feature_-diameter)

AIM element: PATH

Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)

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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.13.5 circular_pattern to numeric_parameter (as base_feature_rotation)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.13.6 circular_pattern to numeric_parameter (as number_of_features)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.14 Closed_slot

AIM element: slot
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: slot <=

{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.14.1 closed_slot to general_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.14.2 closed_slot to complete_circular_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.14.3 closed_slot to loop_slot_end_type (as end_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.15 Compound_feature

AIM element: compound_feature
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'

5.1.10.15.1 feature_description

AIM element: characterized_object.description
Source: ISO 10303-41
Reference Path: compound_feature <=
feature_definition <=
characterized_object
characterized_object.description

5.1.10.15.2 feature_name

AIM element: characterized_object.name
Source: ISO 10303-41
Reference Path: compound_feature <=
feature_definition <=
characterized_object
characterized_object.name

5.1.10.15.3 compound_feature to compound_feature_element (as element)

#1: as element except for thread

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
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 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
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)

5.1.10.16 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 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
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 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: (applied_area <=
shape_aspect)

5.1.10.16.1 compound_feature_element to machining_feature (as element)

AIM element: IDENTICAL MAPPING

5.1.10.16.2 compound_feature_element to transition_feature (as element)

AIM element: IDENTICAL MAPPING

5.1.10.17 Compound_feature_relationship

AIM element: shape_aspect_relationship
Source: ISO 10303-41
Reference Path: {shape_aspect_relationship
shape_aspect_relationship.name = 'compound feature ordering'}

5.1.10.17.1 compound_feature_relationship to compound_feature_element (as predecessor)

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
(instanced_feature)
(transition_feature)

5.1.10.17.2 compound_feature_relationship to compound_feature_element (as successor)

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(instanced_feature)
(transition_feature)

5.1.10.18 Constant_radius_edge_round

AIM element: edge_round
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: edge_round <=
transition_feature <=
shape_aspect
{shape_aspect
shape_aspect.description = 'constant radius'}

5.1.10.18.1 constant_radius_edge_round to numeric_parameter (as first_-face_offset)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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 =>
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}

5.1.10.18.2 constant_radius_edge_round to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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 =>
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}

5.1.10.18.3 constant_radius_edge_round to numeric_parameter (as second_-face_offset)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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 =>
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}

5.1.10.19 Constant_radius_fillet

AIM element: fillet
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: fillet <=
transition_feature <=
shape_aspect
{shape_aspect
shape_aspect.description = 'constant radius'}

5.1.10.19.1 constant_radius_fillet to numeric_parameter (as first_face_offset)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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 =>
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}

5.1.10.19.2 constant_radius_fillet to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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 =>
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}

5.1.10.19.3 constant_radius_fillet to numeric_parameter (as second_face_-offset)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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 =>
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}

5.1.10.20 Counterbore_hole

AIM element: composite_hole
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: composite_hole <=
compound_feature <=
feature_definition=>
[feature_definition <=
characterized_object]
instanced_feature <=
[{[shape_aspect =>
composite_shape_aspect]
[shape_aspect
shape_aspect.description = 'counterbore']}
shape_aspect]

5.1.10.20.1 counterbore_hole to round_hole (as larger_hole)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.20.2 counterbore_hole to round_hole (as smaller_hole)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.21 Countersunk_hole

AIM element: composite_hole
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: composite_hole <=
compound_feature <=
feature_definition=>
[feature_definition <=
characterized_object]
instanced_feature <=
[{[shape_aspect =>
composite_shape_aspect]
[shape_aspect
shape_aspect.description = 'countersunk']}
shape_aspect]

5.1.10.21.1 countersunk_hole to round_hole (as constant_diameter_hole)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.name = 'constant diameter hole']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
instanced_feature <=
feature_definition =>
round_hole
```


5.1.10.21.2 countersunk_hole to round_hole (as tapered_hole)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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 =>
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}

5.1.10.22 Cutout

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: pocket <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
(characterized_object.description = 'circular cutout')
(characterized_object.description = 'complex cutout')}

5.1.10.22.1 cutout to through_pocket_bottom_condition (as bottom_-condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.23 Defined_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: gear <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.24 Defined_marking

AIM element: marking
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: marking <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.24.1 defined_marking to descriptive_parameter (as font_name)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: marking <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'font name'}
representation_item =>
descriptive_representation_item

5.1.10.24.2 defined_marking to descriptive_parameter (as special_-instructions)

AIM element:	PATH
Rules:	dependent_instantiable_shape_representation – (See 5.2.4.9) representation_subtype_exclusiveness – (See 5.2.4.17) shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path:	marking <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition represented_definition=property_definition represented_definition<- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation=> shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'special instructions'} representation_item => descriptive_representation_item

5.1.10.24.3 defined_marking to numeric_parameter (as character_height)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: marking <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.24.4 defined_marking to numeric_parameter (as character_spacing)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: marking <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.25 Defined_thread

AIM element: thread
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: thread <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.25.1 defined_thread to numeric_parameter (as crest)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: thread <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.25.2 defined_thread to numeric_parameter (as major_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: thread <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.25.3 defined_thread to numeric_parameter (as minor_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: thread <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.25.4 defined_thread to numeric_parameter (as pitch_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: thread <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.26 Diagonal_knurl

AIM element: turned_knurl
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: turned_knurl <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'diagonal'}

5.1.10.26.1 diagonal_knurl to descriptive_parameter (as helix_hand)

AIM element: PATH

Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)

Reference Path:
```
turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'helix hand'}
representation_item =>
descriptive_representation_item
```

5.1.10.26.2 diagonal_knurl to numeric_parameter (as helix_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.27 Diamond_knurl

AIM element: turned_knurl
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: turned_knurl <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'diamond'}

5.1.10.27.1 diamond_knurl to numeric_parameter (as helix_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.28 Edge_round

AIM element: edge_round
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: edge_round <=
transition_feature <=
shape_aspect

5.1.10.28.1 edge_round to face_shape_element (as edge_round_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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

5.1.10.28.2 edge_round to face_shape_element (as first_face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: edge_round <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_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

5.1.10.28.3 edge_round to face_shape_element (as second_face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: edge_round <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_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

5.1.10.29 Fillet

AIM element: fillet
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: fillet <=
transition_feature <=
shape_aspect

5.1.10.29.1 fillet to face_shape_element (as fillet_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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

5.1.10.29.2 fillet to face_shape_element (as first_face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: fillet <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_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

5.1.10.29.3 fillet to face_shape_element (as second_face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: fillet <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_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

5.1.10.30 Gear

AIM element: (gear)
(externally_defined_feature_definition)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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')}])

5.1.10.30.1 internal_or_external_gear

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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')}

5.1.10.30.2 module_or_diametral_pitch

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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')}

5.1.10.30.3 gear to numeric_parameter (as normal_attribute)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.30.4 gear to numeric_parameter (as nominal_tooth_depth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.30.5 gear to numeric_parameter (as reference_pressure_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.30.6 gear to numeric_parameter (as profile_shift)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.30.7 gear to numeric_parameter (as number_of_teeth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.30.8 gear to numeric_parameter (as face_width)

AIM element: PATH

Rules:
dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)

Reference Path:
(gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.30.9 gear to numeric_parameter (as root_fillet_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.30.10 gear to numeric_parameter (as tip_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.30.11 gear to shape (as applied_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
property_definition.definition
property_definition=>
product_definition_shape

5.1.10.31 General_boss

AIM element: boss
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: boss <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'complex'}

5.1.10.31.1 general_boss to angle_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.31.2 general_boss to directed_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.31.3 general_boss to closed_profile (as enclosed_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.32 General_cutout

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: pocket <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'complex cutout'}

5.1.10.32.1 general_cutout to profile (as boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.33 General_outside_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}

5.1.10.33.1 general_outside_profile to profile (as boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.34 General_pattern

AIM element: feature_pattern
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: feature_pattern <=
replicate_feature <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.34.1 general_pattern to orientation (as feature_placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'base feature placement'}
representation_item =>
geometric_representation_item =>
placement

5.1.10.35 General_pocket

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: pocket <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'complex'}

5.1.10.35.1 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

5.1.10.35.2 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

5.1.10.35.3 general_pocket to profile (as boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.36 General_removal_volume

AIM element: removal_volume
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: removal_volume <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.36.1 general_removal_volume to shape_element (as removal_volume)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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

5.1.10.37 General_revolution

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: revolved_profile <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'open profile'}

5.1.10.37.1 general_revolution to general_open_profile (as outer_edge_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.38 General_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}

5.1.10.38.1 general_shape_profile to path (as profile_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.39 Groove

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: revolved_profile <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'groove'}

5.1.10.39.1 groove to open_profile (as sweep)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.40 Helical_bevel_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: gear <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'helix bevel gear'}

5.1.10.40.1 left_or_right_hand_tooth

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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')}

5.1.10.40.2 helical_gear to numeric_parameter (as reference_helix_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.41 Helical_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: gear <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'helix gear'}

5.1.10.41.1 left_or_right_hand_tooth

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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')}

5.1.10.41.2 helical_gear to numeric_parameter (as reference_helix_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.42 Hole

AIM element: (round_hole)
(composite_hole)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: (round_hole <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object)
(composite_hole <=
compound_feature <=
feature_definition=>
[feature_definition <=
characterized_object]
instanced_feature <=
[{[shape_aspect =>
composite_shape_aspect])

5.1.10.43 Knurl

AIM element: (turned_knurl)
(externally_defined_feature_definition)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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'}])

5.1.10.43.1 knurl to partial_area_definition (as partial_profile)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.43.2 knurl to shape_element (as applied_shape)

AIM element: PATH
Source: ISO 10303-522
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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

5.1.10.44 Machining_feature

AIM element:	instanced_feature
Source:	ISO 10303-522
Rules:	machining_feature_life_cycle – (See 5.2.4.14) shape_aspect_subtype_exclusiveness – (See 5.2.4.20) subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path:	instanced_feature <= [shape_aspect] [feature_definition <= characterized_object]

5.1.10.44.1 usage_name

AIM element:	shape_aspect.description
Source:	ISO 10303-41
Reference Path:	instanced_feature <= shape_aspect shape_aspect.description

5.1.10.44.2 machining_feature to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.10.45 Manufacturing_feature

AIM element: characterized_object
Source: ISO 10303-41

5.1.10.46 Manufacturing_feature_group

AIM element: group
Source: ISO 10303-41

5.1.10.46.1 group_description

AIM element: group.description
Source: ISO 10303-41

5.1.10.46.2 group_name

AIM element: group.name
Source: ISO 10303-41

5.1.10.46.3 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)

5.1.10.46.4 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

5.1.10.47 Marking

AIM element: (marking)
(externally_defined_feature_definition)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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'}])

5.1.10.47.1 marking to shape (as applied_to_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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

5.1.10.47.2 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
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 = 'marking text'}
representation_item =>
descriptive_representation_item

5.1.10.48 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

5.1.10.48.1 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
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='maximum feature limit'}
shape_representation=>
planar_shape_representation

5.1.10.49 Open_slot

AIM element: slot
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: slot <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.49.1 open_slot to path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.49.2 open_slot to slot_end_type (as end_conditions)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.50 Outer_diameter

AIM element: outer_round
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: outer_round <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'outer diameter'}

5.1.10.50.1 outer_diameter to angle_taper (as reduced_size)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.50.2 outer_diameter to diameter_taper (as reduced_size)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.50.3 outer_diameter to directed_taper (as reduced_size)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.50.4 outer_diameter to numeric_parameter (as diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.50.5 outer_diameter to numeric_parameter (as feature_length)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.51 Outer_diameter_to_shoulder

AIM element: outer_round
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: outer_round <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'outer diameter to shoulder'}

5.1.10.51.1 outer_diameter_to_shoulder to numeric_parameter (as diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.51.2 outer_diameter_to_shoulder to numeric_parameter (as feature_-length)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.51.3 outer_diameter_to_shoulder to vee_profile (as v_shape_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.52 Outer_round

AIM element: outer_round
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: outer_round <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.53 Partial_circular_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}

5.1.10.53.1 partial_circular_shape_profile to partial_circular_profile (as open_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.54 Planar_face

AIM element: flat_face
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: flat_face <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.54.1 planar_face to direction_element (as removal_direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: flat_face <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'removal direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.10.54.2 planar_face to linear_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.54.3 planar_face to linear_profile (as removal_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.54.4 planar_face to linear_profile (as face_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.54.5 planar_face to numeric_parameter (as removal_depth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: flat_face <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.54.6 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

5.1.10.54.7 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

5.1.10.55 Pocket

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: pocket <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.55.1 pocket to numeric_parameter (as base_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.55.2 pocket to angle_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.55.3 pocket to directed_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.55.4 pocket to linear_path (as pocket_depth)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.55.5 pocket to pocket_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.55.6 pocket to through_pocket_bottom_condition (as bottom_-condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.56 Profile_feature

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: (outside_profile <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object)

5.1.10.56.1 profile_feature to linear_profile (as profile_swept_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.57 Protrusion

AIM element: protrusion
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: protrusion <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.57.1 protrusion to shape_element (as shape_volume)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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

5.1.10.58 Recess

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: pocket <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'recess'}

5.1.10.58.1 recess to pocket_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.58.2 recess to closed_profile (as fillet boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.58.3 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

5.1.10.58.4 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

5.1.10.59 Rectangular_boss

AIM element: boss
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: boss <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'rectangular'}

5.1.10.59.1 rectangular_boss to angle_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.59.2 rectangular_boss to directed_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.59.3 rectangular_boss to closed_profile (as enclosed_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.60 Rectangular_closed_pocket

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: pocket <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'closed rectangular'}

5.1.10.60.1 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

5.1.10.60.2 rectangular_closed_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

5.1.10.60.3 rectangular_closed_pocket to rectangular_closed_profile (as closed_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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
```

5.1.10.61 Rectangular_closed_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}
```


5.1.10.61.1 rectangular_closed_shape_profile to rectangular_closed_profile (as closed_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.62 Rectangular_offset_pattern

AIM element: shape_aspect
Source: ISO 10303-41
Rules: 5.2.4.21
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]

5.1.10.62.1 rectangular_offset_pattern to direction_element (as offset_-direction)

AIM element: PATH
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: shape_aspect <-
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'offset direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.10.62.2 rectangular_offset_pattern to numeric_parameter (as column_-index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
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}

5.1.10.62.3 rectangular_offset_pattern to numeric_parameter (as offset_-distance)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
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}

5.1.10.62.4 rectangular_offset_pattern to numeric_parameter (as row_index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
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}

5.1.10.63 Rectangular_omit_pattern

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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]

5.1.10.63.1 rectangular_omit_pattern to numeric_parameter (as column_-index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
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}

5.1.10.63.2 rectangular_omit_pattern to numeric_parameter (as row_index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
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}

5.1.10.64 Rectangular_open_pocket

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: pocket <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'open rectangular'}

5.1.10.64.1 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

5.1.10.64.2 rectangular_open_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

5.1.10.64.3 rectangular_open_pocket to square_u_profile (as open_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.65 Rectangular_open_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
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'}

5.1.10.65.1 rectangular_open_shape_profile to square_U_profile (as open_-boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.66 Rectangular_pattern

AIM element: rectangular_pattern
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: rectangular_pattern <=
replicate_feature <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.66.1 rectangular_pattern to direction_element (as column_layout_-direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'column layout direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.10.66.2 rectangular_pattern to direction_element (as row_layout_-direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'row layout direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.10.66.3 rectangular_pattern to numeric_parameter (as column_spacing)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.66.4 rectangular_pattern to numeric_parameter (as columns)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.66.5 rectangular_pattern to numeric_parameter (as row_spacing)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.66.6 rectangular_pattern to numeric_parameter (as rows)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}
```


5.1.10.66.7 rectangular_pattern to rectangular_offset_pattern (as relocated_-base_feature)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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]

5.1.10.66.8 rectangular_pattern to rectangular_omit_pattern (as missing_-base_feature)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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]

5.1.10.67 Replicate_base

AIM element: feature_component_relationship
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: feature_component_relationship <=
shape_aspect_relationship

5.1.10.67.1 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

5.1.10.67.2 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

5.1.10.68 Replicate_feature

AIM element: replicate_feature
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)5.2.4.20
Reference Path: replicate_feature <=
{feature_definition<=
instanced_feature}
feature_definition<=
characterized_object

5.1.10.68.1 replicate_feature to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.10.68.2 replicate_feature to replicate_base (as replicate_base_feature)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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'}

5.1.10.69 Revolved_feature

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: revolved_profile <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.69.1 revolved_feature to direction_element (as material_side)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: revolved_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'removal direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.10.69.2 revolved_feature to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: revolved_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.70 Revolved_flat

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: revolved_profile <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'flat'}

5.1.10.70.1 revolved_flat to linear_profile (as flat_edge_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.71 Revolved_round

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: revolved_profile <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'round'}

5.1.10.71.1 revolved_round to partial_circular_profile (as rounded_edge_-shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.72 Rip_top

AIM element: rib_top
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: rib_top <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.72.1 rib_top to direction_element (as removal_direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: ribtop <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name= 'removal direction'}
representation =>
shape_representation =>
direction_shape_representation

5.1.10.72.2 rib_top to rib_top_floor (as floor_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.73 Round_hole

AIM element: round_hole
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: round_hole <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.73.1 round_hole to angle_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.73.2 round_hole to blind_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.73.3 round_hole to circular_closed_profile (as diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.73.4 round_hole to diameter_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.73.5 round_hole to directed_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.73.6 round_hole to linear_path (as hole_depth)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.73.7 round_hole to through_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.74 Rounded_end

AIM element: rounded_end
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: rounded_end <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.74.1 rounded_end to linear_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.74.2 rounded_end to partial_circular_profile (as partial_circular_-boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.75 Shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: outside_profile <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.75.1 shape_profile to profile_floor (as floor_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.75.2 shape_profile to through_profile_floor (as floor_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.75.3 shape_profile to direction_element (as removal_direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'removal direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.10.76 Slot

AIM element: slot
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: slot <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.76.1 slot to open_profile (as sweep_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)

5.1.10.76.2 Spherical_cap

AIM element: spherical_cap
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: spherical_cap <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.76.3 spherical_cap to numeric_parameter (as internal_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: spherical_cap <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.76.4 spherical_cap to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: spherical_cap <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.77 Spur_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: gear <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'spur gear'}

5.1.10.78 Step

AIM element: step
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: step <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.78.1 step to linear_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.78.2 step to vee_profile (as removal_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.78.3 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

5.1.10.78.4 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

5.1.10.79 Straight_bevel_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: gear <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'straight bevel gear'}

5.1.10.80 Straight_knurl

AIM element: turned_knurl
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: turned_knurl <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
{characterized_object
characterized_object.description = 'straight'}

5.1.10.81 Thread

AIM element: (thread)
(externally_defined_feature_definition)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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'}])

5.1.10.81.1 thread to descriptive_parameter (as fit_class)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'fit class'}
representation_item =>
descriptive_representation_item

5.1.10.81.2 thread to descriptive_parameter (as fit_class_2)

AIM element: PATH

Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)

Reference Path:
```
 (thread <=)
(externally_defined_feature_definition <=)
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'fit class 2'}
representation_item =>
descriptive_representation_item
```


5.1.10.81.3 thread to descriptive_parameter (as form)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'form'}
representation_item =>
descriptive_representation_item

5.1.10.81.4 thread to descriptive_parameter (as qualifier)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'qualifier'}
representation_item =>
descriptive_representation_item

5.1.10.81.5 inner_or_outer_thread

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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')}

5.1.10.81.6 thread to numeric_parameter (as number_of_threads)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.81.7 thread to numeric_parameter (as nominal_size)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.81.8 thread to partial_area_definition (as partial_profile)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.81.9 thread to shape (as applied_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object<-
property_definition.definition
property_definition=>
product_definition_shape

5.1.10.81.10 thread to descriptive_parameter (as thread_hand)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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

5.1.10.81.11 thread to thread_runout (as runout)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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

5.1.10.82 Thread_runout

AIM element: thread_runout
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference Path: thread_runout<=
shape_aspect
{shape_aspect
shape_aspect.of_shape->
product_definition_shape}

5.1.10.82.1 thread_runout to numeric_parameter (as length_of_runout)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: thread_runout <=
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 =>
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}

5.1.10.82.2 thread_runout to descriptive_parameter (as included_or_extra)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: thread_runout <=
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 =>
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')

5.1.10.82.3 thread_runout to descriptive_parameter (as pitch_or_dimension)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: thread_runout <=
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 =>
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')

5.1.10.83 Transition_feature

AIM element: transition_feature
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
transition_feature_life_cycle – (See 5.2.4.24)
transition_feature_on_part_boundary – (See 5.2.4.25)
Reference Path: transition_feature <=
shape_aspect

5.1.10.83.1 Turned_knurl

AIM element: turned_knurl
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.14)
shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
subtype_mandatory_characterized_object – (See 5.2.4.23)
Reference Path: turned_knurl <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.10.83.2 turned_knurl to numeric_parameter (as diametral_pitch)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.83.3 turned_knurl to numeric_parameter (as number_of_teeth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.83.4 turned_knurl to numeric_parameter (as major_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.83.5 turned_knurl to numeric_parameter (as nominal_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.83.6 turned_knurl to numeric_parameter (as root_fillet)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.10.83.7 turned_knurl to numeric_parameter (as tooth_depth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation=>
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}

5.1.11 functional_limitations UoF

5.1.11.1 Angular_dimension_tolerance

#1: if directed = true

AIM element: (directed_dimensional_location)
(angular_location)
Source: ISO 10303-219
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 5.2.4.22)

5.1.11.1.1 feature_type

AIM element: descriptive_representation_item
Source: ISO 10303-43
Reference path: angular_location <=
dimensional_location
dimensional_characteristic = dimensional_location
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 = 'actual')
{(descriptive_representation_item = 'nominal')
(descriptive_representation_item = 'datum')}

5.1.11.1.2 angular_dimension_tolerance to orientation (as plane_and_-direction)

#1: if directed = true

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
Reference path: (directed_dimensional_location <=
dimensional_location =>
angular_location)

#2: if directed = false

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
Reference path: (angular_location <=)
dimensional_location <=
shape_aspect_relationship
shape_definition = shape_aspect_relationship
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
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

5.1.11.2 Angular_size_dimension_tolerance

AIM element: angular_size
Source: ISO 10303-47

5.1.11.2.1 full_or_half

AIM element: descriptive_representation_item
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 = 'full angle')
(descriptive_representation_item = 'half angle')}

5.1.11.2.2 major_angle

AIM element: angular_size.angle_selection
Source: ISO 10303-47

5.1.11.3 Angularity_tolerance

AIM element: angularity_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.3.1 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]

5.1.11.3.2 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

5.1.11.4 Circular_runout_tolerance

AIM element: circular_runout_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.4.1 runout_angle

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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]

5.1.11.4.2 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

5.1.11.5 Circularity_tolerance

AIM element: roundness_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.6 Common_datum

AIM element: common_datum
 (see NOTE)
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)

NOTE The datum that represents a Common_datum uses at least four shape_aspect_relationships. At least two shape_-aspect_relationship reference the datum associated to the Common_datum. Each datum referenced by the Common_datum has a shape_aspect_relationship to a datum_feature. The Common_datum has a shape_aspect_relationship that references the same datum_features.

5.1.11.6.1 common_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.relatedshape_aspect ->
shape_aspect =>
datum

5.1.11.7 Concentricity_tolerance

AIM element: concentricity_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.7.1 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

5.1.11.7.2 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

5.1.11.8 Coordinate_tolerance

AIM element: dimensional_location
Source: ISO 10303-47
Reference path: {dimensional_location
dimensional_location.name = 'coordinate tolerance'}

5.1.11.8.1 axial

AIM element: descriptive_representation_item
Source: ISO 10303-43
Reference path: dimensional_location
dimensional_characteristic = dimensional_location
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 = 'axial')}

5.1.11.8.2 direction_type

AIM element: descriptive_representation_item
Source: ISO 10303-43
Reference path: dimensional_location
dimensional_characteristic = dimensional_location
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 = 'x axis')
(descriptive_representation_item = 'y axis')
(descriptive_representation_item = 'z axis')
(descriptive_representation_item = 'radial')
(descriptive_representation_item = 'angle')

5.1.11.8.3 coordinate_tolerance to orientation (as base_coordinate_system)

#1: if directed = true

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
Reference path: dimensional_location

#2: if directed = false

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
Reference path: dimensional_location <=
shape_aspect_relationship
shape_definition = shape_aspect_relationship
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.9 Curved_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference path: {dimensional_size
dimensional_size.name = 'curve length'}

5.1.11.10 Cylindricity_tolerance

AIM element: cylindricity_tolerance
Source: ISO 10303-519

5.1.11.11 Datum

AIM element: datum_reference
Source: ISO 10303-47

5.1.11.11.1 name

AIM element: shape_aspect.name
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference path: datum_reference
datum_reference.referenced_datum ->
datum <=
shape_aspect
shape_aspect.name

5.1.11.11.2 precedence

AIM element: datum_reference.precedence
Source: ISO 10303-47

5.1.11.12 Datum_feature

AIM element: (datum)
([datum]
[datum_feature])
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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)}

5.1.11.12.1 datum_feature to datum_target_set (as datum_representation)

AIM element: PATH
Reference path: datum
datum.established_by_relationships

5.1.11.12.2 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

5.1.11.12.3 datum_feature to shape_element (as datum_representation)

AIM element: IDENTICAL MAPPING

5.1.11.13 Datum_target

AIM element: datum_target
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)

5.1.11.13.1 identifier

AIM element: datum_target.target_id
Source: ISO 10303-47

5.1.11.14 Datum_target_set

AIM element: datum.established_by_relationships
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference path: {datum.established_by_relationships[i] ->
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
datum_target}

5.1.11.14.1 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

5.1.11.14.2 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

5.1.11.15 Diameter_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference path: {dimensional_size
dimensional_size.name = 'diameter'}

5.1.11.16 Dimensional_tolerance

AIM element: shape_dimension_representation
Source: ISO 10303-47

5.1.11.16.1 dimension_value

AIM element: measure_representation_item
Source: ISO 10303-45
Reference path: shape_dimension_representation <=
shape_representation <=
representation
representation.items[i] ->
representation_item =>
measure_representation_item<=
measure_with_unit

5.1.11.16.2 dimension_description

#1 subtype is a Size_tolerance

AIM element: (dimensional_size.name)
Source: ISO 10303-47
ISO 10303-41
Reference path: shape_dimension_representation <-
dimensional_characteristic_representation.representation
dimensional_characteristic_representation
dimensional_characteristic_representation.dimension ->
dimensional_characteristic
dimensional_characteristic = dimensional_size
dimensional_size
dimensional_size.name

#2 subtype is a Location_tolerance

AIM element: (dimensional_size.name)
(shape_aspect_relationship.description)
Source: ISO 10303-47
ISO 10303-41
Reference path: shape_dimension_representation <-
dimensional_characteristic_representation.representation
dimensional_characteristic_representation
dimensional_characteristic_representation.dimension ->
dimensional_characteristic
dimensional_characteristic = dimensional_location
dimensional_location<=
shape_aspect_representation
shape_aspect_representation.description

5.1.11.16.3 dimension_note

AIM element: descriptive_representation_item.description
Source: ISO 10303-43
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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’)

5.1.11.16.4 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[1] ->
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value

5.1.11.16.5 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

5.1.11.16.6 dimensional_tolerance to tolerance_value (as limit)

AIM element: PATH
Reference Path: shape_dimension_representation <-
dimensional_characteristic_representation.representation
dimensional_characteristic_representation

5.1.11.17 Distance_along_curve_tolerance

AIM element: dimensional_location_with_path
Source: ISO 10303-47
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)

5.1.11.17.1 with_curve_direction

AIM element: dimensional_location_with_path.path
Source: ISO 10303-47

5.1.11.17.2 distance_along_curve_tolerance to shape_aspect (as path)

AIM element: PATH
Reference path: dimensional_location_with_path
dimensional_location_with_path.path ->
shape_aspect

5.1.11.18 Externally_defined_size_dimension

AIM element: [externally_defined_dimension_definition]
[dimensional_size_with_path]
Source: ISO 10303-219
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 10303-219
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]

5.1.11.18.1 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

5.1.11.18.2 externally_defined_size_dimension to document_assignment (as tolerance_definition)

AIM element: PATH
Reference path: externally_defined_dimension_definition
externally_defined_dimension_definition= document_reference_item
document_reference_item<-
applied_document_reference.items[i]
applied_document_reference
{applied_document_reference<=
document_reference
document_reference.role->
object_role
object_role.name= 'externally defined dimension specification'}

5.1.11.18.3 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

5.1.11.19 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)}

5.1.11.19.1 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]

5.1.11.20 Geometric_tolerance

AIM element: geometric_tolerance
Source: ISO 10303-47
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)

5.1.11.20.1 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

5.1.11.20.2 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

5.1.11.20.3 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

5.1.11.20.4 has_multiple_datums

AIM element: geometric_tolerance.description
Source: ISO 10303-45
Reference path: geometric_tolerance
geometric_tolerance.description
geometric_tolerance.description='seperate'
geometric_tolerance.description='simultaneous'

5.1.11.20.5 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

5.1.11.20.6 geometric_tolerance to shape_aspect (as applied_shape)

AIM element: PATH
Reference path: geometric_tolerance
geometric_tolerance.toleranced_shape_aspect ->
shape_aspect

5.1.11.20.7 geometric_tolerance to tolerance_zone (as zone_definition)

AIM element: PATH
Reference path: geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone

5.1.11.21 Geometric_tolerance_precedence_relationship

AIM element: geometric_tolerance_relationship
Source: ISO 10303-47
Reference path: {geometric_tolerance_relationship
geometric_tolerance_relationship.name = 'precedence'}

5.1.11.21.1 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

5.1.11.21.2 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

5.1.11.22 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'

5.1.11.22.1 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

5.1.11.23 Length_dimension

#1: if the length_dimension is specified and the optional path is specified

AIM element:	dimensional_size_with_path
Source:	ISO 10303-47
Reference path:	dimensional_size_with_path<= dimensional_size dimensional_size.name='length'

#2: if the length_dimension is specified and the optional path is not specified

AIM element:	dimensional_size
Source:	ISO 10303-47
Reference path:	dimensional_size dimensional_size.name='length'

5.1.11.23.1 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

5.1.11.24 Limits_and_fits

#1: if the limits_and_fits is selected for limits_and_fits

AIM element: limits_and_fits
Source: ISO 10303-47

#2: if the limits_and_fits is selected for numeric_parameter

AIM element: (qualified_representation_item)
Source: ISO 10303-45

5.1.11.24.1 deviation

#1: if the limits_and_fits is selected for limits_and_fits

AIM element: limits_and_fits.form_variance
Source: ISO 10303-47

#2: if the limits_and_fits is selected for numeric_parameter

AIM element: (qualitative_uncertainty.uncertainty_value)
Source: ISO 10303-45
Reference path: {qualified_representation_item <=
representation_item}
qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = uncertainty_qualifier
uncertainty_qualifier.measure_name='form variance'
uncertainty_qualifier =>
qualitative_uncertainty
qualitative_uncertainty.uncertainty_value

5.1.11.24.2 fitting_type

#1: if the limits_and_fits is selected for limits_and_fits

AIM element: limits_and_fits.zone_variance
Source: ISO 10303-47

#2: if the limits_and_fits is selected for numeric_parameter

AIM element: (qualitative_uncertainty.uncertainty_value)
Source: ISO 10303-45
Reference path: {qualified_representation_item <=
representation_item}
qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = uncertainty_qualifier
uncertainty_qualifier.measure_name='zone variance'
uncertainty_qualifier =>
qualitative_uncertainty
qualitative_uncertainty.uncertainty_value

5.1.11.24.3 grade

#1: if the limits_and_fits is selected for limits_and_fits

AIM element: limits_and_fits.grade
Source: ISO 10303-47

#2: if the limits_and_fits is selected for numeric_parameter

AIM element: (qualitative_uncertainty.uncertainty_value)
Source: ISO 10303-45
Reference path: {qualified_representation_item <=
representation_item}
qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = uncertainty_qualifier
uncertainty_qualifier.measure_name='grade'
uncertainty_qualifier =>
qualitative_uncertainty
qualitative_uncertainty.uncertainty_value

5.1.11.25 Linear_profile_tolerance

AIM element: line_profile_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.25.1 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

5.1.11.25.2 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
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 = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.26 Location_dimension_tolerance

#1: if directed = true

AIM element: (directed_dimensional_location)
Source: ISO 10303-219
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
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 5.2.4.21)

5.1.11.26.1 feature_type

AIM element: descriptive_representation_item
Source: ISO 10303-43
Reference path: angular_location <=
dimensional_location
dimensional_characteristic = dimensional_location
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 = 'actual')
{(descriptive_representation_item = 'nominal')
(descriptive_representation_item = 'datum')}

5.1.11.26.2 location_dimension_tolerance to orientation (as plane_and_-direction)

#1: if directed = true

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
Reference path: (directed_dimensional_location <=
dimensional_location <=)

#2: if directed = false

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)
Reference path: (dimensional_location <=)
shape_aspect_relationship
shape_definition = shape_aspect_relationship
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.27 Location_tolerance

AIM element: dimensional_location
Source: ISO 10303-47
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.21)

5.1.11.27.1 directed

AIM element: IDENTICAL MAPPING

5.1.11.27.2 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

5.1.11.27.3 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

5.1.11.28 Material_condition_modifier

AIM element: limit_condition
Source: ISO 10303-47

5.1.11.28.1 material_type

AIM element: IDENTICAL_MAPPING

5.1.11.29 Parallelism_tolerance

AIM element: parallelism_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.29.1 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]

5.1.11.29.2 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

5.1.11.29.3 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
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 = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.30 Perpendicularity_tolerance

AIM element: perpendicularity_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.30.1 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]

5.1.11.30.2 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

5.1.11.30.3 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
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 = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.31 Placed_target

AIM element: placed_datum_target_feature
Source: ISO 10303-219
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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')}

5.1.11.31.1 placed_target cartesian_coordinate_space (as is_defined_in)

AIM element: PATH
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
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.context_of_items ->
representation_context =>
geometric_representation_context

5.1.11.31.2 placed_target to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.32 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

5.1.11.32.1 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

5.1.11.32.2 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

5.1.11.32.3 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

5.1.11.33 Position_tolerance

AIM element: position_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.33.1 value_qualifier

AIM element: tolerance_zone_form.name
Source: ISO 10303-47
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)
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

5.1.11.33.2 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

5.1.11.33.3 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
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 = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.34 Projection

AIM element: projected_zone_definition
Source: ISO 10303-47

5.1.11.34.1 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]

5.1.11.34.2 projection to shape_element (as projection_end)

AIM element: PATH
Reference path: projected_zone_definition
projected_zone_definition.projection_end ->
shape_aspect

5.1.11.35 Radial_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference path: {dimensional_size
dimensional_size.name = 'radius'}

5.1.11.36 Size_tolerance

AIM element: dimensional_size
Source: ISO 10303-47

5.1.11.36.1 envelope

AIM element: representation.name
Source: ISO 10303-43
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation <=
shape_representation <=
representation
representation.name
{representation.name = 'envelope tolerance'}

5.1.11.36.2 size_tolerance to shape_element (as applied_shape)

AIM element: PATH
Reference path: dimensional_size
dimensional_size.applies_to ->
shape_aspect

5.1.11.37 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)}

5.1.11.37.1 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]

5.1.11.37.2 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
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 = '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
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 = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.38 Surface_profile_tolerance

AIM element: surface_profile_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.38.1 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

5.1.11.39 Symmetry_tolerance

AIM element: symmetry_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.39.1 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

5.1.11.39.2 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
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 = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.11.40 Target_area

AIM element: datum_target
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)

5.1.11.40.1 target_area to shape_element (as area_shape)

AIM element: IDENTICAL MAPPING

5.1.11.41 Target_circle

AIM element: placed_datum_target_feature
Source: ISO 10303-219
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
shape_aspect.description = 'circle'}

5.1.11.41.1 target_diameter

AIM element: measure_representation_item
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
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}

5.1.11.42 Target_line

AIM element: placed_datum_target_feature
Source: ISO 10303-219
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
shape_aspect.description = 'line'}

5.1.11.42.1 target_length

AIM element: measure_representation_item
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
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}

5.1.11.43 Target_point

AIM element: placed_datum_target_feature
Source: ISO 10303-219
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
shape_aspect.description = 'point'}

5.1.11.44 Target_rectangle

AIM element: placed_datum_target_feature
Source: ISO 10303-219
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
shape_aspect.description = 'rectangle'}

5.1.11.44.1 target_length

AIM element: measure_representation_item
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
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}

5.1.11.44.2 target_width

AIM element: measure_representation_item
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
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}

5.1.11.45 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'

5.1.11.45.1 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

5.1.11.46 Tolerance_limit

AIM element: type_qualifier
Source: ISO 10303-45

5.1.11.46.1 limit_qualifier

AIM element: type_qualifier.name
Source: ISO 10303-45
Reference path: (type_qualifier.name='maximum')
(type_qualifier.name='minimum')
(type_qualifier.name)

5.1.11.47 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 10303-219
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}

5.1.11.47.1 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

5.1.11.47.2 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

5.1.11.47.3 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

5.1.11.48 Tolerance_value

AIM element: dimensional_characteristic_representation
Source: ISO 10303-45

5.1.11.48.1 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

5.1.11.48.2 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

5.1.11.48.3 tolerance_value to tolerance_limit (as defined_value)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
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

5.1.11.48.4 tolerance_value to tolerance_range (as defined_value)

AIM element: PATH
Reference path: dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation

5.1.11.49 Tolerance_zone

AIM element: tolerance_zone
Source: ISO 10303-47

5.1.11.49.1 common_zone

AIM element: tolerance_zone.defining_tolerance
Source: ISO 10303-47

5.1.11.49.2 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

5.1.11.49.3 tolerance_zone to projection (as extended_shape)

AIM element: PATH
Reference path: tolerance_zone <-
tolerance_zone_definition.zone
tolerance_zone_definition =>
projected_zone_definition

5.1.11.49.4 tolerance_zone to tolerance_zone_definition (as zone_definition)

AIM element: PATH
Reference path: tolerance_zone <-
tolerance_zone_definition.zone
tolerance_zone_definition

5.1.11.50 Tolerance_zone_definition

AIM element: tolerance_zone_definition
Source: ISO 10303-47

5.1.11.50.1 tolerance_zone_definition to shape_element (as first_element)

AIM element: PATH
Reference path: tolerance_zone_definition
tolerance_zone_definition.boundaries[i] ->
shape_aspect

5.1.11.50.2 tolerance_zone_definition to shape_element (as second_element)

AIM element: PATH
Reference path: tolerance_zone_definition
tolerance_zone_definition.boundaries[i] ->
shape_aspect

5.1.11.51 Total_runout_tolerance

AIM element: total_runout_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.13)

5.1.11.51.1 runout

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
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]

5.1.11.51.2 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

5.1.11.52 Width_dimension

#1: if the length_dimension is specified and the optional path is specified

AIM element: dimensional_size_with_path
Source: ISO 10303-47
Reference path: dimensional_size_with_path<=
dimensional_size
dimensional_size.name='width'

#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='width'

5.1.12 part properties UoF

5.1.12.1 Descriptive_parameter

AIM element: descriptive_representation_item
Source: ISO 10303-45

5.1.12.1.1 descriptive_string

AIM element: descriptive_representation_item.description
Source: ISO 10303-45

5.1.12.2 Numeric_parameter

AIM element: measure_representation_item
Source: ISO 10303-45

5.1.12.2.1 parameter_unit

AIM element: named_unit
Source: ISO 10303-41
Rules:
Reference path: measure_representation_item <=
measure_with_unit
measure_with_unit.unit_component ->
unit
unit = named_unit
named_unit

5.1.12.2.2 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

5.1.12.3 Numeric_parameter_with_tolerance

AIM element: [measure_representation_item]
[qualified_representation_item]
Source: ISO 10303-45

5.1.12.3.1 numeric_parameter_with_tolerance to plus_minus_value (as implicit_tolerance)

AIM element: IDENTICAL MAPPING

5.1.12.3.2 numeric_parameter_with_tolerance to tolerance_limit (as implicit_tolerance)

AIM element: IDENTICAL MAPPING
Reference path: qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier

5.1.12.3.3 numeric_parameter_with_tolerance to limits_and_fits (as implicit_-tolerance)

AIM element: IDENTICAL MAPPING

5.1.12.3.4 numeric_parameter_with_tolerance to tolerance_range (as implicit_tolerance)

AIM element: IDENTICAL MAPPING

5.1.12.4 Property_parameter

AIM element: representation_item
Source: ISO 10303-43

5.1.12.4.1 parameter_name

AIM element: representation_item.name
Source: ISO 10303-43

5.1.13 program_run UoF

5.1.13.1 Dm_program_identification

AIM element: dm_program_identification
Source: ISO 10303-219
Reference path: dm_program_identification<=
action_resource_requirement

5.1.13.1.1 angular_units_dml

AIM element: resource_property.description
Source: ISO 10303-49
Reference path: dm_program_identification<=
action_resource_requirement <-
characterized_resource_definition= action_resource_requirement
characterized_resource_definition<-
resource_property.resource
resource_property
resource_property.name='angular units dml'
resource_property.description

5.1.13.1.2 linear_units_dml

AIM element: resource_property.description
Source: ISO 10303-49
Reference path: dm_program_identification<=
action_resource_requirement <-
characterized_resource_definition= action_resource_requirement
characterized_resource_definition<-
resource_property.resource
resource_property
resource_property.name='linear units dml'
resource_property.description

5.1.13.1.3 tolerance_standard_dml

AIM element: resource_property.description
Source: ISO 10303-49
Reference path: dm_program_identification<=
action_resource_requirement <-
characterized_resource_definition= action_resource_requirement
characterized_resource_definition<-
resource_property.resource
resource_property
resource_property.name='tolerance standard dml'
resource_property.description

5.1.13.1.4 identifier

AIM element: action_resource_requirement.name
Source: ISO 10303-49
Reference path: dm_program_identification<=
action_resource_requirement
action_resource_requirement.name

5.1.13.1.5 version

AIM element: action_resource_requirement.description
Source: ISO 10303-49
Reference path: dm_program_identification<=
action_resource_requirement
action_resource_requirement.description

5.1.13.1.6 dm_program_identification to person_in_organization (as program_custodian)

AIM element: PATH
Reference path: dm_program_identification<=
action_resource_requirement
person_and_organization_item= action_resource_requirement <-
person_and_organization_item<-
applied_person_and_organization_assignment.items
applied_person_and_organization_assignment<=
person_and_organization_assignment
{person_and_organization_assignment.role->
person_and_organization_role
person_and_organization_role.name='program custodian'}
person_and_organization_assignment.assigned_person_and_organization->
person_and_organization

5.1.13.1.7 dm_program_identification to organization (as program_-custodian)

AIM element: PATH
Reference path: dm_program_identification<=
action_resource_identification
organization_item= action_resource_identification <-
applied_organization_assignment.items
applied_organization_assignment<=
organization_assignment
{organization_assignment.role->
organization_role
organization_role.name='program custodian'}
organization_assignment.assigned_organization->
orgnization

5.1.13.2 Dm_program_run

AIM element: dm_program_run
Source: ISO 10303-219
Reference path: dm_program_run<=
action_method

5.1.13.2.1 part_inspection_status

AIM element: action_resource.description
Source: ISO 10303-41
Reference path: dm_program_run <=
action_method
supported_item=action_method<-
action_resource.usage
{action_resource.name='part inspection status'}
action_resource.description

5.1.13.2.2 url

AIM element: action_resource.description
Source: ISO 10303-41
Reference path: dm_program_run <=
action_method
supported_item=action_method<-
action_resource.usage
{action_resource.name='url'}
action_resource.description

5.1.13.2.3 dm_program_run to measurement_location (as run_location)

AIM element: PATH
Reference path: dm_program_run <=
action_method
characterized_action_definition=action_method<-
action_property.definition
action_property=>
measurement_location

5.1.13.2.4 dm_program_run to numeric_parameter (as measurement_-humidity)

AIM element: PATH
Reference path: dm_program_run <=
action_method
characterized_action_definition=action_method<-
action_property.definition
action_property<-
action_property_representation.property
action_property_representation
action_property_representation.representation->
representation
representation.items[i]->
representation_item=>
(representation_item.name='measurement humidity')
measure_representation_item<=
measure_with_unit

5.1.13.2.5 dm_program_run to numeric_parameter (as measurement_-temperature)

AIM element: PATH
Reference path: dm_program_run <=
action_method
characterized_action_definition=action_method<-
action_property.definition
action_property<-
action_property_representation.property
action_property_representation
action_property_representation.representation->
representation
representation.items[I]->
representation_item=>
(representation_item.name='measurement temerature')
measure_representation_item<=
measure_with_unit

5.1.13.2.6 dm_program_run to dm_program_identification (as program_id)

AIM element: PATH
Reference path: dm_program_run <=
action_method<-
characterized_action_definition=action_method
characterized_action_definition<-
action_resource_requirement.operations[i]
action_resource_requirement=>
dm_program_identification

5.1.13.2.7 dm_program_run to run_administrator (as run_administrator)

#1 administrator is organization

AIM element: PATH
Reference path: dm_program_run <=
action_method
organization_item= action_method <-
applied_organization_assignment.items
applied_organization_assignment<=
organization_assignment
{organization_assignment.role->
organization_role
organization_role.name='run administrator'}
organization_assignment.assigned_organization->
orgnization

#2 administrator is person_in_organization

AIM element: PATH
Reference path: dm_program_run <=
action_method
person_and_organization_item= action_method <-
person_and_organization_item<-
applied_person_and_organization_assignment.items
applied_person_and_organization_assignment<=
person_and_organization_assignment
{person_and_organization_assignment.role->
person_and_organization_role
person_and_organization_role.name='run administrator'}
person_and_organization_assignment.assigned_person_and_organization->
person_and_organization

5.1.13.2.8 dm_program_run to date_time (as run_start)

AIM element: PATH
Reference path: dm_program_run <=
action_method
date_item=action_method<-
applied_date_assignment.items[i]
applied_date_assignment<=
date_assignment
(date_assignment.role->
date_role
date_role.name='run start')
date_assignment.assigned_date->
date<-
date_and_time.date_component
date_and_time

5.1.13.2.9 dm_program_run to date_time (as run_end)

AIM element: PATH
Reference path: dm_program_run <=
action_method
date_item=action_method<-
applied_date_assignment.items[i]
applied_date_assignment<=
date_assignment
(date_assignment.role->
date_role
date_role.name='run end')
date_assignment.assigned_date->
date<-
date_and_time.date_component
date_and_time

5.1.13.2.10 dm_program_run to date_time (as run_date)

AIM element: PATH
Reference path: dm_program_run <=
action_method
date_item=action_method<-
applied_date_assignment.items[i]
applied_date_assignment<=
date_assignment
(date_assignment.role->
date_role
date_role.name='run date')
date_assignment.assigned_date->
date<-
date_and_time.date_component
date_and_time

5.1.13.2.11 dm_program_run to calendar_date (as run_date)

AIM element: PATH
Reference path: dm_program_run <=
action_method
date_item=action_method<-
applied_date_assignment.items[i]
applied_date_assignment<=
date_assignment
(date_assignment.role->
date_role
date_role.name='run date')
date_assignment.assigned_date->
date=>
calendar_date

5.1.13.2.12 dm_program_run to calendar_date (as run_end)

AIM element: PATH
Reference path: dm_program_run <=
action_method
date_item=action_method<-
applied_date_assignment.items[i]
applied_date_assignment<=
date_assignment
(date_assignment.role->
date_role\
date_role.name='run end')
date_assignment.assigned_date->
date=>
calendar_date

5.1.13.2.13 dm_program_run to calendar_date (as run_start)

AIM element: PATH
Reference path: dm_program_run <=
action_method
date_item=action_method<-
applied_date_assignment.items[i]
applied_date_assignment<=
date_assignment
(date_assignment.role->
date_role
date_role.name='run start')
date_assignment.assigned_date->
date=>
calendar_date

5.1.13.2.14 dm_program_run to numeric_parameter (as workpiece_-temperature)

AIM element: PATH
Reference path: dm_program_run <=
action_method
characterized_action_definition=action_method<-
action_property.definition
action_property<-
action_property_representation.property
action_property_representation
action_property_representation.representation->
representation
representation.items[i]->
representation_item=>
(representation_item.name='workpiece temerature')
measure_representation_item<=
measure_with_unit

5.1.13.3 Measurement_location

AIM element: measurement_location
Source: ISO 10303-219
Reference path: action_property=>
measurement_location

5.1.13.3.1 location

AIM element: action_property.name
Source: ISO 10303-49
Reference path: measurement_location
action_proeprty=>
action_property.name

5.1.13.3.2 machine

AIM element: action_property.description
Source: ISO 10303-49
Reference path: measurement_location
action_proeprty=>
action_property.description

5.1.13.4 Run_administrator

#1 administrator is organization

AIM element: applied_organization_assignment
Source: ISO 10303-219
Reference path: applied_organization_assignment<=
organization_assignment
organization_assignment.role->
organization_role
organization_role.name='run administrator'

#2 administrator is person_in_organization

AIM element: applied_person_and_organization_assignment
Source: ISO 10303-219
Reference path: applied_person_and_organization_assignment <=
person_and_organization_assignment
person_and_organization_assignment.role->
person_and_organization_role
person_and_organization_role.name='run administrator'

5.1.13.4.1 shift

#1 administrator is organization

AIM element: organization_role.description
Source: ISO 10303-219
Reference path: applied_organization_assignment<=
organization_assignment
organization_assignment.role->
organization_role
organization_role.name='shift'
organization_role.description

#2 administrator is person_in_organization

AIM element: person_and_organization _role.description
Source: ISO 10303-219
Reference path: applied_ person_and_organization_assignment <=
person_and_organization_assignment
person_and_organization_assignment.role->
person_and_organization_role
person_and_organization.name='shift'
person_and_organization.description

5.1.13.4.2 run_administrator to organization (as administrator)

#1 administrator is organization

AIM element: organization
Source: ISO 10303-219
Reference path: applied_organization_assignment<=
organization_assignment
organization_assignment.assigned_organization
organization

5.1.13.4.3 run_administrator to person_in_organization (as administrator)

#2 administrator is person_in_organization

AIM element: person_and_organization
Source: ISO 10303-219
Reference path: applied_ person_and_organization_assignment <=
person_and_organization_assignment
person_and_organization_assignment.assigned.person_and_organization
person_and_organization

5.1.14 shape_representation_for_machining UoF

5.1.14.1 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]}

5.1.14.2 Block_base_shape

AIM element: block_shape_representation
Source: ISO 10303-219
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: block_shape_representation <=
shape_representation_with_parameters <=
shape_representation

5.1.14.2.1 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}

5.1.14.2.2 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}

5.1.14.3 Brep_model

AIM element: manifold_solid_brep
Source: ISO 10303-42
Reference path: manifold_solid_brep
{manifold_solid_brep=>
faceted_brep}
{manifold_solid_brep=>
brep_with_voids}

5.1.14.4 Brep_model_element

AIM element: (geometric_representation_item)
(topological_representation_item)
Source: ISO 10303-42

5.1.14.4.1 Brep_model_element to Brep_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

5.1.14.5 Brep_shape_aspect_representation

AIM element: advanced_brep_shape_representation
Source: ISO 10303-514
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: advanced_brep_shape_representation <=
shape_representation

5.1.14.5.1 Brep_shape_aspect_representation to Brep_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

5.1.14.6 Brep_shape_representation

AIM element: advanced_brep_shape_representation
Source: ISO 10303-514
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: advanced_brep_shape_representation <=
shape_representation

5.1.14.6.1 Brep_shape_representation to Brep_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

5.1.14.7 Cartesian_coordinate_space

AIM element: geometric_representation_context
Source: ISO 10303-42

5.1.14.7.1 units

AIM element: geometric_representation_context.coordinate_space_dimension
Source: ISO 10303-42

5.1.14.8 Cartesian_point

AIM element: cartesian_point
Source: ISO 10303-42

5.1.14.8.1 x

AIM element: cartesian_point.coordinates[1]
Source: ISO 10303-42

5.1.14.8.2 y

AIM element: cartesian_point.coordinates[2]
Source: ISO 10303-42

5.1.14.8.3 z

AIM element: cartesian_point.coordinates[3]
Source: ISO 10303-42

5.1.14.9 Cartesian_vector

AIM element: direction
Source: ISO 10303-42

5.1.14.9.1 i

AIM element: direction.direction_ratios[1]
Source: ISO 10303-42

5.1.14.9.2 j

AIM element: direction.direction_ratios[2]
Source: ISO 10303-42

5.1.14.9.3 k

AIM element: direction.direction_ratios[3]
Source: ISO 10303-42

5.1.14.10 Cylindrical_base_shape

AIM element: cylindrical_shape_representation
Source: ISO 10303-219
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: cylindrical_shape_representation <=
shape_representation_with_parameters <=
shape_representation

5.1.14.10.1 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}

5.1.14.11 Derived_shape_element

AIM element: derived_shape_aspect
Source: ISO 10303-47

5.1.14.11.1 role

AIM element: shape_aspect.description
Source: ISO 10303-47
Reference path: derived_shape_aspect<=
shape_aspect
shape_aspect.description

5.1.14.11.2 derived_shape_element to shape_element (as is_derived_from)

AIM element: PATH
Reference path: derived_shape_aspect<-
shape_aspect_deriving_relationship.relating_shape_aspect
shape_aspect_deriving_relationship
shape_aspect_deriving_relationship.related_shape_aspect
shape_aspect

5.1.14.12 Direction_element

AIM element: direction_shape_representation
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference path: direction_shape_representation <=
shape_representation<=
representation
{representation.items[i]->
representation_item=>
geometric_representation_item=>
direction}

5.1.14.13 Explicit_base_shape_representation

AIM element: shape_representation
Source: ISO 10303-41
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)

5.1.14.13.1 explicit_base_shape_representation to Brep_shape_representation (as Brep_form)

AIM element: IDENTICAL MAPPING

5.1.14.14 Face_shape_element

AIM element: face_shape_representation
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference path: face_shape_representation <=
shape_representation<=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
(face_surface)
(oriented_face)

5.1.14.15 Face_shape_element_relationship

AIM element: face_shape_representation_relationship
Source: ISO 10303-219
Reference path: face_shape_representation_relationship<=
representation_relationship

5.1.14.15.1 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

5.1.14.15.2 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

5.1.14.16 Geometric_model

AIM element: shape_representation
Source: ISO 10303-41
Reference path: shape_representation<=
representation

5.1.14.16.1 accuracy

AIM element: descriptive_representation_item.description
Source: ISO 10303-41
Reference path: shape_representation<=
representation
representation.items[i]->
representation_item<=
{representation_item.name='accuracy'}
descriptive_representation_item
descriptive_representation_item.description

5.1.14.16.2 description

AIM element: representation.description
Source: ISO 10303-41
Reference path: shape_representation<=
representation
representation.description

5.1.14.16.3 model_extent

AIM element: representation.name
Source: ISO 10303-41
Reference path: shape_representation<=
representation
representation.name

5.1.14.16.4 model_id

AIM element: representation.id
Source: ISO 10303-41
Reference path: shape_representation<=
representation
representation.id

5.1.14.16.5 role

AIM element: description_attribute.attributer_value
Source: ISO 10303-41
Reference path: shape_representation<=
representation
description_attribute_select=representation
description_attribute_select<-
description_attribute.described_item
description_attribute
description_attribute.attribute_value

5.1.14.16.6 version_id

AIM element: id_attribute.attributer_value
Source: ISO 10303-41
Reference path: shape_representation<=
representation
id_attribute_select=representation
id_attribute_select<-
id_attribute.identified_item
id_attribute
id_attribute.attribute_value

5.1.14.16.7 geometric_model to Cartesian_coordinate_space (as is_definied_-in)

AIM element: PATH
Reference path: shape_representation<=
representation
representation.context_of_items
representation_context=>
geometric_representation_context

5.1.14.17 Implicit_base_shape_representation

AIM element: shape_representation_with_parameters
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: shape_representation_with_parameters <=
shape_representation

5.1.14.17.1 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}

5.1.14.17.2 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

5.1.14.18 Location_element

AIM element: location_shape_representation
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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}

5.1.14.19 Ngon_base_shape

AIM element: ngon_shape_representation
Source: ISO 10303-219
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: ngon_shape_representation <=
shape_representation_with_parameters <=
shape_representation

5.1.14.19.1 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')

5.1.14.19.2 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}

5.1.14.19.3 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}

5.1.14.19.4 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}

5.1.14.20 Offset_shape_element

AIM element: derived_shape_aspect
Source: ISO 10303-47
Reference path: derived_shape_aspect<=
shape_aspect
shape_aspect.name='offset shape element'

5.1.14.20.1 offset_shape_element to Numeric_parameter (as offset)

AIM element: PATH
Reference path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition<-
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

5.1.14.20.2 offset_shape_element to direction_element (as offset_direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition=>
product_definition_shape}
represented_definition=property_definition
represented_definition<-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name= 'offset direction'}
representation =>
shape_representation =>
direction_shape_representation

5.1.14.21 Orientation

AIM element: placement
Source: ISO 10303-42

5.1.14.21.1 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))

5.1.14.21.2 location

AIM element: placement.location
Source: ISO 10303-42

5.1.14.22 Path_element

AIM element: path_shape_representation
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.17)
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)

5.1.14.23 Planar_element

AIM element: planar_shape_representation
Source: ISO 10303-522
Reference path: planar_shape_representation <=
shape_representation<=
representation

5.1.14.23.1 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

5.1.14.23.2 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

5.1.14.24 Shape

AIM element: product_definition_shape
Source: ISO 10303-41

5.1.14.24.1 shape to base_shape (as base_shape_definition)

AIM element: PATH
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

5.1.14.24.2 shape to Brep_shape_representation (as Brep_form)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: product_definition_shape <=
property_definition
represented_definition=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

5.1.14.24.3 shape to shape_aspect (as element)

AIM element: PATH
Reference path: product_definition_shape <-
shape_aspect.of_shape
shape_aspect

5.1.14.25 Shape_aspect

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)

5.1.14.25.1 shape_aspect to Brep_model_element (as Brep_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
represented_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)

5.1.14.25.2 shape_aspect to Brep_shape_aspect_representation (as Brep_-form)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.9)
representation_subtype_exclusiveness – (See 5.2.4.17)
shape_representation_subtype_exclusiveness – (See 5.2.4.22)
Reference path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation =>
advanced_brep_shape_representation

5.1.14.25.3 shape_aspect to shape_element (as element)

AIM element: IDENTICAL MAPPING

5.1.14.26 Shape_element

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.20)
Reference path: {shape_aspect
shape_aspect.product_definitional = 'TRUE'}
(shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition
represented_definition=property_definition<-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation)

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 Dimensional_Inspection_Schema;
USE FROM aic_machining_feature;                   -- ISO 10303-522
USE FROM aic_geometric_tolerances;                -- ISO 10303-519

USE FROM aic_advanced_brep;                       -- ISO 10303-514

USE FROM action_schema                            -- ISO 10303-41
   (action,
    action_method,
    action_resource,
    action_relationship,
    executed_action);
USE FROM application_context_schema              -- ISO 10303-41
      (application_context,
       application_context_element,
       application_protocol_definition,
       product_definition_context);

USE FROM date_time_schema                        -- ISO 10303-41
      (calendar_date,
       coordinated_universal_time_offset,
       date,
       date_and_time,
       date_role,
       date_time_select,
       local_time);

USE FROM document_schema                              -- ISO 10303-41
       (document,
        document_usage_constraint,
        document_with_class);
  REFERENCE FROM geometry_schema                       -- ISO 10303-42
       (dummy_gri);

USE FROM effectivity_schema                            -- ISO 10303-41
      (effectivity,
      dated_effectivity,
      lot_effectivity,
      serial_numbered_effectivity);

USE FROM geometry_schema                                -- ISO 10303-42
      (axis1_placement,
      axis2_placement_3d,
      b_spline_curve_with_knots,
```



```
     bezier_curve,
     bezier_surface,
     cartesian_point,
     circle,
     elementary_surface,
     ellipse,
     geometric_representation_item,
     hyperbola,
     line,
     oriented_surface,
     parabola,
     placement,
     plane,
     point,
     quasi_uniform_curve,
     quasi_uniform_surface,
     surface);
USE FROM geometric_model_schema                        -- ISO 10303-42
     (faceted_brep,
      manifold_solid_brep);
USE FROM process_property_representation_schema        -- ISO 10303-49
     (action_property_representation);
USE FROM material_property_definition_schema           -- ISO 10303-45
     (property_definition_relationship);

USE FROM group_schema                                  -- ISO 10303-41
(group,
 group_relationship);

USE FROM management_resources_schema                   -- ISO 10303-41
     (action_assignment,
     date_assignment,
     document_reference,
     document_usage_constraint_assignment,
     effectivity_assignment,
     group_assignment,
     organization_assignment,
     person_and_organization_assignment,
     security_classification_assignment);

 USE FROM measure_schema                                -- ISO 10303-41
      (amount_of_substance_measure,
      area_measure,
      celsius_temperature_measure,
      conversion_based_unit,
      context_dependent_unit,
      count_measure,
      derived_unit,
      electric_current_measure,
      global_unit_assigned_context,
      length_measure,
      length_measure_with_unit,
      length_unit,
      luminous_intensity_measure,
      mass_measure_with_unit,
      mass_unit,
```

```
      mass_measure,
      measure_value,
      measure_with_unit,
      unit,
      named_unit,
      plane_angle_measure,
      plane_angle_measure_with_unit,
      plane_angle_unit,
      positive_plane_angle_measure,
      ratio_measure,
      ratio_unit,
      si_unit,
      solid_angle_measure,
      solid_angle_unit,
      time_measure,
      time_measure_with_unit,
      time_unit,
      thermodynamic_temperature_measure,
      volume_measure);

USE FROM person_organization_schema                  -- ISO 10303-41
     (address,
      organization,
      organization_role,
      organizational_address,
      organizational_project,
      person,
      person_and_organization,
      person_organization_select,
      personal_address);

USE FROM process_property_schema                      -- ISO 10303-49
     (action_property,
      action_resource_requirement,
      resource_property);

  USE FROM product_definition_schema                  -- ISO 10303-41
      (product,
       product_definition,
       product_definition_formation);
USE FROM product_property_representation_schema       -- ISO 10303-41
     (property_definition_representation,
      represented_definition,
      shape_definition_representation,
      shape_representation);
USE FROM qualified_measure_schema                     -- ISO 10303-45
     (descriptive_representation_item,
      expanded_uncertainty,
      measure_qualification,
      measure_representation_item,
      precision_qualifier,
      qualified_representation_item,
      qualitative_uncertainty,
      standard_uncertainty,
      type_qualifier,
      uncertainty_qualifier,
```



```
      value_qualifier);

USE FROM representation_schema                           -- ISO 10303-43
     (compound_representation_item,
      global_uncertainty_assigned_context,
      representation_item,
      set_representation_item,
      representation,
      representation_relationship,
      uncertainty_measure_with_unit);
USE FROM security_classification_schema                    -- ISO 10303-41
     (security_classification,
      security_classification_level);

  USE FROM shape_dimension_schema                          -- ISO 10303-47
      (angular_location,
       angular_size,
       dimensional_characteristic_representation,
       dimensional_location,
       dimensional_location_with_path,
       dimensional_size,
       dimensional_size_with_path,
       shape_dimension_representation);
USE FROM shape_tolerance_schema                            -- ISO 10303-47
     (geometric_tolerance,
      limits_and_fits,
      plus_minus_tolerance,
      tolerance_value,
      tolerance_zone);
  REFERENCE FROM topology_schema                           -- ISO 10303-42
      (dummy_tri);
USE FROM topology_schema                                   -- ISO 10303-42
     (closed_shell,
      connected_face_set,
      edge,
      edge_loop,
      face,
      face_surface,
      open_shell,
      poly_loop,
      shell,
      topological_representation_item);
(*
```

NOTE The schemas referenced above can be found in the following parts of ISO 10303:

aic_machining_feature	-- ISO 10303-522
aic_geometric_tolerances	-- ISO 10303-519
aic_advanced_brep	-- ISO 10303-514
action_schema	-- ISO 10303-41
application_context_schema	-- ISO 10303-41
date_time_schema	-- ISO 10303-41
document_schema	-- ISO 10303-41
effectivity_schema	-- ISO 10303-41
geometry_schema	-- ISO 10303-42
geometric_model_schema	-- ISO 10303-42
group_schema	-- ISO 10303-41
process_property_schema	-- ISO 10303-49
material_property_definition_schema	-- ISO 10303-45

management_resources_schema	-- ISO 10303-41
measure_schema	-- ISO 10303-41
person_organization_schema	-- ISO 10303-41
process_property_schema	-- ISO 10303-49
product_definition_schema	-- ISO 10303-41
product_property_representation_schema	-- ISO 10303-41
qualified_measure_schema	-- ISO 10303-45
representation_schema	-- ISO 10303-43
security_classification_schema	-- ISO 10303-41
shape_dimension_schema	-- ISO 10303-47
shape_tolerance_schema	-- ISO 10303-47
topology_schema	-- ISO 10303-42

5.2.1 Fundamental concepts and assumptions

This part of ISO 10303 is intended for representing and exchanging the results and analyses of dimensional inspection activities. Its fundamental addition to ISO 10303 is the set of dimensional inspection features that characterize the organization of information required by metrologists and others who operate dimensional measurement equipment and use dimensional measurement data. Accordingly, the set of dimensional inspection features included here was developed in consultation with experts of the committee developing ISO 22093 on the Dimensional Measuring Interface Standard, experts of the Dimensional Markup Language Project operating under the Automotive Industry Action Group, and other dimensional measurement experts.

5.2.2 Dimensional inspection schema types

5.2.2.1 action_item

An **action_item** identifies a **product_definition** to which a referenced **action** may be assigned.

<u>EXPRESS specification:</u>

```
*)
TYPE action_item = SELECT
      (product_definition);
END_TYPE;
(*
```

5.2.2.2 date_item

A **date_item** identifies a **dm_program_run** to which a referenced **date** may be assigned.

<u>EXPRESS specification:</u>

```
*)
TYPE date_item = SELECT
     (dm_program_run);
END_TYPE;
(*
```

5.2.2.3 document_reference_item

A **document_reference_item** identifies a **product_definition_formation**, **externally_defined_-feature_definition**, or an **externally_defined_dimension_definition** to which a referenced **document** may be assigned.

EXPRESS specification:

```
*)
TYPE document_reference_item = SELECT
     (product_definition_formation,
       externally_defined_feature_definition,
       externally_defined_dimension_definition);
END_TYPE;
(*
```

5.2.2.4 effectivity_item

An **effectivity_item** identifies a **product_definition_formation** to which a referenced **effectivity** may be assigned.

EXPRESS specification:

```
*)
TYPE effectivity_item = SELECT
      (product_definition_formation);
END_TYPE;
(*
```

5.2.2.5 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;
(*
```

5.2.2.6 organization_item

An **organization_item** identifies a **dm_program_identification** or **dm_proram_run** to which a referenced **organization** may be assigned.

EXPRESS specification:

```
*)
TYPE organization_item = SELECT
     (dm_program_identification,
       dm_program_run);
END_TYPE;
(*
```

5.2.2.7 person_and_organization_item

A **person_and_organization_item** identifies a **dm_program_identification** or **dm_proram_run** to which a referenced **person_and_organization** may be assigned.

EXPRESS specification:

```
*)
TYPE person_and_organization_item = SELECT
     (dm_program_identification,
       dm_program_run);
END_TYPE;
(*
```

5.2.2.8 security_classification_item

A **security_classification_item** identifies a **product_definition_formation** to which a referenced **security_classification_item** may be assigned.

EXPRESS specification:

```
*)
TYPE security_classification_item = SELECT
 (product_definition_formation);
END_TYPE;
(*
```

5.2.3 Dimensional inspection schema entities

5.2.3.1 Dimensional inspection schema entity definitions

5.2.3.1.1 applied_action_assignment

An **applied_action_assignment** specifies those **action_item**s to which a **action** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_action_assignment
      SUBTYPE OF (action_assignment);
      items: SET[1:?] OF action_item;
END_ENTITY;
 (*
```

Attribute definition:

items: the set of **action_item**s to which an **action** is assigned.

5.2.3.1.2 applied_date_assignment

An **applied_date_assignment** specifies those **date_item**s to which a **date** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_date_assignment
      SUBTYPE OF (date_assignment);
      items: SET[1:?] OF date_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **date_item**s to which an **date** is assigned.

5.2.3.1.3 applied_document_reference

An **applied_document_reference** specifies those **document_reference_item**s to which a **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 **document** is assigned.

5.2.3.1.4 applied_document_usage_constraint_assignment

An **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_effectivity_assignment

An **applied_effectivity_assignment** specifies those **effectivity_item**s to which a **effectivity** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_effectivity_assignment
      SUBTYPE OF (effectivity_assignment);
      items: SET[1:?] OF effectivity_item;
END_ENTITY;
 (*
```

Attribute definition:

items: the set of **effectivity_item**s to which an **effectivity** 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 applied_organization_assignment

An **applied_organization_assignment** specifies those **organization_item**s to which an **organization** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_organization_assignment
      SUBTYPE OF (organization_assignment);
      items: SET[1:?] OF organization_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **organization_item**s to which an **organization** is assigned.

5.2.3.1.8 applied_person_and_organization_assignment

An **applied_person_and_organization_assignment** specifies those **person_and_organization_item**s to which a **person_and_organization** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_person_AND_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF person_and_organization_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **person_and_organization_item**s to which a **person_and_organization** is assigned.

5.2.3.1.9 applied_security_classification_assignment

An **applied_security_classification_assignment** specifies those **security_classification_item**s to which an **security_classification** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
  items : SET [1:?] OF security_classification_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **person_and_organization_item**s to which a **person_and_organization** is assigned.

5.2.3.1.10 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 clause 4.2.9 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 |
           (('DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT'
            IN TYPEOF(it)) AND (it.name = 'orientation')))) = 1);
      wr3: (SIZEOF(QUERY (it <* SELF.items |
           ((SIZEOF(
            ['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'length')))) = 1);
      wr4: (SIZEOF(QUERY (it <* SELF.items |
           ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'width')))) = 1);
      wr5: (SIZEOF(QUERY (it <* SELF.items |
           ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'height')))) = 1);
  END_ENTITY; -- block_shape_representation

(*
```

Formal propositions:

WR1: The **block_shape_representation** shall contain exactly four **representation_item**s in its set of **items**.

WR2: One of the **representation_item**s used for the implicit representation of a **block_shape_-representation** shall be of type **placement** with a **name** of 'orientation'.

WR3: One of the **representation_item**s used for the implicit representation of a **block_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

WR4: One of the **representation_item**s used for the implicit representation of a **block_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

WR5: One of the **representation_item**s used for the implicit representation of a **block_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'height'.

Informal propositions:

IP1: The **block_shape_representation** shall be defined at the center of the rectangular area in the X-Y plane with the width of the rectangle in the X direction, the height of the rectangle in the Y direction, and the length of the rectangle in the Z direction.

5.2.3.1.11 cylindrical_shape_representation

A **cylindrical_shape_representation** specifies representation of a shape that is a cylindrical volume defined as a circular area of a defined length. The enclosed area is defined by a circle with a specified radius.

EXPRESS specification:

```
*)
  ENTITY cylindrical_shape_representation
    SUBTYPE OF (shape_representation_with_parameters);
    WHERE
      wr1: (SIZEOF(SELF.items) = 3);
      wr2: (SIZEOF(QUERY (it <* SELF.items | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT'
           IN TYPEOF(it)) AND (it.name = 'orientation')))) = 1);
      wr3: (SIZEOF(QUERY (it <* SELF.items | ((SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
           'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'length')))) = 1);
      wr4: (SIZEOF(QUERY (it <* SELF.items | ((SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
           'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'diameter')))) = 1);
  END_ENTITY; -- cylindrical_shape_representation
(*
```

Formal propositions:

WR1: The **cylindrical_shape_representation** shall contain exactly three **representation_item**s in its set of **items**.

WR2: One of the **representation_item**s used for the implicit representation of a **cylindrical_shape_-representation** shall be of type **placement** with a **name** of 'orientation'.

WR3: One of the **representation_item**s used for the implicit representation of a **cylindrical_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

WR4: One of the **representation_item**s used for the implicit representation of a **cylindrical_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'diameter'.

Informal propositions:

IP1: The location of the **cylindrical_shape_representation** shall be defined to be at the center of the circle that defines the cylinder.

IP2: The **cylindrical_shape_representation** shall be definied by forming a circular profile in the X-Y plane, and the length along the z direction.

5.2.3.1.12 dimensional_measurement_representation

A **dimensional_measurement_representation** specifies a type of **representation** that defines tolerance analysis and feature analysis parameters.

EXPRESS specification:

```
*)

ENTITY dimensional_measurement_representation
     SUBTYPE OF (representation);
END_ENTITY;
(*
```

5.2.3.1.13 directed_dimensional_location

A **directed_dimensional_location** specifies a type of **dimensional_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.

5.2.3.1.14 dm_analysis_dofs_dml

The dm_analysis_dofs_dml specifies a type of dimensional_measurement_representation that definies degrees of freedom parameters.

EXPRESS specification:

```
*)

ENTITY dm_analysis_dofs_dml
     SUBTYPE OF (dimensional_measurement_representation);
 WHERE

  WR1: SIZEOF(QUERY( repi <* SELF.items |((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(repi)) = 2)))) = 6;

(* transx  *)

 wr2: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['transx upper limit', 'transx lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;(* transy  *)

 wr3: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['transy upper limit', 'transy lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) ))<=1 )) )<= 1;
```

```
(* transz  *)

 wr4: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['transz upper limit', 'transz lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;

(* rotx  *)

 wr5:SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['rotx upper limit', 'rotx lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;

(* roty  *)

 wr6: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['roty upper limit', 'roty lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;
```

```
(* rotz  *)

 wr7: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['rotz upper limit', 'rotz lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;
END_ENTITY;

(*
```

Formal propositions:

WR1: there shall be exactly six implicit representation of a of type **measure_representation_item** and **qualified_measure_with_unit**

WR2: Zero or one of the **representation_item**s used for the implicit representation of a **dm_analysis_-dofs_dml** shall be of type **measure_representation_item** with a **name** of 'transx upper limit' or 'transx lower limit'that has zero or one references through the **qualifiers** attribute a **type_qualifier** of **name** 'degree of freedom limited', 'degree of freedom not allowed' or 'degree of freedom allowed'.

WR3: Zero or one of the **representation_item**s used for the implicit representation of a **dm_analysis_-dofs_dml** shall be of type **measure_representation_item** with a **name** of 'transy upper limit' or 'transy lower limit'that has zero or one references through the **qualifiers** attribute a **type_qualifier** of **name** 'degree of freedom limited', 'degree of freedom not allowed' or 'degree of freedom allowed'.

WR4: Zero or one of the **representation_item**s used for the implicit representation of a **dm_analysis_-dofs_dml** shall be of type **measure_representation_item** with a **name** of 'transz upper limit' or 'transz lower limit'that has zero or one references through the **qualifiers** attribute a **type_qualifier** of **name** 'degree of freedom limited', 'degree of freedom not allowed' or 'degree of freedom allowed'.

WR5: Zero or one of the **representation_item**s used for the implicit representation of a **dm_analysis_-dofs_dml** shall be of type **measure_representation_item** with a **name** of 'rotx upper limit' or 'rotx lower limit'that has zero or one references through the **qualifiers** attribute a **type_qualifier** of **name** 'degree of freedom limited', 'degree of freedom not allowed' or 'degree of freedom allowed'.

WR6: Zero or one of the **representation_item**s used for the implicit representation of a **dm_analysis_-dofs_dml** shall be of type **measure_representation_item** with a **name** of 'roty upper limit' or 'roty lower limit'that has zero or one references through the **qualifiers** attribute a **type_qualifier** of **name** 'degree of freedom limited', 'degree of freedom not allowed' or 'degree of freedom allowed'.

WR7: Zero or one of the **representation_item**s used for the implicit representation of a **dm_analysis_-dofs_dml** shall be of type **measure_representation_item** with a **name** of 'rotz upper limit' or 'rotz lower limit'that has zero or one references through the **qualifiers** attribute a **type_qualifier** of **name** 'degree of freedom limited', 'degree of freedom not allowed' or 'degree of freedom allowed'.

5.2.3.1.15 dmf_arc

A **dmf_arc** is a type of **dm_feature_definition** that defines an arc dimensional feature (see 4.2.91).

EXPRESS specification:

```
*)
ENTITY dmf_arc
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(* dmf_arc to dm_point_parameter (as center_point)*)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_arc to dm_vector_parameter (as start_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start vector')
      )) =1 )) )) =1 )) ) =0));
```



```
(* dmf_arc to dm_vector_parameter (as end_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='end vector')
      )) =1 )) )) =1 )) ) =0));

(* dmf_arc to dm_vector_parameter (as axis_direction_vector) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

(* dmf_arc to dm_dimension_parameter (as radius) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='radius')
      )) =1 )) )) <=1 )) ) =0));
```

```
(* inner_outer *)

 WR7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_arc
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_arc** shall contain exactly two **representation_item**s.

WR2: The **dmf_arc** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_-definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'center point'.

WR3: The **dmf_arc** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_-definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'start vector'.

WR4: The **dmf_arc** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_-definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'end vector'.

WR5: The **dmf_arc** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_-definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'axis direction vector'.

WR6: The **dmf_arc** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_-definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'radius'.

WR7: Exactly one **representation_item** used for the implicit representation of the **dmf_arc** shall be of type **descriptive_representation_item** with a **name** of 'inner or outer' and a **description** of either 'inner' or 'outer'.

5.2.3.1.16 dmf_circle

A **dmf_circle** is a type of **dm_feature_definition** that defines an circle dimensional feature (see 4.2.92).

EXPRESS specification:

```
*)
  ENTITY dmf_circle
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*  dmf_circle to dm_point_parameter (as center_point)  *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));
```

```
 (*  dmf_circle to dm_vector_parameter (as axis_direction_vector)  *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

 (*  dmf_circle to dm_dimension_parameter (as diameter)  *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='diameter')
      )) =1 )) )) <=1 )) ) =0));

(* inner_outer *)

 WR5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_circle
 (*
```


Formal propositions:

WR1: The implicit representation of the **dmf_circle** shall contain exactly two **representation_item**s.

WR2: The **dmf_circle** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'center point'.

WR3: The **dmf_circle** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'axis direction vector'.

WR4: The **dmf_circle** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'diameter'.

WR5: Exactly one **representation_item** used for the implicit representation of the **dmf_circle** shall be of type **descriptive_representation_item** with a **name** of 'inner or outer' and a **description** of either 'inner' or 'outer'.

5.2.3.1.17 dmf_cone

A **dmf_cone** is a type of **dm_feature_definition** that defines an cone dimensional feature (see 4.2.93).

EXPRESS specification:

```
*)
ENTITY dmf_cone
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;
```

```
(*  dmf_cone to dm_point_parameter (as start_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start point')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_point_parameter (as end_point) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='end point')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_point_parameter (as apex_point) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='apex point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cone to dm_vector_parameter (as start_vector) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start vector')
      )) =1 )) )) <=1 )) ) =0));
```



```
(*  dmf_cone to dm_vector_parameter (as end_vector) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='end vector')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_vector_parameter (as axis_direction_vector) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cone to dm_dimension_parameter (as diameter) *)

 wr8: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='diameter')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cone to dm_dimension_parameter (as start_length_dml) *)

 wr9: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start length dml')
      )) =1 )) )) =1 )) ) =0));
```

```
(*  dmf_cone to dm_dimension_parameter (as end_length_dml) *)

 wr10: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
       IN TYPEOF(pdr.used_representation)) AND
       (SIZEOF(QUERY( na <* USEDIN(pdr,
       'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
       (na.attribute_value='end length dml')
       )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_dimension_parameter (as included_angle) *)

 wr11: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
       IN TYPEOF(pdr.used_representation)) AND
       (SIZEOF(QUERY( na <* USEDIN(pdr,
       'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
       (na.attribute_value='included angle')
       )) =1 )) )) =1 )) ) =0));

(* inner_outer *)


 WR12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.'+
       'SHAPE_REPRESENTATION_WITH_PARAMETERS')
       IN TYPEOF(pdr.used_representation)) ) |
       (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND
       (it.name = 'inner or outer')) AND
       ((it.description = 'inner') OR
       (it.description = 'outer')))
       )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_cone
 (*
```


Formal propositions:

WR1: The implicit representation of the **dmf_cone** shall contain exactly two **representation_item**s.

WR2: The **dmf_cone** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'start point'.

WR3: The **dmf_cone** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value off 'end point'.

WR4: The **dmf_cone** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'apex point'.

WR5: The **dmf_cone** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'start vector'.

WR6: The **dmf_cone** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'end vector'.

WR7: The **dmf_cone** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'axis direction vector'.

WR8: The **dmf_cone** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'diameter'.

WR9: The **dmf_cone** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'start length dml'.

WR10: The **dmf_cone** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'end length dml'.

WR11: The **dmf_cone** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'included angle'.

WR12: Exactly one **representation_item** used for the implicit representation of the **dmf_cone** shall be of type **descriptive_representation_item** with a **name** of 'inner or outer' and a **description** of either 'inner' or 'outer'.

5.2.3.1.18 dmf_cylinder

A **dmf_cylinder** is a type of **dm_feature_definition** that defines an cylinder dimensional feature (see 4.2.93).

EXPRESS specification:

```
*)
ENTITY dmf_cylinder
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;
```



```
(*  dmf_cylinder to dm_point_parameter (as center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cylinder to dm_point_parameter (as start_point) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cylinder to dm_point_parameter (as end_point) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='end point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cylinder to dm_point_parameter (as axis direction vector) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));
```

```
(*  dmf_cylinder to dm_dimension_parameter (as diameter) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='diameter')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cylinder to dm_dimension_parameter (as length_dml) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='length dml')
      )) =1 )) )) <=1 )) ) =0));

(* inner_outer *)

 WR8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_cylinder
(*
```

Formal propositions:

WR1: The implicit representation of the **dmf_cylinder** shall contain exactly two **representation_item**s.

WR2: The **dmf_cylinder** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'center point'.

WR3: The **dmf_cylinder** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'start point'.

WR4: The **dmf_cylinder** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'end point'.

WR5: The **dmf_cylinder** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'axis direction vector'.

WR6: The **dmf_cylinder** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'diameter'.

WR7: The **dmf_cylinder** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'length dml'.

WR8: Exactly one **representation_item** used for the implicit representation of the **dmf_cylinder** shall be of type **descriptive_representation_item** with a **name** of 'inner or outer' and a **description** of either 'inner' or 'outer'.

5.2.3.1.19 dmf_edge_point

A **dmf_edge_point** is a type of **dm_feature_definition** that defines edge point dimensional feature (see 4.2.95).

EXPRESS specification:

```
*)
ENTITY dmf_edge_point
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

 (*  dmf_edge_point to dm_point_parameter (location_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='location point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_edge_point to dm_vector_parameter (edge_normal_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='edge normal vector')
      )) =1 )) )) =1 )) ) =0));
```



```
(*  dmf_edge_point to dm_vector_parameter (surface_normal_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='surface normal vector')
      )) =1 )) )) =1 )) ) =0));

(*  point_type_dml*)

 WR5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'point type dml')) AND
      ((it.description = 'h edge') OR
      (it.description = 't edge')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_edge_point
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_edge_point** shall contain exactly one **representation_item**s.

WR2: The **dmf_edge_point** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'location point'.

WR3: The **dmf_edge_point** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'edge normal vector'.

WR4: The **dmf_edge_point** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'surface normal vector'.

WR5: There shall be at most one implicit representation of the **dmf_edge_point** shall be of type **descriptive_representation_item** with a **name** of 'point type dml' and a **description** of either 'h edge' or 't edge'.

5.2.3.1.20 dmf_ellipse

A **dmf_ellipse** is a type of **dm_feature_definition** that defines ellipse dimensional feature (see 4.2.96).

<u>EXPRESS specification:</u>

```
*)
ENTITY dmf_ellipse
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*   dmf_ellipse to dm_point_parameter (as center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));
```



```
(*   dmf_ellipse to dm_point_parameter (as focus_point_one_dml) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='focus point one dml')
      )) =1 )) )) <=1 )) ) =0));

(*   dmf_ellipse to dm_point_parameter (as focus_point_two_dml) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='focus point two dml')
      )) =1 )) )) <=1 )) ) =0));

(*   dmf_ellipse to dm_vector_parameter (as major_axis_vector) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='major axis vector')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_ellipse to dm_vector_parameter (as normal_direction) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='normal direction')
      )) =1 )) )) =1 )) ) =0));
```

```
(*   dmf_ellipse to dm_dimension_parameter (as minor_diameter) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='minor diameter')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_ellipse to dm_dimension_parameter (as major_diameter) *)

 wr8: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='major diameter')
      )) =1 )) )) =1 )) ) =0));

(* inner_outer *)

 WR9: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;
  END_ENTITY; -- dmf_ellipse
 (*
```


Formal propositions:

WR1: The implicit representation of the **dmf_ellipse** shall contain exactly two **representation_item**s.

WR2: The **dmf_ellipse** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'center point'.

WR3: The **dmf_ellipse** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'focus point one dml'.

WR4: The **dmf_ellipse** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'focus point two dml'.

WR5: The **dmf_ellipse** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'major axis vector'.

WR6: The **dmf_ellipse** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'normal direction'.

WR7: The **dmf_ellipse** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'minor diameter'.

WR8: The **dmf_ellipse** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'major diameter'.

WR9: Exactly one **representation_item** used for the implicit representation of the **dmf_ellipse** shall be of type **descriptive_representation_item** with a **name** of 'inner or outer' and a **description** of either 'inner' or 'outer'.

5.2.3.1.21 dmf_generic_feature

A **dmf_generic_feature** is a type of **dm_feature_definition** that defines generic dimensional feature (see 4.2.97).

EXPRESS specification:

```
*)
ENTITY dmf_generic_feature
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*  dmf_generic_feature to dm_point_parameter (as center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_generic_feature  dm_vector_parameter (vector_from_center_of_object) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='vector from center of object')
      )) =1 )) )) =1 )) ) =0));
```



```
(*  dmf_generic_feature to dm_vector_parameter (as secondary_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='secondary vector')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_generic_feature to dm_dimension_parameter (parameters) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='parameter')
      )) =1 )) )) <=1 )) ) =0));

(* description *)

 WR6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'description')))
      )) <= 1)) )) = 0)) )) = 0;
END_ENTITY; -- dmf_generic_feature

(*
```

Formal propositions:

WR1: The implicit representation of the **dmf_generic_feature** shall contain exactly two **representation_item**s.

WR2: The **dmf_generic_feature** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'center point'.

WR3: The **dmf_generic_feature** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'vector from center of object'.

WR4: The **dmf_generic_feature** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'secondary vector'.

WR5: The **dmf_generic_feature** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'parameters'.

WR6: At most one **representation_item** used for the implicit representation of the **dmf_generic_feature** shall be of type **descriptive_representation_item** with a **name** of 'description'.

5.2.3.1.22 dmf_geometric_curve

A **dmf_geometric_curve** is a type of **dm_feature_definition** that defines a generic curves dimensional feature (see 4.2.98).

EXPRESS specification:

```
*)
ENTITY dmf_geometric_curve
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(* dmf_geometric_curve to dm_point_parameter (as point_on_plane_of_curve) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on plane of curve')
      )) =1 )) )) =1 )) ) =0));

(* dmf_geometric_curve to dm_vector_parameter (as curve_plane_direction) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='curve plane direction')
      )) =1 )) )) =1 )) ) =0));
```

```
(*  dmf_geometric_curve to cartesian_point (as data_points)  *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINITION_REPRESENTATION'
      IN TYPEOF(pdr)) AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='data point'))) =1 ) AND
      (SIZEOF(QUERY (sr <* pdr.used_representation.items |
      ('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT' IN TYPEOF(sr)))) >=0)
      ))) >=0 ))) =0));

  END_ENTITY; -- dmf_geometric_curve
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_geometric_curve** shall contain exactly two **representation_item**s.

WR2: The **dmf_geometric_curce** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point on plane of curve'.

WR3: The **dmf_geometric_curce** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vectror_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'curve plane direction'.

WR4: The **dmf_geometric_curce** is reference by zero one or more **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **cartesian_point**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'data points'.

5.2.3.1.23 dmf_geometric_surface

A **dmf_geometric_surface** is a type of **dm_feature_definition** that defines geometric surfaces dimensional feature (see 4.2.99).

EXPRESS specification:

```
*)
ENTITY dmf_geometric_surface
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(* dmf_geometric_surface to dm_point_parameter (point_on_surface) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on surface')
      )) =1 )) )) =1 )) ) =0));

(* dmf_geometric_surface to dm_vector_parameter (as local_surface_normal) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='local surface normal')
      )) =1 )) )) =1 )) ) =0));
```

```
(* dmf_geometric_surface to cartesian_point (as data_points) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINITION_REPRESENTATION'
      IN TYPEOF(pdr)) AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='data point') )) =1 ) AND
      (SIZEOF(QUERY (sr <* pdr.used_representation.items |
      ('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT' IN TYPEOF(sr)))) >=0)
      ))) >=0 ))) =0));

  END_ENTITY; -- dmf_geometric_surface
(*
```

Formal propositions:

WR1: The implicit representation of the **dmf_geometric_surface** shall contain exactly one **representation_item**s.

WR2: The **dmf_geometric_surface** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point on plane of curve'.

WR3: The **dmf_geometric_surface** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'curve plane direction'.

WR4: The **dmf_geometric_surface** is reference by zero one or more **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **cartesian_point**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'data points'.

5.2.3.1.24 dmf_line_bounded

A **dmf_line_bounded** is a type of **dm_feature_definition** that defines a bounded line dimensional feature (see 4.2.100).

EXPRESS specification:

```
*)
ENTITY dmf_line_bounded
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(* dmf_line_bounded to dm_point_parameter (as first_end_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='first end point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_line_bounded to dm_point_parameter (as second_end_point) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='second end point')
      )) =1 )) )) =1 )) ) =0));
```

```
(* dmf_line_bounded to dm_vector_parameter (as surface_approach_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='surface approach vector')
      )) =1 )) )) =1 )) ) =0));

(* dmf_line_bounded to dm_vector_parameter (as vector_dml) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='vector dml')
      )) =1 )) )) <=1 )) ) =0));

(* dmf_line_bounded to dm_dimension_parameter (length_dml) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='length dml')
      )) =1 )) )) <=1 )) ) =0));
  END_ENTITY; -- dmf_line_bounded
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_line_bounded** shall contain exactly one **representation_-items**.

WR2: The **dmf_line_bounded** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'first end point'.

WR3: The **dmf_line_bounded** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'second end point'.

WR4: The **dmf_line_bounded** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'surface approach vector'.

WR5: The **dmf_line_bounded** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'vector dml'.

WR6: The **dmf_line_bounded** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'length dml'.

5.2.3.1.25 dmf_line_closed_parallel

A **dmf_line_closed_parallel** is a type of **dm_feature_definition** that defines closed parallel line dimensional feature (see 4.2.101).

EXPRESS specification:

```
*)
ENTITY dmf_line_closed_parallel
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   -----------   3 attributes  -------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      ({1 <= SIZEOF(pdr.used_representation.items) <= 3} ) )) = 1 )) = 1;
```

```
(*  dmf_line_closed_parallel to dm_point_parameter (as center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_line_closed_parallel to dm_vector_parameter (axis_direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_line_closed_parallel to dm_vector_parameter (as longitude_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='longitude vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_line_closed_parallel to dm_dimension_parameter (feature_length) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='feature length')
      )) =1 )) )) =1 )) ) =0));
```



```
(*  dmf_line_closed_parallel to dm_dimension_parameter (width)*)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='width')
      )) =1 )) )) =1 )) ) =0));

(*  end_kind *)

 WR7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'end kind')) AND
      ((it.description = 'round') OR
      (it.description = 'square')))
      )) = 1)) )) = 0)) )) = 0;

(* inner_outer *)

 WR8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) = 1)) )) = 0)) )) = 0;
  END_ENTITY; -- dmf_line_closed_parallel
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_line_closed_parallel** shall contain exactly three **representation_item**s.

WR2: The **dmf_line_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'center point'.

WR3: The **dmf_line_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'axis direction vector'.

WR4: The **dmf_line_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'longitude vector'.

WR5: The **dmf_line_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'feature length'.

WR6: The **dmf_line_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'width'.

WR7: Exactly one **representation_item** used for the implicit representation of the **dmf_-line_closed_parallel** shall be of type **descriptive_representation_item** with a **name** of 'end kind' and a **description** of either 'round' or 'square'

WR8: Exactly one **representation_item** used for the implicit representation of the **dmf_-line_closed_parallel** shall be of type **descriptive_representation_item** with a **name** of 'inner or outer' and a **description** of either 'inner' or 'outer'.

5.2.3.1.26 dmf_line_unbounded

A **dmf_line_unbounded** is a type of **dm_feature_definition** that defines unbounded line dimensional feature (see 4.2.102).

EXPRESS specification:

```
*)
ENTITY dmf_line_unbounded
  SUBTYPE OF (dm_feature_definition);
  WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(*  dmf_line_unbounded to dm_point_parameter (point on line) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on line')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_line_unbounded to dm_vector_parameter (direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector')
      )) =1 )) )) =1 )) ) =0));
```

```
(*  dmf_line_unbounded to dm_vector_parameter (surface_approach_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='surface approach vector')
      )) =1 )) )) =1 )) ) =0));

  END_ENTITY; -- dmf_line_unbounded
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_line_closed_parallel** shall contain exactly one **representation_item**s.

WR2: The **dmf_line_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point on line'.

WR3: The **dmf_line_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'direction vector'.

WR4: The **dmf_line_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'surface approach vector'.

5.2.3.1.27 dmf_pattern

A **dmf_pattern** is a type of **dm_feature_definition** that defines a pattern dimensional feature (see 4.2.103).

EXPRESS specification:

```
*)
ENTITY dmf_pattern
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(*   dmf_pattern to dm_point_parameter (point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_pattern todm_vector_parameter (direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector')
      )) =1 )) )) =1 )) ) =0));
```

```
(*   dmf_pattern to dm_feature (as features) *)

 wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
      'DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_RELATIONSHIP'
      IN TYPEOF(pd)) )) >= 2;
  END_ENTITY; -- dmf_pattern
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_pattern** shall contain exactly one **representation_item**s.

WR2: The **dmf_pattern** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point'.

WR3: The **dmf_pattern** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'direction vector'.

WR4: At least two or more **shape_aspect_relationship**s of type **dm_feature_relationship** shall reference a **dmf_pattern** through the **relating_shape_aspect** attribute.

5.2.3.1.28 dmf_plane

A **dmf_plane** is a type of **dm_feature_definition** that defines plane dimensional feature (see 4.2.104).

EXPRESS specification:

```
*)
ENTITY dmf_plane
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;
```



```
(* dmf_plane to dm_point_parameter (point_on_plane) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on plane')
      )) =1 )) )) =1 )) ) =0));

(* dmf_plane to dm_vector_parameter (direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector')
      )) =1 )) )) =1 )) ) =0));
  END_ENTITY; -- dmf_plane
 (*
```

<u>Formal propositions</u>:

WR1: The implicit representation of the **dmf_plane** shall contain exactly one **representation_item**s in its set of **items**.

WR2: The **dmf_plane** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point on plane'.

WR3: The **dmf_plane** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'direction vector'.

5.2.3.1.29 dmf_plane_closed_parallel

A **dmf_plane_closed_parallel** is a type of **dm_feature_definition** that defines a parallel closed plane dimensional feature (see 4.2.105).

EXPRESS specification:

```
*)
ENTITY dmf_plane_closed_parallel
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   3 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =3 ))) =1 ))=1;

(*  dmf_plane_closed_parallel to dm_point_parameter (center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_plane_closed_parallel to dm_vector_parameter(axis_direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));
```

```
(*  dmf_plane_closed_parallel to dm_vector_parameter (longitude_vector) *)

wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
     (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'DIMENSIONAL_INSPECTION_SCHEMA.'+
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
     IN TYPEOF(pdr.used_representation)) AND
     (SIZEOF(QUERY( na <* USEDIN(pdr,
     'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
     (na.attribute_value='longitude vector')
     )) =1 )) )) =1 )) ) =0));

(*  dmf_plane_closed_parallel to dm_vector_parameter (normal_dml) *)

wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
     (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'DIMENSIONAL_INSPECTION_SCHEMA.'+
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
     IN TYPEOF(pdr.used_representation)) AND
     (SIZEOF(QUERY( na <* USEDIN(pdr,
     'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
     (na.attribute_value='normal dml')
     )) =1 )) )) <=1 )) ) =0));

(*  dmf_plane_closed_parallel to dm_dimension_parameter (feature_length) *)

wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
     (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'DIMENSIONAL_INSPECTION_SCHEMA.'+
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
     IN TYPEOF(pdr.used_representation)) AND
     (SIZEOF(QUERY( na <* USEDIN(pdr,
     'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
     (na.attribute_value='feature length')
     )) =1 )) )) =1 )) ) =0));

(*  dmf_plane_closed_parallel to dm_dimension_parameter (height) *)

wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
     (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'DIMENSIONAL_INSPECTION_SCHEMA.'+
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
     IN TYPEOF(pdr.used_representation)) AND
     (SIZEOF(QUERY( na <* USEDIN(pdr,
     'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
     (na.attribute_value='height')
     )) =1 )) )) <=1 )) ) =0));
```

```
(*  dmf_plane_closed_parallel to dm_dimension_parameter (width) *)

 wr8: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='width')
      )) =1 )) )) =1 )) ) =0));

 (* end_kind *)

  WR9: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.'+
       'SHAPE_REPRESENTATION_WITH_PARAMETERS')
       IN TYPEOF(pdr.used_representation)) ) |
       (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND
       (it.name = 'end kind')) AND
       ((it.description = 'round') OR
       (it.description = 'square')))
       )) = 1)) )) = 0)) )) = 0;

(*  inner_outer *)

  WR10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.'+
       'SHAPE_REPRESENTATION_WITH_PARAMETERS')
       IN TYPEOF(pdr.used_representation)) ) |
       (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND
       (it.name = 'inner or outer')) AND
       ((it.description = 'inner') OR
       (it.description = 'outer')))
       )) = 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_plane_closed_parallel
 (*
```


Formal propositions:

WR1: The implicit representation of the **dmf_plane_closed_parallel** shall contain exactly three **representation_item**s in its set of **items**.

WR2: The **dmf_plane_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'center point'.

WR3: The **dmf_plane_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'axis direction vector'.

WR4: The **dmf_plane_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'longitude vector'.

WR5: The **dmf_plane_closed_parallel** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'normal dml'.

WR6: The **dmf_plane_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'feature length'.

WR7: The **dmf_plane_closed_parallel** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'height'.

WR8: The **dmf_plane_closed_parallel** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'width'.

WR9: Exactly one **representation_item** used for the implicit representation of the **dmf_plane_closed_-parallel** shall be of type **descriptive_representation_item** with a **name** of 'end kind' and a **description** of either 'round' or 'square'

WR10: Exactly one **representation_item** used for the implicit representation of the **dmf_plane_closed_-parallel** shall be of type **descriptive_representation_item** with a **name** of 'inner or outer' and a **description** of either 'inner' or 'outer'.

5.2.3.1.30 dmf_plane_symmetric

A **dmf_plane_symmetric** is a type of **dm_feature_definition** that defines symmetric plane dimensional feature (see 4.2.106).

EXPRESS specification:

```
*)
ENTITY dmf_plane_symmetric
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*    dmf_plane_symmetric to dm_point_parameter (point_on_one_side) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on one side')
      )) =1 )) )) =1 )) ) =0));
```



```
(*    dmf_plane_symmetric to dm_point_parameter (point_on_side_two) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on side two')
      )) =1 )) )) =1 )) ) =0));

(*    dmf_plane_symmetric to dm_point_parameter (point_on_mid_plane) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on mid plane')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_plane_symmetric to dm_vector_parameter (direction_vector_side_one) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector side one')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_plane_symmetric to dm_vector_parameter (direction_vector_side_two) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector side two')
      )) =1 )) )) =1 )) ) =0));
```

```
(*    dmf_plane_symmetric to dm_dimension_parameter (width_at_mid_point) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='width at mid point')
      )) =1 )) )) =1 )) ) =0));

(*  inner_outer *)

 WR8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_plane_symmetric
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_plane_symmetric** shall contain exactly two **representation_item**s in its set of **items**.

WR2: The **dmf_plane_symmetric** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point on side one'.

WR3: The **dmf_plane_symmetric** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point on side two'.

WR4: The **dmf_plane_symmetric** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point on mid side'.

WR5: The **dmf_plane_symmetric** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'direction vector side one'.

WR6: The **dmf_plane_symmetric** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'direction vector side two'.

WR7: The **dmf_plane_symmetric** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of'width at mid point'.

WR8: Exactly one **representation_item** used for the implicit representation of the **dmf_plane_-symmetric** shall be of type **descriptive_representation_item** with a **name** of 'inner orouter' and a **description** of either 'inner' or 'outer'.

5.2.3.1.31 dmf_point

A **dmf_point** is a type of **dm_feature_definition** that defines point dimensional feature (see 4.2.107).

EXPRESS specification:

```
*)
ENTITY dmf_point
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;
```



```
(*  dmf_point to dm_point_parameter (point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_point to dm_vector_parameter (direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector')
      )) =1 )) )) =1 )) ) =0));

  END_ENTITY; -- dmf_point
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_point** shall contain either one or two **representation_item**s in its set of **items**.

WR2: The **dmf_plane_symmetric** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'point'.

WR3: The **dmf_plane_symmetric** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'direction vector'.

5.2.3.1.32 dmf_sphere

A **dmf_sphere** is a type of **dm_feature_definition** that defines a sphere dimensional feature (see 4.2.108).

EXPRESS specification:

```
*)
ENTITY dmf_sphere
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*  dmf_sphere to dm_point_parameter (center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_sphere to dm_vector_parameter (north_pole_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='north pole vector')
       )) =1 )) )) =1 )) ) =0));
```

```
(*  dmf_sphere to dm_vector_parameter (prime_meridian_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='prime meridian vector')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (longitude_start_angle) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='longitude start angle')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (longitude_stop angle) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='longitude stop angle')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (latitude_start_angle) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='latitude start angle')
      )) =1 )) )) <=1 )) ) =0));
```



```
(*  dmf_sphere to dm_dimension_parameter (latitude_stop_angle) *)

 wr8: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='latitude stop angle')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (diameter) *)

 wr9: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='diameter')
      )) =1 )) )) =1 )) ) =0));

(*  inner_outer *)

 WR10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;
  END_ENTITY; -- dmf_sphere
 (*
```

Formal propositions:

WR1: The implicit representation of the **dmf_sphere** shall contain exactly two **representation_item**s in its set of **items**.

WR2: The **dmf_sphere** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'center point'.

WR3: The **dmf_sphere** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'north pole vector'.

WR4: The **dmf_sphere** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'prime meridian vector'.

WR5: The **dmf_sphere** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'longitude start angle'.

WR6: The **dmf_sphere** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'longitude stop angle'.

WR7: The **dmf_sphere** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'latitude start angle'.

WR8: The **dmf_sphere** is reference by at most one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'latitude stop angle'.

WR9: The **dmf_sphere** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'diameter'.

WR10: Exactly one **representation_item** used for the implicit representation of the **dmf_plane_-symmetric** shall be of type **descriptive_representation_item** with a **name** of 'inner orouter' and a **description** of either 'inner' or 'outer'.

5.2.3.1.33 dmf_surface_of_revolution_dml

A **dmf_surface_of_revolution_dml** is a type of **dm_feature_definition** that defines a surface of revolution dimensional feature (see 4.2.109).

EXPRESS specification:

```
*)

ENTITY dmf_surface_of_revolution_dml
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      ({1 <= SIZEOF(pdr.used_representation.items) <= 2} ) )) = 1 )) = 1;
```

```
(*  dmf_sphere to dm_point_parameter (profile_curve) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='profile curve')
      )) =1 )) )) >=2 )) ) =0));

(*  dmf_sphere to dm_vector_parameter (axis_of_revolution) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis of revolution')
      )) =1 )) )) =1 )) ) =0));

  END_ENTITY; -- dmf_surface_of_revolution_dml
(*
```

Formal propositions:

WR1: The implicit representation of the **dmf_surface_of_revolution_dml** shall contain exactly one **representation_item**s in its set of **items**.

WR2: The **dmf_surface_of_revolution_dml** is reference by two or more **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_-parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'profile curve'.

WR3: The **dmf_surface_of_revolution_dml** is reference by exactly one **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of'axis of revolution'.

5.2.3.1.34 dmf_torus

A **dmf_torus** is a type of **dm_feature_definition** that defines torus dimensional feature (see 4.2.110).

<u>EXPRESS specification:</u>

```
*)
ENTITY dmf_torus
  SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*   dmf_torus to dm_point_parameter (center_point)  *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_torus to dm_vector_parameter (vector_or_plane)  *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='vector of plane')
      )) =1 )) )) =1 )) ) =0));
```

```
(*   dmf_torus to dm_dimension_parameter (major_diameter)  *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='major diameter')
      )) =1 )) )) <=1 )) ) =0));

(*   dmf_torus to dm_dimension_parameter (minor_diameter)  *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='minor diameter')
      )) =1 )) )) <=1 )) ) =0));

(*  inner_outer  *)

 WR6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_torus
 (*
```


Formal propositions:

WR1: The implicit representation of the **dmf_torus** shall contain exactly two **representation_item**s in its set of **items**.

WR2: The **dmf_torus** is reference by two or more **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_point_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of ' center point'.

WR3: The **dmf_torus** is reference by two or more **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_vector_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'vector on plane'.

WR4: The **dmf_torus** is reference by two or more **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'major diameter'.

WR5: The **dmf_torus** is reference by two or more **property_definition** through the **definition** attribute that is referenced by a **property_definition_representation** through the **definition** attribute that refernces through the **used_representation** a **representation** of type **dm_dimension_parameter**. The **property_definition_representation** shall be reference by a **name_attribute** through the **named_item** attribute with an attribute_value of 'minor diameter'.

WR6: Exactly one **representation_item** used for the implicit representation of the **dmf_torus** shall be of type **descriptive_representation_item** with a **name** of 'inner or outer' and a **description** of either 'inner' or 'outer'

5.2.3.1.35 dm_dimension_parameter

A **dm_dimension_parameter** is a type of **dm_parameter** that defines an dimension parameter for a dimensional feature (see 4.2.74).

EXPRESS specification:

```
*)

  ENTITY dm_dimension_parameter
    SUBTYPE OF (dm_result_parameter);
    WHERE

(*  dm_dimension_parameter to numeric_parameter (as calculated_value) *)

 wr1: SIZEOF(QUERY ( sri <* SELF.items | ((
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
      IN TYPEOF(sri)) AND (sri.name = 'calculated value')) )) = 1;
```

```
(*dm_dimension_parameter to measurement_uncertainty (as value_uncertainty)*)

 wr2: SIZEOF(QUERY ( repi <* SELF.items |
      (((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(repi)) = 2) AND
      (SIZEOF(QUERY ( uq <* repi.qualifiers |
      (SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.STANDARD_UNCERTAINTY']
      * TYPEOF(uq)) = 1) AND
      (uq.measure_name = 'value uncertainty'))) =1 ))))) <=1;

  END_ENTITY; -- dm_dimension_parameter
(*
```

Formal propositions:

WR1: There shall be exactly one reference through **items** to a **measure_representation_item** with **name** of ‘calculated value’.

WR2: There shall be zero or one reference through **items** to a **measure_representation_item** and **qualified_measure_with_unit** that references through the **qualifiers** attribute an **uncertainty_qualifier** with **measurement_name** of 'value uncertainty' that is of type **standard_uncertainty** or **expanded_-uncertainty**.

5.2.3.1.36 dm_execution_input

A **dm_execution_input** is a type of **execution_action** that defines an the input to execution an inspection program (see 4.2.75).

EXPRESS specification:

```
*)
ENTITY dm_execution_input
   SUBTYPE OF (executed_action);

WHERE

(*  dm_execution_input to dm_program_run (as program_run) *)

  wr1: ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PROGRAM_RUN'
       IN TYPEOF (SELF.chosen_method));

(*   dm_execution_input to dm_execution_result (as result) *)

  wr2: (SIZEOF(QUERY( ap <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_RELATIONSHIP.RELATING_ACTION')|
       ('DIMENSIONAL_INSPECTION_SCHEMA.DM_EXECUTION_RESULT'
       IN TYPEOF(ap.related_action))))=1);

(*   dm_execution_input to part (as as_designed_part) *)
```



```
  wr3: (SIZEOF(QUERY( ap <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_ASSIGNMENT.ASSIGNED_ACTION')|
       (('DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_ACTION_ASSIGNMENT'
       IN TYPEOF(ap)) AND (SIZEOF(QUERY ( it <* ap.items |
       ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION'
        IN TYPEOF(it))))=1))))
       =1);

  END_ENTITY; -- dm_execution_input
(*
```

Formal propositions:

WR1: There shall be a **dm_program_run** referenced through the **chosen_method** attribute.

WR2: There shall be a reference from **action_relationship** through the **relating_action** that references a **dm_executed_result** through the **related_action**.

WR3: There shall be a reference from **action_assignment** through the **assigned_action** that is of type **applied_action_assignment** that references a **product_definition** through the **items** attribute.

5.2.3.1.37 dm_execution_result

A **dm_execution_result** is a type of **execution_action** that defines an the result of execution an inspection program (see 4.2.76).

EXPRESS specification:

```
*)
ENTITY dm_execution_result
   SUBTYPE OF (executed_action);

WHERE

(*  dm_execution_result to dm_program_run (as program_run) *)

  wr1: ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PROGRAM_RUN'
        IN TYPEOF (SELF.chosen_method));

(* dm_execution_result to dm_execution_result_measurement(dm_measurements) *)

  wr2: (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.DM_EXECUTION_RESULT_MEASUREMENT'
       IN TYPEOF (RP))
       ))>=1);

  END_ENTITY; -- dm_execution_result

(*
```

Formal propositions:

WR1: There shall be a **dm_program_run** referenced through the **chosen_method** attribute.

WR2: The **dm_execution_result** shall be referenced by one or more **action_property** entities of type **dm_execution_result_measurement**

5.2.3.1.38 dm_execution_result_measurement

A **dm_execution_result_measurement** is a type of **action_property** that defines a measurement that is the result of execution an inspection program (see 4.2.77).

EXPRESS specification:

```
*)
  ENTITY dm_execution_result_measurement
    SUBTYPE OF (action_property);
    WHERE

(* dm_execution_result_measurement dm_data_acquisition_software (software) *)

wr1: (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
     (NOT(SIZEOF(QUERY ( it <* apr.representation.items |
     ((it.name = 'dm data acquisition software') AND
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'application name')))
     )) <= 1)) )) = 0);

wr2: (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
     (NOT(SIZEOF(QUERY ( it <* apr.representation.items |
     ((it.name = 'dm data acquisition software') AND
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'application version')))
     )) <= 1)) )) = 0);

wr3: (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
     (NOT(SIZEOF(QUERY ( it <* apr.representation.items |
     ((it.name = 'dm data acquisition software') AND
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'vendor name')))
     )) <= 1)) )) = 0);

(*  dm_execution_result_measurement to dm_point (as measurement_points) *)

wr4: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
    ((apr.representation.name = 'measurement points') AND
    ('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT'
    IN TYPEOF(apr.representation))) )) >= 1;

(*  dm_execution_result_measurement to dm_result_parameter (as parameter) *)
```



```
wr5: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
     ('DIMENSIONAL_INSPECTION_SCHEMA.DM_RESULT_PARAMETER'
     IN TYPEOF(apr.representation)) )) = 1;

(*  compensation *)

wr6: SELF.name IN ['TRUE','FALSE'];

  END_ENTITY; -- dm_execution_result_measurement

(*
```

Formal propositions:

WR1: Exactly one **representation_item** used for the representation of an **dm_execution_result_-measurement** shall be of type **descriptive_representation_item** with a name of 'application name'.

WR2: Exactly one **representation_item** used for the representation of an **dm_execution_result_-measurement** shall be of type **descriptive_representation_item** with a name of 'application version'.

WR3: Exactly one **representation_item** used for the representation of an **dm_execution_result_-measurement** shall be of type **descriptive_representation_item** with a name of 'vendor or user name'.

WR4: There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references through **representation** a **representation** that has in the list of **items** a **dm_point** with name 'measurement points'.

WR5: There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references through **representation** a **representation** that has in the list of **items** a **dm_result_parameter**.

WR6: the **name** attribute shall have values of ‘TRUE’ or ‘FALSE’.

5.2.3.1.39 dm_feature_analysis_mode_dml

A **dm_feature_analysis_mode_dml** is a type of **dimensional_measurement_representation** that defines feature analysis parameters (see 4.2.79).

EXPRESS specification:

```
*)

ENTITY dm_feature_analysis_mode_dml
     SUBTYPE OF (dimensional_measurement_representation);
WHERE
```

```
(* axis_method  *)

wr1: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode axis') AND
     (repi.description IN ['least square','minimum radius',
     'cross section centers'])
     ))) <= 1;

(*  axis_extrapolate  *)

wr2: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode axis extrapolate') AND
     (repi.description IN ['TRUE','FALSE','unknown'])
     ))) <= 1;

(* aelpr_analysis *)

wr3: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode aelpr') AND
     (repi.description IN ['least square','minimum maximum'])
     ))) <= 1;

(* ccpst_analysis  *)

wr4: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode ccpst') AND
     (repi.description IN ['least square','minimum maximum',
      'minimum size circumscribed','maximum size inscribed'])
     ))) <= 1;

(* curve_analysis  *)

wr5: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode curve') AND
     (repi.description IN ['least square','minimum maximum','bspline'])
     ))) <= 1;

(* limits_of_size  *)

wr6: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode limits') AND
     (repi.description IN ['functional','mrs average','two point'])
     ))) <= 1;
```



```
(* surface_analysis  *)

wr7: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode surface') AND
     (repi.description IN ['least square','minimum maximum',
      'nurbs','bezier'])
     ))) <= 1;

(* axis_method default  *)

wr8: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode axis default') AND
     (repi.description IN ['least square'])
     ))) = 1;

(*  axis_extrapolate default  *)

wr9: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode axis extrapolate default') AND
     (repi.description IN ['TRUE'])
     ))) = 1;

(* aelpr_analysis *)

wr10: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode aelpr default') AND
     (repi.description IN ['least square'])
     ))) = 1;

(* ccpst_analysis default *)

wr11: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode ccpst default') AND
     (repi.description IN ['least square'])
     ))) = 1;

(* curve_analysis default  *)

wr12: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode curve default') AND
     (repi.description IN ['least square'])
     ))) = 1;
```

```
(* limits_of_size default  *)

wr13: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode limits default') AND
     (repi.description IN ['functional'])
     ))) = 1;

(* surface_analysis default  *)

wr14: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode surface default') AND
     (repi.description IN ['least square'])
     ))) = 1;

  END_ENTITY; -- dm_feature_analysis_mode_dml
 (*
```

Formal propositions:

WR1: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode axis' and **description** of 'least square', 'minimum radius' or 'cross section centers'.

WR2: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode axis extrapolate' and **description** of 'TRUE' 'FALSE' or ‘unknown’.

WR3: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode aelpr' and **description** of 'least square', 'minimum maximum'.

WR4: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode ccpst' and **description** of 'least square', 'minimum maximum', 'minimum size circumscribed' or 'maximum size inscribed'.

WR5: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode curve' and **description** of 'least square', 'minimum maximum' or 'bspline'.

WR6: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode limits' and **description** of 'functional', 'mrs average' or 'two point'.

WR7: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode surface'and **description** of 'least square', 'minimum maximum' or 'nurbs', 'bezier'.

WR8: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode axis default' and **description** of 'least square'.

WR9: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode axis extrapolate default' and **description** of 'TRUE'.

WR10: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode aelpr default' and **description** of 'least square'.

WR11: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode ccpst default' and **description** of 'least square'.

WR12: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode curve default' and **description** of 'least square'.

WR13: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode limits default' and **description** of 'functional'.

WR14: One of the **representation_item**s used for the implicit representation of a **dm_feature_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature analysis mode surface default' and **description** of 'least square'.

5.2.3.1.40 dm_feature_definition

A **dm_feature_definition** is a type of **characterized_object** that defines a dimensional feature (see 4.2.78).

EXPRESS specification:

```
*)
  ENTITY dm_feature_definition
    SUBTYPE OF (characterized_object);
    WHERE
 wr1: SIZEOF(QUERY ( pdr <*
      get_property_definition_representations(SELF) |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) )) >= 1;
```

```
 wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      (('DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
      (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;

 wr3: SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.DMF_ARC',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_LINE_UNBOUNDED',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_CIRCLE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_PATTERN',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_CONE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_PLANE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_PLANE_CLOSED_PARALLEL',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_CYLINDER',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_ELLIPSE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_PLANE_SYMMETRIC',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_GEOMETRIC_CURVE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_POINT',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_GEOMETRIC_SURFACE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_EDGE_POINT',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_SPHERE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_GENERIC_FEATURE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_LINE_BOUNDED',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_TORUS',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_LINE_CLOSED_PARALLEL',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_SURFACE_OF_REVOLUTION_DML']
              * TYPEOF(SELF)) = 1;
  END_ENTITY; -- dm_feature_definition

(*
```

<u>Formal propositions</u>:

WR1: The dm_**feature_definition** shall have exactly one implicit **representation** which shall be of type **dm_shape_representation_with_parameters**.

WR2: Exactly one **representation_item** used for the implicit representation of a **dm_feature_definition** shall be of type **placement** with a **name** of 'orientation'. The **placement** shall define a reference location and orientation for the origin of the **dm_feature_definition.**

WR3: The **dm_feature_definition** shall be either a **dmf_arc, dmf_line_unbounded, dmf_circle, dmf_pattern, dmf_cone, dmf_plane,.dmf_plane_closed_parallel,, dmf_cylinder, dmf_ellipse, dmf_-plane_symmetric, dmf_geometric_curve, dmf_point, dmf_geometric_surface, dmf_edge_point, dmf_sphere, dmf_generic_feature, dmf_line_bounded', dmf_torus, dmf_line_closed_parallel** or **dmf_surface_of_revolution_dml.**

5.2.3.1.41 dm_feature_relationship

A **dm_feature_relationship** defines a feature relationship of an instance of a feature in a feature pattern.

EXPRESS specification:

```
*)
  ENTITY dm_feature_relationship
    SUBTYPE OF (shape_aspect_relationship);
    WHERE
 wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_DEFINITION' IN
                TYPEOF(SELF.relating_shape_aspect);
 wr2: (SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_DEFINITION',
      'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_DEFINITION']
      * TYPEOF(SELF.related_shape_aspect)) = 1);

  END_ENTITY; -- dm_feature_relationship
 (*
```

Formal propositions:

WR1: The **relating_shape_aspect** attribute shall reference a **dm_feature_definition**.

WR2: The **related_shape_aspect** shall reference a **manufacturing_feature** or a **dimensional_-inspection_feature**.

5.2.3.1.42 dm_instanced_feature

An **dm_instanced_feature** is a type of **shape_aspect** and of **feature_definition** that is the representation of a preconceived pattern for the dimensional inspection application purpose.

EXPRESS specification:

```
*)

  ENTITY dm_instanced_feature
    SUBTYPE OF (dm_feature_definition, shape_aspect);
    WHERE

 wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(
      SELF.of_shape.definition);

 wr2: SELF.product_definitional;
```

```
(*  dm_feature to cartesian_point (as boundary) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINITION_REPRESENTATION'
      IN TYPEOF(pdr)) AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND
      (pdr.used_representation.name='boundary') AND
      (SIZEOF(QUERY (sr <* pdr.used_representation.items |
      ('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT'
      IN TYPEOF(sr)))) >=0)
      ))) <=1 ))) =0));


  END_ENTITY; -- dm_instanced_feature
 (*
```

Formal propositions:

WR1: The **dm_instanced_feature** shall be an aspect of the shape of a **product_definition**.

WR2: The **dm_instanced_feature** shall lie on the boundary of the part.

WR3: Exactly one **representation_item** used for the implicit representation that shall be of type **cartesian_point** with a **name** of 'boundary'.

Informal propositions:

IP1: If an **dm_instanced_feature** has a **shape_representation** and if the **product_definition_shape** that is referenced as **of_shape** by the **dm_instanced_feature** has a **shape_representation** then these **shape_representation**s shall have the same **geometric_representation_context**.

5.2.3.1.43 dm_tolerance_analysis_mode_dml

A **dm_tolerance_analysis_mode_dml** is a type of **dimensional_measurement_representation** that defines feature tolerance parameters (see 4.2.89).

EXPRESS specification:

```
*)

  ENTITY dm_tolerance_analysis_mode_dml
    SUBTYPE OF (dimensional_measurement_representation);
WHERE
```



```
(* method  *)

wr1: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode method') AND
     (repi.description IN ['least square 3d','least square normal',
     'minimum deviation 3d','standard'])
     ))) <= 1;

(* option *)

wr2: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode option') AND
     (repi.description IN ['high point','best fit'])
     ))) <= 1;

(* setting  *)

wr3: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode setting') AND
     (repi.description IN ['inner','outer','standard'])
     ))) <= 1;

(* method default *)

wr4: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode method default') AND
     (repi.description IN ['standard'])
     ))) = 1;

(* option default *)

wr5: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode option default') AND
     (repi.description IN ['best fit'])
     ))) = 1;
```

```
(* setting default  *)

wr6: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode setting default') AND
     (repi.description IN ['standard'])
     ))) = 1;

  END_ENTITY; -- dm_tolerance_analysis_mode_dml

(*
```

Formal propositions:

WR1: One of the **representation_item**s used for the implicit representation of a **dm_tolerance_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature tolerance mode method' and **description** of 'least square 3d', 'least square normal' or 'minimum deviation 3d', 'standard'.

WR2: One of the **representation_item**s used for the implicit representation of a **dm_tolerance_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature tolerance mode option' and **description** of 'high point' or 'best fit'.

WR3: One of the **representation_item**s used for the implicit representation of a **dm_tolerance_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature tolerance mode setting' and **description** of 'inner', 'outer' or 'standard'.

WR4: One of the **representation_item**s used for the implicit representation of a **dm_tolerance_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature tolerance mode method default' and **description** of 'standard'.

WR5: One of the **representation_item**s used for the implicit representation of a **dm_tolerance_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature tolerance mode option default' and **description** of 'best fit'.

WR6: One of the **representation_item**s used for the implicit representation of a **dm_tolerance_-analysis_mode_dml** may be of type **descriptive_representation_item** with a **name** of 'feature tolerance mode setting default' and **description** of 'standard'.

5.2.3.1.44 dm_result_parameter

A **dm_result_parameter** is a type of **representation** that defines parameters for dimensional feature (see 4.2.87).

EXPRESS Specification:

```
*)
  ENTITY dm_result_parameter
    SUBTYPE OF (representation);
 WHERE
```



```
(* dm_result_parameter to dm_parameter_value_limits (as limits) *)

 wr1:SIZEOF(QUERY( sri <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1' ) |
     ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PARAMETER_VALUE_LIMITS'
      IN TYPEOF(sri.rep_2)
     ))) <= 1;

(*   calculation_method *)

 wr2:SIZEOF(QUERY( sri <* SELF.items |
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(sri)) AND (sri.name='calculation method'))  )) <=1;

(* dm_result_parameter to dm_parameter_analysis_dml (analysis_conditions) *)

 wr3: SIZEOF(QUERY ( apr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'ACTION_PROPERTY_REPRESENTATION.REPRESENTATION')|
      ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PARAMETER_ANALYSIS_DML'
      IN TYPEOF(apr.property ))))<= 1;

(* dm_result_parameter to ??? (as specification) *)

 wr4: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.'+
      'USED_REPRESENTATION') |
     ((pdr.name='specification') AND
     ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
      IN TYPEOF(pdr.definition)) AND
     (SIZEOF(QUERY( sa <* USEDIN(pdr.definition,
     'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
     (SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.TRANSITION_FEATURE',
              'DIMENSIONAL_INSPECTION_SCHEMA.INSTANCED_FEATURE']
      * TYPEOF(sa)) = 1)
      )) =1 )))) <=1;

wr5:   (SIZEOF(QUERY( pdr <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
         ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DIMENSION_REPRESENTATION'
         IN TYPEOF (pdr.rep_2)))) <=1 );
```

```
 WR6: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.'+
      'USED_REPRESENTATION') |
     ((pdr.name='specification') AND
     ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
     IN TYPEOF(pdr.definition)) AND
     (SIZEOF(QUERY( sa <* USEDIN(pdr.definition,
     'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
     (SIZEOF(QUERY(gtol <* USEDIN(sa,
      'DIMENSIONAL_INSPECTION_SCHEMA.GEOMETRIC_TOLERANCE.'+
      'TOLERANCED_SHAPE_ASPECT') |
     ('DIMENSIONAL_INSPECTION_SCHEMA.GEOMETRIC_TOLERANCE'
      IN TYPEOF(gtol)))) =1 )
           )) =1 )) )) <=1;
END_ENTITY;
(*
```

Formal proposition:

WR1: There shall be exactly one reference from a **representation_relationship** through the **rep_1** attribute that references through the **rep_2** attribute a **dm_parameter_value_limit.**

WR2: There shall be one or less **descriptive_representation_item** with **name** of ‘calculation method’ referenced through the **items** attribute.

WR3: There may be a reference from an **action_property_representation** through the **representation** attribute that references a **dm_parameter_analysis_dml** through the **property** attribute.

WR4: There may be a reference from a **property_definition_representation** through the **used_representation** attribute that references a **property_definition** through the **definition** attribute that references a **shape_aspect** through the **definition** attributeattribute that references a **shape_aspect** through the **definition** attribute that is of type **transition_feature** or **instanced_feature**.

WR5: There shall be exactly one reference from a **representation_relationship** through the **rep_1** attribute that references through the **rep_2** attribute a **shape_dimension_representation.**

WR6: There may be a reference from a **property_definition_representation** through the **used_representation** attribute that references a **property_definition** through the **definition** attribute that references a **shape_aspect** through the **definition** attributeattribute that references a **shape_aspect** through the **definition** attribute that references a **geometric_tolerance** through the **toleranced_shape_-aspect** attribute.

5.2.3.1.45 dm_parameter_analysis_dml

A **dm_parameter_analysis_dml** is a type of **action_property** that defines a defines parameters for data analysis for a program run

EXPRESS Specification:

```
*)

  ENTITY dm_parameter_analysis_dml
    SUBTYPE OF (action_property);
    WHERE

(* to dm_feature_analysis_mode_dml (feature_analysis) *)

 wr1: SIZEOF(QUERY ( apr <* USEDIN(SELF,

'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
      ('DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_ANALYSIS_MODE_DML'
      IN TYPEOF(apr.representation)) )) <= 1;

(*  to dm_tolerance_analysis_mode_dml (tolerance_analysis) *)

 wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
     ('DIMENSIONAL_INSPECTION_SCHEMA.DM_TOLERANCE_ANALYSIS_MODE_DML'
     IN TYPEOF(apr.representation)) )) <= 1;

(* dm_parameter_analysis_dml to dm_analysis_dofs_dml (as dofs) *)

 wr3: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
     ('DIMENSIONAL_INSPECTION_SCHEMA.DM_ANALYSIS_DOFS_DML' IN
     TYPEOF(apr.representation)) )) <= 1;

  END_ENTITY; -- dm_parameter_analysis_dml
(*
```

Formal proposition:

WR1: There shall be exactly one **action_property_representation** referenced through the **propert**y attribute that references a **dm_feature_analysis_mode_dml** through the **representation** attribute.

WR2: There shall be exactly one **action_property_representation** referenced through the **propert**y attribute that references a **dm_tolerance_analysis_mode_dml** through the **representation** attribute.

WR3: There shall be exactly one **action_property_representation** referenced through the **propert**y attribute that references a **dm_analysis_dofs_dml** through the **representation** attribute.

5.2.3.1.46 dm_parameter_value_limits

A **dm_parameter_value_limits** is a type of **compound_representation_item** that specifies calculation limits for **dm_result_parameters**

EXPRESS Specification:

```
*)

   ENTITY dm_parameter_value_limits
    SUBTYPE OF (representation);

 WHERE

 (* limits_method *)

 wr1 : SIZEOF(QUERY ( repi <* SELF.items |
       (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(repi)) AND
       (repi.name = 'limits method')
       )))=1;

 (* dm_parameter_value_limits to numeric_parameter (as calculated_limits) *)

 WR2: (SIZEOF(QUERY ( it <* SELF.items |
      ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'] *
      TYPEOF(it)) = 1) AND
      (it.name = 'calculated limits')
      )))>=2);
  END_ENTITY; -- dm_parameter_value_limits
 (*
```

Formal proposition:

WR1: One of the **representation_item**s shall be of type **descriptive_representation_item** with **name** of 'limit method'.

WR2: One of the **representation_item**s shall be of type **measure_representation_item** with a **name** of 'calculated limits'.

5.2.3.1.47 dm_point

A **dm_point** is a type of **representation** that defines a dimensional measurement point (see 4.2.83).

EXPRESS Specification:

```
*)
  ENTITY dm_point
    SUBTYPE OF (representation);
WHERE

(*   dm_point to cartesian_point (as measured_point) *)

 WR1: (SIZEOF( QUERY( repi <* SELF.items |
      NOT (('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT'
      IN TYPEOF(repi)) AND (repi.name='measured point'))
      ))=1);

(*   dm_point to cartesian_point (as expected_point) *)

 WR2: (SIZEOF( QUERY( repi <* SELF.items |
      NOT (('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT'
      IN TYPEOF(repi)) AND (repi.name='expected point'))
      ))<=1);

  END_ENTITY; -- dm_point

(*
```

Formal proposition:

WR1: There shall be exactly one **representation_item**s in its set of **items** that contains a **cartesian_point** with name of 'measured point'.

WR2: There shall be exactly one **representation_item**s in its set of **items** that contains a **cartesian_point** with name of 'expected point'.

5.2.3.1.48 dm_point_parameter

A **dm_point_parameter** is a type of **dm_parameter** that defines a defines point parameters for dimensional feature (see 4.2.84).

EXPRESS Specification:

```
*)
  ENTITY dm_point_parameter
    SUBTYPE OF (dm_result_parameter);
WHERE

(* dm_point_parameter to cartesian_point (as calculated_point) *)

  wr1:  SIZEOF(QUERY( repi <* SELF.items |
        ('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT'
        IN TYPEOF (repi))))=1;

(*  dm_point_parameter to measurement_uncertainty (as value_uncertainty) *)

 wr2: SIZEOF(QUERY ( repi <* SELF.items | (
    (SIZEOF([
    'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
    * TYPEOF(repi)) = 2) AND
    (SIZEOF(QUERY ( uq <* repi.qualifiers | (
    (SIZEOF([
    'DIMENSIONAL_INSPECTION_SCHEMA.STANDARD_UNCERTAINTY'] * TYPEOF(uq))=1) AND
    (uq.measure_name = 'value uncertainty') AND
    (uq.description IN ['x uncertainty','y uncertainty', 'z uncertainty']) )
     )) <=3 )
     ))) <=1;

  END_ENTITY; -- dm_point_parameter
  (*
```

Formal proposition:

WR1: There shall be exactly one **representation_item**s in its set of **items** that contains a **cartesian_point**.

WR2: There shall be zero to three reference through **items** to a **measure_representation_item** and **qualified_measure_with_unit** that references through the **qualifiers** attribute an **uncertainty_qualifier** with **measurement_name** of type **standard_uncertainty** or **expanded_uncertainty** with a **measure_-name** of 'calculated value' and a **description** of 'x uncertainty', 'y uncertainty' or 'z uncertainty'.

5.2.3.1.49 dm_program_identification

A **dm_program_identification** is a type of **action_resource_requirement** that defines the identification of a dimensional measurement program (see 4.2.85).

EXPRESS Specification:

```
*)
  ENTITY dm_program_identification
    SUBTYPE OF (action_resource_requirement);
WHERE

(*  angular_units_dml *)

 wr1: (SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      (rp.name='angular units dml')))<=1);

(*  linear_units_dml *)

 wr2: (SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      (rp.name='linear units dml')))<=1);

(*  tolerance_standard_dml *)

 wr3: (SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      (rp.name='tolerance standard dml')))<=1);

(* dm_program_identification to person_in_organization (program_custodian) *)
(* dm_program_identification to organization (program_custodian) *)

 wr4: ((SIZEOF(QUERY (adr <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
      ((adr.role.name='program custodian') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.ORGANIZATION' IN TYPEOF
      (adr.assigned_organization)))))<=1) OR
      (SIZEOF(QUERY (adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS')|
      ((adr.role.name='program custodian') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.PERSON_AND_ORGANIZATION' IN TYPEOF
      (adr.assigned_person_and_organization)))))<=1));

  END_ENTITY; -- dm_program_identification

(*
```

Formal proposition:

WR1: There shall be at most one reference by a **resource_property** through the **resource** attribute with a **name** of 'angular units dml'.

WR2: There shall be at most one reference by a **resource_property** through the **resource** attribute with a **name** of 'linear units dml'.

WR3: There shall be at most one reference by a **resource_property** through the **resource** attribute with a **name** of 'tolerance standard dml'.

WR4: There shall be at least one reference by an **applied_organization_assignment** through the **items** attribute that references an **organization_role** through the **role** attribute with **name** of 'program custodian' and references an **organization** through the **assigned_organization** attribute, or there shall be at least one reference by an **applied_person_and_organization_assignment** through the **items** attribute that references an **person_and_organization_role** through the **role** attribute with **name** of 'program custodian' and references a **person_and_organization** through the **assigned_person_and_organization** attribute.

5.2.3.1.50 dm_program_run

A **dm_program_run** is a type of **action_method** that defines the information of a dimensional measurement program run (see 4.2.86).

EXPRESS Specification:

```
*)
  ENTITY dm_program_run
    SUBTYPE OF (action_method);
WHERE

(*  dm_program_run to numeric_parameter (as measurement_temperature) *)

wr1 : (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION')|
      (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
      (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(it)) = 2) AND (it.name = 'measurement temperature')) ))
      = 1)) ) = 0)) )) = 0));
```



```
(*  dm_program_run to numeric_parameter (as workpiece_temperature) *)

wr2 : (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION')|
      (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
      (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(it)) = 2) AND (it.name = 'workpiece temperature')) ))
      <= 1)) ) = 0)) )) = 0));

(*  dm_program_run to numeric_parameter (as measurement_humidity) *)

wr3 : (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION')|
      (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
      (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(it)) = 2) AND (it.name = 'measurement humidity')) ))
      = 1)) ) = 0)) )) = 0));

(*   part_inspection_status *)

wr4 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_RESOURCE.USAGE') |
      (ap.name = 'part inspection status') )) <= 1;

(*   url*)

wr5 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_RESOURCE.USAGE') |
      (ap.name = 'url') )) <= 1;

(*   dm_program_run to dm_program_identification (as program_id) *)

wr6 : SIZEOF(QUERY ( arr <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PROGRAM_IDENTIFICATION'
      IN TYPEOF(arr)) )) = 1;
```



```
(*   dm_program_run to run_administrator (as run_administrator) *)

wr7 : (SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
      ((adr.role.name = 'run administrator') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.ORGANIZATION'
      IN TYPEOF(adr.assigned_organization))) )) = 1) OR
      (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
      ((adr.role.name = 'run administrator') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.PERSON_AND_ORGANIZATION'
      IN TYPEOF(adr.assigned_person_and_organization))) )) = 1);

(*    dm_program_run to measurement_location (as run_location) *)

wr8 : SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION')
      | ('DIMENSIONAL_INSPECTION_SCHEMA.MEASUREMENT_LOCATION' IN
      TYPEOF(rp)) )) <= 1;

(*    dm_program_run to date_time (as run_end) *)

wr9 : (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS')|
      ((adr.role.name = 'run end') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.DATE_AND_TIME'
      IN TYPEOF(adr.assigned_date))) )) <= 1) OR
      (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
      ((adr.role.name = 'run end') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.CALENDAR_DATE'
      IN TYPEOF(adr.assigned_date))) )) <= 1);

(*    dm_program_run to date_time (as run_start) *)

wr10: (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS')|
      ((adr.role.name = 'run end') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.DATE_AND_TIME'
      IN TYPEOF(adr.assigned_date))) )) <= 1) OR
      (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
      ((adr.role.name = 'run start') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.CALENDAR_DATE'
      IN TYPEOF(adr.assigned_date))) )) <= 1);
```



```
(*    dm_program_run to date_time (as run_date) *)

wr11: (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS')|
      ((adr.role.name = 'run date') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.DATE_AND_TIME'
      IN TYPEOF(adr.assigned_date))) )) = 1) OR
      (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS')|
      ((adr.role.name = 'run date') AND (
      'DIMENSIONAL_INSPECTION_SCHEMA.CALENDAR_DATE'
      IN TYPEOF(adr.assigned_date))) )) = 1);

  END_ENTITY; -- dm_program_run
 (*
```

Formal proposition:

WR1: There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **measure_representation_item** and **qualified_representation_item** with a **name** of 'measurement temperature'.

WR2: There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **measure_representation_item** and **qualified_representation_item** with a **name** of 'workpiece temperature'.

WR3: There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **measure_representation_item** and **qualified_representation_item** with a **name** of 'measure humitity'.

WR4: There shall be at most one reference by a **action_resource** through **usage** with a name of 'part inspection status'.

WR5: There shall be at most one reference by a **action_resource** through **usage** with a name of 'url'.

WR6: There shall be exactly one reference by a **dm_program_identification** through **operations** attribute.

WR7: There shall be at least one reference by an **applied_organization_assignment** through the **items** attribute that references an **organization_role** through the **role** attribute with **name** of 'run administrator' and references an **organization** through the **assigned_organization** attribute, or there shall be at least one reference by an **applied_person_and_organization_assignment** through the **items** attribute that references an **person_and_organization_role** through the **role** attribute with **name** of 'run administrator' and references a **person_and_organization** through the **assigned_person_and_organization** attribute.

WR8: There shall be at most one reference by a **measurement_location** through **definition** attribute.

WR9: There shall be at least one reference by an **applied_date_assignment** through the **items** attribute that references an **date_role** through the **role** attribute with **name** of 'run end' and references an **date** through the **assigned_date** attribute, or there shall be at least one reference by an **applied_-date_assignment** through the **items** attribute that references an **date_role** through the **role** attribute with **name** of 'run end' and references a **calendar_date** through the **assigned_date** attribute.

WR10: There shall be at least one reference by an **applied_date_assignment** through the **items** attribute that references an **date_role** through the **role** attribute with **name** of 'run start' and references an **date** through the **assigned_date** attribute, or there shall be at least one reference by an **applied_-date_assignment** through the **items** attribute that references an **date_role** through the **role** attribute with **name** of 'run start' and references a **calendar_date** through the **assigned_date** attribute.

WR11: There shall be at least one reference by an **applied_date_assignment** through the **items** attribute that references an **date_role** through the **role** attribute with **name** of 'run date' and references an **date** through the **assigned_date** attribute, or there shall be at least one reference by an **applied_-date_assignment** through the **items** attribute that references an **date_role** through the **role** attribute with **name** of 'run date' and references a **calendar_date** through the **assigned_date** attribute.

5.2.3.1.51 dm_vector_parameter

A **dm_vector_parameter** is a type of **dm_parameter** that defines a defines vector parameters for dimensional feature (see 4.2.90).

EXPRESS Specification:

```
*)
  ENTITY dm_vector_parameter
    SUBTYPE OF (dm_result_parameter);
  WHERE

(*  dm_vector_parameter to cartesian_vector (as calculated_vector)  *)

  wr1:  SIZEOF(QUERY( repi <* SELF.items |
        ('DIMENSIONAL_INSPECTION_SCHEMA.VECTOR'
        IN TYPEOF (repi))))=1;
```



```
(*  dm_point_parameter to measurement_uncertainty (as value_uncertainty) *)

 wr2: SIZEOF(QUERY ( repi <* SELF.items | (
      (SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(repi)) = 2) AND
      (SIZEOF(QUERY ( uq <* repi.qualifiers | (
      (SIZEOF([
    'DIMENSIONAL_INSPECTION_SCHEMA.STANDARD_UNCERTAINTY'] * TYPEOF(uq))=1) AND
      (uq.measure_name = 'value uncertainty') AND
      (uq.description IN ['i uncertainty','j uncertainty', 'k uncertainty']) )
      )) <=3 )
      ))) <=1;

  END_ENTITY; -- dm_vector_parameter
  (*
```

Formal proposition:

WR1: There shall be exactly one **representation_item**s in its set of **items** that contains a **vector**.

WR2: There shall be zero to three reference through **items** to a **measure_representation_item** and **qualified_measure_with_unit** that references through the **qualifiers** attribute an **uncertainty_qualifier** with **measurement_name** of type **standard_uncertainty** or **expanded_uncertainty** with a **measure_-name** of 'calculated value' and a **description** of 'i uncertainty', 'j uncertainty' or 'k uncertainty'.

5.2.3.1.52 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'.

5.2.3.1.53 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 dimension specification';

  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
      (adr.assigned_document.description =
      'externally dimension specification')
      ) ) <= 1;

END_ENTITY;
 (*
```


Formal propositions:

WR1: The **source** attribute shall reference an **external_reference** with a **description** of 'externally defined dimension specification''.

WR2: The **externally_defined_dimension_definition** shall be reference by a through the **items** attribute by a **document_reference** of type **applied_document_reference** that references through the **assigned_-document** attribute a **document** with **description** of 'externally defined dimension specification'.

5.2.3.1.54 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**.

5.2.3.1.55 measurement_location

A **measurement_location** is a type of **action_property** that defines a defines run locations for a program run (see 4.2.167).

EXPRESS Specification:

```
*)
ENTITY measurement_location
   SUBTYPE OF (action_property);
WHERE
  wr1: ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PROGRAM_RUN'
  IN TYPEOF (SELF.definition));
END_ENTITY;
 (*
```

Formal proposition:

WR1: There shall be exactly one reference through the **definition** attribute to a **dm_program_run.**

5.2.3.1.56 ngon_shape_representation

An **ngon_shape_representation** specifies representation of a shape that is a volume defined as a ngon area of a defined length. The enclosed area is defined by three or more straight sides.

EXPRESS specification:

```
*)
  ENTITY ngon_shape_representation
    SUBTYPE OF (shape_representation_with_parameters);
    WHERE
 wr1: SIZEOF(SELF.items) = 5;
 wr2: SIZEOF(QUERY (it <* SELF.items | ((
      'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
      name = 'orientation')))) = 1;
 wr3: SIZEOF(QUERY (it <* SELF.items | ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'length')))) = 1;
 wr4: SIZEOF(QUERY (it <* SELF.items | ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'corner radius')))) = 1;
 wr5: SIZEOF(QUERY (it <* SELF.items | ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name IN ['circumscribed diameter',
      'diameter across flats'])))) = 1;
 wr6: SIZEOF(QUERY (it <* SELF.items | (((
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND ('DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN
      TYPEOF(it\measure_with_unit.value_component))) AND (it.name
      = 'number of sides')))) = 1;
  END_ENTITY; -- ngon_shape_representation
(*
```

Formal propositions:

WR1: The **ngon_shape_representation** shall contain exactly five **representation_item**s in its set of **items**.

WR2: One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **placement** with a **name** of 'orientation'.

WR3: One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

WR4: One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'corner radius'.

WR5: One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'circumscribed diameter' or 'diameter across flats'.

WR6: One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of sides'.

Informal propositions:

IP1: The location of the **ngon_shape_representation** shall be defined at the center of the enclosed area.
IP2: The **ngon_shape_representation** shall be defined with the enclosed area in the X-Y plane with one of the sides of the ngon parallel to the X direction intersecting the negative Y axis. The length is along the Z direction.

5.2.3.1.57 placed_data_target_feature

A **placed_datum_target_feature** is a type of **datum_target** that represents the implicit definition.

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,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);

  wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
        (('DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT'
        IN TYPEOF(it)) AND (it.name = 'orientation')) )) = 1)) ))
        = 0)) )) = 0);
```

```
wr4 : ((NOT (SELF.description = 'point')) OR
      (SIZEOF(QUERY ( pd <*  USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <*
      QUERY ( pdr <* USEDIN(pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(dtm_rep.used_representation.items) = 1)) ))
      = 0)) )) = 0));

wr5 : ((NOT (SELF.description IN ['line','circle'])) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(dtm_rep.used_representation.items) = 2)) ))
       = 0)) )) = 0));

wr6 : ((NOT (SELF.description = 'rectangle')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(dtm_rep.used_representation.items) = 3)) ))
      = 0)) )) = 0));

wr7 : ((NOT (SELF.description = 'circle')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
      ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'target diameter')) ))
      = 1)) )) = 0)) )) = 0));
```



```
  wr8 : ((NOT (SELF.description = 'line')) OR
        (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
        ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
        = 0)) )) = 0));

  wr9 : ((NOT (SELF.description = 'rectangle')) OR
        (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
        ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
        = 0)) )) = 0));

  wr10: ((NOT (SELF.description = 'rectangle')) OR
        (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
        ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'target width'))))= 1))))
        = 0)) ))=0));
  END_ENTITY; -- placed_datum_target_feature
(*
```

Formal propositions:

WR1: The **description** for the **placed_datum_target_feature** shall be either 'point', 'line', 'rectangle' or 'circle'.

WR2: A **placed_datum_target_feature** shall have exactly one implicit representation.

WR3: Exactly one **representation_item** used for the representation of the **placed_datum_target_-feature** shall be of type **placement** with a **name** of 'orientation'.

WR4: If the **placed_datum_target_feature** is a point, the **representation** shall contain exactly one **representation_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'.

5.2.3.1.58 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 |
     ('DIMENSIONAL_INSPECTION_SCHEMA.SET_REPRESENTATION_ITEM'
      IN TYPEOF (sri))) |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
      IN TYPEOF (mri))
      ))=2;

wr2:  SIZEOF(QUERY (mri <* QUERY(sri <* SELF.item_element |
     ('DIMENSIONAL_INSPECTION_SCHEMA.SET_REPRESENTATION_ITEM'
      IN TYPEOF (sri))) |
 (('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
  IN TYPEOF (mri)) AND
     (mri.name='lower limit'))
      ))=1;

wr3:  SIZEOF(QUERY (mri <* QUERY(sri <* SELF.item_element |
     ('DIMENSIONAL_INSPECTION_SCHEMA.SET_REPRESENTATION_ITEM'
      IN TYPEOF (sri))) |
 (('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
  IN TYPEOF (mri)) AND
     (mri.name='upper limit'))
      ))=1;

wr4: SIZEOF(QUERY(i1 <* SELF.item_element |
 ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
 IN TYPEOF (i1)) AND
    (SIZEOF (QUERY (i2 <* SELF.item_element |
 ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
 IN TYPEOF (i2)) AND
    (i1 :<>: i2) AND
    (i1\measure_with_unit.unit_component :=:
      i2\measure_with_unit.unit_component)
    )) = 1))) = 2 ;

END_ENTITY;
(*
```

Formal propositions:

WR1: The **set_representation_item** shall have exactly two **representation_items** of the same type in its set that is **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 Dimensional inspection schema imported entity modifications

5.2.3.2.1 application_context

The base definition of the **application_context** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **application_context** entity:

— application_context_requires_ap_definition (See 5.2.4.1).

5.2.3.2.2 application_protocol_definition

The base definition of the **application_protocol_definition** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **application_protocol_-definition** entity:

— application_context_requires_ap_definition (See 5.2.4.1).

5.2.3.2.3 chamfer

The base definition of the **chamfer** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **chamfer** entity:

— chamfer_requires_faces (See 5.2.4.2).

5.2.3.2.4 chamfer_offset

The base definition of the **chamfer_offset** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **chamfer_offset** entity:

— chamfer_offset_requires_faces (See 5.2.4.3).

5.2.3.2.5 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 5.2.4.23).

5.2.3.2.6 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 5.2.4.4).

5.2.3.2.7 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.

5.2.3.2.8 edge_round

The base definition of the **edge_round** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **edge_round** entity:

— edge_round_requires_faces (See 5.2.4.12).

5.2.3.2.9 externally_defined_feature_definition

The base definition of the **externally_defined_feature_definition** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **externally_defined_feature_-definition** entity:

— restrict_externally_defined_feature_definition (See 5.2.4.19).

5.2.3.2.10 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 5.2.4.13).

5.2.3.2.11 instanced_feature

The base definition of the **instanced_feature** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **instanced_feature** entity:

— machining_feature_life_cycle (See 5.2.4.14).

5.2.3.2.12 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 5.2.4.5).

5.2.3.2.13 precision_qualifier

The base definition of the **precision_qualifier** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **precision_qualifier t** entity:

— dependent_instantiable_ precision_qualifier (See 5.2.4.6).

5.2.3.2.14 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 rules defined in this part of ISO 10303 apply to the **product**entity:

— product_requires_version (see 5.2.4.15);

5.2.3.2.15 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:

— product_definition_formation_requires_security_classification (see 5.2.4.16);

— product_requires_version (see 5.2.4.15).

5.2.3.2.16 product_definition_representation

The base definition of the **product_definition_representation** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **product_definition_-representation** entity:

— chamfer_offset_requires_faces (See 5.2.4.3);

— edge_round_requires_faces (See 5.2.4.12);

— chamfer_requires_faces (See 5.2.4.2).

5.2.3.2.17 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 5.2.4.17).

5.2.3.2.18 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 5.2.4.10)

— restrict_security_classification_level (see 5.2.4.18).

5.2.3.2.19 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 5.2.4.20).

5.2.3.2.20 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 5.2.4.21).

5.2.3.2.21 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 5.2.4.9);

— shape_representation_subtype_exclusiveness (see5.2.4.22).

5.2.3.2.22 transition_feature

The base definition of the **transition_feature** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **transition_feature** entity:

— transition_feature_life_cycle (See 5.2.4.24);

— transition_feature_on_part_boundary (see 5.2.4.25).

5.2.3.2.23 type_qualifier

The base definition of the **type_qualifier** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **type_qualifier t** entity:

— dependent_instantiable_type_qualifier (See 5.2.4.7).

5.2.3.2.24 uncertainty_qualifier

The base definition of the **uncertainty_qualifier** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **uncertainty_qualifier t** entity:

— dependent_instantiable_uncertainty_qualifier (See 5.2.4.8).

5.2.4 Dimensional inspection schema rules

5.2.4.1 application_context_requires_ap_definition

The **application_context_requires_ap_definition** rule specifies that each instance of **application_-context** shall be referenced by exactly one **application_protocol_definition** that specifies this part of ISO 10303.

EXPRESS specification:

```
*)
RULE application_context_requires_ap_definition FOR
  (application_context, application_protocol_definition);
WHERE
  WR1: SIZEOF (QUERY (ac <* application_context |
       NOT (SIZEOF (QUERY (apd <* application_protocol_definition |
       (ac :=: apd.application)
       AND
       (apd.application_interpreted_model_schema_name =
       'dimensional inspection schema'))) = 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

application_context: the set of all instances of **application_context** entities.

application_protocol_definition: the set of all instances of **application_protocol_definition** entities.

Formal propositions:

WR1: For each instance of **application_context**, there shall be exactly one instance of **application_-protocol_definition** that references the instance of **application_context** as its **application** with a value of 'dimensional inspection schema' as its **application_interpreted_model_schema_name**.

5.2.4.2 chamfer_requires_faces

The **chamfer_requires_faces** rule specifies that all instances of **chamfer** require **face_shape_-representation**.

EXPRESS specification:

```
*)
 RULE chamfer_requires_faces FOR (chamfer,
           property_definition_representation);
```



```
 WHERE
    wr1: SIZEOF(QUERY ( er <* chamfer |
         (SIZEOF (QUERY (pd <* USEDIN (er,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
       IN TYPEOF (pdr.used_representation)) AND
       (pdr.used_representation.name = 'chamfer face')))=1)))=1)))=0;
END_RULE;
(*
```

Argument definitions:

chamfer: the set of all instances of **chamfer** entities.

property_definition_representation: the set of all instances of property_definition_representation entities.

Formal propositions:

WR1: The **chamfer** shall have exactly one **face_shape_representation**s in the role of the chamfer face.

5.2.4.3 chamfer_offset_requires_faces

The **chamfer_offset_requires_faces** rule specifies that all instances of **chamfer_offset** require **face_shape_representation**.

EXPRESS specification:

```
*)
RULE chamfer_offset_requires_faces FOR (chamfer_offset,
           property_definition_representation);
WHERE
        WR1: SIZEOF(QUERY ( co <* chamfer_offset |
         (( (co.description = 'first offset')) AND
         (SIZEOF(QUERY ( pd <* USEDIN(co,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
         (NOT  (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         (('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
         IN TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.name = 'first chamfer face')) )) = 1)) ))
         = 1)) ))=0;
```

```
    WR2:  SIZEOF(QUERY ( co <* chamfer_offset |
         (( (co.description = 'second offset')) AND
         (SIZEOF(QUERY ( pd <* USEDIN(co,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
         (NOT  (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         (('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
         IN TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.name = 'second chamfer face')) )) = 1)) ))
         = 1)) ))=0;
END_RULE;
(*
```

Argument definitions:

chamfer_offset: the set of all instances of **chamfer_offset** entities.

property_definition_representation: the set of all instances of property_definition_representation entities.

Formal propositions:

WR1: If the **description** of the **chamfer_offset** is 'first offset', the **chamfer_offset** shall have at most one one **face_shape_representation** with a **name** of 'first face shape' or if the **description** of the **chamfer_offset** is 'second offset', the **chamfer_offset** shall have at most one one **face_shape_-representation** with a **name** of 'first face shape'.

WR2: 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'.

5.2.4.4 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.

5.2.4.5 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.

5.2.4.6 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.

5.2.4.7 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.

5.2.4.8 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.

5.2.4.9 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.

5.2.4.10 dependent_instantiable_security_classification_level

The **dependent_instantiable_security_classification_level** rule specifies that all instances of **security_-classification_level** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_security_classification_level FOR
 (security_classification_level);
WHERE
  WR1: SIZEOF (QUERY (scl <* security_classification_level |
       NOT (SIZEOF (USEDIN (scl, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

security_classification_level: the set of all instances of **security_classification_level**.

Formal proposition:

WR1: For each instance of **security_classification_level**, there shall be a reference to the **security_-classification_level** instance from an attribute of another entity.

5.2.4.11 dimensional_measurement_part

The **dimensional_measurement_part** rule specifies that all instances of **product_definiton_formation** require constraint.

EXPRESS specification:

```
*)
RULE dimensional_measurement_part FOR
     (product_definition_formation,
      product_definition,
      product,
      applied_security_classification_assignment,
      applied_effectivity_assignment);

 WHERE

(*  product definition  *)

  wr1: SIZEOF (QUERY (pdf <* product_definition_formation |
       NOT (SIZEOF (QUERY (pd <* product_definition |
       ((pdf :=: pd.formation ) AND
       (pd.frame_of_reference.name='part definition'))))
        = 0 )  )) = 1;

(*  product_requires_version  *)

  WR2: SIZEOF (QUERY (prod <* product |
       NOT (SIZEOF (QUERY (pdf <* product_definition_formation |
       prod :=: pdf.of_product )) >= 1 ))) = 0;

(* product_definition_formation_requires_security_classification *)

  WR3: SIZEOF (QUERY (pdf <* product_definition_formation |
       NOT (SIZEOF (QUERY (asca <*
       applied_security_classification_assignment |
       pdf IN asca.items )) <= 1 ))) = 0;

(*   serial_number_DML *)

    wr4: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
             SIZEOF(QUERY ( aea <* applied_effectivity_assignment | ((pdf
             IN aea.items) AND (
             'DIMENSIONAL_INSPECTION_SCHEMA.SERIAL_NUMBERED_EFFECTIVITY' IN
              TYPEOF(aea.assigned_effectivity))) )) <= 1)) )) = 0;
```



```
(*   lot_number_DML *)

    wr5: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
            SIZEOF(QUERY ( aea <* applied_effectivity_assignment | ((pdf
            IN aea.items) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.LOT_EFFECTIVITY' IN TYPEOF(aea.
            assigned_effectivity))) )) <= 1)) )) = 0;

(* part to shape (as physical_form) *)

  wr6: SIZEOF (QUERY (pdf <* product_definition_formation |
       NOT (SIZEOF (QUERY (pd <* product_definition |
       ((pdf :=: pd.formation ) AND
       (SIZEOF(QUERY( prop_def <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
       IN TYPEOF(prop_def)) )) =1 )))) = 0)))=1;

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.

applied_security_classification_assignment: the set of all instances of **applied_security_-classification_assignment** entities.

applied_effectivity_assignment: the set of all instances of **applied_effectivity_assignment** entities.

Formal propositions:

WR1: For each **product_definition_formation** there shall be a reference from a **product_definition** through the **formation** attribute that reference a **product_definition_context** with **name** of ‘part definition’ through the **frame_of_reference** attribute.

WR2: 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**.

WR3: For each instance of **product_definition_formation**, there shall be exactly one instance of **applied_security_classification_assignment** that contains the instance of **product_definition_-formation** in its set of **items**.

WR4: For each instance of **product_definition_formation**, there shall be exactly one reference by an **applied_effectivity_assignment** through the **items** attribute that references a **serial_numbered_-effectivity** through the **assigned_effectivity** attribute.

WR5: For each instance of **product_definition_formation**, there shall be exactly one reference by an **applied_effectivity_assignment** through the **items** attribute that references a **lot_effectivity** through the **assigned_effectivity** attribute.

WR6: For each **product_definition_formation** there shall be a reference from a **product_definition** through the **formation** attribute that is referenced by a **property_definiton** of type **product_definition_-shape** through the **definition** attribute.

5.2.4.12 edge_round_requires_faces

The **edge_round_requires_faces** rule specifies that all instances of **edge_round** require **face_shape_-representation**.

EXPRESS specification:

```
*)
RULE edge_round_requires_faces FOR
      (edge_round, property_definition_representation);
WHERE
    wr1: SIZEOF(QUERY( er <* edge_round |
        (SIZEOF(QUERY ( pd <* USEDIN(er,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.'+
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
         IN TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.name = 'edge round face')) ))
         <= 1)) )) = 1)))=0;

    wr2: SIZEOF(QUERY( er <* edge_round |
        (SIZEOF(QUERY ( pd <* USEDIN(er,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.'+
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
         IN TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.name = 'first face shape')) ))
         <= 1)) )) = 1)))=0;

    wr3: SIZEOF(QUERY( er <* edge_round |
        (SIZEOF(QUERY ( pd <* USEDIN(er,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.'+
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
         IN TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.name = 'second face shape')) ))
         <= 1)) )) = 1)))=0;
  END_RULE; -- edge_round_requires_faces
(*
```


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'.

5.2.4.13 geometric_tolerance_subtype_exclusiveness

The **geometric_tolerance_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **geometric_tolerance** shall be only one of **angularity_tolerance**, **circular_runout_tolerance**, **concentricity_tolerance**, **cylindricity_tolerance**, **flatness_tolerance**, **line_profile_tolerance**, **parallelism_tolerance**, **perpendicularity_tolerance**, **position_tolerance**, **roundness_tolerance**, **straightness_tolerance**, **surface_profile_tolerance**, **symmetry_tolerance**, or **total_runout_tolerance**.

EXPRESS specification:

```
*)
RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);
WHERE
  WR1: SIZEOF (QUERY (gt <* geometric_tolerance |
       NOT (SIZEOF (TYPEOF (gt) *
       ['DIMENSIONAL_INSPECTION_SCHEMA.ANGULARITY_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_RUNOUT_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.CONCENTRICITY_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.CYLINDRICITY_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.FLATNESS_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.LINE_PROFILE_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.PARALLELISM_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.PERPENDICULARITY_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.POSITION_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDNESS_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.STRAIGHTNESS_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.SURFACE_PROFILE_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.SYMMETRY_TOLERANCE',
       'DIMENSIONAL_INSPECTION_SCHEMA.TOTAL_RUNOUT_TOLERANCE'])
       <= 2))) = 0;
END_RULE;
(*
```

Argument definitions:

geometric_tolerance: the set of all instances of **geometric_tolerance** entities.

Formal propositions:

WR1: Each instance of the subtypes of **geometric_tolerance** shall be one of **angularity_tolerance**, **circular_runout_tolerance**, **concentricity_tolerance**, **cylindricity_tolerance**, **flatness_tolerance**, **line_profile_tolerance**, **parallelism_tolerance**, **perpendicularity_tolerance**, **position_tolerance**, **roundness_tolerance**, **straightness_tolerance**, **surface_profile_tolerance**, **symmetry_tolerance**, or **total_runout_tolerance**.

5.2.4.14 machining_feature_life_cycle

The **machining_feature_life_cycle** rule specifies that each instance of **instanced_feature** shall be defined for the manufacturing planning stage of the part on which it is specified.

EXPRESS specification:

```
*)
RULE machining_feature_life_cycle FOR
 (instanced_feature);
WHERE
  WR1: SIZEOF (QUERY (mf <* instanced_feature |
       NOT (mf.of_shape.definition.frame_of_reference.life_cycle_stage =
       'manufacturing planning'))) = 0;
END_RULE;
(*
```

Argument definitions:

instanced_feature: the set of all instances of **instanced_feature** entities.

Formal propositions:

WR1: For each instance of **instanced_feature**, the **life_cycle_stage** of the **product_definition** for which it is defined has a value of 'manufacturing planning'.

5.2.4.15 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**.

5.2.4.16 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,
  applied_security_classification_assignment);
WHERE
  WR1: SIZEOF (QUERY (pdf <* product_definition_formation |
       NOT (SIZEOF (QUERY (asca <*
       applied_security_classification_assignment |
       pdf IN asca.items)) <= 1))) = 0;
END_RULE;
(*
```


Argument definitions:

product_definition_formation: the set of all instances of **product_definition_formation** entities.

applied_security_classification_assignment: the set of all instances of **applied_security_-classification_assignment** entities.

Formal propositions:

WR1: For each instance of **product_definition_formation**, there shall be exactly one instance of **applied_security_classification_assignment** that contains the instance of **product_definition_-formation** in its set of **items**.

5.2.4.17 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) *
       ['DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION',
       'DIMENSIONAL_INSPECTION_SCHEMA.DEFINITIONAL_REPRESENTATION',
       'DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION',
       'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION',
       'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION',
       'DIMENSIONAL_INSPECTION_SCHEMA.LOCATION_SHAPE_REPRESENTATION',
       'DIMENSIONAL_INSPECTION_SCHEMA.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.

5.2.4.18 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.

5.2.4.19 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,
         'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
         ('DIMENSIONAL_INSPECTION_SCHEMA.DOCUMENT_FILE'
         IN TYPEOF(adr.assigned_document)))) >=1)))) =0) OR
         (SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
         (NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
         'DIMENSIONAL_INSPECTION_SCHEMA.'+
         'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS')|
         ('DIMENSIONAL_INSPECTION_SCHEMA.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.

5.2.4.20 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**, or **datum_target**.

EXPRESS specification:

```
*)
RULE shape_aspect_subtype_exclusiveness FOR (shape_aspect);
  WHERE
    wr1: (SIZEOF(QUERY (sr <* shape_aspect | (NOT (SIZEOF(TYPEOF(sr) * [
             'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT',
             'DIMENSIONAL_INSPECTION_SCHEMA.SLOT_END',
             'DIMENSIONAL_INSPECTION_SCHEMA.POCKET_BOTTOM',
             'DIMENSIONAL_INSPECTION_SCHEMA.PROFILE_FLOOR',
             'DIMENSIONAL_INSPECTION_SCHEMA.RIB_TOP_FLOOR',
             'DIMENSIONAL_INSPECTION_SCHEMA.BOSS_TOP',
             'DIMENSIONAL_INSPECTION_SCHEMA.HOLE_BOTTOM',
             'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_AREA',
             'DIMENSIONAL_INSPECTION_SCHEMA.TAPER',
             'DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER_OFFSET',
             'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.NGON_CLOSED_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_PATH_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.LINEAR_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.TEE_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_U_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_PATH_PROFILE',
             'DIMENSIONAL_INSPECTION_SCHEMA.INSTANCED_FEATURE',
             'DIMENSIONAL_INSPECTION_SCHEMA.REPLICATE_FEATURE',
             'DIMENSIONAL_INSPECTION_SCHEMA.TRANSITION_FEATURE',
             'DIMENSIONAL_INSPECTION_SCHEMA.DATUM',
             'DIMENSIONAL_INSPECTION_SCHEMA.DATUM_FEATURE',
             'DIMENSIONAL_INSPECTION_SCHEMA.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**.

5.2.4.21 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) *
       ['DIMENSIONAL_INSPECTION_SCHEMA.DIMENSIONAL_LOCATION',
       'DIMENSIONAL_INSPECTION_SCHEMA.GEOMETRIC_TOLERANCE_RELATIONSHIP',
       'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP',
       'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'])
       <= 2))) = 0;
END_RULE;
(*
```

Argument definitions:

shape_aspect_relationship: the set of all instances of **shape_aspect_relationship** entities.

Formal propositions:

WR1: Each instance of the subtypes of **shape_aspect_relationship** shall be one of the **dimensional_-location**, **geometric_tolerance_relationship**, **feature_component_relationship**, or **shape_defining_-relationship**.

5.2.4.22 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) *
      ['DIMENSIONAL_INSPECTION_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION',
     'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS',
     'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DIMENSION_REPRESENTATION'])
       <= 2))) = 0;
END_RULE;
(*
```


Argument definitions:

shape_representation: the set of all instances of **shape_representation** entities.

Formal propositions:

WR1: Each instance of the subtypes of **shape_representation** shall be one of the **advanced_brep_-shape_representation**, **shape_representation_with_parameters**, or **shape_dimension_representation**.

5.2.4.23 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([
         'DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_DEFINITION',
         'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_DEFINITION',
         'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'] *
         TYPEOF(csa)) = 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**.

5.2.4.24 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'.

5.2.4.25 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;
(*
```


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:

— dimensional inspection 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 dimensional_inspection_schema;

  CONSTANT
  dummy_gri : geometric_representation_item := representation_item('') ||
                 geometric_representation_item();
  dummy_tri : topological_representation_item := representation_item('')
                 || topological_representation_item();
  END_CONSTANT;

  TYPE action_item = SELECT
    (product_definition);
  END_TYPE; -- action_item

  TYPE ahead_or_behind = ENUMERATION OF
    (exact,
     ahead,
     behind);
  END_TYPE; -- ahead_or_behind

  TYPE amount_of_substance_measure = REAL;
  END_TYPE; -- amount_of_substance_measure

  TYPE angle_relator = ENUMERATION OF
    (equal,
     small,
     large);
  END_TYPE; -- angle_relator

  TYPE area_measure = REAL;
  END_TYPE; -- area_measure

  TYPE attribute_type = SELECT
    (label,
     text);
  END_TYPE; -- attribute_type

  TYPE axis2_placement = SELECT
    (axis2_placement_2d,
     axis2_placement_3d);
  END_TYPE; -- axis2_placement

  TYPE b_spline_curve_form = ENUMERATION OF
    (elliptic_arc,
```



```
   polyline_form,
   parabolic_arc,
   circular_arc,
   unspecified,
   hyperbolic_arc);
END_TYPE; -- b_spline_curve_form

TYPE b_spline_surface_form = ENUMERATION OF
  (surf_of_linear_extrusion,
   plane_surf,
   generalised_cone,
   toroidal_surf,
   conical_surf,
   spherical_surf,
   unspecified,
   ruled_surf,
   surf_of_revolution,
   cylindrical_surf,
   quadric_surf);
END_TYPE; -- b_spline_surface_form

TYPE boolean_operand = SELECT
  (solid_model);
END_TYPE; -- boolean_operand

TYPE celsius_temperature_measure = REAL;
END_TYPE; -- celsius_temperature_measure

TYPE characterized_action_definition = SELECT
  (action,
   action_method,
   action_relationship);
END_TYPE; -- characterized_action_definition

TYPE characterized_definition = SELECT
  (characterized_object,
   characterized_product_definition,
   shape_definition);
END_TYPE; -- characterized_definition

TYPE characterized_product_definition = SELECT
  (product_definition);
END_TYPE; -- characterized_product_definition

TYPE characterized_resource_definition = SELECT
  (action_resource,
   action_resource_requirement);
END_TYPE; -- characterized_resource_definition

TYPE compound_item_definition = SELECT
  (set_representation_item);
END_TYPE; -- compound_item_definition

TYPE count_measure = NUMBER;
END_TYPE; -- count_measure
```

```
TYPE curve_on_surface = SELECT
  (pcurve,
   surface_curve,
   composite_curve_on_surface);
END_TYPE; -- curve_on_surface

TYPE date_item = SELECT
  (dm_program_run);
END_TYPE; -- date_item

TYPE date_time_or_event_occurrence = SELECT
  (date_time_select);
END_TYPE; -- date_time_or_event_occurrence

TYPE date_time_select = SELECT
  (date,
   local_time,
   date_and_time);
END_TYPE; -- date_time_select

TYPE day_in_month_number = INTEGER;
WHERE
  wr1: (1 <= SELF) AND (SELF <= 31);
END_TYPE; -- day_in_month_number

TYPE derived_property_select = SELECT
  (property_definition,
   action_property,
   resource_property);
END_TYPE; -- derived_property_select

TYPE description_attribute_select = SELECT
  (application_context,
   date_role,
   effectivity,
   external_source,
   organization_role,
   person_and_organization_role,
   person_and_organization,
   property_definition_representation,
   representation);
END_TYPE; -- description_attribute_select

TYPE dimension_count = INTEGER;
WHERE
  wr1: SELF > 0;
END_TYPE; -- dimension_count

TYPE dimensional_characteristic = SELECT
  (dimensional_location,
   dimensional_size);
END_TYPE; -- dimensional_characteristic

TYPE document_reference_item = SELECT
  (product_definition_formation,
   externally_defined_feature_definition,
```



```
   externally_defined_dimension_definition);
END_TYPE; -- document_reference_item

TYPE effectivity_item = SELECT
  (product_definition_formation);
END_TYPE; -- effectivity_item

TYPE electric_current_measure = REAL;
END_TYPE; -- electric_current_measure

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;

TYPE hour_in_day = INTEGER;
WHERE
  wr1: (0 <= SELF) AND (SELF < 24);
END_TYPE; -- hour_in_day

TYPE id_attribute_select = SELECT
  (action,
   address,
   group,
   property_definition,
   shape_aspect,
   shape_aspect_relationship,
   application_context,
   organizational_project,
   representation);
END_TYPE; -- id_attribute_select

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 luminous_intensity_measure = REAL;
END_TYPE; -- luminous_intensity_measure

TYPE mass_measure = REAL;
END_TYPE; -- mass_measure

TYPE measure_value = SELECT
  (length_measure,
   mass_measure,
   time_measure,
   electric_current_measure,
   thermodynamic_temperature_measure,
   celsius_temperature_measure,
   amount_of_substance_measure,
   luminous_intensity_measure,
   plane_angle_measure,
   solid_angle_measure,
   area_measure,
   volume_measure,
   ratio_measure,
   parameter_value,
   positive_length_measure,
   positive_plane_angle_measure,
   count_measure);
END_TYPE; -- measure_value

TYPE minute_in_hour = INTEGER;
WHERE
  wr1: (0 <= SELF) AND (SELF <= 59);
END_TYPE; -- minute_in_hour

TYPE month_in_year_number = INTEGER;
WHERE
  wr1: (1 <= SELF) AND (SELF <= 12);
END_TYPE; -- month_in_year_number

TYPE name_attribute_select = SELECT
  (address,
   derived_unit,
   effectivity,
   person_and_organization,
   product_definition,
```

```
    property_definition_representation);
END_TYPE; -- name_attribute_select

TYPE organization_item = SELECT
   (dm_program_identification,
    dm_program_run);
END_TYPE; -- organization_item

TYPE parameter_value = REAL;
END_TYPE; -- parameter_value

TYPE pcurve_or_surface = SELECT
   (pcurve,
    surface);
END_TYPE; -- pcurve_or_surface

TYPE person_and_organization_item = SELECT
   (dm_program_identification,
    dm_program_run);
END_TYPE; -- person_and_organization_item

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 property_or_shape_select = SELECT
   (property_definition,
    shape_definition);
END_TYPE; -- property_or_shape_select
```

```
TYPE ratio_measure = REAL;
END_TYPE; -- ratio_measure

TYPE represented_definition = SELECT
  (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,
   document_reference,
   effectivity_assignment,
   group_assignment,
   security_classification_assignment);
END_TYPE; -- role_select

TYPE second_in_minute = REAL;
WHERE
  wr1: (0 <= SELF) AND (SELF <= 60);
END_TYPE; -- second_in_minute

TYPE security_classification_item = SELECT
  (product_definition_formation);
END_TYPE; -- security_classification_item

TYPE set_of_reversible_topology_item = SET [0:?] OF
            reversible_topology_item;
END_TYPE; -- set_of_reversible_topology_item

TYPE set_representation_item = SET [1:?] OF representation_item;
END_TYPE; -- set_representation_item

TYPE shape_definition = SELECT
  (product_definition_shape,
   shape_aspect,
   shape_aspect_relationship);
END_TYPE; -- shape_definition
```

```
TYPE shell = SELECT
  (open_shell,
   closed_shell);
END_TYPE; -- shell

TYPE si_prefix = ENUMERATION OF
  (exa,
   pico,
   mega,
   femto,
   atto,
   centi,
   nano,
   hecto,
   micro,
   tera,
   giga,
   milli,
   peta,
   deci,
   kilo,
   deca);
END_TYPE; -- si_prefix

TYPE si_unit_name = ENUMERATION OF
  (hertz,
   degree_celsius,
   siemens,
   sievert,
   lux,
   watt,
   ohm,
   second,
   becquerel,
   pascal,
   henry,
   tesla,
   volt,
   joule,
   kelvin,
   ampere,
   gram,
   steradian,
   mole,
   lumen,
   gray,
   candela,
   farad,
   radian,
   newton,
   metre,
   weber,
   coulomb);
END_TYPE; -- si_unit_name

TYPE solid_angle_measure = REAL;
```

```
END_TYPE; -- solid_angle_measure

TYPE source_item = SELECT
  (identifier);
END_TYPE; -- source_item

TYPE supported_item = SELECT
  (action,
   action_method);
END_TYPE; -- supported_item

TYPE text = STRING;
END_TYPE; -- text

TYPE thermodynamic_temperature_measure = REAL;
END_TYPE; -- thermodynamic_temperature_measure

TYPE time_measure = REAL;
END_TYPE; -- time_measure

TYPE tolerance_method_definition = SELECT
  (tolerance_value,
   limits_and_fits);
END_TYPE; -- tolerance_method_definition

TYPE tolerance_select = SELECT
  (geometric_tolerance,
   plus_minus_tolerance);
END_TYPE; -- tolerance_select

TYPE transformation = SELECT
  (functionally_defined_transformation);
END_TYPE; -- transformation

TYPE transition_code = ENUMERATION OF
  (discontinuous,
   cont_same_gradient_same_curvature,
   cont_same_gradient,
   continuous);
END_TYPE; -- transition_code

TYPE trimming_select = SELECT
  (cartesian_point,
   parameter_value);
END_TYPE; -- trimming_select

TYPE unit = SELECT
  (named_unit,
   derived_unit);
END_TYPE; -- unit

TYPE value_qualifier = SELECT
  (precision_qualifier,
   type_qualifier,
   uncertainty_qualifier);
END_TYPE; -- value_qualifier
```



```
TYPE vector_or_direction = SELECT
  (vector,
   direction);
END_TYPE; -- vector_or_direction

TYPE volume_measure = REAL;
END_TYPE; -- volume_measure

TYPE 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,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- action

ENTITY action_assignment
  ABSTRACT SUPERTYPE;
    assigned_action : action;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- action_assignment

ENTITY action_method;
    name        : label;
    description : OPTIONAL text;
    consequence : text;
    purpose     : text;
END_ENTITY; -- action_method

ENTITY action_property;
    name        : label;
    description : text;
    definition  : characterized_action_definition;
END_ENTITY; -- action_property

ENTITY action_property_representation;
    name           : label;
    description    : text;
    property       : action_property;
    representation : representation;
END_ENTITY; -- action_property_representation

ENTITY action_relationship;
    name            : label;
    description     : OPTIONAL text;
```

```
      relating_action : action;
      related_action  : action;
  END_ENTITY; -- action_relationship

  ENTITY action_resource;
      name        : label;
      description : OPTIONAL text;
      usage       : SET [1:?] OF supported_item;
      kind        : action_resource_type;
  END_ENTITY; -- action_resource

  ENTITY action_resource_requirement;
      name        : label;
      description : text;
      kind        : resource_requirement_type;
      OPERATIONS  : SET [1:?] OF characterized_action_definition;
  END_ENTITY; -- action_resource_requirement

  ENTITY action_resource_type;
      name : label;
  END_ENTITY; -- action_resource_type

  ENTITY address;
      internal_location       : OPTIONAL label;
      street_number           : OPTIONAL label;
      street                  : OPTIONAL label;
      postal_box              : OPTIONAL label;
      town                    : OPTIONAL label;
      region                  : OPTIONAL label;
      postal_code             : OPTIONAL label;
      country                 : OPTIONAL label;
      facsimile_number        : OPTIONAL label;
      telephone_number        : OPTIONAL label;
      electronic_mail_address : OPTIONAL label;
      telex_number            : OPTIONAL label;
    DERIVE
      name : label := get_name_value(SELF);
      url  : identifier := get_id_value(SELF);
    WHERE
      wr1: ((((((((((EXISTS(internal_location) OR EXISTS(street_number))
               OR EXISTS(street)) OR EXISTS(postal_box)) OR EXISTS(town))
               OR EXISTS(region)) OR EXISTS(postal_code)) OR EXISTS(country))
               OR EXISTS(facsimile_number)) OR EXISTS(telephone_number)) OR
                EXISTS(electronic_mail_address)) OR EXISTS(telex_number);
  END_ENTITY; -- address

  ENTITY advanced_brep_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MANIFOLD_SOLID_BREP',
               'DIMENSIONAL_INSPECTION_SCHEMA.FACETED_BREP',
               'DIMENSIONAL_INSPECTION_SCHEMA.MAPPED_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.AXIS2_PLACEMENT_3D'] *
               TYPEOF(it)) = 1)) )) = 0;
      wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
```



```
             'DIMENSIONAL_INSPECTION_SCHEMA.MANIFOLD_SOLID_BREP',
             'DIMENSIONAL_INSPECTION_SCHEMA.MAPPED_ITEM'] * TYPEOF(it)) =
              1) )) > 0;
    wr3: SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
             'DIMENSIONAL_INSPECTION_SCHEMA.MANIFOLD_SOLID_BREP' IN
             TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb)
              | (NOT (SIZEOF(QUERY ( fcs <* csh\connected_face_set.
             cfs_faces | (NOT (
          'DIMENSIONAL_INSPECTION_SCHEMA.ADVANCED_FACE' IN TYPEOF(fcs))) ))
             = 0)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
             'DIMENSIONAL_INSPECTION_SCHEMA.MANIFOLD_SOLID_BREP' IN
             TYPEOF(it)) ) | (
             'DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_CLOSED_SHELL' IN
             TYPEOF(msb\manifold_solid_brep.outer)) )) = 0;
    wr5: SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
           'DIMENSIONAL_INSPECTION_SCHEMA.BREP_WITH_VOIDS' IN TYPEOF(it)) )
              | (NOT (SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids |
             csh\oriented_closed_shell.orientation )) = 0)) )) = 0;
    wr6: SIZEOF(QUERY ( mi <* QUERY ( it <* items | (
      'DIMENSIONAL_INSPECTION_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) )
| (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION'
             IN TYPEOF(mi\mapped_item.mapping_source.
             mapped_representation))) )) = 0;
END_ENTITY; -- advanced_brep_shape_representation

ENTITY advanced_face
  SUBTYPE OF (face_surface);
  WHERE
    wr1 : SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.ELEMENTARY_SURFACE',
              'DIMENSIONAL_INSPECTION_SCHEMA.B_SPLINE_SURFACE',
              'DIMENSIONAL_INSPECTION_SCHEMA.SWEPT_SURFACE'] * TYPEOF(
              face_geometry)) = 1;
    wr2 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
              'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.
              bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
              .edge_list | (NOT (
              'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_CURVE' IN TYPEOF(oe\
              oriented_edge.edge_element))) )) = 0)) )) = 0;
    wr3 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
              'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.
              bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
              .edge_list | (NOT (SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.LINE',
              'DIMENSIONAL_INSPECTION_SCHEMA.CONIC',
              'DIMENSIONAL_INSPECTION_SCHEMA.POLYLINE',
              'DIMENSIONAL_INSPECTION_SCHEMA.SURFACE_CURVE',
              'DIMENSIONAL_INSPECTION_SCHEMA.B_SPLINE_CURVE'] * TYPEOF(oe
              .edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) ))
              = 0;
    wr4 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
              'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.
              bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
              .edge_list | (NOT (((
              'DIMENSIONAL_INSPECTION_SCHEMA.VERTEX_POINT' IN TYPEOF(oe\
              edge.edge_start)) AND (
```

```
                'DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT' IN TYPEOF(
                oe\edge.edge_start\vertex_point.vertex_geometry))) AND ((
                'DIMENSIONAL_INSPECTION_SCHEMA.VERTEX_POINT' IN TYPEOF(oe\
                edge.edge_end)) AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT' IN TYPEOF(
               oe\edge.edge_end\vertex_point.vertex_geometry))))) )) = 0)) ))
                = 0;
     wr5 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.
                bound)) ) | ('DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_PATH'
                IN TYPEOF(elp_fbnds.bound)) )) = 0;
     wr6 : (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.SWEPT_SURFACE' IN TYPEOF(
                face_geometry))) OR (SIZEOF([
                'DIMENSIONAL_INSPECTION_SCHEMA.LINE',
                'DIMENSIONAL_INSPECTION_SCHEMA.CONIC',
                'DIMENSIONAL_INSPECTION_SCHEMA.POLYLINE',
                'DIMENSIONAL_INSPECTION_SCHEMA.B_SPLINE_CURVE'] * TYPEOF(
                face_geometry\swept_surface.swept_curve)) = 1);
     wr7 : SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* bounds | (
                'DIMENSIONAL_INSPECTION_SCHEMA.VERTEX_LOOP' IN TYPEOF(bnds.
                bound)) ) | (NOT ((
                'DIMENSIONAL_INSPECTION_SCHEMA.VERTEX_POINT' IN TYPEOF(
                vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex)) AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT' IN TYPEOF(
                vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex\
                vertex_point.vertex_geometry)))) )) = 0;
     wr8 : SIZEOF(QUERY ( bnd <* bounds | (NOT (SIZEOF([
                'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_LOOP',
                'DIMENSIONAL_INSPECTION_SCHEMA.VERTEX_LOOP'] * TYPEOF(bnd.
                bound)) = 1)) )) = 0;
     wr9 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.
                bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
                .edge_list | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.SURFACE_CURVE' IN TYPEOF(oe\
                oriented_edge.edge_element\edge_curve.edge_geometry)) AND (
                NOT (SIZEOF(QUERY ( sc_ag <* oe.edge_element\edge_curve.
                edge_geometry\surface_curve.associated_geometry | (NOT (
                'DIMENSIONAL_INSPECTION_SCHEMA.PCURVE' IN TYPEOF(sc_ag))) ))
                = 0))) )) = 0)) )) = 0;
     wr10: ((NOT ('DIMENSIONAL_INSPECTION_SCHEMA.SWEPT_SURFACE' IN
                TYPEOF(face_geometry))) OR ((NOT (
                'DIMENSIONAL_INSPECTION_SCHEMA.POLYLINE' IN TYPEOF(
                face_geometry\swept_surface.swept_curve))) OR (SIZEOF(
                face_geometry\swept_surface.swept_curve\polyline.points) >=
                 3))) AND (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <*
                bounds | ('DIMENSIONAL_INSPECTION_SCHEMA.EDGE_LOOP' IN
                TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
                elp_fbnds.bound\path.edge_list | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.POLYLINE' IN TYPEOF(oe\
                oriented_edge.edge_element\edge_curve.edge_geometry)) AND (
                NOT (SIZEOF(oe\oriented_edge.edge_element\edge_curve.
                edge_geometry\polyline.points) >= 3))) )) = 0)) )) = 0);
  END_ENTITY; -- advanced_face

  ENTITY angular_location
```



```
    SUBTYPE OF (dimensional_location);
      angle_selection : angle_relator;
  END_ENTITY; -- angular_location

  ENTITY angular_size
    SUBTYPE OF (dimensional_size);
      angle_selection : angle_relator;
  END_ENTITY; -- angular_size

  ENTITY angularity_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
      wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
               datum_system) < 3;
  END_ENTITY; -- angularity_tolerance

  ENTITY apex
    SUBTYPE OF (derived_shape_aspect);
  END_ENTITY; -- apex

  ENTITY application_context;
      application : label;
    DERIVE
      description : text := get_description_value(SELF);
      id          : identifier := get_id_value(SELF);
    INVERSE
      context_elements : SET [1:?] OF application_context_element FOR
                           frame_of_reference;
    WHERE
      wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
      wr2: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
  END_ENTITY; -- application_context

  ENTITY application_context_element
    SUPERTYPE OF (ONEOF (product_context,product_definition_context));
      name               : label;
      frame_of_reference : application_context;
  END_ENTITY; -- application_context_element

  ENTITY application_protocol_definition;
      status                                    : label;
      application_interpreted_model_schema_name : label;
      application_protocol_year                 : year_number;
      application                               : application_context;
  END_ENTITY; -- application_protocol_definition

  ENTITY applied_action_assignment
    SUBTYPE OF (action_assignment);
      items : SET [1:?] OF action_item;
  END_ENTITY; -- applied_action_assignment

  ENTITY applied_area
    SUBTYPE OF (shape_aspect);
    WHERE
```

```
wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(SELF.of_shape);
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) )) = 1)) )) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT ((2 <= SIZEOF(impl_rep.
         used_representation.items)) AND (SIZEOF(impl_rep.
         used_representation.items) <= 3))) )) = 0)) )) = 0;
wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
         used_representation.items | (NOT (srwp_i.name IN [
       'orientation','effective length','maximum length'])) )) > 0)) ))
        = 0) )) = 0;
wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | ((SIZEOF([
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
         'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'effective length')) )) = 1)) ))
         = 0)) )) <= 1;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | ((SIZEOF([
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
         'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'maximum length')) )) <= 1)) ))
```



```
              = 0)) )) = 0;
     wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
               (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
END_ENTITY; -- applied_area

ENTITY applied_date_assignment
  SUBTYPE OF (date_assignment);
    items : SET [1:?] OF date_item;
END_ENTITY; -- applied_date_assignment

ENTITY applied_document_reference
  SUBTYPE OF (document_reference);
    items : SET [1:?] OF document_reference_item;
END_ENTITY; -- applied_document_reference

ENTITY applied_document_usage_constraint_assignment
  SUBTYPE OF (document_usage_constraint_assignment);
    items : SET [1:?] OF document_reference_item;
END_ENTITY; -- applied_document_usage_constraint_assignment

ENTITY applied_group_assignment
      SUBTYPE OF (group_assignment);
      items : SET [1:?] OF group_item;
END_ENTITY; -- applied_group_assignment


ENTITY applied_effectivity_assignment
  SUBTYPE OF (effectivity_assignment);
    items : SET [1:?] OF effectivity_item;
END_ENTITY; -- applied_effectivity_assignment

ENTITY applied_organization_assignment
  SUBTYPE OF (organization_assignment);
    items : SET [1:?] OF organization_item;
END_ENTITY; -- applied_organization_assignment

ENTITY applied_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
    items : SET [1:?] OF person_and_organization_item;
END_ENTITY; -- applied_person_and_organization_assignment

ENTITY applied_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
    items : SET [1:?] OF security_classification_item;
END_ENTITY; -- applied_security_classification_assignment

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: ((('DIMENSIONAL_INSPECTION_SCHEMA.UNIFORM_CURVE' IN TYPEOF(SELF))
               OR ('DIMENSIONAL_INSPECTION_SCHEMA.QUASI_UNIFORM_CURVE' IN
               TYPEOF(SELF))) OR (
               'DIMENSIONAL_INSPECTION_SCHEMA.BEZIER_CURVE' IN TYPEOF(SELF)))
               OR (
               'DIMENSIONAL_INSPECTION_SCHEMA.B_SPLINE_CURVE_WITH_KNOTS' IN
                TYPEOF(SELF));
```

```
END_ENTITY; -- b_spline_curve

ENTITY b_spline_curve_with_knots
  SUBTYPE OF (b_spline_curve);
    knot_multiplicities : LIST [2:?] OF INTEGER;
    knots               : LIST [2:?] OF parameter_value;
    knot_spec           : knot_type;
  DERIVE
    upper_index_on_knots : INTEGER := SIZEOF(knots);
  WHERE
    wr1: constraints_param_b_spline(degree,upper_index_on_knots,
             upper_index_on_control_points,knot_multiplicities,knots);
    wr2: SIZEOF(knot_multiplicities) = upper_index_on_knots;
END_ENTITY; -- b_spline_curve_with_knots

ENTITY b_spline_surface
  SUPERTYPE OF (ONEOF (b_spline_surface_with_knots,uniform_surface,
      quasi_uniform_surface,bezier_surface) ANDOR
      rational_b_spline_surface)
  SUBTYPE OF (bounded_surface);
    u_degree            : INTEGER;
    v_degree            : INTEGER;
    control_points_list : LIST [2:?] OF LIST [2:?] OF cartesian_point;
    surface_form        : b_spline_surface_form;
    u_closed            : LOGICAL;
    v_closed            : LOGICAL;
    self_intersect      : LOGICAL;
  DERIVE
    u_upper        : INTEGER := SIZEOF(control_points_list) - 1;
    v_upper        : INTEGER := SIZEOF(control_points_list[1]) - 1;
    control_points : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF
                         cartesian_point := make_array_of_array(
                        control_points_list,0,u_upper,0,v_upper);
  WHERE
    wr1: ((('DIMENSIONAL_INSPECTION_SCHEMA.UNIFORM_SURFACE' IN TYPEOF(
             SELF)) OR (
             'DIMENSIONAL_INSPECTION_SCHEMA.QUASI_UNIFORM_SURFACE' IN
             TYPEOF(SELF))) OR (
             'DIMENSIONAL_INSPECTION_SCHEMA.BEZIER_SURFACE' IN TYPEOF(
             SELF))) OR (
             'DIMENSIONAL_INSPECTION_SCHEMA.B_SPLINE_SURFACE_WITH_KNOTS'
             IN TYPEOF(SELF));
END_ENTITY; -- b_spline_surface

ENTITY b_spline_surface_with_knots
  SUBTYPE OF (b_spline_surface);
    u_multiplicities : LIST [2:?] OF INTEGER;
    v_multiplicities : LIST [2:?] OF INTEGER;
    u_knots          : LIST [2:?] OF parameter_value;
    v_knots          : LIST [2:?] OF parameter_value;
    knot_spec        : knot_type;
  DERIVE
    knot_u_upper : INTEGER := SIZEOF(u_knots);
    knot_v_upper : INTEGER := SIZEOF(v_knots);
  WHERE
    wr1: constraints_param_b_spline(SELF\b_spline_surface.u_degree,
```

```
             knot_u_upper,SELF\b_spline_surface.u_upper,u_multiplicities,
             u_knots);
    wr2: constraints_param_b_spline(SELF\b_spline_surface.v_degree,
             knot_v_upper,SELF\b_spline_surface.v_upper,v_multiplicities,
             v_knots);
    wr3: SIZEOF(u_multiplicities) = knot_u_upper;
    wr4: SIZEOF(v_multiplicities) = knot_v_upper;
END_ENTITY; -- b_spline_surface_with_knots

ENTITY bezier_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- bezier_curve

ENTITY bezier_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- bezier_surface

ENTITY block_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
    wr1: SIZEOF(SELF.items) = 4;
    wr2: SIZEOF(QUERY ( it <* SELF.items | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
              (it.name = 'orientation')) )) = 1;
    wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
    wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1;
    wr5: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'height')) )) = 1;
END_ENTITY; -- block_shape_representation

ENTITY boss
  SUBTYPE OF (feature_definition);
  WHERE
    wr1 : SELF\characterized_object.description IN ['circular','complex',
              'rectangular'];
    wr2 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'boss height occurrence') AND (SIZEOF(
              QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'path feature component usage') AND ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (((
```



```
        'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT' IN
        TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
        relating_shape_aspect.description = 'linear')) AND (sdr.
        name = 'boss height')) )) = 1)) )) = 1)) )) = 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) AND ((1 <= SIZEOF(pdr.
        used_representation.items)) AND (SIZEOF(pdr.
        used_representation.items) <= 2))) )) = 1) )) = 1;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
        used_representation.items | ((srwp_i.name = 'orientation')
        OR (srwp_i.name = 'fillet radius')) )) = SIZEOF(pdr.
        used_representation.items))) )) = 1) )) = 1;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
     'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'fillet radius')) )) <= 1)) ))
        = 0)) )) = 0;
wr6 : (NOT (SELF\characterized_object.description = 'circular')) OR
        (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (
        'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
        sa_occ.description = 'circular profile occurrence') AND (
        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND
     ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
        IN TYPEOF(sar))) ) | (
        'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
        TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
wr7 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

```
                | (
               'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'top condition occurrence') AND (
               SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT
')
                | ((sar.description = 'boss top usage') AND
          ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (
               'DIMENSIONAL_INSPECTION_SCHEMA.BOSS_TOP' IN TYPEOF(fcr.
               relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
      wr8 : (NOT (SELF\characterized_object.description = 'circular')) OR
               (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (
               'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'change in diameter occurrence') AND (
               SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'taper usage') AND
         ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.TAPER' IN TYPEOF(fcr.
               related_shape_aspect)) AND (
               'DIMENSIONAL_INSPECTION_SCHEMA.BOSS' IN TYPEOF(fcr.
               relating_shape_aspect))) )) = 1)) )) <= 1)) )) = 0);
      wr9 : (NOT (SELF\characterized_object.description = 'complex')) OR (
               SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (
               'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'enclosed boundary occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND
         ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.NGON_CLOSED_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_PATH_PROFILE'] *
               TYPEOF(sdr.relating_shape_aspect)) = 1) )) = 1)) )) = 1)) ))
               = 0);
      wr10: (NOT (SELF\characterized_object.description IN ['complex',
               'rectangular'])) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
               USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (
               'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
```



```
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'change in boundary occurrence') AND (
              SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATING_SHAPE_ASPECT') | ((sar.description =
              'taper usage') AND
         ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.TAPER' IN TYPEOF(fcr.
              related_shape_aspect)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.BOSS' IN TYPEOF(fcr.
              relating_shape_aspect))) AND (fcr.related_shape_aspect.
              description IN ['angle taper','directed taper'])) )) = 1)) ))
              <= 1)) )) = 0);
    wr11: (NOT (SELF\characterized_object.description = 'rectangular'))
              OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'rectangular profile occurrence') AND
              (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'profile usage') AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (
              'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE'
              IN TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) ))
              = 0);
    wr12: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
      SELF) | (('DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
              IN TYPEOF(pdr.used_representation)) AND (pdr.
              used_representation.name = 'maximum feature limit')) )) >=
              0;
END_ENTITY; -- boss

ENTITY boss_top
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
              TYPEOF(SELF.of_shape.definition);
    wr2: SELF.description IN ['planar','complex'];
    wr3: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
             USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
    wr4: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
             USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
```

```
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('DIMENSIONAL_INSPECTION_SCHEMA.LOCATION_SHAPE_REPRESENTATION'
              IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
    wr5: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
              'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
               TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
    wr6: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATING_SHAPE_ASPECT') | (((sar.description =
              'boss top usage') AND (sar.name IN ['boss height start',
              'boss height end'])) AND
          ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description
              = 'top condition occurrence') AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.BOSS' IN TYPEOF(fcr.
              related_shape_aspect.of_shape.definition))) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.BOSS_TOP' IN TYPEOF(fcr.
              relating_shape_aspect))) )) >= 1;
    wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) <= 1)) )) = 0;
    wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(impl_rep.
              used_representation.items) = 1)) )) = 0)) )) = 0;
    wr9: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.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 centre_of_symmetry
  SUBTYPE OF (derived_shape_aspect);
  WHERE
```

```
    wr1: SIZEOF(QUERY ( sadr <* deriving_relationships | (NOT (
             'DIMENSIONAL_INSPECTION_SCHEMA.SYMMETRIC_SHAPE_ASPECT' IN
             TYPEOF(sadr.related_shape_aspect))) )) = 0;
END_ENTITY; -- centre_of_symmetry

ENTITY chamfer
  SUBTYPE OF (transition_feature);
  WHERE
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
              TYPEOF(pdr.used_representation)) AND (pdr.
             used_representation.name = 'chamfer face')) )) <= 1)) )) = 0;
    wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
              'RELATING_SHAPE_ASPECT') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
             IN TYPEOF(sar)) ) | (((
             'DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER_OFFSET' IN TYPEOF(fcr
             .related_shape_aspect)) AND (
             'DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER' IN TYPEOF(fcr.
             relating_shape_aspect))) AND (fcr.related_shape_aspect.
             description = 'first offset')) )) = 1;
    wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
              'RELATING_SHAPE_ASPECT') |
         ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
             IN TYPEOF(sar)) ) | (((
             'DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER_OFFSET' IN TYPEOF(fcr
             .related_shape_aspect)) AND (
             'DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER' IN TYPEOF(fcr.
             relating_shape_aspect))) AND (fcr.related_shape_aspect.
             description = 'second offset')) )) = 1;
END_ENTITY; -- chamfer

ENTITY chamfer_offset
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: SELF.description IN ['first offset','second offset'];
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
```



```
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(impl_rep.
          used_representation.items) = 1)) )) = 0)) )) = 0;
wr4: (NOT (SELF.description = 'first offset')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'offset amount')) )) = 1)) ))
          = 0)) )) = 0);
wr5: (NOT (SELF.description = 'first offset')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation)) AND (pdr.
          used_representation.name = 'first face shape')) )) <= 1)) ))
          = 0);
wr6: (NOT (SELF.description = 'second offset')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | (((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'offset amount')) OR ((
          SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
         * TYPEOF(it)) = 2) AND (it.name = 'offset angle'))) )) = 1)) ))
          = 0)) )) = 0);
wr7: (NOT (SELF.description = 'second offset')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation)) AND (pdr.
          used_representation.name = 'second face shape')) )) <= 1)) ))
          = 0);
wr8: SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(SELF,
```

```
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) ) | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER' IN TYPEOF(sdr.
              relating_shape_aspect)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER_OFFSET' IN TYPEOF(sdr
              .related_shape_aspect))) )) = 1;
  END_ENTITY; -- chamfer_offset

  ENTITY characterized_object;
      name        : label;
      description : OPTIONAL text;
  END_ENTITY; -- characterized_object

  ENTITY circle
    SUBTYPE OF (conic);
      radius : positive_length_measure;
  END_ENTITY; -- circle

  ENTITY circular_closed_profile
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
               TYPEOF(SELF.of_shape.definition);
      wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(impl_rep.
              used_representation.items) = 2)) )) = 0)) )) = 0;
      wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
               (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
      wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
```



```
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1)) )) = 0)) ))
              = 0;
END_ENTITY; -- circular_closed_profile

ENTITY circular_pattern
  SUBTYPE OF (replicate_feature);
  WHERE
    wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
              SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATING_SHAPE_ASPECT') |
          ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) ) | (
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
              related_shape_aspect)) )) = 1) )) <= 3)) )) = 0;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT ((SIZEOF(impl_rep.
              used_representation.items) >= 3) AND (SIZEOF(impl_rep.
              used_representation.items) <= 5))) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
```

```
              TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) <= 1)) )) =
              0)) )) = 0;
      wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'base feature rotation')) ))
              <= 1)) )) = 0)) )) = 0;
      wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
              IN TYPEOF(it)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
              measure_with_unit.value_component))) AND (it.name =
              'number of features')) )) = 1)) )) = 0)) )) = 0;
      wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
            * TYPEOF(it)) = 2) AND (it.name = 'angular spacing')) )) = 1)) ))
              = 0)) )) = 0;
      wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
               (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  END_ENTITY; -- circular_pattern

  ENTITY circular_runout_tolerance
```



```
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
      wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
               datum_system) <= 2;
  END_ENTITY; -- circular_runout_tolerance

  ENTITY closed_path_profile
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
                TYPEOF(SELF.of_shape.definition);
      wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
               'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
               'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(impl_rep.
               used_representation.items) = 1)) )) = 0)) )) = 0;
      wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
                (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
      wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
               'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
               'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
               'DIMENSIONAL_INSPECTION_SCHEMA.PATH_SHAPE_REPRESENTATION' IN
                TYPEOF(pdr.used_representation)) )) = 1)) )) = 0;
  END_ENTITY; -- closed_path_profile

  ENTITY closed_shell
    SUBTYPE OF (connected_face_set);
  END_ENTITY; -- closed_shell

  ENTITY coaxiality_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
      wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
```



```
              datum_system) <= 2;
  END_ENTITY; -- coaxiality_tolerance

  ENTITY common_datum
    SUBTYPE OF (composite_shape_aspect, datum);
    WHERE
      wr1: SIZEOF(SELF.component_relationships) = 2;
      wr2: SIZEOF(QUERY ( sar <* SELF.component_relationships | (NOT ((
              'DIMENSIONAL_INSPECTION_SCHEMA.DATUM' IN TYPEOF(sar.
              related_shape_aspect)) AND (NOT (
              'DIMENSIONAL_INSPECTION_SCHEMA.COMMON_DATUM' IN TYPEOF(sar.
              related_shape_aspect))))) )) = 0;
  END_ENTITY; -- common_datum

  ENTITY composite_curve
    SUBTYPE OF (bounded_curve);
      segments       : LIST [1:?] OF composite_curve_segment;
      self_intersect : LOGICAL;
    DERIVE
      n_segments   : INTEGER := SIZEOF(segments);
      closed_curve : LOGICAL := segments[n_segments].transition <>
                       discontinuous;
    WHERE
      wr1: ((NOT closed_curve) AND (SIZEOF(QUERY ( temp <* segments | (
              temp.transition = discontinuous) )) = 1)) OR (closed_curve
              AND (SIZEOF(QUERY ( temp <* segments | (temp.transition =
              discontinuous) )) = 0));
  END_ENTITY; -- composite_curve

  ENTITY composite_curve_on_surface
    SUBTYPE OF (composite_curve);
    DERIVE
      basis_surface : SET [0:2] OF surface := get_basis_surface(SELF);
    WHERE
      wr1: SIZEOF(basis_surface) > 0;
      wr2: constraints_composite_curve_on_surface(SELF);
  END_ENTITY; -- composite_curve_on_surface

  ENTITY composite_curve_segment
    SUBTYPE OF (founded_item);
      transition   : transition_code;
      same_sense   : BOOLEAN;
      parent_curve : curve;
    INVERSE
      using_curves : BAG [1:?] OF composite_curve FOR segments;
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.BOUNDED_CURVE' IN TYPEOF(
              parent_curve);
  END_ENTITY; -- composite_curve_segment

  ENTITY composite_hole
    SUBTYPE OF (compound_feature);
    WHERE
      wr1: SELF\characterized_object.description IN ['counterbore',
              'countersunk'];
      wr2: SIZEOF(QUERY ( pds <* USEDIN(SELF,
```



```
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
              TYPEOF(csa)) AND (SIZEOF(QUERY ( sar <* csa.
              component_relationships |
            (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.ROUND_HOLE' IN TYPEOF(sar.
              related_shape_aspect))) )) = 2)) )) = 1)) )) = 1;
      wr3: (NOT (SELF\characterized_object.description = 'countersunk'))
              OR (SIZEOF(QUERY ( pds <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
              TYPEOF(csa)) AND (SIZEOF(QUERY ( sar <* csa.
              component_relationships | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.ROUND_HOLE' IN TYPEOF(sar.
              related_shape_aspect)) AND (NOT (SIZEOF(QUERY ( pds <*
              QUERY ( pd <* USEDIN(sar.related_shape_aspect,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'change in diameter occurrence') AND (
              SIZEOF(QUERY ( fcr2 <* QUERY ( sar2 <* USEDIN(sa_occ,
'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPEC
T')
               | ((sar2.description = 'taper usage') AND
             ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar2))) ) | ('DIMENSIONAL_INSPECTION_SCHEMA.TAPER'
              IN TYPEOF(fcr2.related_shape_aspect)) )) = 1)) )) = 0)) )) =
               0))) )) = 1)) )) = 1)) )) = 1);
  END_ENTITY; -- composite_hole

  ENTITY composite_shape_aspect
    SUBTYPE OF (shape_aspect);
    INVERSE
      component_relationships : SET [2:?] OF shape_aspect_relationship FOR
                                  relating_shape_aspect;
  END_ENTITY; -- composite_shape_aspect

  ENTITY compound_feature
    SUBTYPE OF (feature_definition);
    WHERE
      wr1: SIZEOF(QUERY ( pds <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
```

```
            csa.name = 'compound feature in solid') AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(csa))) )) = 1)) )) = 1;
    wr2: SIZEOF(QUERY ( pds <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(csa)) )) = 1)) )) = 1;
    wr3: SIZEOF(QUERY ( pds <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(csa)) AND (SIZEOF(QUERY ( fcr <* csa.
            component_relationships | (NOT
        ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(fcr))) )) = 0)) )) = 1)) )) = 1;
    wr4: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(csa)) )) = 1)) )) = 1;
    wr5: SIZEOF(QUERY ( pds <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(csa)) AND (SIZEOF(QUERY ( sar <* csa.
            component_relationships | (
            'DIMENSIONAL_INSPECTION_SCHEMA.THREAD' IN TYPEOF(sar.
            related_shape_aspect)) )) = 0)) )) = 1)) )) = 1;
    wr6: SIZEOF(QUERY ( pds <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(csa)) AND (SIZEOF(QUERY ( sar <* csa.
            component_relationships | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.COMPOUND_FEATURE' IN TYPEOF(
            sar.related_shape_aspect)) AND (sar.related_shape_aspect\
          characterized_object.name <> SELF\characterized_object.name)) ))
            = 0)) )) = 1)) )) = 1;
END_ENTITY; -- compound_feature
```



```
ENTITY compound_representation_item
  SUBTYPE OF (representation_item);
    item_element : compound_item_definition;
END_ENTITY; -- compound_representation_item

ENTITY concentricity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
             datum_system) = 1;
END_ENTITY; -- concentricity_tolerance

ENTITY conic
  SUPERTYPE OF (ONEOF (circle,ellipse,hyperbola,parabola))
  SUBTYPE OF (curve);
    position : axis2_placement;
END_ENTITY; -- conic

ENTITY conical_surface
  SUBTYPE OF (elementary_surface);
    radius     : length_measure;
    semi_angle : plane_angle_measure;
  WHERE
    wr1: radius >= 0;
END_ENTITY; -- conical_surface

ENTITY connected_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 coordinated_universal_time_offset;
    hour_offset   : INTEGER;
    minute_offset : OPTIONAL INTEGER;
    sense         : ahead_or_behind;
  DERIVE
    actual_minute_offset : INTEGER := NVL(minute_offset,0);
  WHERE
    wr1: (0 <= hour_offset) AND (hour_offset < 24);
    wr2: (0 <= actual_minute_offset) AND (actual_minute_offset <= 59);
    wr3: NOT (((hour_offset <> 0) OR (actual_minute_offset <> 0)) AND (
             sense = exact));
END_ENTITY; -- coordinated_universal_time_offset
```

```
ENTITY curve
  SUPERTYPE OF (ONEOF (line,conic,pcurve,surface_curve))
  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 | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
              (it.name = 'orientation')) )) = 1;
    wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
    wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1;
END_ENTITY; -- cylindrical_shape_representation

ENTITY cylindrical_surface
  SUBTYPE OF (elementary_surface);
    radius : positive_length_measure;
END_ENTITY; -- cylindrical_surface

ENTITY cylindricity_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: NOT (('DIMENSIONAL_INSPECTION_SCHEMA.' +
             'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
END_ENTITY; -- cylindricity_tolerance

ENTITY date
  SUPERTYPE OF (calendar_date);
    year_component : year_number;
END_ENTITY; -- date

ENTITY date_and_time;
    date_component : date;
    time_component : local_time;
END_ENTITY; -- date_and_time

ENTITY date_assignment
  ABSTRACT SUPERTYPE;
    assigned_date : date;
    role          : date_role;
END_ENTITY; -- date_assignment

ENTITY date_role;
    name : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
```



```
              'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- date_role

ENTITY dated_effectivity
  SUBTYPE OF (effectivity);
    effectivity_end_date   : OPTIONAL date_time_or_event_occurrence;
    effectivity_start_date : date_time_or_event_occurrence;
END_ENTITY; -- dated_effectivity

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) * [
             'DIMENSIONAL_INSPECTION_SCHEMA.DATUM_FEATURE',
             'DIMENSIONAL_INSPECTION_SCHEMA.DATUM_TARGET']) <> 1) )) = 0;
END_ENTITY; -- datum

ENTITY datum_feature
  SUBTYPE OF (shape_aspect);
  INVERSE
    feature_basis_relationship : shape_aspect_relationship FOR
                                 relating_shape_aspect;
  WHERE
    wr1: SIZEOF(QUERY ( sar <* bag_to_set(USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
              'RELATING_SHAPE_ASPECT')) | (NOT (
             'DIMENSIONAL_INSPECTION_SCHEMA.DATUM' IN TYPEOF(sar.
             related_shape_aspect))) )) = 0;
    wr2: SELF.product_definitional = TRUE;
END_ENTITY; -- datum_feature

ENTITY datum_reference;
    precedence       : INTEGER;
    referenced_datum : datum;
  WHERE
    wr1: precedence > 0;
END_ENTITY; -- datum_reference

ENTITY datum_target
  SUBTYPE OF (shape_aspect);
    target_id : identifier;
  INVERSE
    target_basis_relationship : shape_aspect_relationship FOR
                                relating_shape_aspect;
  WHERE
    wr1: SIZEOF(QUERY ( sar <* bag_to_set(USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
              'RELATING_SHAPE_ASPECT')) | (NOT (
             'DIMENSIONAL_INSPECTION_SCHEMA.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: 'DIMENSIONAL_INSPECTION_SCHEMA.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_shape_aspect
    SUPERTYPE OF (ONEOF (apex,centre_of_symmetry,geometric_alignment,
        geometric_intersection,parallel_offset,perpendicular_to,extension,
        tangent))
    SUBTYPE OF (shape_aspect);
    INVERSE
      deriving_relationships : SET [1:?] OF
                                   shape_aspect_deriving_relationship FOR
                                  relating_shape_aspect;
  END_ENTITY; -- derived_shape_aspect

  ENTITY derived_unit;
      elements : SET [1:?] OF derived_unit_element;
    DERIVE
      name : label := get_name_value(SELF);
    WHERE
      wr1: (SIZEOF(elements) > 1) OR ((SIZEOF(elements) = 1) AND (elements
               [1].exponent <> 1));
      wr2: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  END_ENTITY; -- derived_unit

  ENTITY derived_unit_element;
      unit     : named_unit;
      exponent : REAL;
  END_ENTITY; -- derived_unit_element

  ENTITY description_attribute;
      attribute_value : text;
      described_item  : description_attribute_select;
  END_ENTITY; -- description_attribute

  ENTITY descriptive_representation_item
    SUBTYPE OF (representation_item);
      description : text;
  END_ENTITY; -- descriptive_representation_item

  ENTITY dimension_related_tolerance_zone_element;
      related_dimension : dimensional_location;
```



```
    related_element   : tolerance_zone_definition;
END_ENTITY; -- dimension_related_tolerance_zone_element

ENTITY dimensional_characteristic_representation;
    dimension      : dimensional_characteristic;
    representation : shape_dimension_representation;
END_ENTITY; -- dimensional_characteristic_representation

ENTITY dimensional_exponents;
    length_exponent                    : REAL;
    mass_exponent                      : REAL;
    time_exponent                      : REAL;
    electric_current_exponent          : REAL;
    thermodynamic_temperature_exponent : REAL;
    amount_of_substance_exponent       : REAL;
    luminous_intensity_exponent        : REAL;
END_ENTITY; -- dimensional_exponents

ENTITY dimensional_location
  SUPERTYPE OF (ONEOF (angular_location,dimensional_location_with_path))
  SUBTYPE OF (shape_aspect_relationship);
END_ENTITY; -- dimensional_location

ENTITY dimensional_location_with_path
  SUBTYPE OF (dimensional_location);
    path : shape_aspect;
END_ENTITY; -- dimensional_location_with_path

ENTITY dimensional_measurement_representation
  SUBTYPE OF (representation);
END_ENTITY; -- dimensional_measurement_representation

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_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 (
                'DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION' IN TYPEOF(it))) ))
                = 0;
  END_ENTITY; -- direction_shape_representation

  ENTITY dm_analysis_dofs_dml
    SUBTYPE OF (dimensional_measurement_representation);
 WHERE

  WR1: SIZEOF(QUERY( repi <* SELF.items |((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(repi)) = 2)))) <= 6;

(* transx  *)

 wr2:  SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['transx upper limit', 'transx lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;

(* transy  *)

 wr3: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['transy upper limit', 'transy lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) ))<=1 )) )<= 1;

(* transz  *)

 wr4: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['transz upper limit', 'transz lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
```



```
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;

(* rotx  *)

 wr5: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['rotx upper limit', 'rotx lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;

(* roty  *)

 wr6: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['roty upper limit', 'roty lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;

(* rotz  *)

 wr7: SIZEOF(QUERY ( repi <* SELF.items |
     ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
      (repi.name IN ['rotz upper limit', 'rotz lower limit']) AND
      (SIZEOF(QUERY ( tq <* repi.qualifiers |
      (('DIMENSIONAL_INSPECTION_SCHEMA.TYPE_QUALIFIER' IN TYPEOF(tq)) AND
      (tq.name IN
      ['degree of freedom limited',
      'degree of freedom not allowed',
      'degree of freedom allowed']
      )) )) <=1 )) )<= 1;

  END_ENTITY; -- dm_analysis_dofs_dml

  ENTITY dm_dimension_parameter
    SUBTYPE OF (dm_result_parameter);
    WHERE

(*  dm_dimension_parameter to numeric_parameter (as calculated_value) *)

 wr1: SIZEOF(QUERY ( sri <* SELF.items | ((
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
```

```
      IN TYPEOF(sri)) AND (sri.name = 'calculated value')) )) = 1;


(*dm_dimension_parameter to measurement_uncertainty (as value_uncertainty) *)

wr2: SIZEOF(QUERY ( repi <* SELF.items |
      (((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(repi)) = 2) AND
      (SIZEOF(QUERY ( uq <* repi.qualifiers |
      (SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.STANDARD_UNCERTAINTY']
      * TYPEOF(uq)) = 1) AND
      (uq.measure_name = 'value uncertainty'))) =1 ))))) <=1;

  END_ENTITY; -- dm_dimension_parameter

  ENTITY dm_execution_input
    SUBTYPE OF (executed_action);
WHERE

(*  dm_execution_input to dm_program_run (as program_run) *)

  wr1: ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PROGRAM_RUN'
       IN TYPEOF (SELF.chosen_method));

(*   dm_execution_input to dm_execution_result (as result) *)

  wr2: (SIZEOF(QUERY( ap <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_RELATIONSHIP.RELATING_ACTION')|
       ('DIMENSIONAL_INSPECTION_SCHEMA.DM_EXECUTION_RESULT'
       IN TYPEOF(ap.related_action))))=1);

(*   dm_execution_input to part (as as_designed_part) *)

  wr3: (SIZEOF(QUERY( ap <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_ASSIGNMENT.ASSIGNED_ACTION')|
       (('DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_ACTION_ASSIGNMENT'
       IN TYPEOF(ap)) AND (SIZEOF(QUERY ( it <* ap.items |
       ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION'
        IN TYPEOF(it))))=1))))
       =1);

  END_ENTITY; -- dm_execution_input

  ENTITY dm_execution_result
    SUBTYPE OF (executed_action);
WHERE

(*  dm_execution_result to dm_program_run (as program_run) *)

  wr1: ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PROGRAM_RUN'
        IN TYPEOF (SELF.chosen_method));

(* dm_execution_result to dm_execution_result_measurement(dm_measurements) *)
```



```
  wr2: (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.DM_EXECUTION_RESULT_MEASUREMENT'
       IN TYPEOF (RP))
       ))>=1);

  END_ENTITY; -- dm_execution_result

  ENTITY dm_execution_result_measurement
    SUBTYPE OF (action_property);
    WHERE

(*   dm_execution_result_measurement   to   dm_data_acquisition_software   (as
software) *)

wr1: (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
     (NOT(SIZEOF(QUERY ( it <* apr.representation.items |
     ((it.name = 'dm data acquisition software') AND
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'application name')))
     )) <= 1)) )) = 0);
wr2: (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
     (NOT(SIZEOF(QUERY ( it <* apr.representation.items |
     ((it.name = 'dm data acquisition software') AND
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'application version')))
     )) <= 1)) )) = 0);
wr3: (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
     (NOT(SIZEOF(QUERY ( it <* apr.representation.items |
     ((it.name = 'dm data acquisition software') AND
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'vendor name')))
     )) <= 1)) )) = 0);

(*  dm_execution_result_measurement to dm_point (as measurement_points) *)

wr4: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
    ((apr.representation.name = 'measurement points') AND
    ('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT'
    IN TYPEOF(apr.representation))) )) >= 1;

(*  dm_execution_result_measurement to dm_result_parameter (as parameter) *)

wr5: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
     ('DIMENSIONAL_INSPECTION_SCHEMA.DM_RESULT_PARAMETER'
     IN TYPEOF(apr.representation)) )) = 1;

(*  compensation *)

wr6: SELF.name IN ['TRUE','FALSE'];
```

```
  END_ENTITY; -- dm_execution_result_measurement

  ENTITY dm_feature_analysis_mode_dml
    SUBTYPE OF (dimensional_measurement_representation);
WHERE

(* axis_method  *)

wr1: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode axis') AND
     (repi.description IN ['least square','minimum radius',
     'cross section centers'])
     ))) <= 1;

(*  axis_extrapolate  *)

wr2: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode axis extrapolate') AND
     (repi.description IN ['TRUE','FALSE','unknown'])
     ))) <= 1;

(* aelpr_analysis *)

wr3: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode aelpr') AND
     (repi.description IN ['least square','minimum maximum'])
     ))) <= 1;

(* ccpst_analysis  *)

wr4: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode ccpst') AND
     (repi.description IN ['least square','minimum maximum',
      'minimum size circumscribed','maximum size inscribed'])
     ))) <= 1;

(* curve_analysis  *)

wr5: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode curve') AND
     (repi.description IN ['least square','minimum maximum','bspline'])
     ))) <= 1;

(* limits_of_size  *)
```



```
wr6: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode limits') AND
     (repi.description IN ['functional','mrs average','two point'])
     ))) <= 1;

(* surface_analysis  *)

wr7: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode surface') AND
     (repi.description IN ['least square','minimum maximum',
      'nurbs','bezier'])
     ))) <= 1;

(* axis_method default  *)

wr8: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode axis default') AND
     (repi.description IN ['least square'])
     ))) = 1;

(*  axis_extrapolate default  *)

wr9: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode axis extrapolate default') AND
     (repi.description IN ['TRUE'])
     ))) = 1;

(* aelpr_analysis *)

wr10: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode aelpr default') AND
     (repi.description IN ['least square'])
     ))) = 1;

(* ccpst_analysis default *)

wr11: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode ccpst default') AND
     (repi.description IN ['least square'])
     ))) = 1;

(* curve_analysis default  *)

wr12: SIZEOF(QUERY ( repi <* SELF.items | (
```

```
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode curve default') AND
     (repi.description IN ['least square'])
     ))) = 1;

(* limits_of_size default  *)

wr13: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode limits default') AND
     (repi.description IN ['functional'])
     ))) = 1;

(* surface_analysis default  *)

wr14: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature analysis mode surface default') AND
     (repi.description IN ['least square'])
     ))) = 1;

  END_ENTITY; -- dm_feature_analysis_mode_dml

  ENTITY dm_feature_definition
    SUBTYPE OF (characterized_object);
    WHERE
 wr1: SIZEOF(QUERY ( pdr <*
      get_property_definition_representations(SELF) |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) )) >= 1;

 wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      (('DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
      (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;

 wr3: SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.DMF_ARC',
             'DIMENSIONAL_INSPECTION_SCHEMA.DMF_LINE_UNBOUNDED',
             'DIMENSIONAL_INSPECTION_SCHEMA.DMF_CIRCLE',
             'DIMENSIONAL_INSPECTION_SCHEMA.DMF_PATTERN',
             'DIMENSIONAL_INSPECTION_SCHEMA.DMF_CONE',
             'DIMENSIONAL_INSPECTION_SCHEMA.DMF_PLANE',
             'DIMENSIONAL_INSPECTION_SCHEMA.DMF_PLANE_CLOSED_PARALLEL',
             'DIMENSIONAL_INSPECTION_SCHEMA.DMF_CYLINDER',
             'DIMENSIONAL_INSPECTION_SCHEMA.DMF_ELLIPSE',
```



```
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_PLANE_SYMMETRIC',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_GEOMETRIC_CURVE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_POINT',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_GEOMETRIC_SURFACE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_EDGE_POINT',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_SPHERE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_GENERIC_FEATURE',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_LINE_BOUNDED',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_TORUS',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_LINE_CLOSED_PARALLEL',
              'DIMENSIONAL_INSPECTION_SCHEMA.DMF_SURFACE_OF_REVOLUTION_DML']
              * TYPEOF(SELF)) = 1;
  END_ENTITY; -- dm_feature_definition

  ENTITY dm_feature_relationship
    SUBTYPE OF (shape_aspect_relationship);
    WHERE
 wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_DEFINITION' IN
              TYPEOF(SELF.relating_shape_aspect);
 wr2: (SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_DEFINITION',
      'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_DEFINITION']
      * TYPEOF(SELF.related_shape_aspect)) = 1);

  END_ENTITY; -- dm_feature_relationship

  ENTITY dm_instanced_feature
    SUBTYPE OF (dm_feature_definition, shape_aspect);
    WHERE

 wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(
      SELF.of_shape.definition);

 wr2: SELF.product_definitional;

(*  dm_feature to cartesian_point (as boundary) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINITION_REPRESENTATION'
      IN TYPEOF(pdr)) AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND
      (pdr.used_representation.name='boundary') AND
      (SIZEOF(QUERY (sr <* pdr.used_representation.items |
      ('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT'
      IN TYPEOF(sr)))) >=0)
      ))) <=1 ))) =0));


  END_ENTITY; -- dm_instanced_feature

  ENTITY dm_parameter_analysis_dml
```

```
    SUBTYPE OF (action_property);
    WHERE

(*       dm_parameter_analysis_dml       to       dm_feature_analysis_mode_dml
(feature_analysis)  *)

 wr1: SIZEOF(QUERY ( apr <* USEDIN(SELF,

'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
      ('DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_ANALYSIS_MODE_DML'
      IN TYPEOF(apr.representation)) )) <= 1;

(*      dm_parameter_analysis_dml      to      dm_tolerance_analysis_mode_dml
(tolerance_analysis) *)

 wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
|
     ('DIMENSIONAL_INSPECTION_SCHEMA.DM_TOLERANCE_ANALYSIS_MODE_DML'
     IN TYPEOF(apr.representation)) )) <= 1;

(* dm_parameter_analysis_dml to dm_analysis_dofs_dml (as dofs) *)

 wr3: SIZEOF(QUERY ( apr <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
|
     ('DIMENSIONAL_INSPECTION_SCHEMA.DM_ANALYSIS_DOFS_DML' IN
     TYPEOF(apr.representation)) )) <= 1;

  END_ENTITY; -- dm_parameter_analysis_dml

  ENTITY dm_parameter_value_limits
    SUBTYPE OF (representation);

 WHERE

 (* limits_method *)

 wr1 : SIZEOF(QUERY ( repi <* SELF.items |
       (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(repi)) AND
       (repi.name = 'limits method')
       )))=1;

 (* dm_parameter_value_limits to numeric_parameter (as calculated_limits) *)

 WR2: (SIZEOF(QUERY ( it <* SELF.items |
      ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'] *
      TYPEOF(it)) = 1) AND
      (it.name = 'calculated limits')
      )))>=2);
  END_ENTITY; -- dm_parameter_value_limits

  ENTITY dm_point
    SUBTYPE OF (representation);
```



```
WHERE

(*   dm_point to cartesian_point (as measured_point) *)

 WR1: (SIZEOF( QUERY( repi <* SELF.items |
      NOT (('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT'
      IN TYPEOF(repi)) AND (repi.name='measured point'))
      ))=1);

(*   dm_point to cartesian_point (as expected_point) *)

 WR2: (SIZEOF( QUERY( repi <* SELF.items |
      NOT (('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT'
      IN TYPEOF(repi)) AND (repi.name='expected point'))
      ))<=1);

  END_ENTITY; -- dm_point

  ENTITY dm_point_parameter
    SUBTYPE OF (dm_result_parameter);
WHERE

(* dm_point_parameter to cartesian_point (as calculated_point) *)

  wr1:  SIZEOF(QUERY( repi <* SELF.items |
        ('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT'
        IN TYPEOF (repi))))=1;

(*  dm_point_parameter to measurement_uncertainty (as value_uncertainty) *)

 wr2: SIZEOF(QUERY ( repi <* SELF.items | (
     (SIZEOF([
     'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
     * TYPEOF(repi)) = 2) AND
     (SIZEOF(QUERY ( uq <* repi.qualifiers | (
     (SIZEOF([
   'DIMENSIONAL_INSPECTION_SCHEMA.STANDARD_UNCERTAINTY'] * TYPEOF(uq))=1) AND
     (uq.measure_name = 'value uncertainty') AND
     (uq.description IN ['x uncertainty','y uncertainty', 'z uncertainty']) )
     )) <=3 )
     ))) <=1;
  END_ENTITY; -- dm_point_parameter

  ENTITY dm_program_identification
    SUBTYPE OF (action_resource_requirement);
WHERE

(*  angular_units_dml *)

 wr1: (SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      (rp.name='angular units dml')))<=1);

(*  linear_units_dml *)
```

```
 wr2: (SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      (rp.name='linear units dml')))<=1);

(*  tolerance_standard_dml *)

 wr3: (SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      (rp.name='tolerance standard dml')))<=1);

(* dm_program_identification to person_in_organization (program_custodian) *)
(* dm_program_identification to organization (program_custodian) *)

 wr4: ((SIZEOF(QUERY (adr <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
      ((adr.role.name='program custodian') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.ORGANIZATION' IN TYPEOF
      (adr.assigned_organization)))))<=1) OR
      (SIZEOF(QUERY (adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS')|
      ((adr.role.name='program custodian') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.PERSON_AND_ORGANIZATION' IN TYPEOF
      (adr.assigned_person_and_organization)))))<=1));

  END_ENTITY; -- dm_program_identification

  ENTITY dm_program_run
    SUBTYPE OF (action_method);
WHERE

(*  dm_program_run to numeric_parameter (as measurement_temperature) *)

wr1 : (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION')|
      (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
      (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(it)) = 2) AND (it.name = 'measurement temperature')) ))
      = 1)) ) = 0)) )) = 0));

(*  dm_program_run to numeric_parameter (as workpiece_temperature) *)

wr2 : (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION')|
      (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
      (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(it)) = 2) AND (it.name = 'workpiece temperature')) ))
      <= 1)) ) = 0)) )) = 0));

(*  dm_program_run to numeric_parameter (as measurement_humidity) *)
```



```
wr3 : (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION')|
      (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
      (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
      * TYPEOF(it)) = 2) AND (it.name = 'measurement humidity')) ))
      = 1)) ) = 0)) )) = 0));

(*   part_inspection_status *)

wr4 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_RESOURCE.USAGE') |
      (ap.name = 'part inspection status') )) <= 1;

(*   url*)

wr5 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_RESOURCE.USAGE') |
      (ap.name = 'url') )) <= 1;

(*   dm_program_run to dm_program_identification (as program_id) *)

wr6 : SIZEOF(QUERY ( arr <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PROGRAM_IDENTIFICATION'
      IN TYPEOF(arr)) )) = 1;

(*   dm_program_run to run_administrator (as run_administrator) *)

wr7 : (SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
      ((adr.role.name = 'run administrator') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.ORGANIZATION'
      IN TYPEOF(adr.assigned_organization))) )) = 1) OR
      (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
      ((adr.role.name = 'run administrator') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.PERSON_AND_ORGANIZATION'
      IN TYPEOF(adr.assigned_person_and_organization))) )) = 1);

(*    dm_program_run to measurement_location (as run_location) *)

wr8 : SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.ACTION_PROPERTY.DEFINITION')
      | ('DIMENSIONAL_INSPECTION_SCHEMA.MEASUREMENT_LOCATION' IN
      TYPEOF(rp)) )) <= 1;

(*    dm_program_run to date_time (as run_end) *)

wr9 : (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS')|
      ((adr.role.name = 'run end') AND
```

```
      ('DIMENSIONAL_INSPECTION_SCHEMA.DATE_AND_TIME'
      IN TYPEOF(adr.assigned_date))) )) <= 1) OR
      (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
      ((adr.role.name = 'run end') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.CALENDAR_DATE'
      IN TYPEOF(adr.assigned_date))) )) <= 1);

(*    dm_program_run to date_time (as run_start) *)

wr10: (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS')|
      ((adr.role.name = 'run end') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.DATE_AND_TIME'
      IN TYPEOF(adr.assigned_date))) )) <= 1) OR
      (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
      ((adr.role.name = 'run start') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.CALENDAR_DATE'
      IN TYPEOF(adr.assigned_date))) )) <= 1);

(*    dm_program_run to date_time (as run_date) *)

wr11: (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS')|
      ((adr.role.name = 'run date') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.DATE_AND_TIME'
      IN TYPEOF(adr.assigned_date))) )) = 1) OR
      (SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS')|
      ((adr.role.name = 'run date') AND (
      'DIMENSIONAL_INSPECTION_SCHEMA.CALENDAR_DATE'
      IN TYPEOF(adr.assigned_date))) )) = 1);

  END_ENTITY; -- dm_program_run

  ENTITY dm_result_parameter
    SUBTYPE OF (representation);
 WHERE

(* dm_result_parameter to dm_parameter_value_limits (as limits) *)

 wr1:SIZEOF(QUERY( sri <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1' ) |
     ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PARAMETER_VALUE_LIMITS'             IN
TYPEOF(sri.rep_2)
     ))) <= 1;

(*   calculation_method *)

 wr2:SIZEOF(QUERY( sri <* SELF.items |
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(sri)) AND (sri.name='calculation method'))  )) <=1;

(* dm_result_parameter to dm_parameter_analysis_dml (as analysis_conditions)
*)
```



```
 wr3: SIZEOF(QUERY ( apr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'ACTION_PROPERTY_REPRESENTATION.REPRESENTATION')|
      ('DIMENSIONAL_INSPECTION_SCHEMA.DM_PARAMETER_ANALYSIS_DML'
      IN TYPEOF(apr.property ))))<= 1;

(* dm_result_parameter to ??? (as specification) *)

 wr4: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.'+
      'USED_REPRESENTATION') |
     ((pdr.name='specification') AND
     ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
      IN TYPEOF(pdr.definition)) AND
     (SIZEOF(QUERY( sa <* USEDIN(pdr.definition,
     'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
     (SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.TRANSITION_FEATURE',
              'DIMENSIONAL_INSPECTION_SCHEMA.INSTANCED_FEATURE']
      * TYPEOF(sa)) = 1)
      )) =1 )))) <=1;

wr5:   (SIZEOF(QUERY( pdr <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
         ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DIMENSION_REPRESENTATION'
         IN TYPEOF (pdr.rep_2)))) <=1 );

 WR6: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.'+
       'USED_REPRESENTATION') |
     ((pdr.name='specification') AND
     ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
      IN TYPEOF(pdr.definition)) AND
     (SIZEOF(QUERY( sa <* USEDIN(pdr.definition,
     'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |

     (SIZEOF(QUERY(gtol <* USEDIN(sa,
     'DIMENSIONAL_INSPECTION_SCHEMA.GEOMETRIC_TOLERANCE.'+
     'TOLERANCED_SHAPE_ASPECT') |
     ('DIMENSIONAL_INSPECTION_SCHEMA.GEOMETRIC_TOLERANCE'
      IN TYPEOF(gtol)))) =1 )
      )) =1 )) )) <=1;
  END_ENTITY; -- dm_result_parameter

  ENTITY dm_tolerance_analysis_mode_dml
    SUBTYPE OF (dimensional_measurement_representation);
WHERE

(* method  *)

wr1: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode method') AND
     (repi.description IN ['least square 3d','least square normal',
     'minimum deviation 3d','standard'])
```

```
     ))) <= 1;

(* option *)

wr2: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode option') AND
     (repi.description IN ['high point','best fit'])
     ))) <= 1;

(* setting  *)

wr3: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode setting') AND
     (repi.description IN ['inner','outer','standard'])
     ))) <= 1;

(* method default *)

wr4: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode method default') AND
     (repi.description IN ['standard'])
     ))) = 1;

(* option default *)

wr5: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode option default') AND
     (repi.description IN ['best fit'])
     ))) = 1;

(* setting default  *)

wr6: SIZEOF(QUERY ( repi <* SELF.items | (
     ('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(repi)) AND
     (repi.name = 'feature tolerance mode setting default') AND
     (repi.description IN ['standard'])
     ))) = 1;

  END_ENTITY; -- dm_tolerance_analysis_mode_dml

  ENTITY dm_vector_parameter
    SUBTYPE OF (dm_result_parameter);
  WHERE

(*  dm_vector_parameter to cartesian_vector (as calculated_vector)  *)

  wr1:  SIZEOF(QUERY( repi <* SELF.items |
```



```
        ('DIMENSIONAL_INSPECTION_SCHEMA.VECTOR'
        IN TYPEOF (repi))))=1;

(*  dm_point_parameter to measurement_uncertainty (as value_uncertainty) *)

 wr2: SIZEOF(QUERY ( repi <* SELF.items | (
     (SIZEOF([
     'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'DIMENSIONAL_INSPECTION_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
     * TYPEOF(repi)) = 2) AND
     (SIZEOF(QUERY ( uq <* repi.qualifiers | (
     (SIZEOF([
   'DIMENSIONAL_INSPECTION_SCHEMA.STANDARD_UNCERTAINTY'] * TYPEOF(uq))=1) AND
     (uq.measure_name = 'value uncertainty') AND
     (uq.description IN ['i uncertainty','j uncertainty', 'k uncertainty']) )
     )) <=3 )
     ))) <=1;
  END_ENTITY; -- dm_vector_parameter

  ENTITY dmf_arc
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(* dmf_arc to dm_point_parameter (as center_point)*)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_arc to dm_vector_parameter (as start_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start vector')
      )) =1 )) )) =1 )) ) =0));

(* dmf_arc to dm_vector_parameter (as end_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='end vector')
      )) =1 )) )) =1 )) ) =0));

(* dmf_arc to dm_vector_parameter (as axis_direction_vector) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

(* dmf_arc to dm_dimension_parameter (as radius) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='radius')
      )) =1 )) )) <=1 )) ) =0));

(* inner_outer *)

 WR7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```



```
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;
(* inner_outer *)

 WR7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_arc

  ENTITY dmf_circle
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*  dmf_circle to dm_point_parameter (as center_point)  *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
```

```
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

 (*  dmf_circle to dm_vector_parameter (as axis_direction_vector)  *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

 (*  dmf_circle to dm_dimension_parameter (as diameter)  *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='diameter')
      )) =1 )) )) <=1 )) ) =0));

(* inner_outer *)

 WR5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_circle

  ENTITY dmf_cone
    SUBTYPE OF (dm_feature_definition);
    WHERE
```



```
(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*  dmf_cone to dm_point_parameter (as start_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start point')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_point_parameter (as end_point) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='end point')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_point_parameter (as apex_point) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='apex point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cone to dm_vector_parameter (as start_vector) *)
```

```
 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start vector')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_vector_parameter (as end_vector) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='end vector')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_vector_parameter (as axis_direction_vector) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cone to dm_dimension_parameter (as diameter) *)

 wr8: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='diameter')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cone to dm_dimension_parameter (as start_length_dml) *)
```



```
 wr9: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start length dml')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_dimension_parameter (as end_length_dml) *)

 wr10: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
       IN TYPEOF(pdr.used_representation)) AND
       (SIZEOF(QUERY( na <* USEDIN(pdr,
       'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
       (na.attribute_value='end length dml')
       )) =1 )) )) <=1 )) ) =0));

(*  dmf_cone to dm_dimension_parameter (as included_angle) *)

 wr11: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
       IN TYPEOF(pdr.used_representation)) AND
       (SIZEOF(QUERY( na <* USEDIN(pdr,
       'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
       (na.attribute_value='included angle')
       )) =1 )) )) =1 )) ) =0));

(* inner_outer *)


 WR12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.'+
       'SHAPE_REPRESENTATION_WITH_PARAMETERS')
       IN TYPEOF(pdr.used_representation)) ) |
       (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND
       (it.name = 'inner or outer')) AND
       ((it.description = 'inner') OR
```

```
       (it.description = 'outer')))
       )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_cone

  ENTITY dmf_cylinder
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*  dmf_cylinder to dm_point_parameter (as center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='start point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cylinder to dm_point_parameter (as start_point) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
```

```
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='end point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cylinder to dm_point_parameter (as end_point) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cylinder to dm_dimension_parameter (as diameter) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='diameter')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_cylinder to dm_dimension_parameter (as length_dml) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='length dml')
      )) =1 )) )) <=1 )) ) =0));

(* inner_outer *)

 WR8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
```

```
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_cylinder

  ENTITY dmf_edge_point
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

 (*  dmf_edge_point to dm_point_parameter (location_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='location point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_edge_point to dm_vector_parameter (edge_normal_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='edge normal vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_edge_point to dm_vector_parameter (surface_normal_vector) *)
```

```
 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='surface normal vector')
      )) =1 )) )) =1 )) ) =0));

(*  point_type_dml*)

 WR5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'point type dml')) AND
      ((it.description = 'h edge') OR
      (it.description = 't edge')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_edge_point

  ENTITY dmf_ellipse
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*   dmf_ellipse to dm_point_parameter (as center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_ellipse to dm_point_parameter (as focus_point_one_dml) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='focus point one dml')
      )) =1 )) )) <=1 )) ) =0));

(*   dmf_ellipse to dm_point_parameter (as focus_point_two_dml) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='focus point two dml')
      )) =1 )) )) <=1 )) ) =0));

(*   dmf_ellipse to dm_vector_parameter (as major_axis_vector) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='major axis vector')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_ellipse to dm_vector_parameter (as normal_direction) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```



```
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='normal direction')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_ellipse to dm_dimension_parameter (as minor_diameter) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='minor diameter')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_ellipse to dm_dimension_parameter (as major_diameter) *)

 wr8: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='major diameter')
      )) =1 )) )) =1 )) ) =0));

(* inner_outer *)

 WR9: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;
  END_ENTITY; -- dmf_ellipse

  ENTITY dmf_generic_feature
    SUBTYPE OF (dm_feature_definition);
```

```
   WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*  dmf_generic_feature to dm_point_parameter (as center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(*            dmf_generic_feature       to      dm_vector_parameter       (as
vector_from_center_of_object) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='vector from center of object')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_generic_feature to dm_vector_parameter (as secondary_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='secondary vector')
      )) =1 )) )) <=1 )) ) =0));
```

```
(*  dmf_generic_feature to dm_dimension_parameter (parameters) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='parameter')
      )) =1 )) )) <=1 )) ) =0));

(* description *)

 WR6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'description')))
      )) <= 1)) )) = 0)) )) = 0;
         END_ENTITY; -- dmf_generic_feature

  ENTITY dmf_geometric_curve
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(* dmf_geometric_curve to dm_point_parameter (as point_on_plane_of_curve) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
```

```
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on plane of curve')
      )) =1 )) )) =1 )) ) =0));

(* dmf_geometric_curve to dm_vector_parameter (as curve_plane_direction) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='curve plane direction')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_geometric_curve to cartesian_point (as data_points)  *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINITION_REPRESENTATION'
      IN TYPEOF(pdr)) AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='data point'))) =1 ) AND
      (SIZEOF(QUERY (sr <* pdr.used_representation.items |
      ('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT' IN TYPEOF(sr)))) >=0)
      ))) >=0 ))) =0));

  END_ENTITY; -- dmf_geometric_curve

  ENTITY dmf_geometric_surface
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;
```



```
(* dmf_geometric_surface to dm_point_parameter (point_on_surface) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on surface')
      )) =1 )) )) =1 )) ) =0));

(* dmf_geometric_surface to dm_vector_parameter (as local_surface_normal) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='local surface normal')
      )) =1 )) )) =1 )) ) =0));

(* dmf_geometric_surface to cartesian_point (as data_points) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINITION_REPRESENTATION'
      IN TYPEOF(pdr)) AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='data point') )) =1 ) AND
      (SIZEOF(QUERY (sr <* pdr.used_representation.items |
      ('DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT' IN TYPEOF(sr)))) >=0)
      ))) >=0 ))) =0));

  END_ENTITY; -- dmf_geometric_surface

  ENTITY dmf_line_bounded
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
```

```
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(* dmf_line_bounded to dm_point_parameter (as first_end_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='first end point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_line_bounded to dm_point_parameter (as second_end_point) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='second end point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_line_bounded to dm_vector_parameter (as surface_approach_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='surface approach vector')
      )) =1 )) )) =1 )) ) =0));

(* dmf_line_bounded to dm_vector_parameter (as vector_dml) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
```

```
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='vector dml')
      )) =1 )) )) <=1 )) ) =0));

(* dmf_line_bounded to dm_dimension_parameter (length_dml) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='length dml')
      )) =1 )) )) <=1 )) ) =0));
  END_ENTITY; -- dmf_line_bounded

  ENTITY dmf_line_closed_parallel
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   3 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =3 ))) =1 ))=1;

(*  dmf_line_closed_parallel to dm_point_parameter (as center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_line_closed_parallel to dm_vector_parameter (axis_direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
```

```
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_line_closed_parallel to dm_vector_parameter (as longitude_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='longitude vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_line_closed_parallel to dm_dimension_parameter (feature_length) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='feature length')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_line_closed_parallel to dm_dimension_parameter (width)*)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='width')
      )) =1 )) )) =1 )) ) =0));

(*  end_kind *)

 WR7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
```



```
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'end kind')) AND
      ((it.description = 'round') OR
      (it.description = 'square')))
      )) = 1)) )) = 0)) )) = 0;

(* inner_outer *)

 WR8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) = 1)) )) = 0)) )) = 0;
  END_ENTITY; -- dmf_line_closed_parallel

  ENTITY dmf_line_unbounded
    SUBTYPE OF (dm_feature_definition);
  WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(*  dmf_line_unbounded to dm_point_parameter (point on line) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
```

```
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on line')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_line_unbounded to dm_vector_parameter (direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_line_unbounded to dm_vector_parameter (surface_approach_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='surface approach vector')
      )) =1 )) )) =1 )) ) =0));

  END_ENTITY; -- dmf_line_unbounded

  ENTITY dmf_pattern
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(*   dmf_pattern to dm_point_parameter (point) *)
```

```
 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_pattern todm_vector_parameter (direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_pattern to dm_feature (as features) *)

 wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
      'DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_RELATIONSHIP'
      IN TYPEOF(pd)) )) >= 2;
  END_ENTITY; -- dmf_pattern

  ENTITY dmf_plane
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(* dmf_plane to dm_point_parameter (point_on_plane) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
```

```
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on plane')
      )) =1 )) )) =1 )) ) =0));

(* dmf_plane to dm_vector_parameter (direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector')
      )) =1 )) )) =1 )) ) =0));
  END_ENTITY; -- dmf_plane

  ENTITY dmf_plane_closed_parallel
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   3 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =3 ))) =1 ))=1;

(*  dmf_plane_closed_parallel to dm_point_parameter (center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(* dmf_plane_closed_parallel to dm_vector_parameter(axis_direction_vector) *)
```



```
 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis direction vector')
      )) =1 )) )) =1 )) ) =0));

 (*  dmf_plane_closed_parallel to dm_vector_parameter (longitude_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='longitude vector')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_plane_closed_parallel to dm_vector_parameter (normal_dml) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='normal dml')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_plane_closed_parallel to dm_dimension_parameter (feature_length) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='feature length')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_plane_closed_parallel to dm_dimension_parameter (height) *)
```

```
 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='height')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_plane_closed_parallel to dm_dimension_parameter (width) *)

 wr8: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='width')
      )) =1 )) )) =1 )) ) =0));

 (* end_kind *)

  WR9: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.'+
       'SHAPE_REPRESENTATION_WITH_PARAMETERS')
       IN TYPEOF(pdr.used_representation)) ) |
       (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND
       (it.name = 'end kind')) AND
       ((it.description = 'round') OR
       (it.description = 'square')))
       )) = 1)) )) = 0)) )) = 0;

(*  inner_outer *)

  WR10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.'+
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.'+
       'SHAPE_REPRESENTATION_WITH_PARAMETERS')
       IN TYPEOF(pdr.used_representation)) ) |
       (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
```



```
       IN TYPEOF(it)) AND
       (it.name = 'inner or outer')) AND
       ((it.description = 'inner') OR
       (it.description = 'outer')))
       )) = 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_plane_closed_parallel

  ENTITY dmf_plane_symmetric
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;


(*    dmf_plane_symmetric to dm_point_parameter (point_on_one_side) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on one side')
      )) =1 )) )) =1 )) ) =0));

(*    dmf_plane_symmetric to dm_point_parameter (point_on_side_two) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on side two')
      )) =1 )) )) =1 )) ) =0));

(*    dmf_plane_symmetric to dm_point_parameter (point_on_mid_point) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
```

```
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point on mid plane')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_plane_symmetric to dm_vector_parameter (direction_vector_side_one) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector side one')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_plane_symmetric to dm_vector_parameter (direction_vector_side_two) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector side two')
      )) =1 )) )) =1 )) ) =0));

(*    dmf_plane_symmetric to dm_dimension_parameter (width_at_mid_point) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='width at mid point')
      )) =1 )) )) =1 )) ) =0));

(*  inner_outer *)

 WR8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
```



```
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;

  END_ENTITY; -- dmf_plane_symmetric

  ENTITY dmf_point
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;


(*  dmf_point to dm_point_parameter (point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='point')
      )) =1 )) )) =1 )) ) =0));


(*  dmf_point to dm_vector_parameter (direction_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='direction vector')
      )) =1 )) )) =1 )) ) =0));

  END_ENTITY; -- dmf_point

  ENTITY dmf_sphere
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*  dmf_sphere to dm_point_parameter (center_point) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(*  dmf_sphere to dm_vector_parameter (north_pole_vector) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='north pole vector')
       )) =1 )) )) =1 )) ) =0));

(*  dmf_sphere to dm_vector_parameter (prime_meridian_vector) *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
```



```
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='prime meridian vector')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (longitude_start_angle) *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='longitude start angle')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (longitude_stop angle) *)

 wr6: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='longitude stop angle')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (latitude_start_angle) *)

 wr7: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='latitude start angle')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (latitude_stop_angle) *)

 wr8: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
```

```
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='latitude stop angle')
      )) =1 )) )) <=1 )) ) =0));

(*  dmf_sphere to dm_dimension_parameter (diameter) *)

 wr9: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='diameter')
      )) =1 )) )) =1 )) ) =0));

(*  inner_outer *)

 WR10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND
      (it.name = 'inner or outer')) AND
      ((it.description = 'inner') OR
      (it.description = 'outer')))
      )) <= 1)) )) = 0)) )) = 0;
  END_ENTITY; -- dmf_sphere

  ENTITY dmf_surface_of_revolution_dml
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   1 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
```



```
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =1 ))) =1 ))=1;

(*  dmf_sphere to dm_point_parameter (profile_curve) *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='profile curve')
      )) =1 )) )) >=2 )) ) =0));

(*  dmf_sphere to dm_vector_parameter (axis_of_revolution) *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='axis of revolution')
      )) =1 )) )) =1 )) ) =0));

  END_ENTITY; -- dmf_surface_of_revolution_dml

  ENTITY dmf_torus
    SUBTYPE OF (dm_feature_definition);
    WHERE

(*   ------------   2 attributes  --------------------- *)

 wr1: SIZEOF( QUERY( pd <* USEDIN( SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      SIZEOF( QUERY( pdr <* USEDIN( pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.'+
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(pdr.used_representation.items) =2 ))) =1 ))=1;

(*   dmf_torus to dm_point_parameter (center_point)  *)

 wr2: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
```

```
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_POINT_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='center point')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_torus to dm_vector_parameter (vector_or_plane)  *)

 wr3: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_VECTOR_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='vector of plane')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_torus to dm_dimension_parameter (major_diameter)  *)

 wr4: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='major diameter')
      )) =1 )) )) =1 )) ) =0));

(*   dmf_torus to dm_dimension_parameter (minor_diameter)  *)

 wr5: (NOT(SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DM_DIMENSION_PARAMETER'
      IN TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY( na <* USEDIN(pdr,
      'DIMENSIONAL_INSPECTION_SCHEMA.NAME_ATTRIBUTE.NAMED_ITEM') |
      (na.attribute_value='minor diameter')
      )) =1 )) )) =1 )) ) =0));

(*  inner_outer  *)

 WR6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
```



```
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     (('DIMENSIONAL_INSPECTION_SCHEMA.'+
     'SHAPE_REPRESENTATION_WITH_PARAMETERS')
     IN TYPEOF(pdr.used_representation)) ) |
     (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
     ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND
     (it.name = 'inner or outer')) AND
     ((it.description = 'inner') OR
     (it.description = 'outer')))
     )) <= 1)) )) = 0)) )) = 0;

 END_ENTITY; -- dmf_torus

 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,
     '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

 ENTITY document_reference
   ABSTRACT SUPERTYPE;
```

```
    assigned_document : document;
    source            : label;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- document_reference

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,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
    wr2: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
             SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | ((NOT (SIZEOF(
             impl_rep.used_representation.items) >= 1)) AND (SIZEOF(
             impl_rep.used_representation.items) <= 3)) )) = 0)) )) = 0);
    wr3: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
             QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
             ['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
             = 0);
    wr4: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
             QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
             ['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'first offset')) )) <= 1)) ))
             = 0)) )) = 0);
    wr5: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
             QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

```
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
               IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
               QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
               ['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) <= 1)) ))
               = 0)) )) = 0);
      wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
                TYPEOF(pdr.used_representation)) AND (pdr.
               used_representation.name = 'edge round face')) )) <= 1)) ))
               = 0;
      wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
                TYPEOF(pdr.used_representation)) AND (pdr.
               used_representation.name = 'first face shape')) )) <= 1)) ))
               = 0;
      wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
                TYPEOF(pdr.used_representation)) AND (pdr.
               used_representation.name = 'second face shape')) )) <= 1)) ))
               = 0;
  END_ENTITY; -- edge_round

  ENTITY effectivity
    SUPERTYPE OF (ONEOF (serial_numbered_effectivity,dated_effectivity,
        lot_effectivity));
      id : identifier;
    DERIVE
      name        : label := get_name_value(SELF);
      description : text := get_description_value(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
      wr2: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  END_ENTITY; -- effectivity

  ENTITY effectivity_assignment
    ABSTRACT SUPERTYPE;
      assigned_effectivity : effectivity;
```



```
    DERIVE
      role : object_role := get_role(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
  END_ENTITY; -- effectivity_assignment

  ENTITY elementary_surface
    SUPERTYPE OF (ONEOF (plane,cylindrical_surface,conical_surface,
        spherical_surface,toroidal_surface))
    SUBTYPE OF (surface);
      position : axis2_placement_3d;
  END_ENTITY; -- elementary_surface

  ENTITY ellipse
    SUBTYPE OF (conic);
      semi_axis_1 : positive_length_measure;
      semi_axis_2 : positive_length_measure;
  END_ENTITY; -- ellipse

  ENTITY executed_action
    SUBTYPE OF (action);
  END_ENTITY; -- executed_action

  ENTITY extension
    SUBTYPE OF (derived_shape_aspect);
    WHERE
      wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
  END_ENTITY; -- extension

  ENTITY expanded_uncertainty
    SUBTYPE OF (standard_uncertainty);
      coverage_factor : REAL;
  END_ENTITY; -- expanded_uncertainty

  ENTITY external_source;
      source_id : source_item;
    DERIVE
      description : text := get_description_value(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  END_ENTITY; -- external_source

  ENTITY externally_defined_dimension_definition
    SUBTYPE OF (externally_defined_item, dimensional_size);
 WHERE

 wr1: SELF.source.description = 'externally dimension specification';

  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
      (adr.assigned_document.description =
      'externally 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,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) AND ((5 <= SIZEOF(pdr.
              used_representation.items)) AND (SIZEOF(pdr.
              used_representation.items) <= 10))) )) = 1) )) = 1);
    wr3 : (NOT (SELF\characterized_object.description = 'marking')) OR (
              SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) AND (SIZEOF(pdr.used_representation.
              items) = 2)) )) = 1) )) = 1);
    wr4 : (NOT (SELF\characterized_object.description = 'knurl')) OR (
              SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) AND (SIZEOF(pdr.used_representation.
              items) = 1)) )) = 1) )) = 1);
    wr5 : (NOT (SELF\characterized_object.description IN ['knurl',
              'thread'])) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(
              SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```



```
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'partial area occurrence') AND (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'applied area usage') AND
            ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_AREA' IN TYPEOF(sdr.
            relating_shape_aspect)) )) = 1)) )) <= 1)) )) = 0);
  wr6 : (NOT (SELF\characterized_object.description = 'marking')) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'marking text')) )) = 1)) ))
            = 0)) )) = 0);
  wr7 : (NOT (SELF\characterized_object.description = 'thread')) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
     ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'removal direction')) AND ((
            it.description = 'internal') OR (it.description =
            'external'))) )) = 1)) )) = 0)) )) = 0);
  wr8 : (NOT (SELF\characterized_object.description = 'thread')) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
    (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'qualifier')) )) <= 1)) )) =
            0)) )) = 0);
  wr9 : (NOT (SELF\characterized_object.description = 'thread')) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

```
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items |
    (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
           IN TYPEOF(it)) AND (it.name = 'hand')) )) = 1)) )) = 0)) ))
           = 0);
  wr10: (NOT (SELF\characterized_object.description = 'thread')) OR (
           SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items |
    (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'fit class')) )) = 1)) )) = 0)) ))
           = 0);
  wr11: (NOT (SELF\characterized_object.description = 'thread')) OR (
           SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items |
     (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
           IN TYPEOF(it)) AND (it.name = 'form')) )) = 1)) )) = 0)) ))
           = 0);
  wr12: (NOT (SELF\characterized_object.description = 'thread')) OR (
           SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items |
  ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
           'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
         TYPEOF(it)) = 2) AND (it.name = 'major diameter')) )) <= 1)) ))
           = 0)) )) = 0);
  wr13: (NOT (SELF\characterized_object.description = 'thread')) OR (
           SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

```
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'number of threads')) )) =
            1)) )) = 0)) )) = 0);
 wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
    (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'fit class 2')) )) <= 1)) ))
            = 0)) )) = 0;
 wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'nominal size')) )) <= 1)) ))
            = 0)) )) = 0;
 wr16: (NOT (SELF\characterized_object.description IN ['knurl','gear',
            'thread'])) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(
            SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'applied shape') AND
        ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
            relating_shape_aspect)) )) = 1) )) <= 1)) )) = 0);
 wr17: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
```

```
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) AND (10 <= SIZEOF(pdr.
            used_representation.items))) AND (SIZEOF(pdr.
            used_representation.items) >= 11)) )) = 1) )) = 1);
  wr18: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
  ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'number of teeth')) )) = 1)) ))
            = 0)) )) = 0);
  wr19: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
  ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
         'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
         * TYPEOF(it)) = 2) AND (it.name = 'reference presure angle')) ))
            = 1)) )) = 0)) )) = 0);
  wr20: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'rake shift factor')) )) =
             1)) )) = 0)) )) = 0);
  wr21: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
            SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
```



```
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
           'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'nominal tool depth')) ))
           = 1)) )) = 0)) )) = 0);
wr22: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
           SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items | ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
           'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'face width')) )) = 1)) ))
           = 0)) )) = 0);
wr23: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
           SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items | ((SIZEOF([
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
           'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'tip diameter')) )) = 1)) ))
           = 0)) )) = 0);
wr24: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
           SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items |
     ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
           IN TYPEOF(it)) AND (it.name = 'module or diametral pitch'))
           AND ((it.description = 'module') OR (it.description =
           'diameter pitch'))) )) = 1)) )) = 0)) )) = 0);
wr25: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
           SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'DIMENSIONAL_INSPECTION_SCHEMA.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items | ((SIZEOF([
```



```
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                TYPEOF(it)) = 2) AND (it.name = 'normal attribute')) )) =
               1)) )) = 0)) )) = 0);
     wr26: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
               SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
               IN TYPEOF(it)) AND (it.name = 'internal or external gear'))
               AND ((it.description = 'internal') OR (it.description =
               'external'))) )) = 1)) )) = 0)) )) = 0);
     wr27: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
               SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'root fillet radius')) ))
               <= 1)) )) = 0)) )) = 0);
 END_ENTITY; -- externally_defined_feature_definition

 ENTITY externally_defined_item;
     item_id : source_item;
     source  : external_source;
 END_ENTITY; -- externally_defined_item

 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 | (
              'DIMENSIONAL_INSPECTION_SCHEMA.FACE_OUTER_BOUND' IN TYPEOF(
              temp)) )) <= 1;
 END_ENTITY; -- face

 ENTITY face_bound
   SUBTYPE OF (topological_representation_item);
     bound       : loop;
     orientation : BOOLEAN;
 END_ENTITY; -- face_bound
```



```
  ENTITY face_outer_bound
    SUBTYPE OF (face_bound);
  END_ENTITY; -- face_outer_bound

  ENTITY face_shape_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: SIZEOF(SELF.items) >= 1;
      wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT ((
               'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SURFACE' IN TYPEOF(it))
               OR ('DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_FACE' IN TYPEOF(
               it)))) )) = 0;
  END_ENTITY; -- face_shape_representation

ENTITY face_shape_representation_relationship
  SUBTYPE OF (representation_relationship);
      WHERE
      wr1: ('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
            IN TYPEOF(SELF.rep_1));
      wr2: ('DIMENSIONAL_INSPECTION_SCHEMA.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 ('DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_SURFACE' IN
               TYPEOF(face_geometry));
  END_ENTITY; -- face_surface

  ENTITY faceted_brep
    SUBTYPE OF (manifold_solid_brep);
  END_ENTITY; -- faceted_brep

  ENTITY feature_component_definition
    SUBTYPE OF (characterized_object);
    WHERE
      wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE')) = 1)) ))
               = 0;
  END_ENTITY; -- feature_component_definition

  ENTITY feature_component_relationship
    SUPERTYPE OF (ONEOF (pattern_omit_membership,pattern_offset_membership))
    SUBTYPE OF (shape_aspect_relationship);
    WHERE
      wr1: ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT',
               'DIMENSIONAL_INSPECTION_SCHEMA.REPLICATE_FEATURE',
               'DIMENSIONAL_INSPECTION_SCHEMA.TRANSITION_FEATURE',
               'DIMENSIONAL_INSPECTION_SCHEMA.MODIFIED_PATTERN'] * TYPEOF(
               SELF.relating_shape_aspect)) = 1) OR (
```

```
               'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_DEFINITION' IN
               TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
               ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
               IN TYPEOF(SELF.relating_shape_aspect.of_shape.definition));
  END_ENTITY; -- feature_component_relationship

  ENTITY feature_definition
    SUBTYPE OF (characterized_object);
    WHERE
      wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
               SELF) |
       ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
               IN TYPEOF(pdr.used_representation)) )) <= 1;
      wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
                (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
      wr3: SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.BOSS',
               'DIMENSIONAL_INSPECTION_SCHEMA.TURNED_KNURL',
               'DIMENSIONAL_INSPECTION_SCHEMA.THREAD',
               'DIMENSIONAL_INSPECTION_SCHEMA.GEAR',
               'DIMENSIONAL_INSPECTION_SCHEMA.MARKING',
               'DIMENSIONAL_INSPECTION_SCHEMA.RIB_TOP',
               'DIMENSIONAL_INSPECTION_SCHEMA.ROUND_HOLE',
               'DIMENSIONAL_INSPECTION_SCHEMA.OUTSIDE_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.POCKET',
               'DIMENSIONAL_INSPECTION_SCHEMA.REMOVAL_VOLUME',
               'DIMENSIONAL_INSPECTION_SCHEMA.REVOLVED_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.OUTER_ROUND',
               'DIMENSIONAL_INSPECTION_SCHEMA.FLAT_FACE',
               'DIMENSIONAL_INSPECTION_SCHEMA.PROTRUSION',
               'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_END',
               'DIMENSIONAL_INSPECTION_SCHEMA.SLOT',
               'DIMENSIONAL_INSPECTION_SCHEMA.SPHERICAL_CAP',
               'DIMENSIONAL_INSPECTION_SCHEMA.STEP',
               'DIMENSIONAL_INSPECTION_SCHEMA.COMPOUND_FEATURE',
               'DIMENSIONAL_INSPECTION_SCHEMA.REPLICATE_FEATURE',
       'DIMENSIONAL_INSPECTION_SCHEMA.EXTERNALLY_DEFINED_FEATURE_DEFINITION']
               * TYPEOF(SELF)) = 1;
      wr4: (NOT (SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.ROUND_HOLE',
               'DIMENSIONAL_INSPECTION_SCHEMA.BOSS',
               'DIMENSIONAL_INSPECTION_SCHEMA.OUTSIDE_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.REMOVAL_VOLUME',
               'DIMENSIONAL_INSPECTION_SCHEMA.FLAT_FACE',
               'DIMENSIONAL_INSPECTION_SCHEMA.POCKET',
               'DIMENSIONAL_INSPECTION_SCHEMA.PROTRUSION',
               'DIMENSIONAL_INSPECTION_SCHEMA.RIB_TOP',
               'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_END',
               'DIMENSIONAL_INSPECTION_SCHEMA.SLOT',
```



```
              'DIMENSIONAL_INSPECTION_SCHEMA.STEP'] * TYPEOF(SELF)) = 1))
              OR (SIZEOF(QUERY ( pdr <*
              get_property_definition_representations(SELF) | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.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,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
              used_representation.items | (NOT (
            'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(srwp_i))) ))
              > 0)) )) > 0) )) = 0;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.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,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0);
    wr2: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
              SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

```
          used_representation)) ) | ((NOT (SIZEOF(impl_rep.
          used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
          used_representation.items) <= 3)) )) = 0)) )) = 0);
wr3: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
         TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
          = 0);
wr4: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'first offset')) )) <= 1)) ))
          = 0)) )) = 0);
wr5: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) <= 1)) ))
          = 0)) )) = 0);
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
          'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation)) AND (pdr.
          used_representation.name = 'fillet face')) )) = 1)) )) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
          'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
```



```
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
                 TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'first face shape')) )) = 1)) ))
                = 0;
      wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
                'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.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,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'removal direction')) )) = 1)) ))
                = 0;
      wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
                sa_occ.description = 'course of travel occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'path feature component usage') AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (((
                'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT' IN
                TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
                relating_shape_aspect.description = 'linear')) AND (sdr.name
                = 'course of travel')) )) = 1)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
                sa_occ.description = 'removal boundary occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
```

```
            IN TYPEOF(sar))) ) | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.LINEAR_PROFILE' IN TYPEOF(sdr
           .relating_shape_aspect)) AND (sdr.name = 'removal boundary')) ))
            = 1)) )) = 1)) )) = 0;
    wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'enclosed boundary occurrence') AND (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
             'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.NGON_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_PATH_PROFILE'] *
            TYPEOF(sdr.relating_shape_aspect)) = 1) AND (sdr.
            relating_shape_aspect.description = 'boundary')) )) = 1)) ))
            <= 1)) )) = 0;
    wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
            SELF) | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
            IN TYPEOF(pdr.used_representation)) AND (pdr.
            used_representation.name = 'maximum feature limit')) )) >= 0;
    wr6: SIZEOF(QUERY ( pds <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (((
            'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(csa)) AND (csa.name = 'uncut area')) AND (SIZEOF(
            QUERY ( sar <* csa.component_relationships |
          (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar)) AND (SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.BOSS',
            'DIMENSIONAL_INSPECTION_SCHEMA.PROTRUSION'] * TYPEOF(sar.
            related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1;
    wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'removal depth')) )) <= 1)) ))
            = 0)) )) = 0;
  END_ENTITY; -- flat_face
```

```
  ENTITY flatness_tolerance
    SUBTYPE OF (geometric_tolerance);
    WHERE
      wr1: NOT (('DIMENSIONAL_INSPECTION_SCHEMA.' +
               'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
  END_ENTITY; -- flatness_tolerance

  ENTITY founded_item;
  END_ENTITY; -- founded_item

  ENTITY functionally_defined_transformation;
      name        : label;
      description : OPTIONAL text;
  END_ENTITY; -- functionally_defined_transformation

  ENTITY gear
    SUBTYPE OF (feature_definition);
    WHERE
      wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
    | ((('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) AND (10 <= SIZEOF(pdr.
                used_representation.items))) AND (SIZEOF(pdr.
                used_representation.items) >= 13)) )) = 1) )) = 1;
      wr2 : (NOT (SELF\characterized_object.description IN [
                'straight bevel gear','helical bevel gear','spur gear',
                'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
     | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'number of teeth')) )) = 1)) ))
                = 0)) )) = 0);
      wr3 : (NOT (SELF\characterized_object.description IN [
                'straight bevel gear','helical bevel gear','spur gear',
                'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
      | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
```

```
                'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name =
               'reference pressure angle')) )) = 1)) )) = 0)) )) = 0);
      wr4 : (NOT (SELF\characterized_object.description IN [
               'straight bevel gear','helical bevel gear','spur gear',
               'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
     | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
               IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
               QUERY ( it <* impl_rep.used_representation.items | ((
               SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                TYPEOF(it)) = 2) AND (it.name = 'rake shift factor')) )) =
                1)) )) = 0)) )) = 0);
      wr5 : (NOT (SELF\characterized_object.description IN [
               'straight bevel gear','helical bevel gear','spur gear',
               'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
      | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
               IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
               QUERY ( it <* impl_rep.used_representation.items | ((
               SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                TYPEOF(it)) = 2) AND (it.name = 'nominal tooth depth')) ))
               = 1)) )) = 0)) )) = 0);
      wr6 : (NOT (SELF\characterized_object.description IN [
               'straight bevel gear','helical bevel gear','spur gear',
               'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
     | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
               IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
               QUERY ( it <* impl_rep.used_representation.items | ((
               SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                TYPEOF(it)) = 2) AND (it.name = 'face width')) )) = 1)) ))
               = 0)) )) = 0);
      wr7 : (NOT (SELF\characterized_object.description IN [
               'straight bevel gear','helical bevel gear','spur gear',
               'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```



```
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
      | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items | ((
             SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'tip diameter')) )) = 1)) ))
             = 0)) )) = 0);
      wr8 : (NOT (SELF\characterized_object.description IN [
             'straight bevel gear','helical bevel gear','spur gear',
             'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
     | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items |
         ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (it.name = 'module or diametral pitch'))
             AND ((it.description = 'module') OR (it.description =
             'diametral pitch'))) )) = 1)) )) = 0)) )) = 0);
      wr9 : (NOT (SELF\characterized_object.description IN [
             'straight bevel gear','helical bevel gear','spur gear',
             'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
     | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items |
        ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (it.name = 'internal or external gear'))
             AND ((it.description = 'internal') OR (it.description =
             'external'))) )) = 1)) )) = 0)) )) = 0);
      wr10: (NOT (SELF\characterized_object.description IN [
             'straight bevel gear','helical bevel gear','spur gear',
             'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
      | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items | ((
             SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
```

```
                 TYPEOF(it)) = 2) AND (it.name = 'normal attribute')) )) =
                1)) )) = 0)) )) = 0);
      wr11: (NOT (SELF\characterized_object.description IN [
                'straight bevel gear','helical bevel gear','spur gear',
                'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
      | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                 TYPEOF(it)) = 2) AND (it.name = 'root fillet radius')) ))
                <= 1)) )) = 0)) )) = 0);
      wr12: (NOT (SELF\characterized_object.description IN ['helix gear',
                'helical bevel gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(
                SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
     | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'reference helix angle')) ))
                = 1)) )) = 0)) )) = 0);
      wr13: (NOT (SELF\characterized_object.description IN ['helix gear',
                'helical bevel gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(
                SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
     | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items |
           ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (it.name = 'gear tooth')) AND ((it.
                description = 'left hand tooth') OR (it.description =
                'right hand tooth'))) )) = 1)) )) = 0)) )) = 0);
      wr14: (NOT (SELF\characterized_object.description IN [
                'straight bevel gear','helical bevel gear'])) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
```



```
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
      | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'tip angle')) )) = 1)) ))
                = 0)) )) = 0);
      wr15: (NOT (SELF\characterized_object.description IN [
                'straight bevel gear','helical bevel gear'])) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
)
      | ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'root angle')) )) = 1)) ))
                = 0)) )) = 0);
      wr16: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                 TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT
')
             | ((sar.description = 'applied shape') AND
             ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
                relating_shape_aspect)) )) = 1) )) = 1)) )) = 0;
  END_ENTITY; -- gear

  ENTITY geometric_alignment
    SUBTYPE OF (derived_shape_aspect);
    WHERE
      wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1;
  END_ENTITY; -- geometric_alignment

  ENTITY geometric_intersection
    SUBTYPE OF (derived_shape_aspect);
    WHERE
      wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1;
  END_ENTITY; -- geometric_intersection

  ENTITY geometric_representation_context
    SUBTYPE OF (representation_context);
```

```
    coordinate_space_dimension : dimension_count;
END_ENTITY; -- geometric_representation_context

ENTITY geometric_representation_item
  SUPERTYPE OF (ONEOF (point,direction,vector,placement,
      cartesian_transformation_operator,curve,surface,edge_curve,
      face_surface,poly_loop,vertex_point,solid_model))
  SUBTYPE OF (representation_item);
  DERIVE
    dim : dimension_count := dimension_of(SELF);
  WHERE
    wr1: SIZEOF(QUERY ( using_rep <* using_representations(SELF) | (NOT
        ('DIMENSIONAL_INSPECTION_SCHEMA.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 group;
    name        : label;
    description : OPTIONAL text;
  DERIVE
    id : identifier := get_id_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- group

ENTITY group_assignment
  ABSTRACT SUPERTYPE;
```



```
    assigned_group : group;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- group_assignment

ENTITY group_relationship;
    name           : label;
    description    : OPTIONAL text;
    relating_group : group;
    related_group  : group;
END_ENTITY; -- group_relationship

ENTITY 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 hole_bottom
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
              IN TYPEOF(SELF.of_shape.definition);
    wr2 : SELF.description IN ['through','flat','flat with radius',
              'flat with taper','spherical','conical'];
    wr3 : (NOT (SELF.description = 'through')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(USEDIN(pd,'DIMENSIONAL_INSPECTION_SCHEMA.'
              + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 0)) ))
              = 0);
    wr4 : (NOT (SELF.description IN ['flat with radius',
              'flat with taper','spherical','conical'])) OR (SIZEOF(
              QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0);
    wr5 : (NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

```
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 0)) )) = 0)) )) = 0);
  wr6 : (NOT (SELF.description IN ['flat with radius','spherical']))
            OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 1)) )) = 0)) )) = 0);
  wr7 : (NOT (SELF.description = 'flat with taper')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 2)) )) = 0)) )) = 0);
  wr8 : (NOT (SELF.description = 'conical')) OR (SIZEOF(QUERY ( pd <*
            USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | ((NOT (SIZEOF(impl_rep.
            used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
            used_representation.items) <= 2)) )) = 0)) )) = 0);
  wr9 : (SELF.description = 'through') OR (SIZEOF(QUERY ( fcr <*
            QUERY ( sar <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATING_SHAPE_ASPECT') | ((sar.description =
            'hole bottom usage') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (fcr.name IN ['hole depth start',
            'hole depth end']) )) >= 1);
  wr10: (NOT (SELF.description = 'flat with radius')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
   ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) = 1)) ))
            = 0)) )) = 0);
  wr11: (NOT (SELF.description = 'spherical')) OR (SIZEOF(
```



```
          QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
          = 0);
wr12: (NOT (SELF.description = 'conical')) OR (SIZEOF(QUERY ( pd <*
          USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'tip radius')) )) <= 1)) ))
          = 0)) )) = 0);
wr13: (NOT (SELF.description = 'conical')) OR (SIZEOF(QUERY ( pd <*
          USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
 ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(it)) = 2) AND (it.name = 'tip angle')) )) = 1)) ))
          = 0)) )) = 0);
wr14: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATING_SHAPE_ASPECT') | ((sar.description =
          'hole bottom usage') AND
    ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
          IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.
          description = 'bottom condition occurrence') AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.ROUND_HOLE' IN TYPEOF(fcr.
          related_shape_aspect.of_shape.definition))) AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.HOLE_BOTTOM' IN TYPEOF(fcr.
          relating_shape_aspect))) )) >= 1;
wr15: (NOT (SELF.description = 'flat with taper')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
```



```
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'final diameter')) )) = 1)) ))
                = 0)) )) = 0);
      wr16: (NOT (SELF.description = 'flat with taper')) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
             * TYPEOF(it)) = 2) AND (it.name = 'taper diameter')) )) = 1)) ))
                = 0)) )) = 0);
  END_ENTITY; -- hole_bottom

  ENTITY hyperbola
    SUBTYPE OF (conic);
      semi_axis      : positive_length_measure;
      semi_imag_axis : positive_length_measure;
  END_ENTITY; -- hyperbola

  ENTITY id_attribute;
      attribute_value : identifier;
      identified_item : id_attribute_select;
  END_ENTITY; -- id_attribute

  ENTITY instanced_feature
    SUBTYPE OF (feature_definition, shape_aspect);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(
               SELF.of_shape.definition);
      wr2: SELF.product_definitional;
  END_ENTITY; -- instanced_feature

  ENTITY length_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_UNIT' IN TYPEOF(SELF\
               measure_with_unit.unit_component);
  END_ENTITY; -- length_measure_with_unit

  ENTITY length_unit
    SUBTYPE OF (named_unit);
    WHERE
      wr1: (((((SELF\named_unit.dimensions.length_exponent = 1) AND (SELF
```



```
              \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
              named_unit.dimensions.time_exponent = 0)) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0)) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0)) AND (SELF\
              named_unit.dimensions.amount_of_substance_exponent = 0)) AND
               (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- length_unit

ENTITY 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 (('DIMENSIONAL_INSPECTION_SCHEMA.' +
              'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF)))
              OR (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
              datum_system) <= 3);
    wr2: SIZEOF(QUERY ( sar <* USEDIN(SELF\geometric_tolerance.
              toleranced_shape_aspect,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
              name IN ['affected plane association',
              'resulting intersection curve association']) )) = 1;
END_ENTITY; -- line_profile_tolerance

ENTITY linear_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
               TYPEOF(SELF.of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

```
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(impl_rep.
              used_representation.items) = 2)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
               (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'profile length')) )) = 1)) ))
              = 0)) )) = 0;
END_ENTITY; -- linear_profile

ENTITY local_time;
    hour_component   : hour_in_day;
    minute_component : OPTIONAL minute_in_hour;
    second_component : OPTIONAL second_in_minute;
    zone             : coordinated_universal_time_offset;
  WHERE
    wr1: valid_time(SELF);
END_ENTITY; -- local_time

ENTITY location_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SIZEOF(SELF.items) = 1;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT (
              'DIMENSIONAL_INSPECTION_SCHEMA.POINT' IN TYPEOF(it))) )) = 0;
END_ENTITY; -- location_shape_representation

ENTITY loop
  SUPERTYPE OF (ONEOF (vertex_loop,edge_loop,poly_loop))
  SUBTYPE OF (topological_representation_item);
END_ENTITY; -- loop

ENTITY lot_effectivity
  SUBTYPE OF (effectivity);
    effectivity_lot_id   : identifier;
    effectivity_lot_size : measure_with_unit;
```



```
END_ENTITY; -- lot_effectivity

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,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) AND ((2 <= SIZEOF(pdr.
             used_representation.items)) AND (SIZEOF(pdr.
             used_representation.items) <= 6))) )) = 1) )) = 1;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (it.name = 'marking text')) )) = 1)) )) =
              0)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items |
        (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (it.name = 'special instructions')) )) <=
              1)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
```

```
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
          (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (it.name = 'font name')) )) <= 1)) )) = 0)) ))
                = 0;
      wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'character height')) )) <= 1)) ))
                = 0)) )) = 0;
      wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                TYPEOF(it)) = 2) AND (it.name = 'character spacing')) )) <=
                1)) )) = 0)) )) = 0;
      wr7: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'applied shape') AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
                relating_shape_aspect)) )) = 1) )) = 1)) )) = 0;
  END_ENTITY; -- marking

  ENTITY mass_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.MASS_UNIT' IN TYPEOF(SELF\
                measure_with_unit.unit_component);
  END_ENTITY; -- mass_measure_with_unit
```

```
ENTITY mass_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF
              \named_unit.dimensions.mass_exponent = 1)) AND (SELF\
              named_unit.dimensions.time_exponent = 0)) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0)) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0)) AND (SELF\
              named_unit.dimensions.amount_of_substance_exponent = 0)) AND
               (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- mass_unit

ENTITY measure_qualification;
    name              : label;
    description       : text;
    qualified_measure : measure_with_unit;
    qualifiers        : SET [1:?] OF value_qualifier;
  WHERE
    wr1: SIZEOF(QUERY ( temp <* qualifiers | (
              'DIMENSIONAL_INSPECTION_SCHEMA.PRECISION_QUALIFIER' IN
              TYPEOF(temp)) )) < 2;
END_ENTITY; -- measure_qualification

ENTITY measure_representation_item
  SUBTYPE OF (representation_item, measure_with_unit);
END_ENTITY; -- measure_representation_item

ENTITY measure_with_unit
  SUPERTYPE OF (ONEOF (length_measure_with_unit,mass_measure_with_unit,
      time_measure_with_unit,plane_angle_measure_with_unit,
      ratio_measure_with_unit));
    value_component : measure_value;
    unit_component  : unit;
  WHERE
    wr1: valid_units(SELF);
END_ENTITY; -- measure_with_unit

ENTITY measurement_location
  SUBTYPE OF (action_property);
  WHERE
    wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.DM_PROGRAM_RUN'
    IN TYPEOF(SELF.definition);
END_ENTITY; -- measurement_location

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,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATING_SHAPE_ASPECT') |
```

```
          ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) ) | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.REPLICATE_FEATURE',
               'DIMENSIONAL_INSPECTION_SCHEMA.INSTANCED_FEATURE'] * TYPEOF(
               fcr.related_shape_aspect.of_shape.definition)) >= 1) AND (
               fcr.description = 'base shape')) )) = 1;
      wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATING_SHAPE_ASPECT') |
            ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) ) | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN',
               'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN'] *
               TYPEOF(fcr.related_shape_aspect.of_shape.definition)) = 1)
               AND (fcr.description = 'base pattern')) )) = 1;
      wr3: SIZEOF(QUERY ( sar <* USEDIN(SELF,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATING_SHAPE_ASPECT') | (SIZEOF(QUERY ( msar <* USEDIN(
               sar.related_shape_aspect,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATED_SHAPE_ASPECT') | (((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.PATTERN_OFFSET_MEMBERSHIP',
               'DIMENSIONAL_INSPECTION_SCHEMA.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,mass_unit,
        time_unit,plane_angle_unit,solid_angle_unit,ratio_unit));
      dimensions : dimensional_exponents;
  END_ENTITY; -- named_unit

  ENTITY ngon_closed_profile
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
                TYPEOF(SELF.of_shape.definition);
      wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
```



```
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT ((SIZEOF(impl_rep.
          used_representation.items) >= 3) AND (SIZEOF(impl_rep.
          used_representation.items) <= 4))) )) = 0)) )) = 0;
wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
          used_representation.items | ((((srwp_i.name = 'orientation')
          OR (srwp_i.name = 'number of sides')) OR (srwp_i.name =
          'circumscribed diameter')) OR (srwp_i.name = 'corner radius'))
          OR (srwp_i.name = 'diameter across flats')) )) = SIZEOF(pdr.
          used_representation.items))) )) = 1) )) = 1;
wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
           (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | (((
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
          measure_with_unit.value_component))) AND (it.name =
          'number of sides')) )) = 1)) )) = 0)) )) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name IN ['circumscribed diameter',
```

```
              'diameter across flats'])) )) = 1)) )) = 0)) )) = 0;
    wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) <= 1)) ))
              = 0)) )) = 0;
END_ENTITY; -- ngon_closed_profile

ENTITY ngon_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
    wr1: SIZEOF(SELF.items) = 5;
    wr2: SIZEOF(QUERY ( it <* SELF.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
               (it.name = 'orientation')) )) = 1;
    wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
    wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) = 1;
    wr5: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name IN ['circumscribed diameter',
              'diameter across flats'])) )) = 1;
    wr6: SIZEOF(QUERY ( it <* SELF.items | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
              IN TYPEOF(it)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.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: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
               TYPEOF(SELF.of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```



```
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(impl_rep.
             used_representation.items) = 1)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
              (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
             'DIMENSIONAL_INSPECTION_SCHEMA.PATH_SHAPE_REPRESENTATION' IN
              TYPEOF(pdr.used_representation)) )) = 1)) )) = 0;
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) AND (pdr.
             used_representation.name = 'profile limit')) )) <= 1)) )) =
             0;
END_ENTITY; -- open_path_profile

ENTITY open_shell
  SUBTYPE OF (connected_face_set);
END_ENTITY; -- open_shell

ENTITY organization;
    id          : OPTIONAL identifier;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- organization

ENTITY organization_assignment
  ABSTRACT SUPERTYPE;
```



```
    assigned_organization : organization;
    role                  : organization_role;
END_ENTITY; -- organization_assignment

ENTITY organization_role;
    name : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- organization_role

ENTITY organizational_address
  SUBTYPE OF (address);
    organizations : SET [1:?] OF organization;
    description   : OPTIONAL text;
END_ENTITY; -- organizational_address

ENTITY organizational_project;
    name                        : label;
    description                 : OPTIONAL text;
    responsible_organizations : SET [1:?] OF organization;
  DERIVE
    id : identifier := get_id_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- organizational_project

ENTITY oriented_closed_shell
  SUBTYPE OF (closed_shell);
    closed_shell_element : closed_shell;
    orientation          : BOOLEAN;
  DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                         conditional_reverse(SELF.
                                         orientation,SELF.
                                         closed_shell_element.cfs_faces);
  WHERE
    wr1: NOT ('DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_CLOSED_SHELL' IN
             TYPEOF(SELF.closed_shell_element));
END_ENTITY; -- oriented_closed_shell

ENTITY oriented_edge
  SUBTYPE OF (edge);
    edge_element : edge;
    orientation  : BOOLEAN;
  DERIVE
    SELF\edge.edge_start : vertex := boolean_choose(SELF.orientation,
                              SELF.edge_element.edge_start,SELF.
                              edge_element.edge_end);
    SELF\edge.edge_end   : vertex := boolean_choose(SELF.orientation,
                              SELF.edge_element.edge_end,SELF.
                              edge_element.edge_start);
  WHERE
```



```
    wr1: NOT ('DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_EDGE' IN TYPEOF(
             SELF.edge_element));
END_ENTITY; -- oriented_edge

ENTITY oriented_face
  SUBTYPE OF (face);
    face_element : face;
    orientation  : BOOLEAN;
  DERIVE
    SELF\face.bounds : SET [1:?] OF face_bound := conditional_reverse(
                          SELF.orientation,SELF.face_element.bounds);
  WHERE
    wr1: NOT ('DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_FACE' IN TYPEOF(
             SELF.face_element));
END_ENTITY; -- oriented_face

ENTITY oriented_open_shell
  SUBTYPE OF (open_shell);
    open_shell_element : open_shell;
    orientation        : BOOLEAN;
  DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                       conditional_reverse(SELF.
                                       orientation,SELF.
                                       open_shell_element.cfs_faces);
  WHERE
    wr1: NOT ('DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_OPEN_SHELL' IN
             TYPEOF(SELF.open_shell_element));
END_ENTITY; -- oriented_open_shell

ENTITY oriented_path
  SUBTYPE OF (path);
    path_element : path;
    orientation  : BOOLEAN;
  DERIVE
    SELF\path.edge_list : LIST [1:?] OF UNIQUE oriented_edge :=
                          conditional_reverse(SELF.orientation,SELF.
                          path_element.edge_list);
  WHERE
    wr1: NOT ('DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_PATH' IN TYPEOF(
             SELF.path_element));
END_ENTITY; -- oriented_path

ENTITY oriented_surface
  SUBTYPE OF (surface);
    orientation : BOOLEAN;
END_ENTITY; -- oriented_surface

ENTITY outer_round
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (NOT (SELF\characterized_object.description = 'outer diameter'))
             OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
```

```
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
((('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.
          used_representation.items) = 3)) )) = 1) )) = 1);
 wr2: (NOT (SELF\characterized_object.description =
          'outer diameter to shoulder')) OR (SIZEOF(QUERY ( pd <*
          USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) AND ((2 <= SIZEOF(pdr.
          used_representation.items)) AND (SIZEOF(pdr.
          used_representation.items) <= 3))) )) = 1) )) = 1);
 wr3: SELF\characterized_object.description IN ['outer diameter',
          'outer diameter to shoulder'];
 wr4: (NOT (SELF\characterized_object.description = 'outer diameter'))
          OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1)) )) = 0)) ))
          = 0);
 wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1)) )) = 0)) ))
          = 0;
 wr6: (NOT (SELF\characterized_object.description =
          'outer diameter to shoulder')) OR (SIZEOF(QUERY ( pds <*
          QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
          IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
          sa_occ.description = 'v-shape boundary occurrence') AND (
          SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
           'RELATED_SHAPE_ASPECT') | ((sar.description =
          'profile usage') AND (
```



```
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE' IN TYPEOF(sdr.
            relating_shape_aspect)) AND (sdr.relating_shape_aspect.
            description = 'v-shape')) )) = 1)) )) = 1)) )) = 0);
    wr7: (NOT (SELF\characterized_object.description = 'outer diameter'))
            OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'reduced size occurrence') AND (SIZEOF(
            QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
             'RELATED_SHAPE_ASPECT') | ((sar.description = 'taper usage')
      AND ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (((
            'DIMENSIONAL_INSPECTION_SCHEMA.TAPER' IN TYPEOF(sdr.
            relating_shape_aspect)) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.OUTER_ROUND' IN TYPEOF(sdr.
            related_shape_aspect.of_shape.definition))) AND (sdr.name =
            'reduced size')) )) = 1)) )) <= 1)) )) = 0);
    wr8: (NOT (SELF\characterized_object.description =
            'outer diameter to shoulder')) OR (SIZEOF(QUERY ( pd <*
            USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'length')) )) <= 1)) )) = 0)) ))
            = 0);
    wr9: (NOT (SELF\characterized_object.description =
            'outer diameter to shoulder')) OR (SIZEOF(QUERY ( pd <*
            USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'feature length')) )) <= 1)) ))
            = 0)) )) = 0);
END_ENTITY; -- outer_round

ENTITY outside_profile
  SUBTYPE OF (feature_definition);
```

```
WHERE
  wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
   (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
            IN TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.
            used_representation.items) = 1)) )) = 1) )) = 1;
  wr2 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            sa_occ.description IN ['boundary occurrence',
            'non-planar boundary occurrence',
            'partial circular boundary occurrence',
            'closed circular boundary occurrence',
            'open rectangular boundary occurrence',
            'closed rectangular boundary occurrence']) )) = 1)) )) = 0;
  wr3 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            sa_occ.description = 'boundary occurrence') )) = 1)) )) = 0))
            OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND (('DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
            SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.NGON_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_PATH_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.LINEAR_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.TEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_PATH_PROFILE'] *
            TYPEOF(sdr.relating_shape_aspect)) = 1) AND (sdr.
            relating_shape_aspect.description = 'outside boundary')) ))
            = 1) )) = 1)) )) = 0);
  wr4 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```



```
           | (
          'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
          sa_occ.description IN ['complex boundary occurrence',
          'partial circular boundary occurrence',
          'closed circular boundary occurrence',
          'open rectangular boundary occurrence',
          'closed rectangular boundary occurrence']) )) = 1)) )) = 0))
          OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (
          'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
          NOT (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATED_SHAPE_ASPECT') | ((sar.description =
          'profile floor usage') AND
    ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
          IN TYPEOF(sar))) ) | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.PROFILE_FLOOR' IN TYPEOF(sdr
          .relating_shape_aspect)) AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.OUTSIDE_PROFILE' IN TYPEOF(
          sdr.related_shape_aspect.of_shape.definition))) )) = 1)) ))
          = 0)) )) = 0);
wr5 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (
          'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
          sa_occ.description IN ['outside boundary',
          'complex boundary occurrence',
          'partial circular boundary occurrence',
          'closed circular boundary occurrence',
          'open rectangular boundary occurrence',
          'closed rectangular boundary occurrence']) )) = 1)) )) = 0))
          OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (
          'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
          SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATED_SHAPE_ASPECT') | ((sar.description =
          'path feature component usage') AND ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (((
          SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT'] *
          TYPEOF(sdr.relating_shape_aspect)) = 1) AND (sdr.name =
          'profile swept shape')) AND (sdr.relating_shape_aspect.
          description = 'linear')) )) = 1) )) = 1)) )) = 0);
wr6 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
```



```
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
          sa_occ.description = 'complex boundary occurrence') )) = 1)) ))
            = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND
            ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.NGON_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_PATH_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.TEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.LINEAR_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_PATH_PROFILE'] *
            TYPEOF(sdr.relating_shape_aspect)) = 1) )) = 1) )) = 1)) ))
            = 0);
  wr7 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
          sa_occ.description = 'partial circular boundary occurrence') ))
            = 1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
            USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
             TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
  wr8 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
```



```
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
           sa_occ.description = 'closed circular boundary occurrence') ))
            = 1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
            USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
            TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
  wr9 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
          sa_occ.description = 'open rectangular boundary occurrence') ))
            = 1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
            USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND
          ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE' IN TYPEOF(
            sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
  wr10: (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            sa_occ.description =
            'closed rectangular boundary occurrence') )) = 1)) )) = 0))
            OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
```

```
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND
               ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE'
                IN TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) =
                 0);
      wr11: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                 TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
                sa_occ.description IN ['boundary occurrence',
                'complex boundary occurrence',
                'partial circular boundary occurrence',
                'closed circular boundary occurrence',
                'open rectangular boundary occurrence',
                'closed rectangular boundary occurrence']) )) = 1)) )) = 0)
                OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'removal direction')) )) = 1)) ))
                = 0);
      wr12: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
                SELF) |
            (('DIMENSIONAL_INSPECTION_SCHEMA.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 parallel_offset
    SUBTYPE OF (derived_shape_aspect);
      offset : measure_with_unit;
    WHERE
      wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
  END_ENTITY; -- parallel_offset

  ENTITY parallelism_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
      wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
               datum_system) < 3;
```



```
END_ENTITY; -- parallelism_tolerance

ENTITY parametric_representation_context
  SUBTYPE OF (representation_context);
END_ENTITY; -- parametric_representation_context

ENTITY partial_circular_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
              TYPEOF(SELF.of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(impl_rep.
             used_representation.items) >= 3)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
              (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
             = 0;
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
```

```
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'sweep angle')) )) = 1)) ))
               = 0)) )) = 0;
      wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
               IN TYPEOF(pdr.used_representation)) AND (pdr.
               used_representation.name = 'profile limit')) )) <= 1)) )) =
               0;
  END_ENTITY; -- partial_circular_profile

  ENTITY path
    SUPERTYPE OF (ONEOF (edge_loop,oriented_path))
    SUBTYPE OF (topological_representation_item);
      edge_list : LIST [1:?] OF UNIQUE oriented_edge;
    WHERE
      wr1: path_head_to_tail(SELF);
  END_ENTITY; -- path

  ENTITY path_feature_component
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1 : 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
                IN TYPEOF(SELF.of_shape.definition);
      wr2 : SELF.description IN ['partial circular','complete circular',
                'linear','complex'];
      wr3 : (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
                USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) )) = 1)) )) = 0);
      wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it))
                AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
      wr5 : (NOT (SELF.description = 'partial circular')) OR (SIZEOF(
```



```
            QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 3)) )) = 0)) )) = 0);
  wr6 : (NOT (SELF.description = 'partial circular')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
  ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
            = 0);
  wr7 : (NOT (SELF.description = 'partial circular')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
  ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
            * TYPEOF(it)) = 2) AND (it.name = 'sweep angle')) )) = 1)) ))
            = 0)) )) = 0);
  wr8 : (NOT (SELF.description = 'complete circular')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 2)) )) = 0)) )) = 0);
  wr9 : (NOT (SELF.description = 'complete circular')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
```

```
              impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
              = 0);
    wr10: (NOT (SELF.description = 'linear')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(impl_rep.
              used_representation.items) = 2)) )) = 0)) )) = 0);
    wr11: (NOT (SELF.description = 'linear')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'distance')) )) = 1)) )) =
               0)) )) = 0);
    wr12: (NOT (SELF.description = 'linear')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
              IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
    wr13: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.PATH_SHAPE_REPRESENTATION'
              IN TYPEOF(pdr.used_representation)) AND (pdr.
              used_representation.name = 'sweep path')) AND (SIZEOF(
              QUERY ( srwp_i <* pdr.used_representation.items | (srwp_i.
              name = 'profile shape') )) = 1)) )) = 1)) )) = 0);
END_ENTITY; -- path_feature_component

ENTITY path_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SIZEOF(SELF.items) >= 1;
    wr2: SIZEOF(QUERY ( i <* SELF.items | (SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.BOUNDED_CURVE',
```



```
              'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_CURVE',
              'DIMENSIONAL_INSPECTION_SCHEMA.PATH'] * TYPEOF(i)) = 1) ))
              >= 1;
END_ENTITY; -- path_shape_representation

ENTITY pattern_offset_membership
  SUBTYPE OF (feature_component_relationship);
  WHERE
    wr1 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
              relating_shape_aspect,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATING_SHAPE_ASPECT') |
         (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | (SIZEOF(
              QUERY ( pdr <* QUERY ( pd <* USEDIN(fcr.
              related_shape_aspect.of_shape,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN',
              'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN'] *
              TYPEOF(pdr.definition)) = 1) )) = 0) )) = 0;
    wr2 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
              related_shape_aspect,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((fcr.description
              = 'modified pattern') AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(
              fcr.relating_shape_aspect))) )) >= 1;
    wr3 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
              related_shape_aspect,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(
              fcr.relating_shape_aspect)) AND (NOT (SIZEOF(
              QUERY ( modfcr <* QUERY ( modsar <* USEDIN(fcr.
              relating_shape_aspect,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
              SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN',
              'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN'] *
              TYPEOF(modsar.related_shape_aspect.of_shape.definition)) =
              1) AND (modsar :<>: fcr)) ) | (NOT (modfcr.
              related_shape_aspect.of_shape.definition :=: SELF.
              relating_shape_aspect.of_shape.definition)) )) = 0))) )) =
              0;
    wr4 : (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
              TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
               (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(USEDIN(pd,'DIMENSIONAL_INSPECTION_SCHEMA.'
              + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 2)) ))
```

```
            = 0);
  wr5 : (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN' IN
            TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
             (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(USEDIN(pd,'DIMENSIONAL_INSPECTION_SCHEMA.'
            + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
            = 0);
  wr6 : (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN' IN
            TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
             (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
            SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0);
  wr7 : (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN' IN
            TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
             (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
            measure_with_unit.value_component))) AND (it.name =
            'index number')) )) = 1)) )) = 0)) )) = 0);
  wr8 : (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN' IN
            TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
             (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
         (('DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'
            IN TYPEOF(it)) AND (it.name = 'offset')) )) = 1)) )) = 0)) ))
            = 0);
  wr9 : (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
            TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
             (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
```



```
          used_representation.items) = 3)) )) = 0)) )) = 0);
wr10: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
          TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
           (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
     ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
          measure_with_unit.value_component))) AND (it.name =
          'row index')) )) = 1)) )) = 0)) )) = 0);
wr11: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
          TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
           (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
          measure_with_unit.value_component))) AND (it.name =
          'column index')) )) = 1)) )) = 0)) )) = 0);
wr12: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
          TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
           (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
           TYPEOF(it)) AND (it.name = 'offset distance')) )) = 1)) ))
          = 0)) )) = 0);
wr13: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
          TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
           (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
          IN TYPEOF(pdr.used_representation)) AND (pdr.
```

```
                used_representation.name = 'offset direction')) )) = 1)) ))
                = 0);
  END_ENTITY; -- pattern_offset_membership

  ENTITY pattern_omit_membership
    SUBTYPE OF (feature_component_relationship);
    WHERE
      wr1: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
               relating_shape_aspect,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATING_SHAPE_ASPECT') |
         (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | (SIZEOF(
               QUERY ( pdr <* QUERY ( pd <* USEDIN(fcr.related_shape_aspect
               .of_shape,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN',
               'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN'] *
               TYPEOF(pdr.definition)) = 1) )) = 0) )) = 0;
      wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
               related_shape_aspect,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATED_SHAPE_ASPECT') |
          (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((fcr.description =
                'modified pattern') AND (
               'DIMENSIONAL_INSPECTION_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(
               fcr.relating_shape_aspect))) )) >= 1;
      wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
               related_shape_aspect,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATED_SHAPE_ASPECT') |
          (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(
               fcr.relating_shape_aspect)) AND (NOT (SIZEOF(
               QUERY ( modfcr <* QUERY ( modsar <* USEDIN(fcr.
               relating_shape_aspect,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
               SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN',
               'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN'] *
               TYPEOF(modsar.related_shape_aspect.of_shape.definition)) = 1)
               AND (modsar :<>: fcr)) ) | (NOT (modfcr.related_shape_aspect
               .of_shape.definition :=: SELF.relating_shape_aspect.of_shape
               .definition)) )) = 0))) )) = 0;
      wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(USEDIN(pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
               = 0;
      wr5: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN' IN
               TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
               (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```



```
          | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
         SIZEOF(pdr.used_representation.items) = 1)) )) = 0)) )) = 0);
wr6: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_PATTERN' IN
         TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
         (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | (((
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
         IN TYPEOF(it)) AND (
         'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
         measure_with_unit.value_component))) AND (it.name =
         'index number')) )) = 1)) )) = 0)) )) = 0);
wr7: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
         TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
         (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
         SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0);
wr8: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
         TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
         (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | (((
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
         IN TYPEOF(it)) AND (
         'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
         measure_with_unit.value_component))) AND (it.name =
         'row index')) )) = 1)) )) = 0)) )) = 0);
wr9: (NOT ('DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_PATTERN' IN
         TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
         (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | (((
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
```

```
              IN TYPEOF(it)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
              measure_with_unit.value_component))) AND (it.name =
              'column index')) )) = 1)) )) = 0)) )) = 0);
  END_ENTITY; -- pattern_omit_membership

  ENTITY pcurve
    SUBTYPE OF (curve);
      basis_surface      : surface;
      reference_to_curve : definitional_representation;
    WHERE
      wr1: SIZEOF(reference_to_curve\representation.items) = 1;
      wr2: 'DIMENSIONAL_INSPECTION_SCHEMA.CURVE' IN TYPEOF(
              reference_to_curve\representation.items[1]);
      wr3: reference_to_curve\representation.items[1]\
              geometric_representation_item.dim = 2;
  END_ENTITY; -- pcurve

  ENTITY perpendicular_to
    SUBTYPE OF (derived_shape_aspect);
    WHERE
      wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
  END_ENTITY; -- perpendicular_to

  ENTITY perpendicularity_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
      wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
              datum_system) <= 3;
  END_ENTITY; -- perpendicularity_tolerance

  ENTITY person;
      id            : identifier;
      last_name     : OPTIONAL label;
      first_name    : OPTIONAL label;
      middle_names  : OPTIONAL LIST [1:?] OF label;
      prefix_titles : OPTIONAL LIST [1:?] OF label;
      suffix_titles : OPTIONAL LIST [1:?] OF label;
    WHERE
      wr1: EXISTS(last_name) OR EXISTS(first_name);
  END_ENTITY; -- person

  ENTITY person_and_organization;
      the_person       : person;
      the_organization : organization;
    DERIVE
      name        : label := get_name_value(SELF);
      description : text := get_description_value(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
      wr2: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              '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,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               '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 placement
    SUPERTYPE OF (ONEOF (axis1_placement,axis2_placement_2d,
        axis2_placement_3d))
    SUBTYPE OF (geometric_representation_item);
      location : cartesian_point;
  END_ENTITY; -- placement

 ENTITY placed_datum_target_feature
  SUBTYPE OF (datum_target);
   WHERE
  wr1 : (SELF.description IN ['point','line','rectangle','circle']);

  wr2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);

  wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
        (('DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT'
        IN TYPEOF(it)) AND (it.name = 'orientation')) )) = 1)) ))
        = 0)) )) = 0);

  wr4 : ((NOT (SELF.description = 'point')) OR
        (SIZEOF(QUERY ( pd <*  USEDIN(SELF,
```

```
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <*
        QUERY ( pdr <* USEDIN(pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(dtm_rep.used_representation.items) = 1)) ))
        = 0)) )) = 0));

  wr5 : ((NOT (SELF.description IN ['line','circle'])) OR
        (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(dtm_rep.used_representation.items) = 2)) ))
         = 0)) )) = 0));

  wr6 : ((NOT (SELF.description = 'rectangle')) OR
        (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(dtm_rep.used_representation.items) = 3)) ))
        = 0)) )) = 0));

  wr7 : ((NOT (SELF.description = 'circle')) OR
        (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
        ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'target diameter')) ))
        = 1)) )) = 0)) )) = 0));

  wr8 : ((NOT (SELF.description = 'line')) OR
        (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
        'DIMENSIONAL_INSPECTION_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```



```
      (('DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
      ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
      = 0)) )) = 0));

wr9 : ((NOT (SELF.description = 'rectangle')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
      ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
      = 0)) )) = 0));

wr10: ((NOT (SELF.description = 'rectangle')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'DIMENSIONAL_INSPECTION_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
      ((SIZEOF([
      'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'target width'))))= 1))))
      = 0)) ))=0));
END_ENTITY; -- placed_datum_target_feature

ENTITY planar_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SIZEOF(SELF.items) = 1;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (
             'DIMENSIONAL_INSPECTION_SCHEMA.PLANE' IN TYPEOF(it)) )) = 1;
END_ENTITY; -- planar_shape_representation

ENTITY plane
  SUBTYPE OF (elementary_surface);
END_ENTITY; -- plane

ENTITY plane_angle_measure_with_unit
```

```
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_UNIT' IN TYPEOF(SELF
              \measure_with_unit.unit_component);
  END_ENTITY; -- plane_angle_measure_with_unit

  ENTITY plane_angle_unit
    SUBTYPE OF (named_unit);
    WHERE
      wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF
              \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
              named_unit.dimensions.time_exponent = 0)) AND (SELF\
              named_unit.dimensions.electric_current_exponent = 0)) AND (
              SELF\named_unit.dimensions.
              thermodynamic_temperature_exponent = 0)) AND (SELF\
              named_unit.dimensions.amount_of_substance_exponent = 0)) AND
               (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
  END_ENTITY; -- plane_angle_unit

  ENTITY plus_minus_tolerance;
      range                : tolerance_method_definition;
      toleranced_dimension : dimensional_characteristic;
    UNIQUE
      ur1 : toleranced_dimension;
  END_ENTITY; -- plus_minus_tolerance

  ENTITY pocket
    SUBTYPE OF (feature_definition);
    WHERE
      wr1 : SELF\characterized_object.description IN ['closed rectangular',
               'open rectangular','complex','circular cutout',
               'complex cutout','recess'];
      wr2 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'pocket depth occurrence') AND (
              SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | (((sar.description =
              'path feature component usage') AND (sar.name =
              'pocket depth')) AND
            ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT' IN
              TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
              relating_shape_aspect.description = 'linear')) )) = 1)) ))
              = 1)) )) = 0;
      wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(
              SELF) |
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
              IN TYPEOF(pdr.used_representation)) )) = 1;
      wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```



```
            | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) AND ((1 <= SIZEOF(pdr.
            used_representation.items)) AND (SIZEOF(pdr.
            used_representation.items) <= 2))) )) = 1) )) = 1;
  wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
            used_representation.items | ((srwp_i.name = 'orientation')
            OR (srwp_i.name = 'fillet radius')) )) = SIZEOF(pdr.
            used_representation.items))) )) = 1) )) = 1;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'fillet radius')) )) <= 1)) ))
            = 0)) )) = 0;
  wr7 : (NOT (SELF\characterized_object.description IN ['complex',
            'non-circular cutout','recess'])) OR (SIZEOF(
            QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
            | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'boundary occurrence') AND (SIZEOF(
            QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND
          ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.NGON_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_PATH_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.TEE_PROFILE',
```

```
            'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_PATH_PROFILE'] *
            TYPEOF(sdr.relating_shape_aspect)) = 1) )) = 1)) )) = 1)) ))
            = 0);
 wr8 : (NOT (SELF\characterized_object.description =
            'closed rectangular')) OR (SIZEOF(QUERY ( pds <*
            QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'closed boundary occurrence') AND (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND
         ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE'
            IN TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) ))
            = 0);
 wr9 : (NOT (SELF\characterized_object.description =
            'open rectangular')) OR (SIZEOF(QUERY ( pds <*
            QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'open boundary occurrence') AND (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND
         ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE' IN TYPEOF(
            sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
 wr10: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'bottom condition occurrence') AND (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'pocket bottom usage') AND
       ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar))) ) | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.POCKET_BOTTOM' IN TYPEOF(sdr
            .relating_shape_aspect)) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.POCKET' IN TYPEOF(sdr.
         related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
            = 0;
```



```
wr11: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (
          'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
          sa_occ.description = 'change in boundary occurrence') AND (
          SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATING_SHAPE_ASPECT') | ((sar.description =
          'taper usage') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
          IN TYPEOF(sar))) ) | (((
          'DIMENSIONAL_INSPECTION_SCHEMA.TAPER' IN TYPEOF(fcr.
          relating_shape_aspect)) AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.POCKET' IN TYPEOF(fcr.
          related_shape_aspect.of_shape.definition))) AND (fcr.
          related_shape_aspect.description IN ['angle taper',
          'directed taper'])) )) = 1)) )) <= 1)) )) = 0;
wr12: (NOT (SELF\characterized_object.description =
          'circular cutout')) OR (SIZEOF(QUERY ( pds <*
          QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (
          'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
          sa_occ.description = 'enclosed boundary occurrence') AND (
          SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATED_SHAPE_ASPECT') | ((sar.description =
          'profile usage') AND
         ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
          IN TYPEOF(sar))) ) | (
          'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
          TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
wr13: (NOT (SELF\characterized_object.description IN [
          'circular cutout','complex cutout'])) OR (SIZEOF(
          QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (
          'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
          sa_occ.description = 'bottom condition occurrence') AND (
          SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATED_SHAPE_ASPECT') | ((sar.description =
          'pocket bottom usage') AND
      ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
          IN TYPEOF(sar))) ) | (((
          'DIMENSIONAL_INSPECTION_SCHEMA.POCKET_BOTTOM' IN TYPEOF(sdr
          .relating_shape_aspect)) AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.POCKET' IN TYPEOF(sdr.
          related_shape_aspect.of_shape.definition))) AND (sdr.
          relating_shape_aspect.description = 'through')) )) = 1)) ))
```

```
              = 1)) )) = 0);
      wr14: (NOT (SELF\characterized_object.description = 'recess')) OR (
              SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'bottom condition occurrence') AND (
              SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'pocket bottom usage') AND
          ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.POCKET_BOTTOM' IN TYPEOF(sdr
              .relating_shape_aspect)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.POCKET' IN TYPEOF(sdr.
              related_shape_aspect.of_shape.definition))) AND (sdr.
             relating_shape_aspect.description IN ['planar','complex'])) ))
              = 1)) )) = 1)) )) = 0);
      wr15: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
              SELF) |
        (('DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
              IN TYPEOF(pdr.used_representation)) AND (pdr.
              used_representation.name = 'maximum feature limit')) )) >=
              0;
      wr16: (NOT (SELF\characterized_object.description IN [
              'closed rectangular','open rectangular','complex','recess']))
              OR (SIZEOF(QUERY ( pds <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              ('DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
              TYPEOF(csa)) AND (csa.name = 'uncut area')) AND (SIZEOF(
              QUERY ( sar <* csa.component_relationships |
         (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) AND (SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.BOSS',
              'DIMENSIONAL_INSPECTION_SCHEMA.PROTRUSION'] * TYPEOF(sar.
              related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1);
  END_ENTITY; -- pocket

  ENTITY pocket_bottom
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1 : 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
              IN TYPEOF(SELF.of_shape.definition);
      wr2 : SELF.description IN ['planar','complex','through'];
      wr3 : (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
```



```
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
            IN TYPEOF(pdr.used_representation)) AND (pdr.
            used_representation.name = 'floor normal')) )) = 1)) )) = 0);
    wr4 : (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
            USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         (('DIMENSIONAL_INSPECTION_SCHEMA.LOCATION_SHAPE_REPRESENTATION'
            IN TYPEOF(pdr.used_representation)) AND (pdr.
            used_representation.name = 'floor location')) )) = 1)) )) =
             0);
    wr5 : (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
            USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
            IN TYPEOF(pdr.used_representation)) AND (pdr.
            used_representation.name = 'floor face')) )) = 1)) )) = 0);
    wr6 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) = 1)) )) = 0);
    wr7 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) <= 1)) )) = 0)) )) = 0);
    wr8 : (NOT (SELF.description = 'through')) OR (SIZEOF(QUERY ( pd <*
            USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) = 0)) )) = 0);
    wr9 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
```



```
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
      ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                 TYPEOF(it)) = 2) AND (it.name = 'radius')) )) <= 1)) )) =
                0)) )) = 0);
      wr10: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | ((sar.description =
                'pocket bottom usage') AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.
                description = 'bottom condition occurrence') AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.POCKET' IN TYPEOF(fcr.
                related_shape_aspect.of_shape.definition))) AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.POCKET_BOTTOM' IN TYPEOF(fcr
                .relating_shape_aspect))) )) >= 1;
      wr11: (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
                QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | ((sar.description =
                'pocket bottom usage') AND
            ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | ((fcr.related_shape_aspect.description
                = 'bottom condition occurrence') AND (fcr.
                related_shape_aspect.name IN ['pocket depth start',
                'pocket depth end'])) )) = 0);
  END_ENTITY; -- pocket_bottom

  ENTITY point
    SUPERTYPE OF (cartesian_point)
    SUBTYPE OF (geometric_representation_item);
  END_ENTITY; -- point

  ENTITY poly_loop
    SUBTYPE OF (loop, geometric_representation_item);
      polygon : LIST [3:?] OF UNIQUE cartesian_point;
  END_ENTITY; -- poly_loop

  ENTITY polyline
    SUBTYPE OF (bounded_curve);
      points : LIST [2:?] OF cartesian_point;
  END_ENTITY; -- polyline
ENTITY position_tolerance
SUBTYPE OF (geometric_tolerance);
WHERE
WR1: ( NOT ('DIMENSIONAL_INSPECTION_SCHEMA.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF))) OR
(SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
END_ENTITY; -- position_tolerance

  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,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  END_ENTITY; -- product_definition

  ENTITY product_definition_context
    SUBTYPE OF (application_context_element);
      life_cycle_stage : label;
  END_ENTITY; -- product_definition_context

  ENTITY product_definition_formation;
      id          : identifier;
      description : OPTIONAL text;
      of_product  : product;
    UNIQUE
      url1 : id, of_product;
  END_ENTITY; -- product_definition_formation

  ENTITY product_definition_shape
    SUBTYPE OF (property_definition);
    UNIQUE
      url1 : definition;
    WHERE
      wr1:
SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.CHARACTERIZED_PRODUCT_DEFINITION',
               'DIMENSIONAL_INSPECTION_SCHEMA.CHARACTERIZED_OBJECT'] *
               TYPEOF(SELF\property_definition.definition)) > 0;
  END_ENTITY; -- product_definition_shape

  ENTITY profile_floor
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1 : 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
                IN TYPEOF(SELF.of_shape.definition);
      wr2 : SELF.description IN ['planar','complex','through'];
```

```
wr3 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) )) = 1)) )) = 0);
wr4 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | ((NOT (SIZEOF(impl_rep.
          used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
          used_representation.items) <= 2)) )) = 0)) )) = 0);
wr5 : (NOT (SELF.description = 'through')) OR (SIZEOF(QUERY ( pd <*
          USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) )) = 0)) )) = 0);
wr6 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
          = 0);
wr7 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATING_SHAPE_ASPECT') | ((sar.description =
          'profile floor usage') AND
    ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
          IN TYPEOF(sar))) ) | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.OUTSIDE_PROFILE' IN TYPEOF(
          fcr.related_shape_aspect.of_shape.definition)) AND (
          'DIMENSIONAL_INSPECTION_SCHEMA.PROFILE_FLOOR' IN TYPEOF(fcr
          .relating_shape_aspect))) )) >= 1;
wr8 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

```
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
        ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (it.name =
                'shape profile floor orientation')) AND (it.description IN
                ['shape profile start','shape profile end'])) )) = 1)) )) =
                 0)) )) = 0);
      wr9 : (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
                USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'floor')) )) = 1)) )) = 1);
      wr10: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
                USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
             (('DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'floor')) )) = 1)) )) = 1);
  END_ENTITY; -- profile_floor

  ENTITY projected_zone_definition
    SUBTYPE OF (tolerance_zone_definition);
      projection_end   : shape_aspect;
      projected_length : measure_with_unit;
    WHERE
      wr1: projected_length.value_component > 0;
  END_ENTITY; -- projected_zone_definition

  ENTITY property_definition;
      name        : label;
      description : OPTIONAL text;
      definition  : characterized_definition;
    DERIVE
      id : identifier := get_id_value(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
  END_ENTITY; -- property_definition

  ENTITY property_definition_relationship;
      name                         : label;
      description                  : text;
      relating_property_definition : property_definition;
      related_property_definition  : property_definition;
```

```
END_ENTITY; -- property_definition_relationship

ENTITY property_definition_representation;
    definition          : represented_definition;
    used_representation : representation;
  DERIVE
    description : text := get_description_value(SELF);
    name        : label := get_name_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    wr2: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- property_definition_representation

ENTITY protrusion
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(impl_rep.
             used_representation.items) = 1)) )) = 0)) )) = 0;
    wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
             IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
             'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
             sa_occ.description = 'shape volume occurrence') AND (SIZEOF(
             QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
             'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
              'RELATED_SHAPE_ASPECT') | ((sar.description =
             'volume shape usage') AND (
             'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
             IN TYPEOF(sar))) ) | (sdr.relating_shape_aspect.description
             = 'volume shape') )) = 1)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
             SELF) | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) AND (pdr.
             used_representation.name = 'maximum feature limit')) )) >= 0;
END_ENTITY; -- protrusion

ENTITY qualified_representation_item
  SUBTYPE OF (representation_item);
    qualifiers : SET [1:?] OF value_qualifier;
  WHERE
    wr1: SIZEOF(QUERY ( temp <* qualifiers | (
             'DIMENSIONAL_INSPECTION_SCHEMA.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: 'DIMENSIONAL_INSPECTION_SCHEMA.RATIO_UNIT' IN TYPEOF(SELF\
             measure_with_unit.unit_component);
END_ENTITY; -- ratio_measure_with_unit

ENTITY ratio_unit

  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF
             \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
             named_unit.dimensions.time_exponent = 0)) AND (SELF\
             named_unit.dimensions.electric_current_exponent = 0)) AND (
             SELF\named_unit.dimensions.
             thermodynamic_temperature_exponent = 0)) AND (SELF\
             named_unit.dimensions.amount_of_substance_exponent = 0)) AND
              (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- ratio_unit

ENTITY rational_b_spline_curve
  SUBTYPE OF (b_spline_curve);
    weights_data : LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:upper_index_on_control_points] OF REAL :=
                 list_to_array(weights_data,0,
                 upper_index_on_control_points);
  WHERE
    wr1: SIZEOF(weights_data) = SIZEOF(SELF\b_spline_curve.
             control_points_list);
    wr2: curve_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_curve

ENTITY rational_b_spline_surface
  SUBTYPE OF (b_spline_surface);
    weights_data : LIST [2:?] OF LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF REAL :=
                 make_array_of_array(weights_data,0,u_upper,0,v_upper);
  WHERE
    wr1: (SIZEOF(weights_data) = SIZEOF(SELF\b_spline_surface.
```

```
              control_points_list)) AND (SIZEOF(weights_data[1]) = SIZEOF(
              SELF\b_spline_surface.control_points_list[1]));
      wr2: surface_weights_positive(SELF);
  END_ENTITY; -- rational_b_spline_surface

  ENTITY rectangular_closed_profile
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
               TYPEOF(SELF.of_shape.definition);
      wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT ((SIZEOF(impl_rep.
              used_representation.items) >= 3) AND (SIZEOF(impl_rep.
              used_representation.items) <= 4))) )) = 0)) )) = 0;
      wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
              USEDIN(pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
              IN TYPEOF(pdr.used_representation)) AND (SIZEOF(
              QUERY ( srwp_i <* pdr.used_representation.items | ((((srwp_i
              .name = 'orientation') OR (srwp_i.name = 'length')) OR (
              srwp_i.name = 'width')) OR (srwp_i.name = 'corner radius')) ))
              = SIZEOF(pdr.used_representation.items))) )) = 1) )) = 1;
      wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
               (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
      wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
```



```
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
               = 0;
    wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1)) )) = 0)) ))
               = 0;
    wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
               'DIMENSIONAL_INSPECTION_SCHEMA.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,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                 TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') |
         ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar)) ) | (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
                related_shape_aspect)) )) = 1) )) <= 5)) )) = 0;
    wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
        (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
            IN TYPEOF(pdr.used_representation)) AND (pdr.
          used_representation.name = 'row layout direction')) )) = 1)) ))
            = 0;
  wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
            IN TYPEOF(pdr.used_representation)) AND (pdr.
            used_representation.name = 'column layout direction')) )) =
             1)) )) = 0;
  wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
            'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) = 1)) )) = 0;
  wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 5)) )) = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
            measure_with_unit.value_component))) AND (it.name =
            'number of rows')) )) = 1)) )) = 0)) )) = 0;
  wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
       ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
```



```
              measure_with_unit.value_component))) AND (it.name =
              'number of columns')) )) = 1)) )) = 0)) )) = 0;
    wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'row spacing')) )) = 1)) ))
              = 0)) )) = 0;
    wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'column spacing')) )) = 1)) ))
              = 0)) )) = 0;
    wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.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,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```



```
              used_representation)) ) | (NOT (SIZEOF(impl_rep.
              used_representation.items) = 1)) )) = 0)) )) = 0;
    wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'shape volume occurrence') AND (SIZEOF(
              QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | ((sar.description =
              'volume shape usage') AND (('DIMENSIONAL_INSPECTION_SCHEMA.'
              + '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,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
    wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATING_SHAPE_ASPECT') |
           ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) ) | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.REPLICATE_FEATURE',
              'DIMENSIONAL_INSPECTION_SCHEMA.INSTANCED_FEATURE'] * TYPEOF(
              fcr.related_shape_aspect)) >= 1) AND (fcr.name =
              'pattern basis')) )) = 1;
    wr3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATING_SHAPE_ASPECT') | (NOT
          ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) )) + SIZEOF(QUERY ( sar <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | (NOT
           ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ))) = 0;
END_ENTITY; -- replicate_feature

ENTITY representation;
    name             : label;
    items            : SET [1:?] OF representation_item;
    context_of_items : representation_context;
  DERIVE
    id          : identifier := get_id_value(SELF);
```



```
      description : text := get_description_value(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
      wr2: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  END_ENTITY; -- representation

  ENTITY representation_context;
      context_identifier : identifier;
      context_type       : text;
    INVERSE
      representations_in_context : SET [1:?] OF representation FOR
                                       context_of_items;
  END_ENTITY; -- representation_context

  ENTITY representation_item;
      name : label;
    WHERE
      wr1: SIZEOF(using_representations(SELF)) > 0;
  END_ENTITY; -- representation_item

  ENTITY representation_map;
      mapping_origin        : representation_item;
      mapped_representation : representation;
    INVERSE
      map_usage : SET [1:?] OF mapped_item FOR mapping_source;
    WHERE
      wr1: item_in_context(SELF.mapping_origin,SELF.mapped_representation.
               context_of_items);
  END_ENTITY; -- representation_map

ENTITY representation_relationship;
  name        : label;
  description : OPTIONAL text;
  rep_1       : representation;
  rep_2       : representation;
END_ENTITY;


  ENTITY resource_property;
      name        : label;
      description : text;
      resource    : characterized_resource_definition;
  END_ENTITY; -- resource_property

  ENTITY resource_requirement_type;
      name        : label;
      description : text;
  END_ENTITY; -- resource_requirement_type

  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,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(impl_rep.
         used_representation.items) = 2)) )) = 0)) )) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | ((SIZEOF([
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
         'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
         = 0;
wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
         SELF) |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
         IN TYPEOF(pdr.used_representation)) AND (pdr.
         used_representation.name = 'removal direction')) )) = 1;
wr5: (NOT (SELF\characterized_object.description = 'open profile'))
         OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
         IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
         'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
         sa_occ.description = 'outer edge shape occurrence') AND (
         SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
         'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
          'RELATED_SHAPE_ASPECT') | ((sar.description =
         'profile usage') AND (
         'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
         IN TYPEOF(sar))) ) | ((
         'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_PATH_PROFILE' IN TYPEOF(
         sdr.relating_shape_aspect)) AND (sdr.relating_shape_aspect.
         description = 'outer edge shape')) )) = 1)) )) = 1)) )) = 0);
wr6: (NOT (SELF\characterized_object.description = 'flat')) OR (
         SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
          | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
         IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
         'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
         sa_occ.description = 'flat edge shape occurrence') AND (
         SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
         'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
          'RELATED_SHAPE_ASPECT') | ((sar.description =
         'profile usage') AND (
         'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
         IN TYPEOF(sar))) ) | ((
```



```
            'DIMENSIONAL_INSPECTION_SCHEMA.LINEAR_PROFILE' IN TYPEOF(sdr
            .relating_shape_aspect)) AND (sdr.relating_shape_aspect.
            description = 'flat edge shape')) )) = 1)) )) = 1)) )) = 0);
    wr7: (NOT (SELF\characterized_object.description = 'round')) OR (
            SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'rounded edge shape occurrence') AND (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
             'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
            TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
            relating_shape_aspect.description = 'rounded edge shape')) ))
            = 1)) )) = 1)) )) = 0);
    wr8: (NOT (SELF\characterized_object.description = 'groove')) OR (
            SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'sweep occurrence') AND (SIZEOF(
            QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
             'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.TEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(
            sdr.relating_shape_aspect)) = 1) AND (sdr.
            relating_shape_aspect.description = 'sweep')) )) = 1)) )) =
            1)) )) = 0);
    wr9: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
            SELF) | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.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,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
```

```
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'rib top condition occurrence') AND (
              SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | ((sar.description =
              'rib top usage') AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.RIB_TOP_FLOOR' IN TYPEOF(sdr.
              relating_shape_aspect)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.RIB_TOP' IN TYPEOF(sdr.
            related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
              = 0;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
              IN TYPEOF(pdr.used_representation)) AND (pdr.
              used_representation.name = 'removal direction')) )) = 1)) ))
              = 1;
    wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
              SELF) | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.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: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
               TYPEOF(SELF.of_shape.definition);
    wr2: SELF.description IN ['planar','complex'];
    wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATING_SHAPE_ASPECT') | ((sar.description =
              'rib top usage') AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description
              = 'rib top condition occurrence') AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.RIB_TOP' IN TYPEOF(fcr.
              related_shape_aspect.of_shape.definition))) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.RIB_TOP_FLOOR' IN TYPEOF(fcr.
              relating_shape_aspect))) )) >= 1;
    wr4: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
               TYPEOF(pdr.used_representation)) AND (pdr.
              used_representation.name = 'rib top face')) )) = 1)) )) = 0);
    wr5: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
```



```
              USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
              IN TYPEOF(pdr.used_representation)) AND (pdr.
              used_representation.name = 'rib top face')) )) = 1)) )) = 0);
      wr6: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pds <*
              QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'boundary occurrence') AND (SIZEOF(
              QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | ((sar.description =
              'profile usage') AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | ((SIZEOF([
              'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE',
              'DIMENSIONAL_INSPECTION_SCHEMA.NGON_CLOSED_PROFILE',
              'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
              'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_PATH_PROFILE'] *
              TYPEOF(sdr.relating_shape_aspect)) = 1) AND (sdr.
           relating_shape_aspect.description = 'rib top floor boundary')) ))
              = 1)) )) = 1)) )) = 0);
  END_ENTITY; -- rib_top_floor

  ENTITY role_association;
      role           : object_role;
      item_with_role : role_select;
  END_ENTITY; -- role_association

  ENTITY round_hole
    SUBTYPE OF (feature_definition);
    WHERE
      wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'diameter occurrence') AND (SIZEOF(
              QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | ((sar.description =
              'profile usage') AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
              TYPEOF(sdr.relating_shape_aspect)) AND (sdr.name =
              'diameter')) )) = 1)) )) = 1)) )) = 0;
      wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
```

```
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'hole depth occurrence') AND (SIZEOF(
              QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | ((sar.description =
              'path feature component usage') AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT' IN
              TYPEOF(sdr.relating_shape_aspect)) AND (sdr.name =
              'hole depth')) AND (sdr.relating_shape_aspect.description =
              'linear')) )) = 1)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'bottom condition occurrence') AND (
              SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | ((sar.description =
              'hole bottom usage') AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.HOLE_BOTTOM' IN TYPEOF(fcr.
              relating_shape_aspect)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.ROUND_HOLE' IN TYPEOF(fcr.
            related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
              = 0;
      wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'change in diameter occurrence') AND (
              SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
              'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
               'RELATED_SHAPE_ASPECT') | ((sar.description = 'taper usage')
              AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (('DIMENSIONAL_INSPECTION_SCHEMA.TAPER'
              IN TYPEOF(fcr.relating_shape_aspect)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.ROUND_HOLE' IN TYPEOF(fcr.
           related_shape_aspect.of_shape.definition))) )) = 1)) )) <= 1)) ))
              = 0;
      wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
              SELF) |
        ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
              IN TYPEOF(pdr.used_representation)) )) = 1;
  END_ENTITY; -- round_hole

  ENTITY rounded_end
    SUBTYPE OF (feature_definition);
    WHERE
      wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
```



```
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 1)) )) = 0)) )) = 0;
    wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'partial circular boundary occurrence')
            AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
             'RELATED_SHAPE_ASPECT') | ((sar.description =
            'profile usage') AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
            TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'course of travel occurrence') AND (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
             'RELATED_SHAPE_ASPECT') | ((sar.description =
            'path feature component usage') AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT' IN
            TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
            relating_shape_aspect.description = 'linear')) )) = 1)) )) =
             1)) )) = 0;
    wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
            SELF) | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.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: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
             TYPEOF(SELF.of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
```

```
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | ((NOT (SIZEOF(impl_rep.
            used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
            used_representation.items) <= 2)) )) = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
            used_representation.items | ((srwp_i.name = 'orientation')
            OR (srwp_i.name = 'depth')) )) = SIZEOF(pdr.
            used_representation.items))) )) = 1) )) = 1;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
            = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
            IN TYPEOF(pdr.used_representation)) AND (pdr.
            used_representation.name = 'profile limit')) )) <= 1)) )) =
            0;
  wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
```



```
            TYPEOF(it)) = 2) AND (it.name = 'depth')) )) <= 1)) )) = 0)) ))
             = 0;
END_ENTITY; -- rounded_u_profile

ENTITY roundness_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: NOT (('DIMENSIONAL_INSPECTION_SCHEMA.' +
             'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
END_ENTITY; -- roundness_tolerance

ENTITY runout_zone_definition
  SUBTYPE OF (tolerance_zone_definition);
    orientation : runout_zone_orientation;
END_ENTITY; -- runout_zone_definition

ENTITY runout_zone_orientation;
    angle : measure_with_unit;
END_ENTITY; -- runout_zone_orientation

ENTITY runout_zone_orientation_reference_direction
  SUBTYPE OF (runout_zone_orientation);
    orientation_defining_relationship : shape_aspect_relationship;
END_ENTITY; -- runout_zone_orientation_reference_direction

ENTITY 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,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- security_classification_assignment

ENTITY security_classification_level;
    name : label;
END_ENTITY; -- security_classification_level

ENTITY serial_numbered_effectivity
  SUBTYPE OF (effectivity);
    effectivity_start_id : identifier;
    effectivity_end_id   : OPTIONAL identifier;
END_ENTITY; -- serial_numbered_effectivity

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,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
  END_ENTITY; -- shape_aspect

  ENTITY shape_aspect_deriving_relationship
    SUBTYPE OF (shape_aspect_relationship);
    WHERE
      wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.DERIVED_SHAPE_ASPECT' IN TYPEOF(
               SELF\shape_aspect_relationship.relating_shape_aspect);
  END_ENTITY; -- shape_aspect_deriving_relationship

  ENTITY shape_aspect_relationship;
      name                   : label;
      description            : OPTIONAL text;
      relating_shape_aspect : shape_aspect;
      related_shape_aspect  : shape_aspect;
    DERIVE
      id : identifier := get_id_value(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
  END_ENTITY; -- shape_aspect_relationship

  ENTITY shape_defining_relationship
    SUBTYPE OF (shape_aspect_relationship);
  END_ENTITY; -- shape_defining_relationship

  ENTITY shape_definition_representation
    SUBTYPE OF (property_definition_representation);
    WHERE
      wr1: ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(SELF.definition)) OR (
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINITION' IN TYPEOF(
               SELF.definition.definition));
      wr2: 'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION' IN TYPEOF(
               SELF.used_representation);
  END_ENTITY; -- shape_definition_representation

  ENTITY shape_dimension_representation
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: SIZEOF(QUERY ( temp <* SELF.items | (NOT (
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
               IN TYPEOF(temp))) )) = 0;
      wr2: SIZEOF(SELF.items) <= 2;
      wr3: SIZEOF(QUERY ( pos_mri <* QUERY ( real_mri <* SELF.items | (
               'REAL'  IN  TYPEOF(real_mri\measure_with_unit.value_component))
)
                | (NOT (pos_mri\measure_with_unit.value_component > 0)) ))
               = 0;
  END_ENTITY; -- shape_dimension_representation

  ENTITY shape_representation
```



```
    SUBTYPE OF (representation);
  END_ENTITY; -- shape_representation

  ENTITY shape_representation_with_parameters
    SUBTYPE OF (shape_representation);
    WHERE
      wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT',
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',

'DIMENSIONAL_INSPECTION_SCHEMA.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,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(impl_rep.
               used_representation.items) = 1)) )) = 0)) )) = 0;
      wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'swept shape occurrence') AND (SIZEOF(
               QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND (
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_U_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.TEE_PROFILE',
               'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(
               sdr.relating_shape_aspect)) = 1) )) = 1)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

```
                | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'course of travel occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATED_SHAPE_ASPECT') | ((sar.description =
               'path feature component usage') AND ((sar.name =
               'course of travel') AND (
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar)))) ) | (
               'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT' IN
               TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
      wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'end condition occurrence') AND (
               SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATED_SHAPE_ASPECT') | (((sar.description =
               'slot end usage') AND (sar.name IN ['course of travel start',
               'course of travel end'])) AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (((
               'DIMENSIONAL_INSPECTION_SCHEMA.SLOT_END' IN TYPEOF(fcr.
               relating_shape_aspect)) AND (fcr.relating_shape_aspect.
               description IN ['open','radiused','flat','woodruff'])) AND (
               'DIMENSIONAL_INSPECTION_SCHEMA.SLOT' IN TYPEOF(fcr.
             related_shape_aspect.of_shape.definition))) )) = 1)) )) = 2)) ))
               = 0;
      wr5: (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'end condition occurrence') AND (
               SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP' +
               '.RELATED_SHAPE_ASPECT') | (((sar.description =
               'slot end usage') AND (sar.name IN ['course of travel start',
               'course of travel end'])) AND
            ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (((
               'DIMENSIONAL_INSPECTION_SCHEMA.SLOT_END' IN TYPEOF(fcr.
               relating_shape_aspect)) AND (fcr.relating_shape_aspect.
               description IN ['loop'])) AND (
               'DIMENSIONAL_INSPECTION_SCHEMA.SLOT' IN TYPEOF(fcr.
             related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
               = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'course of travel occurrence') AND (
```



```
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'path feature component usage') AND ((sar.name =
                'course of travel') AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar)))) ) | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT' IN
                TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
                relating_shape_aspect.description IN ['complex',
                'complete circular'])) )) = 1)) )) = 1)) )) = 0);
      wr6: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
                SELF) | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'maximum feature limit')) )) >= 0;
  END_ENTITY; -- slot

  ENTITY slot_end
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1 : 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
                IN TYPEOF(SELF.of_shape.definition);
      wr2 : SELF.description IN ['open','radiused','flat','woodruff',
                'loop'];
      wr3 : (NOT (SELF.description IN ['open','radiused','loop'])) OR (
                SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) )) = 0)) )) = 0);
      wr4 : (NOT (SELF.description IN ['flat','woodruff'])) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
      wr5 : (NOT (SELF.description IN ['flat'])) OR (SIZEOF(QUERY ( pd <*
                USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                impl_rep.used_representation.items) = 2)) )) = 0)) )) = 0);
      wr6 : (NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pd <*
                USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
    ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
            IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
            QUERY ( it <* impl_rep.used_representation.items | ((
            SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'first radius')) )) = 1)) ))
            = 0)) )) = 0);
  wr7 : (NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pd <*
            USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
   ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
            IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
            QUERY ( it <* impl_rep.used_representation.items | ((
            SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'second radius')) )) = 1)) ))
            = 0)) )) = 0);
  wr8 : (NOT (SELF.description = 'woodruff')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
            IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
            QUERY ( it <* impl_rep.used_representation.items | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
             TYPEOF(it)) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
            = 0);
  wr9 : (NOT (SELF.description IN ['woodruff'])) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
            IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
            impl_rep.used_representation.items) = 1)) )) = 0)) )) = 0);
  wr10: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATING_SHAPE_ASPECT') | (((sar.description =
            'slot end usage') AND (sar.name IN [
            'course of travel start','course of travel end'])) AND
       ('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.
            description = 'end condition occurrence') AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.SLOT' IN TYPEOF(fcr.
            related_shape_aspect.of_shape.definition))) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.SLOT_END' IN TYPEOF(fcr.
            relating_shape_aspect))) )) >= 1;
END_ENTITY; -- slot_end
```



```
ENTITY solid_angle_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF
             \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
             named_unit.dimensions.time_exponent = 0)) AND (SELF\
             named_unit.dimensions.electric_current_exponent = 0)) AND (
             SELF\named_unit.dimensions.
             thermodynamic_temperature_exponent = 0)) AND (SELF\
             named_unit.dimensions.amount_of_substance_exponent = 0)) AND
              (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- solid_angle_unit

ENTITY solid_model
  SUPERTYPE OF (manifold_solid_brep)
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- solid_model

ENTITY spherical_cap
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(impl_rep.
             used_representation.items) = 3)) )) = 0)) )) = 0;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
             = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
           'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(it)) = 2) AND (it.name = 'internal angle')) )) = 1)) ))
```

```
              = 0)) )) = 0;
END_ENTITY; -- spherical_cap

ENTITY spherical_surface
  SUBTYPE OF (elementary_surface);
    radius : positive_length_measure;
END_ENTITY; -- spherical_surface

ENTITY square_u_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
              IN TYPEOF(SELF.of_shape.definition);
    wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
    wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | ((NOT (SIZEOF(impl_rep.
              used_representation.items) >= 4)) AND (SIZEOF(impl_rep.
              used_representation.items) <= 7)) )) = 0)) )) = 0;
    wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
              used_representation.items | ((((((((srwp_i.name =
              'orientation') OR (srwp_i.name = 'width')) OR (srwp_i.name
              = 'first angle')) OR (srwp_i.name = 'second angle')) OR (
              srwp_i.name = 'first radius')) OR (srwp_i.name =
              'second radius')) OR (srwp_i.name = 'profile limit')) OR (
              srwp_i.name = 'depth')) )) = SIZEOF(pdr.used_representation
              .items))) )) = 1) )) = 1;
    wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it))
```



```
          AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
         TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
          = 0;
    wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'first radius')) )) <= 1)) ))
          = 0)) )) = 0;
    wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
         TYPEOF(it)) = 2) AND (it.name = 'second radius')) )) <= 1)) ))
          = 0)) )) = 0;
    wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
          = 0)) )) = 0;
    wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
```

```
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                 'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
                = 0)) )) = 0;
      wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
                'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
             (('DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'profile limit')) )) <= 1)) )) =
                 0;
      wr12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'depth')) )) <= 1)) )) = 0)) ))
                = 0;
  END_ENTITY; -- square_u_profile

  ENTITY standard_uncertainty
    SUBTYPE OF (uncertainty_qualifier);
      uncertainty_value : REAL;
  END_ENTITY; -- standard_uncertainty

  ENTITY step
    SUBTYPE OF (feature_definition);
    WHERE
      wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(impl_rep.
               used_representation.items) = 1)) )) = 0)) )) = 0;
      wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
```



```
                sa_occ.description = 'course of travel occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'path feature component usage') AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT' IN
                TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
                relating_shape_aspect.description = 'linear')) )) = 1)) )) =
                 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
              'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
                sa_occ.description = 'removal boundary occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE' IN TYPEOF(sdr.
                relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
      wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
                SELF) | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'maximum feature limit')) )) >= 0;
      wr5: SIZEOF(QUERY ( pds <* USEDIN(SELF,
             'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (((
                'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN
                TYPEOF(csa)) AND (csa.name = 'uncut area')) AND (SIZEOF(
                QUERY ( sar <* csa.component_relationships |
            (('DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar)) AND (SIZEOF([
                'DIMENSIONAL_INSPECTION_SCHEMA.BOSS',
                'DIMENSIONAL_INSPECTION_SCHEMA.PROTRUSION'] * TYPEOF(sar.
                related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1;
  END_ENTITY; -- step

  ENTITY straightness_tolerance
    SUBTYPE OF (geometric_tolerance);
    WHERE
      wr1: NOT (('DIMENSIONAL_INSPECTION_SCHEMA.' +
                '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: ('DIMENSIONAL_INSPECTION_SCHEMA.PCURVE' IN TYPEOF(
             associated_geometry[1])) OR (master_representation <>
             pcurve_s1);
    wr3: ('DIMENSIONAL_INSPECTION_SCHEMA.PCURVE' IN TYPEOF(
             associated_geometry[2])) OR (master_representation <>
             pcurve_s2);
    wr4: NOT ('DIMENSIONAL_INSPECTION_SCHEMA.PCURVE' IN TYPEOF(curve_3d));
END_ENTITY; -- surface_curve

ENTITY surface_of_linear_extrusion
  SUBTYPE OF (swept_surface);
    extrusion_axis : vector;
END_ENTITY; -- surface_of_linear_extrusion

ENTITY surface_of_revolution
  SUBTYPE OF (swept_surface);
    axis_position : axis1_placement;
  DERIVE
    axis_line : line := (dummy_gri || curve()) || line(axis_position.
                   location,dummy_gri || vector(axis_position.z,1));
END_ENTITY; -- surface_of_revolution

ENTITY surface_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: (NOT (('DIMENSIONAL_INSPECTION_SCHEMA.' +
             'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF)))
             OR (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
             datum_system) <= 3);
END_ENTITY; -- surface_profile_tolerance

ENTITY swept_surface
  SUPERTYPE OF (ONEOF (surface_of_linear_extrusion,surface_of_revolution))
  SUBTYPE OF (surface);
    swept_curve : curve;
END_ENTITY; -- swept_surface

ENTITY symmetric_shape_aspect
  SUBTYPE OF (shape_aspect);
  INVERSE
    basis_relationships : SET [1:?] OF shape_aspect_relationship FOR
                          relating_shape_aspect;
  WHERE
    wr1: SIZEOF(QUERY ( x <* SELF.basis_relationships | (
             'DIMENSIONAL_INSPECTION_SCHEMA.CENTRE_OF_SYMMETRY' IN
```



```
            TYPEOF(x.related_shape_aspect)) )) >= 1;
END_ENTITY; -- symmetric_shape_aspect

ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
            datum_system) <= 3;
END_ENTITY; -- symmetry_tolerance

ENTITY tangent
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY; -- tangent

ENTITY taper
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
             TYPEOF(SELF.of_shape.definition);
    wr2: SELF.description IN ['angle taper','diameter taper',
            'directed taper'];
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) = 1)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 1)) )) = 0)) )) = 0;
    wr5: (NOT (SELF.description = 'angle taper')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
         'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'taper angle')) )) = 1)) ))
            = 0)) )) = 0);
    wr6: (NOT (SELF.description = 'diameter taper')) OR (SIZEOF(
            QUERY ( pd <* USEDIN(SELF,
```

```
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'final diameter')) )) = 1)) ))
             = 0)) )) = 0);
    wr7: (NOT (SELF.description = 'directed taper')) OR (SIZEOF(
             QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(it)) = 2) AND (it.name = 'angle')) )) = 1)) )) = 0)) ))
             = 0);
    wr8: (NOT (SELF.description = 'directed taper')) OR (SIZEOF(
             QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) AND (pdr.
             used_representation.name = 'direction')) )) = 1)) )) = 0);
END_ENTITY; -- taper

ENTITY tee_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
              IN TYPEOF(SELF.of_shape.definition);
    wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
    wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
```



```
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | ((NOT (SIZEOF(impl_rep.
            used_representation.items) >= 9)) AND (SIZEOF(impl_rep.
            used_representation.items) <= 10)) )) = 0)) )) = 0;
  wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
            used_representation.items | (((((((((srwp_i.name =
            'orientation') OR (srwp_i.name = 'width')) OR (srwp_i.name
            = 'depth')) OR (srwp_i.name = 'cross bar width')) OR (
            srwp_i.name = 'cross bar depth')) OR (srwp_i.name =
            'first offset')) OR (srwp_i.name = 'second offset')) OR (
            srwp_i.name = 'first angle')) OR (srwp_i.name =
            'second angle')) OR (srwp_i.name = 'radius')) )) = SIZEOF(
            pdr.used_representation.items))) )) = 1) )) = 1;
  wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it))
            AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
            = 0;
  wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'depth')) )) = 1)) )) = 0)) ))
```

```
            = 0;
    wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'cross bar width')) )) = 1)) ))
             = 0)) )) = 0;
    wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'cross bar depth')) )) = 1)) ))
             = 0)) )) = 0;
    wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'first offset')) )) = 1)) ))
             = 0)) )) = 0;
    wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) = 1)) ))
             = 0)) )) = 0;
    wr12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
```



```
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
                = 0)) )) = 0;
      wr13: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
             'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
             'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
                = 0)) )) = 0;
      wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                 TYPEOF(it)) = 2) AND (it.name = 'radius')) )) <= 1)) )) =
                0)) )) = 0;
      wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
                'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               (('DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'profile limit')) )) <= 1)) )) =
                 0;
  END_ENTITY; -- tee_profile

  ENTITY thread
    SUBTYPE OF (feature_definition);
    WHERE
      wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
```

```
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) AND (8 <= SIZEOF(pdr.
                used_representation.items))) AND (SIZEOF(pdr.
                used_representation.items) <= 11)) )) = 1) )) = 1;
      wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'major diameter')) )) = 1)) ))
                = 0)) )) = 0;
      wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items                       |
((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
                 TYPEOF(it)) = 2) AND (it.name = 'minor diameter')) )) <= 1))
))
                = 0)) )) = 0;
      wr4          :          SIZEOF(QUERY          (          pd          <*
USEDIN(SELF,'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'pitch diameter')) )) <= 1)) ))
                = 0)) )) = 0;
      wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                'DIMENSIONAL_INSPECTION_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
                TYPEOF(it)) = 2) AND (it.name = 'number of threads')) )) =
```



```
          1)) )) = 0)) )) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
      (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND (it.name = 'fit class')) )) = 1)) )) = 0)) ))
          = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
    (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND (it.name = 'form')) )) = 1)) )) = 0)) ))
          = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
   ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND (it.name = 'hand')) AND (it.description
          IN ['left','right'])) )) = 1)) )) = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
    (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND (it.name = 'qualifier')) )) <= 1)) )) =
          0)) )) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

```
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
        ((('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'thread side')) AND ((it.
          description = 'internal') OR (it.description = 'external'))) ))
            = 1)) )) = 0)) )) = 0;
  wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'crest')) )) <= 1)) )) = 0)) ))
            = 0;
  wr12: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'partial area occurrence') AND (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'applied area usage') AND
           ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_AREA' IN TYPEOF(sdr.
            relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
  wr13: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (
            'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'applied shape') AND
         ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
            relating_shape_aspect)) )) = 1) )) = 1)) )) = 0;
  wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
```



```
              impl_rep.used_representation.items |
         (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
              IN TYPEOF(it)) AND (it.name = 'fit class 2')) )) <= 1)) ))
              = 0)) )) = 0;
    wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'nominal size')) )) <= 1)) ))
              = 0)) )) = 0;
END_ENTITY; -- thread

ENTITY time_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: 'DIMENSIONAL_INSPECTION_SCHEMA.TIME_UNIT' IN TYPEOF(SELF\
             measure_with_unit.unit_component);
END_ENTITY; -- time_measure_with_unit

ENTITY time_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF
             \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
             named_unit.dimensions.time_exponent = 1)) AND (SELF\
             named_unit.dimensions.electric_current_exponent = 0)) AND (
             SELF\named_unit.dimensions.
             thermodynamic_temperature_exponent = 0)) AND (SELF\
             named_unit.dimensions.amount_of_substance_exponent = 0)) AND
              (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- time_unit

ENTITY tolerance_value;
    lower_bound : measure_with_unit;
    upper_bound : measure_with_unit;
  WHERE
    wr1: upper_bound.value_component > lower_bound.value_component;
    wr2: upper_bound.unit_component = lower_bound.unit_component;
END_ENTITY; -- tolerance_value

ENTITY tolerance_zone
  SUBTYPE OF (shape_aspect);
    defining_tolerance : SET [1:?] OF geometric_tolerance;
    form               : tolerance_zone_form;
END_ENTITY; -- tolerance_zone

ENTITY tolerance_zone_definition
  SUPERTYPE OF (ONEOF (projected_zone_definition,runout_zone_definition));
    zone       : tolerance_zone;
```

```
    boundaries : SET [1:?] OF shape_aspect;
END_ENTITY; -- tolerance_zone_definition

ENTITY tolerance_zone_form;
    name : label;
END_ENTITY; -- tolerance_zone_form

ENTITY 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(['DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION',
             'DIMENSIONAL_INSPECTION_SCHEMA.COMPOUND_FEATURE'] * TYPEOF(
             SELF.of_shape.definition)) = 1;
    wr2: SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER',
             'DIMENSIONAL_INSPECTION_SCHEMA.EDGE_ROUND',
             'DIMENSIONAL_INSPECTION_SCHEMA.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,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
             'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) AND ((6 <= SIZEOF(pdr.
             used_representation.items)) AND (SIZEOF(pdr.
             used_representation.items) <= 9))) )) = 1) )) = 1;
    wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
```



```
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
      ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
            measure_with_unit.value_component))) AND (it.name =
            'number of teeth')) )) <= 1)) )) = 0)) )) = 0;
  wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'major diameter')) )) = 1)) ))
            = 0)) )) = 0;
  wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
  ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'nominal diameter')) )) =
            1)) )) = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.' +
            'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
            QUERY ( pdr <* USEDIN(pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
           'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'tooth depth')) )) <= 1)) ))
            = 0)) )) = 0;
  wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
```

```
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'root fillet')) )) <= 1)) ))
          = 0)) )) = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
   'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
        'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'diametral pitch')) )) = 1)) ))
          = 0)) )) = 0;
wr9 : (NOT (SELF\characterized_object.description IN ['diamond',
          'diagonal'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
         'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(it)) = 2) AND (it.name = 'helix angle')) )) = 1)) ))
          = 0)) )) = 0);
wr10: (NOT (SELF\characterized_object.description = 'diagonal')) OR
          (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
   (('DIMENSIONAL_INSPECTION_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND (it.name = 'helix hand')) )) = 1)) )) =
          0)) )) = 0);
wr11: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
           | (
          'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
          sa_occ.description = 'partial area occurrence') AND (
          SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
```



```
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'applied area usage') AND ((
                'DIMENSIONAL_INSPECTION_SCHEMA.' +
                'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
                'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_AREA' IN TYPEOF(sdr.
                relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
      wr12: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
                 TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'applied shape') AND
             ('DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
                relating_shape_aspect)) )) = 1) )) = 1)) )) = 0;
  END_ENTITY; -- turned_knurl

  ENTITY type_qualifier;
      name : label;
  END_ENTITY; -- type_qualifier

  ENTITY uncertainty_measure_with_unit
    SUBTYPE OF (measure_with_unit);
      name        : label;
      description : OPTIONAL text;
    WHERE
      wr1: valid_measure_value(SELF\measure_with_unit.value_component);
  END_ENTITY; -- uncertainty_measure_with_unit

  ENTITY uncertainty_qualifier
    SUPERTYPE OF (standard_uncertainty);
      measure_name : label;
      description  : text;
  END_ENTITY; -- uncertainty_qualifier

  ENTITY uniform_curve
    SUBTYPE OF (b_spline_curve);
  END_ENTITY; -- uniform_curve

  ENTITY uniform_surface
    SUBTYPE OF (b_spline_surface);
  END_ENTITY; -- uniform_surface

  ENTITY value_range
    SUBTYPE OF (compound_representation_item);
 WHERE

 wr1:  SIZEOF(QUERY (mri <* QUERY(sri <* SELF.item_element |
      ('DIMENSIONAL_INSPECTION_SCHEMA.SET_REPRESENTATION_ITEM'
       IN TYPEOF (sri))) |
      ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
```

```
      IN TYPEOF (mri))
      ))=2;

wr2:  SIZEOF(QUERY (mri <* QUERY(sri <* SELF.item_element |
     ('DIMENSIONAL_INSPECTION_SCHEMA.SET_REPRESENTATION_ITEM'
      IN TYPEOF (sri))) |
 (('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
  IN TYPEOF (mri)) AND
     (mri.name='lower limit'))
      ))=1;

wr3:  SIZEOF(QUERY (mri <* QUERY(sri <* SELF.item_element |
     ('DIMENSIONAL_INSPECTION_SCHEMA.SET_REPRESENTATION_ITEM'
      IN TYPEOF (sri))) |
 (('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
  IN TYPEOF (mri)) AND
     (mri.name='upper limit'))
      ))=1;

wr4: SIZEOF(QUERY(i1 <* SELF.item_element |
 ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
 IN TYPEOF (i1)) AND
    (SIZEOF (QUERY (i2 <* SELF.item_element |
 ('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
 IN TYPEOF (i2)) AND
    (i1 :<>: i2) AND
    (i1\measure_with_unit.unit_component :=:
      i2\measure_with_unit.unit_component)
    )) = 1))) = 2 ;
 END_ENTITY; -- value_range

 ENTITY 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 : 'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION'
              IN TYPEOF(SELF.of_shape.definition);
     wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'DIMENSIONAL_INSPECTION_SCHEMA.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
     wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
```



```
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | ((NOT (SIZEOF(impl_rep.
            used_representation.items) >= 3)) AND (SIZEOF(impl_rep.
            used_representation.items) <= 6)) )) = 0)) )) = 0;
    wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
            used_representation.items | (((((srwp_i.name =
            'orientation') OR (srwp_i.name = 'profile angle')) OR (
            srwp_i.name = 'tilt angle')) OR (srwp_i.name =
            'profile radius')) OR (srwp_i.name = 'first length')) OR (
            srwp_i.name = 'second length')) )) = SIZEOF(pdr.
            used_representation.items))) )) = 1) )) = 1;
    wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.PLACEMENT' IN TYPEOF(it))
            AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
         ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (
            'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
             TYPEOF(it\measure_with_unit.value_component))) AND (it.
            name = 'profile radius')) )) <= 1)) )) = 0)) )) = 0;
    wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
            'DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
```

```
            * TYPEOF(it)) = 2) AND (it.name = 'profile angle')) )) = 1)) ))
              = 0)) )) = 0;
    wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
     ((SIZEOF(['DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'tilt angle')) )) = 1)) ))
              = 0)) )) = 0;
    wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           (('DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
              IN TYPEOF(pdr.used_representation)) AND (pdr.
              used_representation.name = 'profile limit')) )) <= 1)) )) =
               0;
    wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
           ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
              IN TYPEOF(it)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
               TYPEOF(it\measure_with_unit.value_component))) AND (it.
              name = 'first length')) )) <= 1)) )) = 0)) )) = 0;
    wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'DIMENSIONAL_INSPECTION_SCHEMA.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
            ((('DIMENSIONAL_INSPECTION_SCHEMA.MEASURE_REPRESENTATION_ITEM'
              IN TYPEOF(it)) AND (
              'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
               TYPEOF(it\measure_with_unit.value_component))) AND (it.
              name = 'second length')) )) <= 1)) )) = 0)) )) = 0;
END_ENTITY; -- vee_profile

ENTITY 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

  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 =
             'dimensional inspection schema')) )) = 1)) )) = 0;

  END_RULE; -- application_context_requires_ap_definition

 RULE chamfer_offset_requires_faces FOR (chamfer_offset,
          property_definition_representation);
WHERE
   WR1: SIZEOF(QUERY ( co <* chamfer_offset |
         (( (co.description = 'first offset')) AND
         (SIZEOF(QUERY ( pd <* USEDIN(co,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
         (NOT  (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         (('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
         IN TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.name = 'first chamfer face')) )) = 1)) ))
         = 1)) ))=0;

    WR2:  SIZEOF(QUERY ( co <* chamfer_offset |
         (( (co.description = 'second offset')) AND
         (SIZEOF(QUERY ( pd <* USEDIN(co,
         'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
         (NOT  (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'DIMENSIONAL_INSPECTION_SCHEMA.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         (('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
         IN TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.name = 'second chamfer face')) )) = 1)) ))
         = 1)) ))=0;
END_RULE;

 RULE chamfer_requires_faces FOR (chamfer,
          property_definition_representation);
 WHERE
   wr1: SIZEOF(QUERY ( er <* chamfer |
```

```
        (SIZEOF (QUERY (pd <* USEDIN (er,
       'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
       'DIMENSIONAL_INSPECTION_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION'
       IN TYPEOF (pdr.used_representation)) AND
       (pdr.used_representation.name = 'chamfer face')))=1)))=1)))=0;

END_RULE;

  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_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 dimensional_measurement_part FOR (product_definition_formation,
           product_definition, product,
           applied_security_classification_assignment,
           applied_effectivity_assignment);

WHERE
  wr1: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
           SIZEOF(QUERY ( pd <* product_definition | ((pdf :=: pd.
        formation) AND (pd.frame_of_reference.name = 'part definition')) ))
           = 0)) )) = 1;
  wr2: SIZEOF(QUERY ( prod <* product | (NOT (SIZEOF(QUERY ( pdf <*
           product_definition_formation | (prod :=: pdf.of_product) )) >=
            1)) )) = 0;
  wr3: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
           SIZEOF(QUERY ( asca <*
           applied_security_classification_assignment | (pdf IN asca.
           items) )) <= 1)) )) = 0;
  wr4: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
           SIZEOF(QUERY ( aea <* applied_effectivity_assignment | ((pdf
           IN aea.items) AND (
           'DIMENSIONAL_INSPECTION_SCHEMA.SERIAL_NUMBERED_EFFECTIVITY' IN
            TYPEOF(aea.assigned_effectivity))) )) <= 1)) )) = 0;
  wr5: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
           SIZEOF(QUERY ( aea <* applied_effectivity_assignment | ((pdf
           IN aea.items) AND (
           'DIMENSIONAL_INSPECTION_SCHEMA.LOT_EFFECTIVITY' IN TYPEOF(aea.
           assigned_effectivity))) )) <= 1)) )) = 0;
```

```
    wr6: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
            SIZEOF(QUERY ( pd <* product_definition | ((pdf :=: pd.
            formation) AND (SIZEOF(QUERY ( prop_def <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | ('DIMENSIONAL_INSPECTION_SCHEMA.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(prop_def)) )) = 1)) )) = 0)) )) = 1;

  END_RULE; -- dimensional_measurement_part

  RULE edge_round_requires_faces FOR (edge_round,
            property_definition_representation);

  WHERE
    wr1:  SIZEOF(QUERY  (  er  <*  edge_round  |  (SIZEOF(QUERY  (  pd  <*
USEDIN(er,'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
            TYPEOF(pdr.used_representation)) AND (pdr.used_representation.
            name = 'edge round face')) )) <= 1)) )) = 1) )) = 0;
    wr2:  SIZEOF(QUERY  (  er  <*  edge_round  |  (SIZEOF(QUERY  (  pd  <*
USEDIN(er,'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
            TYPEOF(pdr.used_representation)) AND (pdr.used_representation.
            name = 'first face shape')) )) <= 1)) )) = 1) )) = 0;
    wr3:  SIZEOF(QUERY  (  er  <*  edge_round  |  (SIZEOF(QUERY  (  pd  <*
USEDIN(er,'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'DIMENSIONAL_INSPECTION_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
            TYPEOF(pdr.used_representation)) AND (pdr.used_representation.
            name = 'second face shape')) )) <= 1)) )) = 1) )) = 0;

  END_RULE; -- edge_round_requires_faces

  RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);

  WHERE
    wr1: SIZEOF(QUERY ( gt <* geometric_tolerance | (NOT (SIZEOF(TYPEOF(gt)
            * ['DIMENSIONAL_INSPECTION_SCHEMA.ANGULARITY_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_RUNOUT_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.CONCENTRICITY_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.CYLINDRICITY_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.FLATNESS_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.LINE_PROFILE_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.PARALLELISM_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.PERPENDICULARITY_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.POSITION_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDNESS_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.STRAIGHTNESS_TOLERANCE',
            'DIMENSIONAL_INSPECTION_SCHEMA.SURFACE_PROFILE_TOLERANCE',
```



```
             'DIMENSIONAL_INSPECTION_SCHEMA.SYMMETRY_TOLERANCE',
             'DIMENSIONAL_INSPECTION_SCHEMA.TOTAL_RUNOUT_TOLERANCE'])  <=  2))
))
             = 0;

  END_RULE; -- geometric_tolerance_subtype_exclusiveness

  RULE machining_feature_life_cycle FOR (instanced_feature);

  WHERE
    wr1: SIZEOF(QUERY ( mf <* instanced_feature | (NOT (mf.of_shape.
             definition.frame_of_reference.life_cycle_stage =
             'manufacturing planning')) )) = 0;

  END_RULE; -- machining_feature_life_cycle

  RULE product_definition_formation_requires_security_classification FOR (
             product_definition_formation,
             applied_security_classification_assignment);

  WHERE
    wr1: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
             SIZEOF(QUERY ( asca <*
             applied_security_classification_assignment | (pdf IN asca.
             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 representation_subtype_exclusiveness FOR (representation);

  WHERE
    wr1: SIZEOF(QUERY ( rep <* representation | (NOT (SIZEOF(TYPEOF(rep) *
              ['DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION',

'DIMENSIONAL_INSPECTION_SCHEMA.DEFINITIONAL_REPRESENTATION','DIMENSIONAL_INSP
ECTION_SCHEMA.DIRECTION_SHAPE_REPRESENTATION',
             'DIMENSIONAL_INSPECTION_SCHEMA.FACE_SHAPE_REPRESENTATION',
             'DIMENSIONAL_INSPECTION_SCHEMA.PLANAR_SHAPE_REPRESENTATION',
             'DIMENSIONAL_INSPECTION_SCHEMA.LOCATION_SHAPE_REPRESENTATION',
             'DIMENSIONAL_INSPECTION_SCHEMA.PATH_SHAPE_REPRESENTATION']) <=
              2)) )) = 0;

  END_RULE; -- representation_subtype_exclusiveness

  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,
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
           | ('DIMENSIONAL_INSPECTION_SCHEMA.DOCUMENT_FILE' IN TYPEOF(
          adr.assigned_document)) )) >= 1)) )) = 0) OR (SIZEOF(
          QUERY ( ex <* externally_defined_feature_definition | (NOT (
          SIZEOF(QUERY ( adr <* USEDIN(ex,
          'DIMENSIONAL_INSPECTION_SCHEMA.' +
          'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS') | (
          'DIMENSIONAL_INSPECTION_SCHEMA.DOCUMENT_FILE' IN TYPEOF(adr\
          document_usage_constraint_assignment.assigned_document_usage.
          source)) )) >= 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) * [
          'DIMENSIONAL_INSPECTION_SCHEMA.DIMENSIONAL_LOCATION',
         'DIMENSIONAL_INSPECTION_SCHEMA.GEOMETRIC_TOLERANCE_RELATIONSHIP',
          'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP',
          'DIMENSIONAL_INSPECTION_SCHEMA.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) * [
          'DIMENSIONAL_INSPECTION_SCHEMA.PATH_FEATURE_COMPONENT',
          'DIMENSIONAL_INSPECTION_SCHEMA.SLOT_END',
          'DIMENSIONAL_INSPECTION_SCHEMA.POCKET_BOTTOM',
          'DIMENSIONAL_INSPECTION_SCHEMA.PROFILE_FLOOR',
          'DIMENSIONAL_INSPECTION_SCHEMA.RIB_TOP_FLOOR',
          'DIMENSIONAL_INSPECTION_SCHEMA.BOSS_TOP',
          'DIMENSIONAL_INSPECTION_SCHEMA.HOLE_BOTTOM',
          'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_AREA',
          'DIMENSIONAL_INSPECTION_SCHEMA.TAPER',
          'DIMENSIONAL_INSPECTION_SCHEMA.CHAMFER_OFFSET',
          'DIMENSIONAL_INSPECTION_SCHEMA.CIRCULAR_CLOSED_PROFILE',
          'DIMENSIONAL_INSPECTION_SCHEMA.NGON_CLOSED_PROFILE',
```

```
            'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_PATH_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.SQUARE_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.LINEAR_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.TEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.VEE_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.ROUNDED_U_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_PATH_PROFILE',
            'DIMENSIONAL_INSPECTION_SCHEMA.INSTANCED_FEATURE',
            'DIMENSIONAL_INSPECTION_SCHEMA.REPLICATE_FEATURE',
            'DIMENSIONAL_INSPECTION_SCHEMA.TRANSITION_FEATURE',
            'DIMENSIONAL_INSPECTION_SCHEMA.DATUM',
            'DIMENSIONAL_INSPECTION_SCHEMA.DATUM_FEATURE',
            'DIMENSIONAL_INSPECTION_SCHEMA.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) * ['DIMENSIONAL_INSPECTION_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION',
      'DIMENSIONAL_INSPECTION_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS',
      'DIMENSIONAL_INSPECTION_SCHEMA.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([
           'DIMENSIONAL_INSPECTION_SCHEMA.DM_FEATURE_DEFINITION',
           'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_DEFINITION',
           'DIMENSIONAL_INSPECTION_SCHEMA.FEATURE_COMPONENT_DEFINITION']
           * TYPEOF(csa)) = 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 | (
        'DIMENSIONAL_INSPECTION_SCHEMA.MAPPED_ITEM' IN TYPEOF(z)) );
    IF SIZEOF(x) > 0 THEN
      REPEAT i := 1 TO HIINDEX(x) BY 1;
        IF x[i]\mapped_item.mapping_source.mapped_representation IN
            parent_set THEN
          RETURN(FALSE);
        END_IF;
        IF NOT acyclic_mapped_representation(parent_set + x[i]\mapped_item
            .mapping_source.mapped_representation,x[i]\mapped_item.
            mapping_source.mapped_representation.items) THEN
          RETURN(FALSE);
        END_IF;
      END_REPEAT;
    END_IF;
    x := children_set - x;
    IF SIZEOF(x) > 0 THEN
      REPEAT i := 1 TO HIINDEX(x) BY 1;
        y := QUERY ( z <* bag_to_set(USEDIN(x[i],'')) | (
            'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION_ITEM'  IN  TYPEOF(z))
);
        IF NOT acyclic_mapped_representation(parent_set,y) THEN
          RETURN(FALSE);
        END_IF;
      END_REPEAT;
    END_IF;
    RETURN(TRUE);

  END_FUNCTION; -- acyclic_mapped_representation

  FUNCTION associated_surface(
               arg: pcurve_or_surface
      ): surface;

    LOCAL
      surf : surface;
    END_LOCAL;
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.PCURVE' IN TYPEOF(arg) THEN
      surf := arg.basis_surface;
    ELSE
      surf := arg;
    END_IF;
    RETURN(surf);

  END_FUNCTION; -- associated_surface
```



```
FUNCTION bag_to_set(
             the_bag: BAG OF GENERIC:intype
    ): SET OF GENERIC:intype;

  LOCAL
    the_set : SET OF GENERIC:intype := [];
  END_LOCAL;
  IF SIZEOF(the_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(the_bag) BY 1;
      the_set := the_set + the_bag[i];
    END_REPEAT;
  END_IF;
  RETURN(the_set);

END_FUNCTION; -- bag_to_set

FUNCTION base_axis(
             dim: INTEGER;
             axis1, axis2, axis3: direction
    ): LIST [2:3] OF direction;

  LOCAL
    u      : LIST [2:3] OF direction;
    d1     : direction;
    d2     : direction;
    factor : REAL;
  END_LOCAL;
  IF dim = 3 THEN
    d1 := NVL(normalise(axis3),dummy_gri || direction([0,0,1]));
    d2 := first_proj_axis(d1,axis1);
    u := [d2,second_proj_axis(d1,d2,axis2),d1];
  ELSE
    IF EXISTS(axis1) THEN
      d1 := normalise(axis1);
      u := [d1,orthogonal_complement(d1)];
      IF EXISTS(axis2) THEN
        factor := dot_product(axis2,u[2]);
        IF factor < 0 THEN
          u[2].direction_ratios[1] := -u[2].direction_ratios[1];
          u[2].direction_ratios[2] := -u[2].direction_ratios[2];
        END_IF;
      END_IF;
    ELSE
      IF EXISTS(axis2) THEN
        d1 := normalise(axis2);
        u := [orthogonal_complement(d1),d1];
        u[1].direction_ratios[1] := -u[1].direction_ratios[1];
        u[1].direction_ratios[2] := -u[1].direction_ratios[2];
      ELSE
        u := [dummy_gri || direction([1,0]),dummy_gri || direction([0,1])];
      END_IF;
    END_IF;
  END_IF;
  RETURN(u);
```

```
END_FUNCTION; -- base_axis

FUNCTION boolean_choose(
             b: BOOLEAN;
             choice1, choice2: GENERIC:item
    ): GENERIC:item;
  IF b THEN
    RETURN(choice1);
  ELSE
    RETURN(choice2);
  END_IF;

END_FUNCTION; -- boolean_choose

FUNCTION build_2axes(
             ref_direction: direction
    ): LIST [2:2] OF direction;

  LOCAL
    d : direction := NVL(normalise(ref_direction),dummy_gri ||
         direction([1,0]));
  END_LOCAL;
  RETURN([d,orthogonal_complement(d)]);

END_FUNCTION; -- build_2axes

FUNCTION build_axes(
             axis, ref_direction: direction
    ): LIST [3:3] OF direction;

  LOCAL
    d1 : direction;
    d2 : direction;
  END_LOCAL;
  d1 := NVL(normalise(axis),dummy_gri || direction([0,0,1]));
  d2 := first_proj_axis(d1,ref_direction);
  RETURN([d2,normalise(cross_product(d1,d2)).orientation,d1]);

END_FUNCTION; -- build_axes

FUNCTION closed_shell_reversed(
             a_shell: closed_shell
    ): oriented_closed_shell;

  LOCAL
    the_reverse : oriented_closed_shell;
  END_LOCAL;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_CLOSED_SHELL' IN TYPEOF(
      a_shell) THEN
    the_reverse := ((dummy_tri || connected_face_set(a_shell\
        connected_face_set.cfs_faces)) || closed_shell()) ||
        oriented_closed_shell(a_shell\oriented_closed_shell.
        closed_shell_element,NOT a_shell\oriented_closed_shell.
        orientation);
  ELSE
    the_reverse := ((dummy_tri || connected_face_set(a_shell\
```



```
        connected_face_set.cfs_faces)) || closed_shell()) ||
        oriented_closed_shell(a_shell,FALSE);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- closed_shell_reversed

FUNCTION conditional_reverse(
             p: BOOLEAN;
             an_item: reversible_topology
    ): reversible_topology;
  IF p THEN
    RETURN(an_item);
  ELSE
    RETURN(topology_reversed(an_item));
  END_IF;

END_FUNCTION; -- conditional_reverse

FUNCTION constraints_composite_curve_on_surface(
             c: composite_curve_on_surface
    ): BOOLEAN;

  LOCAL
    n_segments : INTEGER := SIZEOF(c.segments);
  END_LOCAL;
  REPEAT k := 1 TO n_segments BY 1;
    IF ((NOT ('DIMENSIONAL_INSPECTION_SCHEMA.PCURVE' IN TYPEOF(c\
        composite_curve.segments[k].parent_curve))) AND (NOT (
        'DIMENSIONAL_INSPECTION_SCHEMA.SURFACE_CURVE' IN TYPEOF(c\
        composite_curve.segments[k].parent_curve)))) AND (NOT (
        'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_CURVE_ON_SURFACE' IN
        TYPEOF(c\composite_curve.segments[k].parent_curve))) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
  RETURN(TRUE);

END_FUNCTION; -- constraints_composite_curve_on_surface

FUNCTION constraints_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 'DIMENSIONAL_INSPECTION_SCHEMA.DERIVED_UNIT' IN TYPEOF(x) THEN
    REPEAT i := LOINDEX(x.elements) TO HIINDEX(x.elements) BY 1;
      result.length_exponent := result.length_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.length_exponent);
      result.mass_exponent := result.mass_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.mass_exponent);
      result.time_exponent := result.time_exponent + (x.elements[i].
          exponent * x.elements[i].unit.dimensions.time_exponent);
      result.electric_current_exponent := result.
          electric_current_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.electric_current_exponent);
      result.thermodynamic_temperature_exponent := result.
          thermodynamic_temperature_exponent + (x.elements[i].exponent *
          x.elements[i].unit.dimensions.
          thermodynamic_temperature_exponent);
      result.amount_of_substance_exponent := result.
          amount_of_substance_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.amount_of_substance_exponent);
      result.luminous_intensity_exponent := result.
          luminous_intensity_exponent + (x.elements[i].exponent * x.
          elements[i].unit.dimensions.luminous_intensity_exponent);
    END_REPEAT;
  ELSE
```

```
      result := x.dimensions;
    END_IF;
    RETURN(result);

  END_FUNCTION; -- derive_dimensional_exponents

  FUNCTION dimension_of(
              item: geometric_representation_item
      ): dimension_count;

    LOCAL
      x   : SET OF representation;
      y   : representation_context;
      dim : dimension_count;
    END_LOCAL;
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.CARTESIAN_POINT' IN TYPEOF(item)
        THEN
      dim := SIZEOF(item\cartesian_point.coordinates);
      RETURN(dim);
    END_IF;
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.DIRECTION' IN TYPEOF(item) THEN
      dim := SIZEOF(item\direction.direction_ratios);
      RETURN(dim);
    END_IF;
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.VECTOR' IN TYPEOF(item) THEN
      dim := SIZEOF(item\vector.orientation\direction.direction_ratios);
      RETURN(dim);
    END_IF;
    x := using_representations(item);
    y := x[1].context_of_items;
    dim := y\geometric_representation_context.coordinate_space_dimension;
    RETURN(dim);

  END_FUNCTION; -- dimension_of

  FUNCTION dimensions_for_si_unit(
              n: si_unit_name
      ): dimensional_exponents;
    CASE n OF
      metre       :        RETURN(dimensional_exponents(1,0,0,0,0,0,0));
      gram        :        RETURN(dimensional_exponents(0,1,0,0,0,0,0));
      second      :        RETURN(dimensional_exponents(0,0,1,0,0,0,0));
      ampere      :        RETURN(dimensional_exponents(0,0,0,1,0,0,0));
      kelvin      :        RETURN(dimensional_exponents(0,0,0,0,1,0,0));
      mole        :        RETURN(dimensional_exponents(0,0,0,0,0,1,0));
      candela     :        RETURN(dimensional_exponents(0,0,0,0,0,0,1));
      radian      :        RETURN(dimensional_exponents(0,0,0,0,0,0,0));
      steradian   :        RETURN(dimensional_exponents(0,0,0,0,0,0,0));
      hertz       :        RETURN(dimensional_exponents(0,0,-1,0,0,0,0));
      newton      :        RETURN(dimensional_exponents(1,1,-2,0,0,0,0));
      pascal      :        RETURN(dimensional_exponents(-1,1,-2,0,0,0,0));
      joule       :        RETURN(dimensional_exponents(2,1,-2,0,0,0,0));
      watt        :        RETURN(dimensional_exponents(2,1,-3,0,0,0,0));
      coulomb     :        RETURN(dimensional_exponents(0,0,1,1,0,0,0));
      volt        :        RETURN(dimensional_exponents(2,1,-3,-1,0,0,0));
      farad       :        RETURN(dimensional_exponents(-2,-1,4,1,0,0,0));
```



```
    ohm           :         RETURN(dimensional_exponents(2,1,-3,-2,0,0,0));
    siemens       :         RETURN(dimensional_exponents(-2,-1,3,2,0,0,0));
    weber         :         RETURN(dimensional_exponents(2,1,-2,-1,0,0,0));
    tesla         :         RETURN(dimensional_exponents(0,1,-2,-1,0,0,0));
    henry         :         RETURN(dimensional_exponents(2,1,-2,-2,0,0,0));
    degree_celsius  :         RETURN(dimensional_exponents(0,0,0,0,1,0,0));
    lumen         :         RETURN(dimensional_exponents(0,0,0,0,0,0,1));
    lux           :         RETURN(dimensional_exponents(-2,0,0,0,0,0,1));
    becquerel     :         RETURN(dimensional_exponents(0,0,-1,0,0,0,0));
    gray          :         RETURN(dimensional_exponents(2,0,-2,0,0,0,0));
    sievert       :         RETURN(dimensional_exponents(2,0,-2,0,0,0,0));
    END_CASE;

END_FUNCTION; -- dimensions_for_si_unit

FUNCTION dot_product(
             arg1, arg2: direction
    ): REAL;

  LOCAL
    ndim   : INTEGER;
    scalar : REAL;
    vec1   : direction;
    vec2   : direction;
  END_LOCAL;
  IF (NOT EXISTS(arg1)) OR (NOT EXISTS(arg2)) THEN
    scalar := ?;
  ELSE
    IF arg1.dim <> arg2.dim THEN
      scalar := ?;
    ELSE
      BEGIN
        vec1 := normalise(arg1);
        vec2 := normalise(arg2);
        ndim := arg1.dim;
        scalar := 0;
        REPEAT i := 1 TO ndim BY 1;
          scalar := scalar + (vec1.direction_ratios[i] * vec2.
              direction_ratios[i]);
        END_REPEAT;
      END;
    END_IF;
  END_IF;
  RETURN(scalar);

END_FUNCTION; -- dot_product

FUNCTION edge_reversed(
             an_edge: edge
    ): oriented_edge;

  LOCAL
    the_reverse : oriented_edge;
  END_LOCAL;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_EDGE' IN TYPEOF(an_edge)
     THEN
```

```
      the_reverse := (dummy_tri || edge(an_edge.edge_end,an_edge.
          edge_start)) || oriented_edge(an_edge\oriented_edge.edge_element,
          NOT an_edge\oriented_edge.orientation);
    ELSE
      the_reverse := (dummy_tri || edge(an_edge.edge_end,an_edge.
          edge_start)) || oriented_edge(an_edge,FALSE);
    END_IF;
    RETURN(the_reverse);

  END_FUNCTION; -- edge_reversed

  FUNCTION face_bound_reversed(
               a_face_bound: face_bound
      ): face_bound;

    LOCAL
      the_reverse : face_bound;
    END_LOCAL;
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.FACE_OUTER_BOUND' IN TYPEOF(
        a_face_bound) THEN
      the_reverse := (dummy_tri || face_bound(a_face_bound\face_bound.
          bound,NOT a_face_bound\face_bound.orientation)) ||
          face_outer_bound();
    ELSE
      the_reverse := dummy_tri || face_bound(a_face_bound.bound,NOT
          a_face_bound.orientation);
    END_IF;
    RETURN(the_reverse);

  END_FUNCTION; -- face_bound_reversed

  FUNCTION face_reversed(
               a_face: face
      ): oriented_face;

    LOCAL
      the_reverse : oriented_face;
    END_LOCAL;
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_FACE' IN TYPEOF(a_face)
         THEN
      the_reverse := (dummy_tri || face(set_of_topology_reversed(a_face.
          bounds))) || oriented_face(a_face\oriented_face.face_element,NOT
          a_face\oriented_face.orientation);
    ELSE
      the_reverse := (dummy_tri || face(set_of_topology_reversed(a_face.
          bounds))) || oriented_face(a_face,FALSE);
    END_IF;
    RETURN(the_reverse);

  END_FUNCTION; -- face_reversed

  FUNCTION first_proj_axis(
               z_axis, arg: direction
      ): direction;

    LOCAL
```



```
    x_vec  : vector;
    v      : direction;
    z      : direction;
    x_axis : direction;
  END_LOCAL;
  IF NOT EXISTS(z_axis) THEN
    RETURN(?);
  ELSE
    z := normalise(z_axis);
    IF NOT EXISTS(arg) THEN
      IF z.direction_ratios <> [1,0,0] THEN
        v := dummy_gri || direction([1,0,0]);
      ELSE
        v := dummy_gri || direction([0,1,0]);
      END_IF;
    ELSE
      IF arg.dim <> 3 THEN
        RETURN(?);
      END_IF;
      IF cross_product(arg,z).magnitude = 0 THEN
        RETURN(?);
      ELSE
        v := normalise(arg);
      END_IF;
    END_IF;
    x_vec := scalar_times_vector(dot_product(v,z),z);
    x_axis := vector_difference(v,x_vec).orientation;
    x_axis := normalise(x_axis);
  END_IF;
  RETURN(x_axis);

END_FUNCTION; -- first_proj_axis

FUNCTION 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 'DIMENSIONAL_INSPECTION_SCHEMA.PCURVE' IN TYPEOF(c) THEN
    surfs := [c\pcurve.basis_surface];
  ELSE
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.SURFACE_CURVE' IN TYPEOF(c) THEN
      n := SIZEOF(c\surface_curve.associated_geometry);
      REPEAT i := 1 TO n BY 1;
        surfs := surfs + associated_surface(c\surface_curve.
            associated_geometry[i]);
      END_REPEAT;
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.COMPOSITE_CURVE_ON_SURFACE' IN
      TYPEOF(c) THEN
    n := SIZEOF(c\composite_curve.segments);
```

```
      surfs := get_basis_surface(c\composite_curve.segments[1].
          parent_curve);
      IF n > 1 THEN
        REPEAT i := 2 TO n BY 1;
          surfs := surfs * get_basis_surface(c\composite_curve.segments[i]
              .parent_curve);
        END_REPEAT;
      END_IF;
    END_IF;
    RETURN(surfs);

  END_FUNCTION; -- get_basis_surface

  FUNCTION get_description_value(
               obj: description_attribute_select
      ): text;

    LOCAL
      description_bag : BAG OF description_attribute := USEDIN(obj,(
                         'DIMENSIONAL_INSPECTION_SCHEMA.' +
                         'DESCRIPTION_ATTRIBUTE.') + 'DESCRIBED_ITEM');
    END_LOCAL;
    IF SIZEOF(description_bag) = 1 THEN
      RETURN(description_bag[1].attribute_value);
    ELSE
      RETURN(?);
    END_IF;

  END_FUNCTION; -- get_description_value

  FUNCTION get_id_value(
               obj: id_attribute_select
      ): identifier;

    LOCAL
      id_bag : BAG OF id_attribute := USEDIN(obj,(
                'DIMENSIONAL_INSPECTION_SCHEMA.' + 'ID_ATTRIBUTE.') +
                'IDENTIFIED_ITEM');
    END_LOCAL;
    IF SIZEOF(id_bag) = 1 THEN
      RETURN(id_bag[1].attribute_value);
    ELSE
      RETURN(?);
    END_IF;

  END_FUNCTION; -- get_id_value

  FUNCTION get_name_value(
               obj: name_attribute_select
      ): label;

    LOCAL
      name_bag : BAG OF name_attribute := USEDIN(obj,(
                  'DIMENSIONAL_INSPECTION_SCHEMA.' + 'NAME_ATTRIBUTE.') +
                  'NAMED_ITEM');
    END_LOCAL;
```



```
    IF SIZEOF(name_bag) = 1 THEN
      RETURN(name_bag[1].attribute_value);
    ELSE
      RETURN(?);
    END_IF;

  END_FUNCTION; -- get_name_value

  FUNCTION get_property_definition_representations(
               c_def_instance: characterized_definition
      ): SET OF property_definition_representation;

    LOCAL
      pdr_set : SET OF property_definition_representation := [];
      pd_set  : SET OF property_definition := [];
    END_LOCAL;
    pd_set := bag_to_set(USEDIN(c_def_instance,
        'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITION.DEFINITION'));
    IF SIZEOF(pd_set) < 1 THEN
      RETURN(pdr_set);
    END_IF;
    REPEAT i := 1 TO HIINDEX(pd_set) BY 1;
      pdr_set                  :=                  pdr_set                  +
bag_to_set(USEDIN(pd_set[i],'DIMENSIONAL_INSPECTION_SCHEMA.PROPERTY_DEFINITIO
N_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,(
                'DIMENSIONAL_INSPECTION_SCHEMA.' + 'ROLE_ASSOCIATION.') +
                 'ITEM_WITH_ROLE');
    END_LOCAL;
    IF SIZEOF(role_bag) = 1 THEN
      RETURN(role_bag[1].role);
    ELSE
      RETURN(?);
    END_IF;

  END_FUNCTION; -- get_role

  FUNCTION item_in_context(
               item: representation_item;
               cntxt: representation_context
      ): BOOLEAN;

    LOCAL
      y : BAG OF representation_item;
    END_LOCAL;
    IF SIZEOF(USEDIN(item,
```

```
        'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION.ITEMS') * cntxt.
        representations_in_context) > 0 THEN
      RETURN(TRUE);
    ELSE
      y := QUERY ( z <* USEDIN(item,'') | (
          'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION_ITEM'   IN   TYPEOF(z))
);
      IF SIZEOF(y) > 0 THEN
        REPEAT i := 1 TO HIINDEX(y) BY 1;
          IF item_in_context(y[i],cntxt) THEN
            RETURN(TRUE);
          END_IF;
        END_REPEAT;
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- item_in_context

  FUNCTION leap_year(
             year: year_number
      ): BOOLEAN;
    IF (((year MOD 4) = 0) AND ((year MOD 100) <> 0)) OR ((year MOD 400) =
         0) THEN
      RETURN(TRUE);
    ELSE
      RETURN(FALSE);
    END_IF;

  END_FUNCTION; -- leap_year

  FUNCTION list_face_loops(
             f: face
      ): LIST [0:?] OF loop;

    LOCAL
      loops : LIST [0:?] OF loop := [];
    END_LOCAL;
    REPEAT i := 1 TO SIZEOF(f.bounds) BY 1;
      loops := loops + f.bounds[i].bound;
    END_REPEAT;
    RETURN(loops);

  END_FUNCTION; -- list_face_loops

  FUNCTION list_of_topology_reversed(
             a_list: list_of_reversible_topology_item
      ): list_of_reversible_topology_item;

    LOCAL
      the_reverse : list_of_reversible_topology_item;
    END_LOCAL;
    the_reverse := [];
    REPEAT i := 1 TO SIZEOF(a_list) BY 1;
      the_reverse := topology_reversed(a_list[i]) + the_reverse;
    END_REPEAT;
```



```
    RETURN(the_reverse);

  END_FUNCTION; -- list_of_topology_reversed

  FUNCTION list_to_array(
               lis: LIST [0:?] OF GENERIC:t;
               low, u: INTEGER
      ): ARRAY OF GENERIC:t;

    LOCAL
      n   : INTEGER;
      res : ARRAY [low:u] OF GENERIC:t;
    END_LOCAL;
    n := SIZEOF(lis);
    IF n <> ((u - low) + 1) THEN
      RETURN(?);
    ELSE
      res := [lis[1],n];
      REPEAT i := 2 TO n BY 1;
        res[(low + i) - 1] := lis[i];
      END_REPEAT;
      RETURN(res);
    END_IF;

  END_FUNCTION; -- list_to_array

  FUNCTION list_to_set(
               l: LIST [0:?] OF GENERIC:t
      ): SET OF GENERIC:t;

    LOCAL
      s : SET OF GENERIC:t := [];
    END_LOCAL;
    REPEAT i := 1 TO SIZEOF(l) BY 1;
      s := s + l[i];
    END_REPEAT;
    RETURN(s);

  END_FUNCTION; -- list_to_set

  FUNCTION make_array_of_array(
               lis: LIST [1:?] OF LIST [1:?] OF GENERIC:t;
               low1, u1, low2, u2: INTEGER
      ): ARRAY OF ARRAY OF GENERIC:t;

    LOCAL
      res : ARRAY [low1:u1] OF ARRAY [low2:u2] OF GENERIC:t;
    END_LOCAL;
    IF ((u1 - low1) + 1) <> SIZEOF(lis) THEN
      RETURN(?);
    END_IF;
    IF ((u2 - low2) + 1) <> SIZEOF(lis[1]) THEN
      RETURN(?);
    END_IF;
    res := [list_to_array(lis[1],low2,u2),(u1 - low1) + 1];
    REPEAT i := 2 TO HIINDEX(lis) BY 1;
```

```
      IF ((u2 - low2) + 1) <> SIZEOF(lis[i]) THEN
        RETURN(?);
      END_IF;
      res[(low1 + i) - 1] := list_to_array(lis[i],low2,u2);
    END_REPEAT;
    RETURN(res);

  END_FUNCTION; -- make_array_of_array

  FUNCTION mixed_loop_type_set(
               l: SET [0:?] OF loop
      ): LOGICAL;

    LOCAL
      poly_loop_type : LOGICAL;
    END_LOCAL;
    IF SIZEOF(l) <= 1 THEN
      RETURN(FALSE);
    END_IF;
    poly_loop_type := 'DIMENSIONAL_INSPECTION_SCHEMA.POLY_LOOP' IN TYPEOF(
        l[1]);
    REPEAT i := 2 TO SIZEOF(l) BY 1;
      IF ('DIMENSIONAL_INSPECTION_SCHEMA.POLY_LOOP' IN TYPEOF(l[i])) <>
          poly_loop_type THEN
        RETURN(TRUE);
      END_IF;
    END_REPEAT;
    RETURN(FALSE);

  END_FUNCTION; -- mixed_loop_type_set

  FUNCTION msb_shells(
               brep: manifold_solid_brep
      ): SET [1:?] OF closed_shell;
    IF SIZEOF(QUERY ( msbtype <* TYPEOF(brep) | (msbtype LIKE
        '*BREP_WITH_VOIDS') )) >= 1 THEN
      RETURN(brep\brep_with_voids.voids + brep.outer);
    ELSE
      RETURN([brep.outer]);
    END_IF;

  END_FUNCTION; -- msb_shells

  FUNCTION 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 'DIMENSIONAL_INSPECTION_SCHEMA.VECTOR' IN TYPEOF(arg) THEN
        BEGIN
          v := dummy_gri || direction(arg.orientation.direction_ratios);
          IF arg.magnitude = 0 THEN
            RETURN(?);
          ELSE
            vec := dummy_gri || vector(v,1);
          END_IF;
        END;
      ELSE
        v := dummy_gri || direction(arg.direction_ratios);
      END_IF;
      mag := 0;
      REPEAT i := 1 TO ndim BY 1;
        mag := mag + (v.direction_ratios[i] * v.direction_ratios[i]);
      END_REPEAT;
      IF mag > 0 THEN
        mag := SQRT(mag);
        REPEAT i := 1 TO ndim BY 1;
          v.direction_ratios[i] := v.direction_ratios[i] / mag;
        END_REPEAT;
        IF 'DIMENSIONAL_INSPECTION_SCHEMA.VECTOR' IN TYPEOF(arg) THEN
          vec.orientation := v;
          result := vec;
        ELSE
          result := v;
        END_IF;
      ELSE
        RETURN(?);
      END_IF;
    END_IF;
    RETURN(result);

  END_FUNCTION; -- normalise

  FUNCTION open_shell_reversed(
                 a_shell: open_shell
      ): oriented_open_shell;

    LOCAL
      the_reverse : oriented_open_shell;
    END_LOCAL;
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_OPEN_SHELL' IN TYPEOF(
        a_shell) THEN
      the_reverse := ((dummy_tri || connected_face_set(a_shell\
          connected_face_set.cfs_faces)) || open_shell()) ||
          oriented_open_shell(a_shell\oriented_open_shell.
          open_shell_element,NOT a_shell\oriented_open_shell.orientation);
    ELSE
      the_reverse := ((dummy_tri || connected_face_set(a_shell\
          connected_face_set.cfs_faces)) || open_shell()) ||
          oriented_open_shell(a_shell,FALSE);
    END_IF;
    RETURN(the_reverse);
```

```
END_FUNCTION; -- open_shell_reversed

FUNCTION orthogonal_complement(
             vec: direction
    ): direction;

  LOCAL
    result : direction;
  END_LOCAL;
  IF (vec.dim <> 2) OR (NOT EXISTS(vec)) THEN
    RETURN(?);
  ELSE
    result := dummy_gri || direction([-vec.direction_ratios[2],vec.
        direction_ratios[1]]);
    RETURN(result);
  END_IF;

END_FUNCTION; -- orthogonal_complement

FUNCTION path_head_to_tail(
             a_path: path
    ): BOOLEAN;

  LOCAL
    n : INTEGER;
    p : BOOLEAN := TRUE;
  END_LOCAL;
  n := SIZEOF(a_path.edge_list);
  REPEAT i := 2 TO n BY 1;
    p := p AND (a_path.edge_list[i - 1].edge_end :=: a_path.edge_list[i]
        .edge_start);
  END_REPEAT;
  RETURN(p);

END_FUNCTION; -- path_head_to_tail

FUNCTION path_reversed(
             a_path: path
    ): oriented_path;

  LOCAL
    the_reverse : oriented_path;
  END_LOCAL;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.ORIENTED_PATH' IN TYPEOF(a_path)
       THEN
    the_reverse := (dummy_tri || path(list_of_topology_reversed(a_path.
        edge_list))) || oriented_path(a_path\oriented_path.path_element,
        NOT a_path\oriented_path.orientation);
  ELSE
    the_reverse := (dummy_tri || path(list_of_topology_reversed(a_path.
        edge_list))) || oriented_path(a_path,FALSE);
  END_IF;
  RETURN(the_reverse);

END_FUNCTION; -- path_reversed
```



```
FUNCTION scalar_times_vector(
             scalar: REAL;
             vec: vector_or_direction
    ): vector;

  LOCAL
    v      : direction;
    mag    : REAL;
    result : vector;
  END_LOCAL;
  IF (NOT EXISTS(scalar)) OR (NOT EXISTS(vec)) THEN
    RETURN(?);
  ELSE
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.VECTOR' IN TYPEOF(vec) THEN
      v := dummy_gri || direction(vec.orientation.direction_ratios);
      mag := scalar * vec.magnitude;
    ELSE
      v := dummy_gri || direction(vec.direction_ratios);
      mag := scalar;
    END_IF;
    IF mag < 0 THEN
      REPEAT i := 1 TO SIZEOF(v.direction_ratios) BY 1;
        v.direction_ratios[i] := -v.direction_ratios[i];
      END_REPEAT;
      mag := -mag;
    END_IF;
    result := dummy_gri || vector(normalise(v),mag);
  END_IF;
  RETURN(result);

END_FUNCTION; -- scalar_times_vector

FUNCTION second_proj_axis(
             z_axis, x_axis, arg: direction
    ): direction;

  LOCAL
    temp   : vector;
    v      : direction;
    y_axis : vector;
  END_LOCAL;
  IF NOT EXISTS(arg) THEN
    v := dummy_gri || direction([0,1,0]);
  ELSE
    v := arg;
  END_IF;
  temp := scalar_times_vector(dot_product(v,z_axis),z_axis);
  y_axis := vector_difference(v,temp);
  temp := scalar_times_vector(dot_product(v,x_axis),x_axis);
  y_axis := vector_difference(y_axis,temp);
  y_axis := normalise(y_axis);
  RETURN(y_axis.orientation);

END_FUNCTION; -- second_proj_axis
```

```
FUNCTION set_of_topology_reversed(
             a_set: set_of_reversible_topology_item
    ): set_of_reversible_topology_item;

  LOCAL
    the_reverse : set_of_reversible_topology_item;
  END_LOCAL;
  the_reverse := [];
  REPEAT i := 1 TO SIZEOF(a_set) BY 1;
    the_reverse := the_reverse + topology_reversed(a_set[i]);
  END_REPEAT;
  RETURN(the_reverse);

END_FUNCTION; -- set_of_topology_reversed

FUNCTION shell_reversed(
             a_shell: shell
    ): shell;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.OPEN_SHELL' IN TYPEOF(a_shell) THEN
    RETURN(open_shell_reversed(a_shell));
  ELSE
    IF 'DIMENSIONAL_INSPECTION_SCHEMA.CLOSED_SHELL' IN TYPEOF(a_shell)
         THEN
      RETURN(closed_shell_reversed(a_shell));
    ELSE
      RETURN(?);
    END_IF;
  END_IF;

END_FUNCTION; -- shell_reversed

FUNCTION surface_weights_positive(
             b: rational_b_spline_surface
    ): BOOLEAN;

  LOCAL
    result : BOOLEAN := TRUE;
  END_LOCAL;
  REPEAT i := 0 TO b.u_upper BY 1;
    REPEAT j := 0 TO b.v_upper BY 1;
      IF b.weights[i][j] <= 0 THEN
        result := FALSE;
        RETURN(result);
      END_IF;
    END_REPEAT;
  END_REPEAT;
  RETURN(result);

END_FUNCTION; -- surface_weights_positive

FUNCTION topology_reversed(
             an_item: reversible_topology
    ): reversible_topology;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.EDGE' IN TYPEOF(an_item) THEN
    RETURN(edge_reversed(an_item));
  END_IF;
```



```
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.PATH' IN TYPEOF(an_item) THEN
    RETURN(path_reversed(an_item));
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.FACE_BOUND' IN TYPEOF(an_item) THEN
    RETURN(face_bound_reversed(an_item));
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.FACE' IN TYPEOF(an_item) THEN
    RETURN(face_reversed(an_item));
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.SHELL' IN TYPEOF(an_item) THEN
    RETURN(shell_reversed(an_item));
  END_IF;
  IF 'SET' IN TYPEOF(an_item) THEN
    RETURN(set_of_topology_reversed(an_item));
  END_IF;
  IF 'LIST' IN TYPEOF(an_item) THEN
    RETURN(list_of_topology_reversed(an_item));
  END_IF;
  RETURN(?);

END_FUNCTION; -- topology_reversed

FUNCTION 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,'')) | ((
      'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z))
      OR ('DIMENSIONAL_INSPECTION_SCHEMA.FOUNDED_ITEM' IN TYPEOF(z))) );
  IF SIZEOF(next_items) > 0 THEN
    REPEAT i := 1 TO HIINDEX(next_items) BY 1;
      IF NOT (next_items[i] IN new_check_items) THEN
        result_items := (result_items + next_items[i]) + using_items(
            next_items[i],new_check_items);
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(result_items);

END_FUNCTION; -- using_items

FUNCTION using_representations(
             item: founded_item_select
    ): SET OF representation;

  LOCAL
    results            : SET OF representation;
    intermediate_items : SET OF founded_item_select;
```

```
      result_bag         : BAG OF representation;
    END_LOCAL;
    results := [];
    result_bag := USEDIN(item,
        'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION.ITEMS');
    IF SIZEOF(result_bag) > 0 THEN
      REPEAT i := 1 TO HIINDEX(result_bag) BY 1;
        results := results + result_bag[i];
      END_REPEAT;
    END_IF;
    intermediate_items := using_items(item,[]);
    IF SIZEOF(intermediate_items) > 0 THEN
      REPEAT i := 1 TO HIINDEX(intermediate_items) BY 1;
        result_bag := USEDIN(intermediate_items[i],
            'DIMENSIONAL_INSPECTION_SCHEMA.REPRESENTATION.ITEMS');
        IF SIZEOF(result_bag) > 0 THEN
          REPEAT j := 1 TO HIINDEX(result_bag) BY 1;
            results := results + result_bag[j];
          END_REPEAT;
        END_IF;
      END_REPEAT;
    END_IF;
    RETURN(results);

  END_FUNCTION; -- using_representations

  FUNCTION valid_calendar_date(
               date: calendar_date
      ): LOGICAL;
    CASE date.month_component OF
      1 :         RETURN((1 <= date.day_component) AND (date.day_component
            <= 31));
      2 :         BEGIN
          IF leap_year(date.year_component) THEN
            RETURN((1 <= date.day_component) AND (date.day_component <= 29));
          ELSE
            RETURN((1 <= date.day_component) AND (date.day_component <= 28));
          END_IF;
        END;
      3 :         RETURN((1 <= date.day_component) AND (date.day_component
            <= 31));
      4 :         RETURN((1 <= date.day_component) AND (date.day_component
            <= 30));
      5 :         RETURN((1 <= date.day_component) AND (date.day_component
            <= 31));
      6 :         RETURN((1 <= date.day_component) AND (date.day_component
            <= 30));
      7 :         RETURN((1 <= date.day_component) AND (date.day_component
            <= 31));
      8 :         RETURN((1 <= date.day_component) AND (date.day_component
            <= 31));
      9 :         RETURN((1 <= date.day_component) AND (date.day_component
            <= 30));
      10  :         RETURN((1 <= date.day_component) AND (date.
            day_component <= 31));
      11  :         RETURN((1 <= date.day_component) AND (date.
```



```
          day_component <= 30));
    12  :        RETURN((1 <= date.day_component) AND (date.
          day_component <= 31));
    END_CASE;
  RETURN(FALSE);

END_FUNCTION; -- valid_calendar_date

FUNCTION valid_measure_value(
             m: measure_value
    ): BOOLEAN;
  IF 'REAL' IN TYPEOF(m) THEN
    RETURN(m > 0);
  ELSE
    IF 'INTEGER' IN TYPEOF(m) THEN
      RETURN(m > 0);
    ELSE
      RETURN(TRUE);
    END_IF;
  END_IF;

END_FUNCTION; -- valid_measure_value

FUNCTION valid_time(
             time: local_time
    ): BOOLEAN;
  IF EXISTS(time.second_component) THEN
    RETURN(EXISTS(time.minute_component));
  ELSE
    RETURN(TRUE);
  END_IF;

END_FUNCTION; -- valid_time

FUNCTION valid_units(
             m: measure_with_unit
    ): BOOLEAN;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.LENGTH_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(1,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.MASS_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,1,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.TIME_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,1,0,0,0,0) THEN
      RETURN(FALSE);
```

```
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.ELECTRIC_CURRENT_MEASURE' IN TYPEOF(
      m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,1,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.THERMODYNAMIC_TEMPERATURE_MEASURE'
      IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,1,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.CELSIUS_TEMPERATURE_MEASURE' IN
      TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,1,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.AMOUNT_OF_SUBSTANCE_MEASURE' IN
      TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,1,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.LUMINOUS_INTENSITY_MEASURE' IN
      TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,1) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.PLANE_ANGLE_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.SOLID_ANGLE_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.AREA_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(2,0,0,0,0,0,0) THEN
      RETURN(FALSE);
```



```
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.VOLUME_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(3,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.RATIO_MEASURE' IN TYPEOF(m.
      value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.POSITIVE_LENGTH_MEASURE' IN TYPEOF(m
      .value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(1,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  IF 'DIMENSIONAL_INSPECTION_SCHEMA.POSITIVE_PLANE_ANGLE_MEASURE' IN
      TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  RETURN(TRUE);

END_FUNCTION; -- valid_units

FUNCTION 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 'DIMENSIONAL_INSPECTION_SCHEMA.VECTOR' IN TYPEOF(arg1) THEN
        mag1 := arg1.magnitude;
        vec1 := arg1.orientation;
```



```
        ELSE
          mag1 := 1;
          vec1 := arg1;
        END_IF;
        IF 'DIMENSIONAL_INSPECTION_SCHEMA.VECTOR' IN TYPEOF(arg2) THEN
          mag2 := arg2.magnitude;
          vec2 := arg2.orientation;
        ELSE
          mag2 := 1;
          vec2 := arg2;
        END_IF;
        vec1 := normalise(vec1);
        vec2 := normalise(vec2);
        ndim := SIZEOF(vec1.direction_ratios);
        mag := 0;
        res := dummy_gri || direction(vec1.direction_ratios);
        REPEAT i := 1 TO ndim BY 1;
          res.direction_ratios[i] := (mag1 * vec1.direction_ratios[i]) + (
              mag2 * vec2.direction_ratios[i]);
          mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
        END_REPEAT;
        IF mag > 0 THEN
          result := dummy_gri || vector(res,SQRT(mag));
        ELSE
          result := dummy_gri || vector(vec1,0);
        END_IF;
      END;
    END_IF;
    RETURN(result);

  END_FUNCTION; -- vector_difference

END_SCHEMA; -- dimensional_inspection_schema
```


Annex B

(normative)

AIM short names

Table B.1 provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

Table B.1 — AIM short names of entities

AIM Element	Short name
ACTION	ACTION
ACTION_ASSIGNMENT	ACTASS
ACTION_METHOD	ACTMTH
ACTION_PROPERTY	ACTPRP
ACTION_PROPERTY_REPRESENTATION	ACPRRP
ACTION_RELATIONSHIP	ACTRLT
ACTION_RESOURCE	ACTRSR
ACTION_RESOURCE_REQUIREMENT	ACRSRQ
ACTION_RESOURCE_TYPE	ACRSTY
ADDRESS	ADDRSS
ADVANCED_BREP_SHAPE_REPRESENTATION	ABSR
ADVANCED_FACE	ADVFC
ANGULAR_LOCATION	ANGLCT
ANGULAR_SIZE	ANGSZ
ANGULARITY_TOLERANCE	ANGTLR
APEX	APEX
APPLICATION_CONTEXT	APPCNT
APPLICATION_CONTEXT_ELEMENT	APCNEL
APPLICATION_PROTOCOL_DEFINITION	APPRDF
APPLIED_ACTION_ASSIGNMENT	APACAS
APPLIED_AREA	APPAR
APPLIED_DATE_ASSIGNMENT	APDTAS
APPLIED_DOCUMENT_REFERENCE	APDCRF

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	ADUCA
APPLIED_GROUP_ASSIGNMENT	APGRAS
APPLIED_EFFECTIVITY_ASSIGNMENT	APEFAS
APPLIED_ORGANIZATION_ASSIGNMENT	APORAS
APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT	APAOA
APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT	ASCA
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
CENTRE_OF_SYMMETRY	CNOFSY
CHAMFER	CHMFR
CHAMFER_OFFSET	CHMOFF
CHARACTERIZED_OBJECT	CHROBJ
CIRCLE	CIRCLE

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
CIRCULAR_CLOSED_PROFILE	CRCLPR
CIRCULAR_PATTERN	CRCPTT
CIRCULAR_RUNOUT_TOLERANCE	CRRNTL
CLOSED_PATH_PROFILE	CLPTPR
CLOSED_SHELL	CLSSHL
COAXIALITY_TOLERANCE	CXLTLR
COMMON_DATUM	CMMDTM
COMPOSITE_CURVE	CMPCRV
COMPOSITE_CURVE_ON_SURFACE	CCOS
COMPOSITE_CURVE_SEGMENT	CMCRSG
COMPOSITE_HOLE	CMPHL
COMPOSITE_SHAPE_ASPECT	CMSHAS
COMPOUND_FEATURE	CMPFTR
COMPOUND_REPRESENTATION_ITEM	CMRPIT
CONCENTRICITY_TOLERANCE	CNCTLR
CONIC	CONIC
CONICAL_SURFACE	CNCSRF
CONNECTED_FACE_SET	CNFCST
CONTEXT_DEPENDENT_UNIT	CNDPUN
CONVERSION_BASED_UNIT	CNBSUN
COORDINATED_UNIVERSAL_TIME_OFFSET	CUTO
CURVE	CURVE
CYLINDRICAL_SHAPE_REPRESENTATION	CYSHRP
CYLINDRICAL_SURFACE	CYLSRF
CYLINDRICITY_TOLERANCE	CYLTLR
DATE	DATE
DATE_AND_TIME	DTANTM
DATE_ASSIGNMENT	DTASS
DATE_ROLE	DTRL
DATED_EFFECTIVITY	DTDEFF

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
DATUM	DATUM
DATUM_FEATURE	DTMFTR
DATUM_REFERENCE	DTMRFR
DATUM_TARGET	DTMTRG
DEFINITIONAL_REPRESENTATION	DFNRPR
DEGENERATE_TOROIDAL_SURFACE	DGTRSR
DERIVED_SHAPE_ASPECT	DRSHAS
DERIVED_UNIT	DRVUNT
DERIVED_UNIT_ELEMENT	DRUNEL
DESCRIPTION_ATTRIBUTE	DSCATT
DESCRIPTIVE_REPRESENTATION_ITEM	DSRPIT
DIMENSION_RELATED_TOLERANCE_ZONE_ELEMENT	DRTZE
DIMENSIONAL_CHARACTERISTIC_REPRESENTATION	DMCHRP
DIMENSIONAL_EXPONENTS	DMNEXP
DIMENSIONAL_LOCATION	DMNLCT
DIMENSIONAL_LOCATION_WITH_PATH	DLWP
DIMENSIONAL_MEASUREMENT_REPRESENTATION	DMMSRP
DIMENSIONAL_SIZE	DMNSZ
DIMENSIONAL_SIZE_WITH_PATH	DSWP
DIRECTED_DIMENSIONAL_LOCATION	DRDMLC
DIRECTION	DRCTN
DIRECTION_SHAPE_REPRESENTATION	DRSHRP
DM_ANALYSIS_DOFS_DML	DADD
DM_DIMENSION_PARAMETER	DMDMPR
DM_EXECUTION_INPUT	DMEXIN
DM_EXECUTION_RESULT	DMEXRS
DM_EXECUTION_RESULT_MEASUREMENT	DERM
DM_FEATURE_ANALYSIS_MODE_DML	DFAMD
DM_FEATURE_DEFINITION	DMFTDF
DM_FEATURE_RELATIONSHIP	DMFTRL

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
DM_INSTANCED_FEATURE	DMINFT
DM_PARAMETER_ANALYSIS_DML	DPAD
DM_PARAMETER_VALUE_LIMITS	DPVL
DM_POINT	DMPNT
DM_POINT_PARAMETER	DMPNPR
DM_PROGRAM_IDENTIFICATION	DMPRID
DM_PROGRAM_RUN	DMPRRN
DM_RESULT_PARAMETER	DMRSPR
DM_TOLERANCE_ANALYSIS_MODE_DML	DTAMD
DM_VECTOR_PARAMETER	DMVCPR
DMF_ARC	DMFARC
DMF_CIRCLE	DMFCRC
DMF_CONE	DMFCN
DMF_CYLINDER	DMFCYL
DMF_EDGE_POINT	DMEDPN
DMF_ELLIPSE	DMFELL
DMF_GENERIC_FEATURE	DMGNFT
DMF_GEOMETRIC_CURVE	DMGMCR
DMF_GEOMETRIC_SURFACE	DMGMSR
DMF_LINE_BOUNDED	DMLNBN
DMF_LINE_CLOSED_PARALLEL	DLCP
DMF_LINE_UNBOUNDED	DMLNUN
DMF_PATTERN	DMFPTT
DMF_PLANE	DMFPLN
DMF_PLANE_CLOSED_PARALLEL	DPCP
DMF_PLANE_SYMMETRIC	DMPLSY
DMF_POINT	DMFPNT
DMF_SPHERE	DMFSPH
DMF_SURFACE_OF_REVOLUTION_DML	DSORD
DMF_TORUS	DMFTRS

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
DOCUMENT	DCMNT
DOCUMENT_FILE	DCMFL
DOCUMENT_REFERENCE	DCMRFR
DOCUMENT_REPRESENTATION_TYPE	DCRPTY
DOCUMENT_TYPE	DCMTYP
DOCUMENT_USAGE_CONSTRAINT	DCUSCN
DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	DUCA
DOCUMENT_USAGE_ROLE	DCUSRL
DOCUMENT_WITH_CLASS	DCWTCL
EDGE	EDGE
EDGE_CURVE	EDGCRV
EDGE_LOOP	EDGLP
EDGE_ROUND	EDGRND
EFFECTIVITY	EFFCTV
EFFECTIVITY_ASSIGNMENT	EFFASS
ELEMENTARY_SURFACE	ELMSRF
ELLIPSE	ELLPS
EXECUTED_ACTION	EXCACT
EXPANDED_UNCERTAINTY	EXPUNC
EXTENSION	EXTNSN
EXTERNAL_SOURCE	EXTSRC
EXTERNALLY_DEFINED_DIMENSION_DEFINITION	EDDD
EXTERNALLY_DEFINED_FEATURE_DEFINITION	EDFD
EXTERNALLY_DEFINED_ITEM	EXDFIT
FACE	FACE
FACE_BOUND	FCBND
FACE_OUTER_BOUND	FCOTBN
FACE_SHAPE_REPRESENTATION	FCSHRP
FACE_SHAPE_REPRESENTATION_RELATIONSHIP	FSRR
FACE_SURFACE	FCSRF

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
FACETED_BREP	FCTBR
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
GEOMETRIC_ALIGNMENT	GMTALG
GEOMETRIC_INTERSECTION	GMTINT
GEOMETRIC_REPRESENTATION_CONTEXT	GMRPCN
GEOMETRIC_REPRESENTATION_ITEM	GMRPIT
GEOMETRIC_TOLERANCE	GMTTLR
GEOMETRIC_TOLERANCE_RELATIONSHIP	GMTLRL
GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE	GTWDR
GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT	GTWDU
GROUP	GROUP
GROUP_ASSIGNMENT	GRPASS
GROUP_RELATIONSHIP	GRPRLT
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT	GC
GLOBAL_UNIT_ASSIGNED_CONTEXT	GUAC
HOLE_BOTTOM	HLBTT
HYPERBOLA	HYPRBL
ID_ATTRIBUTE	IDATT
INSTANCED_FEATURE	INSFTR
LENGTH_MEASURE_WITH_UNIT	LMWU
LENGTH_UNIT	LNGUNT

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
LIMITS_AND_FITS	LMANFT
LINE	LINE
LINE_PROFILE_TOLERANCE	LNP0
LINEAR_PROFILE	LNRPRF
LOCAL_TIME	LCLTM
LOCATION_SHAPE_REPRESENTATION	LCSHRP
LOOP	LOOP
LOT_EFFECTIVITY	LTEFF
MANIFOLD_SOLID_BREP	MNSLBR
MAPPED_ITEM	MPPITM
MARKING	MRKNG
MASS_MEASURE_WITH_UNIT	MMWU
MASS_UNIT	MSSUNT
MEASURE_QUALIFICATION	MSRQLF
MEASURE_REPRESENTATION_ITEM	MSRPIT
MEASURE_WITH_UNIT	MSWTUN
MEASUREMENT_LOCATION	MSRLCT
MODIFIED_GEOMETRIC_TOLERANCE	MDGMTL
MODIFIED_PATTERN	MDFPTT
NAME_ATTRIBUTE	NMATT
NAMED_UNIT	NMDUNT
NGON_CLOSED_PROFILE	NGCLPR
NGON_SHAPE_REPRESENTATION	NGSHRP
OBJECT_ROLE	OBJRL
OPEN_PATH_PROFILE	OPPTPR
OPEN_SHELL	OPNSHL
ORGANIZATION	ORGNZT
ORGANIZATION_ASSIGNMENT	ORGASS
ORGANIZATION_ROLE	ORGRL
ORGANIZATIONAL_ADDRESS	ORGADD

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
ORGANIZATIONAL_PROJECT	ORGPRJ
ORIENTED_CLOSED_SHELL	ORCLSH
ORIENTED_EDGE	ORNEDG
ORIENTED_FACE	ORNFC
ORIENTED_OPEN_SHELL	OROPSH
ORIENTED_PATH	ORNPTH
ORIENTED_SURFACE	ORNSRF
OUTER_ROUND	OTRRND
OUTSIDE_PROFILE	OTSPRF
PARABOLA	PRBL
PARALLEL_OFFSET	PRLOFF
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
PERPENDICULAR_TO	PRPT
PERPENDICULARITY_TOLERANCE	PRPTLR
PERSON	PERSON
PERSON_AND_ORGANIZATION	PRANOR
PERSON_AND_ORGANIZATION_ASSIGNMENT	PAOA
PERSON_AND_ORGANIZATION_ROLE	PAOR
PERSONAL_ADDRESS	PRSADD
PLACEMENT	PLCMNT
PLACED_DATUM_TARGET_FEATURE	PDT0
PLANAR_SHAPE_REPRESENTATION	PLSHRP

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
PLANE	PLANE
PLANE_ANGLE_MEASURE_WITH_UNIT	PAMWU
PLANE_ANGLE_UNIT	PLANUN
PLUS_MINUS_TOLERANCE	PLMNTL
POCKET	POCKET
POCKET_BOTTOM	PCKBTT
POINT	POINT
POLY_LOOP	PLYLP
POLYLINE	PLYLN
POSITION_TOLERANCE	PSTTLR
PRECISION_QUALIFIER	PRCQLF
PRODUCT	PRDCT
PRODUCT_CONTEXT	PRDCNT
PRODUCT_DEFINITION	PRDDFN
PRODUCT_DEFINITION_CONTEXT	PRDFCN
PRODUCT_DEFINITION_FORMATION	PRDFFR
PRODUCT_DEFINITION_SHAPE	PRDFSH
PROFILE_FLOOR	PRFFLR
PROJECTED_ZONE_DEFINITION	PRZNDF
PROPERTY_DEFINITION	PRPDFN
PROPERTY_DEFINITION_RELATIONSHIP	PRDFR
PROPERTY_DEFINITION_REPRESENTATION	PRDFRP
PROTRUSION	PRTRSN
QUALIFIED_REPRESENTATION_ITEM	QLRPIT
QUASI_UNIFORM_CURVE	QSUNCR
QUASI_UNIFORM_SURFACE	QSUNSR
RATIO_MEASURE_WITH_UNIT	RMWU
RATIO_UNIT	RTUNT
RATIONAL_B_SPLINE_CURVE	RBSC
RATIONAL_B_SPLINE_SURFACE	RBSS

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
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
RESOURCE_PROPERTY	RSRPRP
RESOURCE_REQUIREMENT_TYPE	RSRQTY
REVOLVED_PROFILE	RVLPRF
RIB_TOP	RBTP
RIB_TOP_FLOOR	RBTPFL
ROLE_ASSOCIATION	RLASS
ROUND_HOLE	RNDHL
ROUNDED_END	RNDEND
ROUNDED_U_PROFILE	RNUPR
ROUNDNESS_TOLERANCE	RNDTLR
RUNOUT_ZONE_DEFINITION	RNZNDF
RUNOUT_ZONE_ORIENTATION	RNZNOR
RUNOUT_ZONE_ORIENTATION_REFERENCE_DIRECTION	RZORD
SECURITY_CLASSIFICATION	SCRCLS
SECURITY_CLASSIFICATION_ASSIGNMENT	SCCLAS
SECURITY_CLASSIFICATION_LEVEL	SCCLLV
SERIAL_NUMBERED_EFFECTIVITY	SRNMEF
SHAPE_ASPECT	SHPASP
SHAPE_ASPECT_DERIVING_RELATIONSHIP	SADR
SHAPE_ASPECT_RELATIONSHIP	SHASRL

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
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_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
SURFACE_OF_LINEAR_EXTRUSION	SL
SURFACE_OF_REVOLUTION	SROFRV
SURFACE_PROFILE_TOLERANCE	SRPRTL
SWEPT_SURFACE	SWPSRF
SYMMETRIC_SHAPE_ASPECT	SYSHAS
SYMMETRY_TOLERANCE	SYMTLR
TANGENT	TNGNT
TAPER	TAPER
TEE_PROFILE	TPRF
THREAD	THREAD
TIME_MEASURE_WITH_UNIT	TMWU
TIME_UNIT	TMUNT

Table B.1 – AIM short names of entities (continued)

AIM Element	Short name
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
VECTOR	VECTOR
VEE_PROFILE	VPRF
VERTEX	VERTEX
VERTEX_LOOP	VRTLP
VERTEX_POINT	VRTPNT

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 'dimensional_inspection_-schema'.

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 an 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.

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(219) version (1)}

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 dimensional-inspection-schema in the open information system, the object identifier are assigned as follows:

{iso standard 10303 part(219) version (1) object (1) dimensional_inspection_schema(1)}

is assigned to the dimensional_inspection_schema (see annex A).

{iso standard 10303 part(219) version (1) object (1) dimensional_inspection_schema(0)}

is assigned to the dimensional_inspection_schema short form schema (see 5.2).

The meaning of this value is defined in ISO 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 [2]. Figure F.1 gives the basic notation. Each activity may be decomposed to provide more detail. If an activity has been decomposed, a separate figure is included.

As with any IDEF0 model, the application activity model is dependent on a particular viewpoint and purpose. The viewpoint of the application activity model is from a manufacturing engineer. The purpose of the application activity model is to clarify the context and scope of this 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.

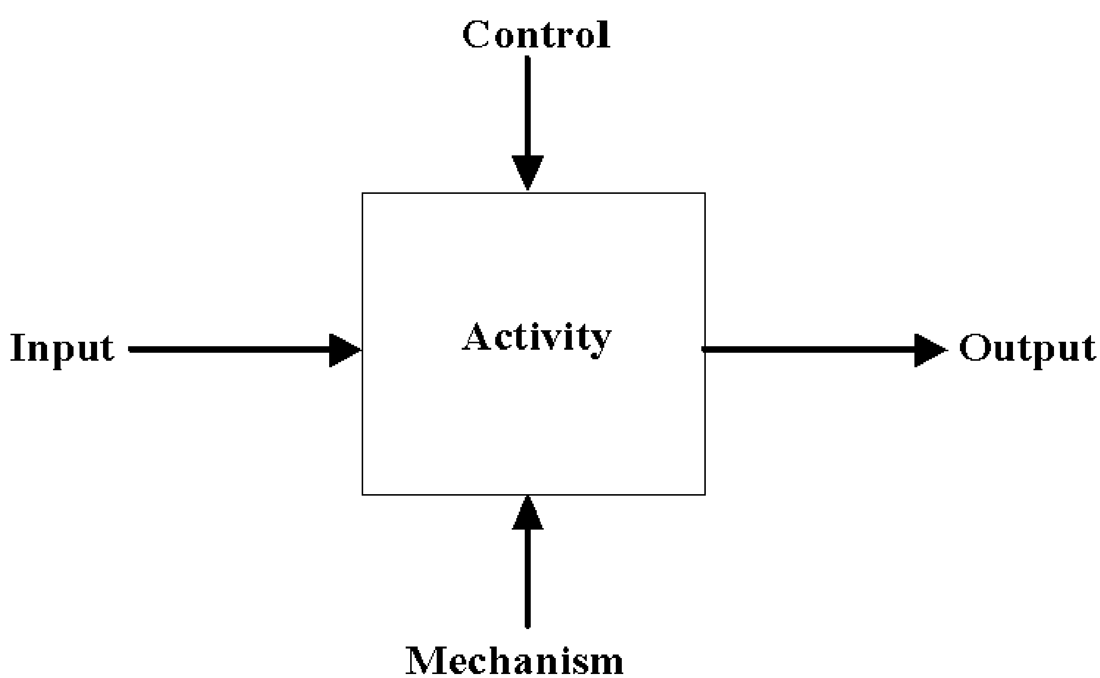


Figure F.1 — IDEF0 Basic notation

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 Accounting Software*: Software tools used for the explicit purpose of archiving and administration of dimensional inspection data.

F.1.2 Acquire DME, Tools and Fixtures*: Requisitions for commercially available equipment, fixtures and tooling as necessary. The requisition will utilize any available databases.

F.1.3 Administrative Data*: Data relating to the organizational and business aspects of a dimensional inspection operation rather than the engineering or metrology aspects.

F.1.4 Algorithm Software: Computer codes for calculating the dimensions of a feature from the measured data points that collectively represent the feature.

F.1.5 Aligned Part*: The orientation of the part with respect to the dimensional measuring equipment as a result of applying setup instructions.

F.1.6 Analysis Results Feedback: Information resulting from performing the dimensional inspection process, which can be used as a control for the planning and the execution functions of that process.

F.1.7 Analysis Results Feedback (w/FLs): Information resulting from performing the dimensional inspection process, which can be used as a control for the planning and the execution functions of that process. This information also includes feature labels.

F.1.8 Approve/Disapprove the Daft Plan (A2432)*: The approval of simulation results, a plan approval is indicated if the results pass inspection; a change request is indicated if the results do not pass inspection.

F.1.9 Archival Results of Analysis: Data, calculated quantities, and other information flowing from the *perform analysis of inspection* function and headed for the *archiving* function. This term includes raw data points, analysis results feedback, and statistical results.

F.1.10 Archived Information: All the dimensional inspection information considered worth being stored. It is important to note that the archive is not really a data stream, but a data base. Data is constantly added to the archival data base but data is occasionally fetched. The database inputs and outputs represent the schema.

F.1.11 Associate (A41): The activity by which the individual measured data points are grouped to form digital representations of part features and fitted to substitute part features.

F.1.12 Calculate (A42): The activity used to combine two or more substitute features in a relationship to generate a composite feature, when that is required for subsequent tolerance evaluation. In addition, the activity may generate a third feature from two or more features, when that is required for subsequent tolerance evaluation or for the generation of a coordinate system.

F.1.13 Calculate Statistics (A44): The task of examining evaluated raw data points and creating statistical results.

F.1.14 Change Request*: A request made by an inspection plan validator to change a draft inspection plan.

F.1.15 Constructed Feature: A feature that is generated from two or more substitute features.

F.1.16 Constructed Feature Relationship: The spatial relationship(s) between the substitute features comprising the constructed feature.

F.1.17 Coordinate System (generated or default): A coordinate system that is associated with individual measured data points. The default system is the coordinate system of the inspection analysis activity, the generated system is a coordinate system created to fit the data points to the part features.

F.1.18 Coordinate System(s): Mathematical system(s) for describing the position and orientation of something in the 3D space. The coordinate systems include the machine's coordinate system and where needed, the coordinate systems for each fixture, part, sensor, carriage, rotary table, and sensor calibration artifact.

F.1.19 Data Structures*: The files, along with their specific formats, that store the data resulting from the development of a dimensional inspection plan and the execution of a dimensional inspection activity.

F.1.20 Database Software*: The software tools required to maintain administrative and archive dimensional inspection data.

F.1.21 Database System*: The software tools required by dimensional inspection planning to maintain information on inspection planning systems, fixture synthesis system, inspection simulation systems, inspection program library, and inspection plan validation.

F.1.22 Decompose Selected Tolerances and Features (A22)*: The dimensional inspection planning activity in which part surfaces are identified for measurements, the result of which may be parameterized for comparison against specified tolerances and features.

F.1.23 Decomposed Tolerances*: Measurable surfaces which are decomposed from a tolerance.

F.1.24 Decomposed Tolerances with points to measure*: Measurable surfaces which are decomposed from a tolerance each of which includes a series of points to be measured.

F.1.25 Derived Coordinate System: The coordinate system considered appropriate for a particular inspection. Choosing a good coordinate system reduces uncertainty in measurements and results. As the first coordinate system used is generally not the best one, a derived coordinate system is generated that suits the inspection task better.

F.1.26 Determine Clamping Devices (A2413)*: The task of determining the types of clamps needed to position the part in the fixture, or on the dimensional measuring equipment.

F.1.27 Determine Fixture Orientation (A2412)*: The task of determining the orientation(s) of the fixture with respect to the part and dimensional measuring equipment.

F.1.28 Determine Inspection Sequence (A2422)*: The task of determining the order in which surfaces and individual data points are measured during an inspection operation.

F.1.29 Determine Measurement Positions (A2421)*: The dimensional inspection planning activity that generates expected data points to be measured from decomposed tolerances

F.1.30 Determine Part Orientation (A2411)*: The task of determining the orientation(s) of the part with respect to the dimensional measuring equipment. Each orientation is often chosen to optimize the number of features that can be measured.

F.1.31 Determine Process Parameters (A2423)*: The activity makes the discussion on type of process parameters that are required for the draft process plan. The process parameter can either be a machine parameter or a sensor parameter. Some examples are machine speed, measuring speed, machine operation mode, sensor orientation, sensor size, etc.

F.1.32 Determine Setup (A241)*: The task of determining the actual configuration of a part to be inspected, the fixtures to hold it, and the chosen orientation of this assembly with respect to the dimensional measuring equipment.

F.1.33 Develop Inspection Process Plan (A24)*: The summation of activities that evaluate the design data, such as engineering product design data (PDD), for a machined part and determine the inspection requirements and associated data required to produce the part. This activity begins upon the release of the engineering PDD to the manufacturing planning department and ends with the delivery of the inspection package to production control. Both engineering PDD and process plan change requests may be generated as a result of the inspection strategy evaluation. This task includes the verification of the inspection information and the administration of the package and associated data. This activity includes determining the inspection requirements, processes and equipment to be utilized. The gaging and measurement resources are selected for the characteristics identified for inspection. This information is organized into the inspection package (NOTE process plan?), verified and released for part manufacture.

F.1.34 Dimensional Inspection Plan (w/Fls): A document of detailed instructions to inspect parts using design model data as input. Contents of the document may include inspection scope, accuracy requirements, specification of appropriate dimensional measurement equipment, part set-up, step-by-step inspection instructions, and plan validation.

F.1.35 Dimensional Inspection Standards: Standards that deal with dimensional inspection methods such as ANSI B89.3.2 Dimensional Measuring Method, and ANSI/ASQC E-2 Guide to Inspection Planning.

F.1.36 Dimensions & Tolerances: Dimensions are defined in this part of ISO 10303 as *a numerical value expressed in appropriate units of measurement and indicated graphically on technical drawings with lines, symbols, and notes*. Dimensions are classified according to the following types: functional dimension; nonfunctional dimension; auxiliary dimension. Tolerances are the permitted variation of dimension or other geometrical parameter. Operationally, it is a data structure specifying a tolerance type, tolerance value, applicable tolerance modifiers (material condition, projected zone, etc.), the toleranced feature (either composite feature or simple feature) and, for certain tolerance types, the datum reference frame.

F.1.37 DME Program (w/Fls)*: The part of the dimensional inspection containing the software code describing the detailed sequence of operations to be performed by automated DME.

F.1.38 DME Program Library*: A set of commonly used software routines that may be called as part of a DME program.

F.1.39 DME, Tools and Fixtures*: The tools, jigs, fixtures and gages required in the manufacturing of a product through the specification of a recognizable reference number.

F.1.40 Draft Inspection Plan*: An inspection plan developed by an inspection planner subject to change by an inspection validator.

F.1.41 Engineering Product Definition Data, with Feature Labels (Fls): The subset of released engineering product definition data (PDD) relevant to NC processing and a superset of change descriptions. It is assumed that the access to this information is managed by a vault function. It includes engineering bill of material (EBOM), shape, interface requirements, change description, geometry model, tolerances, surface finish, material requirements, and specified stock material. The data element is cast as a control to the A0 activities because the process planner does not directly change the engineering PDD.

F.1.42 Environmental Conditions*: The ambient temperature, pressure, and humidity under which an inspection process is carried out.

F.1.43 Equipment Programming Required Resources*: The programming language, such as Dimensional Measuring Interface Standard (DMIS), in which the dimensional measurement process plan is developed.

F.1.44 Equipment Programming Language Standard*: The programming language, such as Dimensional Measuring Interface Standard (DMIS), in which the dimensional measurement process plan is developed.

F.1.45 Evaluate (A43): The activity whereby dimensions and other parameters of a constructed feature or profile are compared with the tolerance zones for that feature.

F.1.46 Execute Inspection (A3)*: The activity whereby a dimensional inspection plan is carried out by the operator and the equipment to produce dimensional inspection results, including at least raw data points and possibly some preliminary results of analysis.

F.1.47 Facility Requirement*: The information regarding the specific materials and supplies for a facility necessary for requisition.

F.1.48 Feature Labels: Identification of the feature being inspected.

F.1.49 Fixture Synthesis System*: A system that designs fixtures for an inspection process.

F.1.50 Generate Data Structures (A244)*: The activity of creating files, along with their specific formats, that store the data resulting from the development of a dimensional inspection plan and the execution of a dimensional inspection activity.

F.1.51 Generate Machine and Sensor Motion Simulation (A2431)*: The activity of creating a time sequence of the calculated positions of a probe with respect to the part, the fixtures, and other components of the dimensional measuring equipment for a draft inspection plan. The purpose of the simulation is to reveal errors in the draft inspection plan.

F.1.52 Generate Output Data (A33)*: The activity takes the resulting data points from measuring the part and creates output data of raw data points and associated feature labels.

F.1.53 Generate Support Data (A25)*: The activity for creating specifications of fixtures, probes and probe accessories, clamps, machines and other resources needed for inspection.

F.1.54 Human Resources*: The operator(s) of a dimensional inspection activity or process.

F.1.55 Identify Inspection Accuracy Requirements (A21)*: The dimensional inspection planning activity of developing accuracy requirements for the machine(s) to be performing a dimensional inspection based on the sizes of the specified tolerance zones

F.1.56 Input Measurement Parameters (A314)*: The inspection setup activity for inputing parameters that describe the operation of a measuring machine, such as the probe speed or table speeds, or on a touch probe, the force required to trigger it.

F.1.57 Inspected Part: A discrete, solid object or assembly, which has dimensional requirements in its specification, has been inspected, and has either passed or failed inspection.

F.1.58 Inspection Data Analysis Programs: Software programs used to evaluate dimensions and other parameters of a constructed feature or profile to compared with the tolerance zones for that feature.

F.1.59 Inspection Order*: The authorization to begin the process of inspection once the scheduling requirements are met. This is a component of the manufacturing process plan. It enables the planning authorization.

F.1.60 Inspection Plan Validator*: A person or a software function that checks the validity of a draft inspection plan and makes the decision to approve, disapprove, or request changes to the draft inspection plan.

F.1.61 Inspection Planning System*: A computer system used to assist an inspection planner to generate inspection plans.

F.1.62 Inspection Process Parameters*: A factor that determines how an inspection process should be performed. The process parameter can either be a machine parameter or a sensor parameter. Some examples are machine speed, measuring speed, machine operation mode, sensor orientation, sensor size, etc.

F.1.63 Inspection Request*: The authorization to begin the process of inspection, once the scheduling requirements are met.

F.1.64 Inspection Sequence*: The order in which surfaces and individual data points are measured during an inspection operation.

F.1.65 Inspection Simulation System*: A system that generates simulated machine and probe motion relative to the part for verification.

F.1.66 Inspection Type*: Two types of inspection are possible: in-process inspection and final part inspection. In-process inspection includes on-machine inspection and between-workstation inspection. In-process inspection is for monitoring, correcting, and controlling manufacturing processes. Final inspection determines conformance of parts to design specification. A *global* inspection of a part can also be performed to check the completeness of all the machined features before the in-process or the final inspection.

F.1.67 Inspection Uncertainty Requirements*: The uncertainties required in the result for each measured dimensional feature taking into account the required tolerance for each feature. The uncertainty requirement is usually significantly smaller than the tolerance.

F.1.68 Knowledge Based System*: Set of criteria for determining uncertainty requirements for dimensional measurements to test adequately whether parts pass or fail different types of dimensional and geometric tolerances

F.1.69 Labeled Coordinate System: coordinate system associated with measurement points acquired for a specific labeled feature

F.1.70 List of Features*: Identification of the feature from the as designed product model.

F.1.71 List of Tolerances*: Identification of the tolerances from the as designed product model.

F.1.72 Load Program (A313)*: The inspection setup activity for loading the program to be used for dimensional inspection

F.1.73 Manage Dimensional Inspection (A0): Activity that includes all aspects of dimensional inspection, in particular, administration, archiving, planning, execution, and analysis.

F.1.74 Manufacturing Knowledge*: Knowledge concerning the kinds of information that can be produced from a manufacturing process.

F.1.75 Manufacturing Process Plan: All of the data on the design for a part, including related tolerances and nominals, nominal part data, design tolerances, and feature labels, as well as all of the data on how the part was manufactured, including particular machine tools and the order in which they are used.

F.1.76 Measure Part (Activity A32)*: The actual task of measuring the part, once part setup has been completed, the results of this task are the raw data points to be evaluated.

F.1.77 Measurement Complete Status*: A status code indicated the measure part activity is or is not yet complete.

F.1.78 Measurement Parameters*: Parameters that describe the operation of a measuring machine, such as the probe speed or table speeds, or on a touch probe, the force required to trigger it.

F.1.79 Part: A discrete, solid object or assembly, having dimensional requirements in its specification that is to be inspected.

F.1.80 Part Orientations*: The orientation(s) of the part with respect to the dimensional measuring equipment. Each orientation is often chosen to optimize the number of features that can be measured.

F.1.81 Part Ready Status*: A status code to indicate if tooling and sensors have successfully been retrieved, the part is ready for inspection, and the program may now be loaded.

F.1.82 Perform Analysis of Inspection (A4): The activity whereby the measured raw data points resulting from the *execute inspection* activity serve as input for the description of features and the calculation of dimensions and parameters, which are then compared with dimensional tolerances.

F.1.83 Perform Calibration & Qualification (A315)*: Dimensional inspection execution activity whereby the dimensional inspection system is calibrated and its measurement uncertainty is characterized in preparation for a dimensional measurement

F.1.84 Perform Inspection Administration and Archiving (A1)*: The activity of organizational and business aspects of a dimensional inspection operation rather than the engineering or metrology aspects.

F.1.85 Perform Inspection Planning (A2)*: The activity that provides a dimensional inspection plan, which includes a DME program for automated dimensional measuring equipment.

F.1.86 Plan Approval*: A flag output from the inspection planning operation indicating that the inspection plan has been validated and may now be executed and archived.

F.1.87 Prepare Part for Inspection (A312)*: The activity whereby the tools and sensors have been made ready and the part is now placed in the fixture or other inspection equipment per the inspection plan and inspection request. The part is made ready for the program run.

F.1.88 Program Loaded Status*: A status code to indicate that the program is now loaded and input measurement parameters may now be loaded.

F.1.89 Purchase Orders for General Equipment and Supplies*: Dimensional inspection administrative activities determines if any equipment needs to be requisitioned for the inspection of the part. Purchase orders are created to obtain necessary equipment needed for the part inspecting.

F.1.90 Raw Data Points: The collection of numbers resulting from the *execute inspection* activity.

F.1.91 Regulations and Policies*: Rules to regulate the inspection plan validation and approval process.

F.1.92 Related Tolerances and Nominals: Tolerances and nominal dimensions associated with a particular designed feature against which are compared the dimensions and geometric forms of substitute or constructed features derived from measurement.

F.1.93 Report of Environmental Conditions*: A record of ambient temperature, pressure, and humidity, as well as part and machine temperatures, date, and time, and other related factors. Also, a document containing all appropriate statements about the environmental conditions.

F.1.94 Report of Environmental Conditions*: A report of the ambient temperature, pressure, and humidity under which an inspection process is carried out.

F.1.95 Results from Execution (w/Fls): The collective output from the execution of dimensional inspection that includes specifically setup results, a report of environmental conditions, feature labels, and raw data points.

F.1.96 Retrieve Identified Tooling & Sensors (A311)*: Action of reviewing the dimensional inspection program and inspection request, and preparing tools and sensors for a part inspecton.

F.1.97 Retrieve Product Model (A211)*: Action of reading or inputting the engineering product definition data for a part.

F.1.98 Scheduling Software*: Software tools used for the explicit purpose of scheduling dimensional inspection activities. It contains equipment routing, together with machine usage, manpower skills and time, tooling, materials, etc. for each station in the routing

F.1.99 Select Data Analysis Functions (A233)*: Selection of mathematical functions and procedures used to analyze measured data points to evaluate actual dimensions and tolerances of the parts.

EXMAPLE: regression analysis

F.1.100 Select DME, Tools, and Fixtures (A231)*: The activity of selecting individual tools, jigs, fixtures and gages required in the manufacturing of a product through the specification of a recognizable reference number. The selection process will utilize, if available, any databases. Requisitions will be initiated for commercially available equipment, fixtures and tooling as necessary.

F.1.101 Select DME, Tools, Fixtures, and Functions (A23)*: Additionally this is the activity of selecting functions and procedures used to analyze measured data points to evaluate actual dimensions and tolerances of the parts

F.1.102 Select Features for Global Inspection (A213)*: A *global* inspection of a part can also be performed to check the completeness of all the machined features before the in-process or the final inspection.

F.1.103 Select Sensors (A232)*: The activity of selecting individual sensor used in the dimensional inspection measurement. The selection process will utilize, if available, any databases. Requisitions will be initiated for commercially available sensors as necessary.

F.1.104 Select Tolerances to be Inspected (A212)*: Tolerance that is selected by the inspection planner as needing to be inspected.

F.1.105 Selected Data Analysis Functions: The Selected of mathematical functions and procedures used to analyze measured data points to evaluate actual dimensions and tolerances of the parts.

F.1.106 Selected Features: The representation form chosen for the part that will be the product of the process plan.

F.1.107 Selected Tolerances: Tolerance that is selected by the inspection planner as needing to be inspected.

F.1.108 Selected Tolerances and Associated Features: Tolerances selected by the inspection planner that reference an inspection feature. The tolerances have a relationship to the representation of features.

F.1.109 Selections of DME, Tools and Fixtures*: The identification of individual tools, jigs, fixtures and gages required in the manufacturing of a product through the specification of a recognizable reference number. The selection process will utilize, if available, any databases. Requisitions will be initiated for commercially available equipment, fixtures and tooling as necessary.

F.1.110 Selections of DME, Tools, Fixtures and Functions*: Additionally identifies functions and procedures used to analyze measured data points to evaluate actual dimensions and tolerances of the parts

F.1.111 Selections of Sensors*: The identification of individual sensor used in the dimensional inspection measurement. The selection process will utilize, if available, any databases. Requisitions will be initiated for commercially available sensors as necessary.

F.1.112 Sensor Calibration Data*: The dimensional information describing a sensor calibration artifact such as coordinate systems, sphere diameter, and other information depending on the sensor technology. This control may not be required for all dimensional measurement systems.

F.1.113 Sensor Configuration*: The description of a specific sensor or a combination of sensors required to accomplish a dimensional inspection task.

F.1.114 Setup Complete Status*: Assertion that states that the setup has been completed and that the system is ready for the next step.

F.1.151 Setup for Inspection (A31)*: The activity of preparing a measuring instrument or DME to perform dimensional inspection. The activity may include retrieving sensors, fixturing the part, fixturing a standard part, and calibrating the instrument.

F.1.116 Setup Results*: The results of measurement checks of the part and sensor setup on the DME as well as the results of recalibration of the DME.

F.1.117 Simulation Result*: The result of simulating motion for dimensional inspection machines and sensors.

F.1.118 Software Tools: Software codes, usually commercially available, for enabling all aspects of the dimensional inspection operation.

F.1.119 Software Tools for Analysis: Software tools used for the explicit purpose of analysis of raw data points as a result of dimensional inspection.

F.1.120 Software Tools for Execution*: Software tools that run dimensional inspection equipment to produce dimensional inspection results, including at least raw data points and possibly some preliminary results of analysis.

F.1.121 Software Tools for Planning*: Software tools that create a dimensional inspection plan, which includes a DME program for automated dimensional measuring equipment.

F.1.122 Specification of DME, Tools and Fixture*: Documents that details the usage of DME, tools and fixtures, including technical limitations, setup procedures, and usage.

F.1.123 Specification of Special Required Resources*: Special information output from the inspection planning function requesting hardware or software resources from the administration function, which are not currently available in the database of resources for the planning function.

F.1.124 Specify Inspection Plan (Activity A242)*: Once an inspection setup has been determined this activity defines an inspection plan and creates a draft inspection plan for approval.

F.1.125 Specify Uncertainty Requirements (A214)*: Standard tooling. Any tooling that can be purchased from a tooling vendor, such as perishable tools and conventional fixtures.

F.1.126 Standards, Specifications, Policies & Procedures*: 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.1.127 Statistical Results (w/FLs): The resultant data from evaluated raw data points and inspected features.

F.1.128 Statistics Software: Software tools used to perform statistical calculations.

F.1.129 Substitute Feature (Geometry)*: An ideal geometric feature (such as a circle, cylinder, plane, cone) or a mathematically defined surface or contour to which the measurement data is fit and which may be used to replace that actual measurement data in subsequent operations.

F.1.130 Support Data*: A general term for specifications of fixtures, probes and probe accessories, clamps, machines and other resources needed for inspection.

F.1.131 Synthesis Plan (A2424)*: Dimensional inspection planning activity to integrate an inspection plan sequence with the labels of the features to be measured and with process parameters of the equipment

F.1.132 Tooling in Place Status*: A status code to indicate if tooling and sensors have successfully been retrieved and are ready for part inspection.

F.1.133 Validate and Approve the Inspection Plan (A243)*: The iterative process of debugging and simulating the inspection plan in order to test its function and correctness and the subsequent act of approving the inspection plan for use.

F.1.134 Work Holding Specifications*: The description of specific models of fixtures, their positions, the associated clamps and fasteners, and their positions and orientations required for holding the part in the chosen orientation(s).

F.2 Application activity model diagrams

The application activity model is given in figures F.2 through F.17. Activities and data flows tat are out of scope are marked with asterisks.

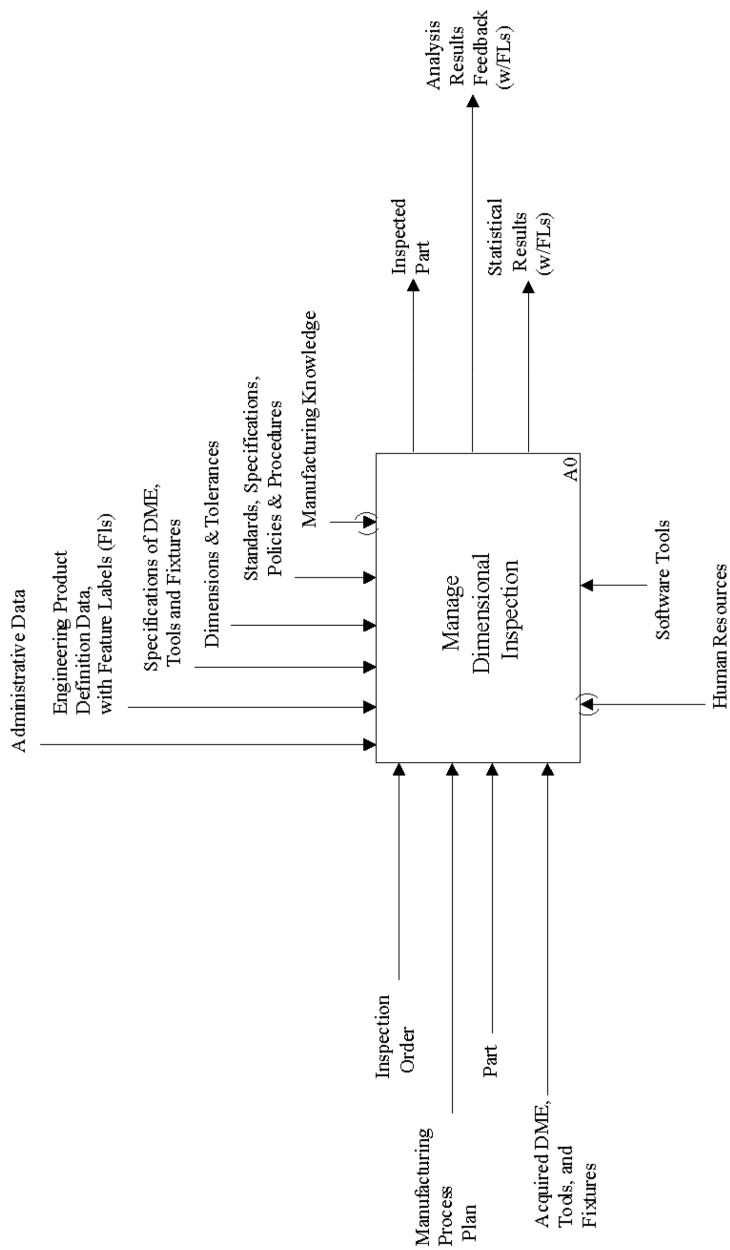


Figure F.2 - A0 Manage Dimensional Inspection

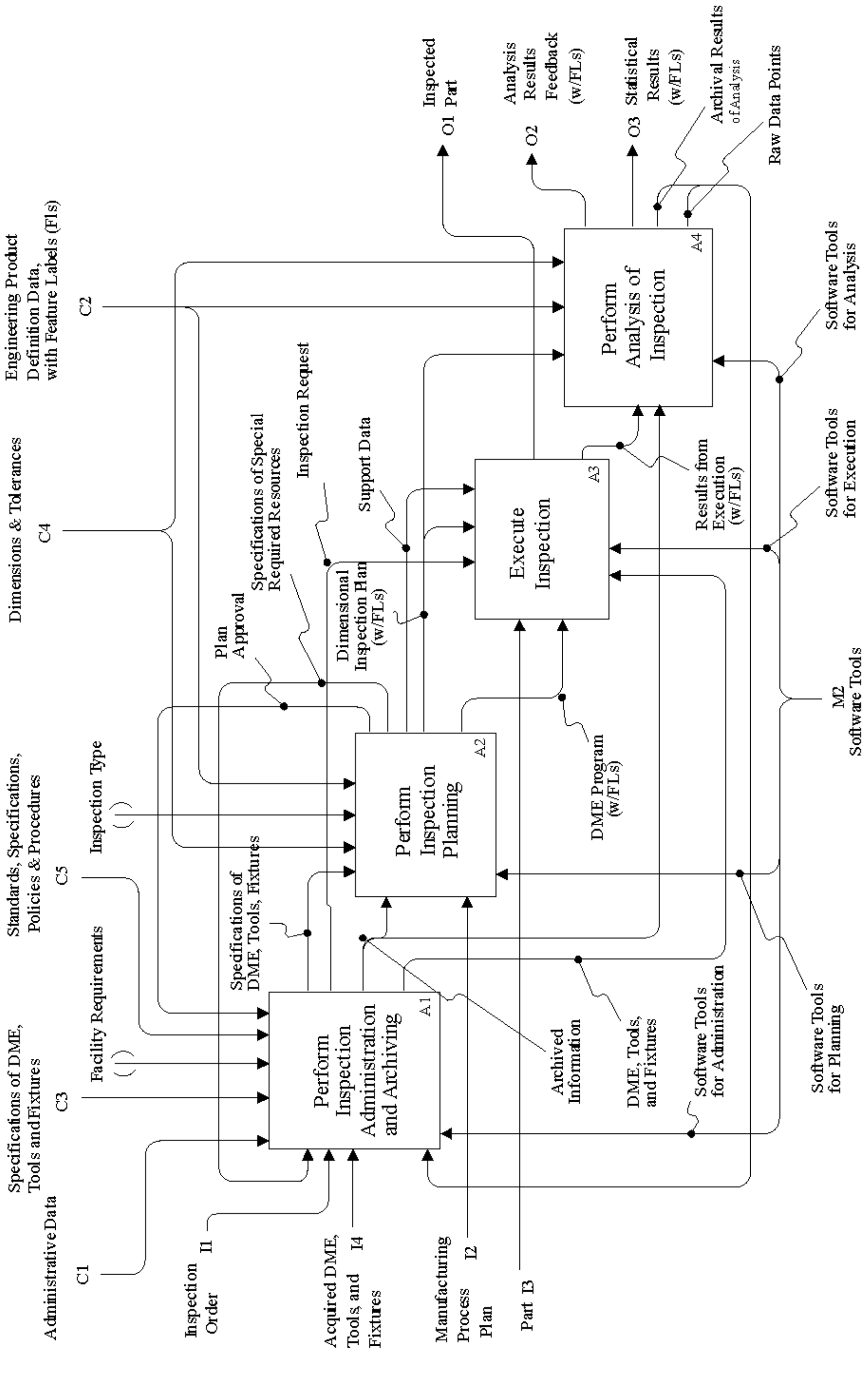


Figure F.3 - A0 Manage Dimensional Inspection

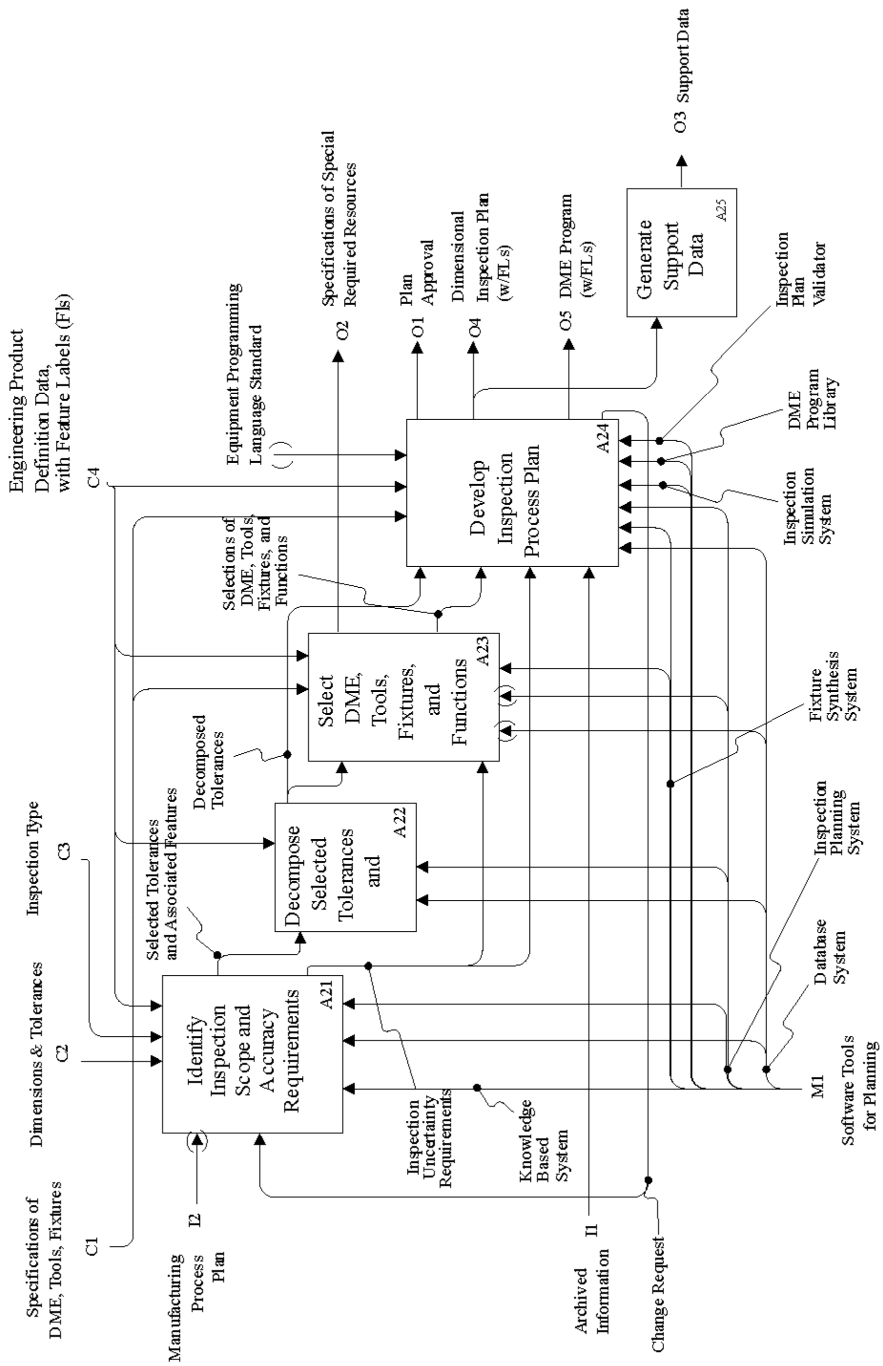


Figure F.4 - A2 Perform Inspection Planning

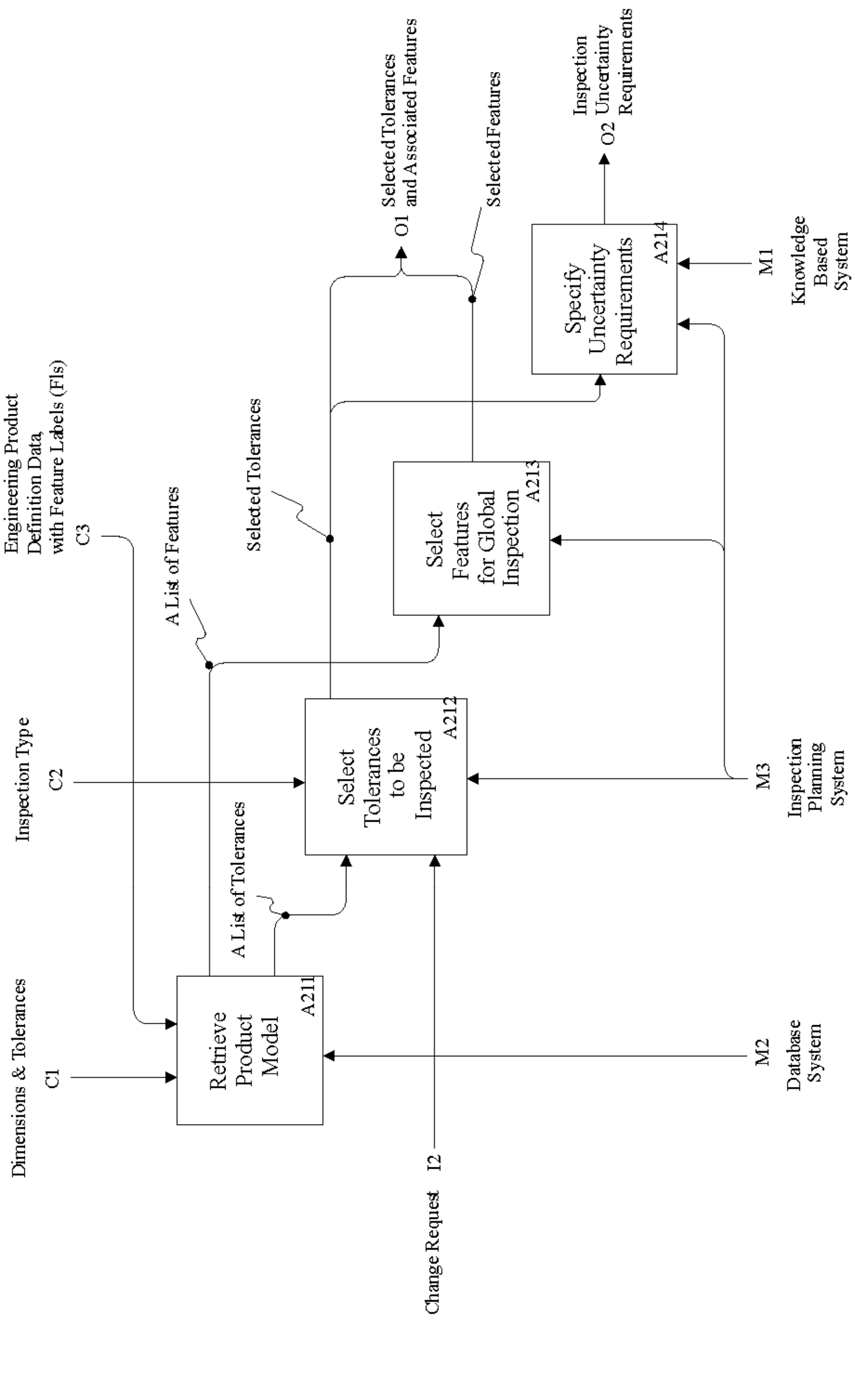


Figure F.5 - A21 Identify Inspection Scope and Accuracy Requirements

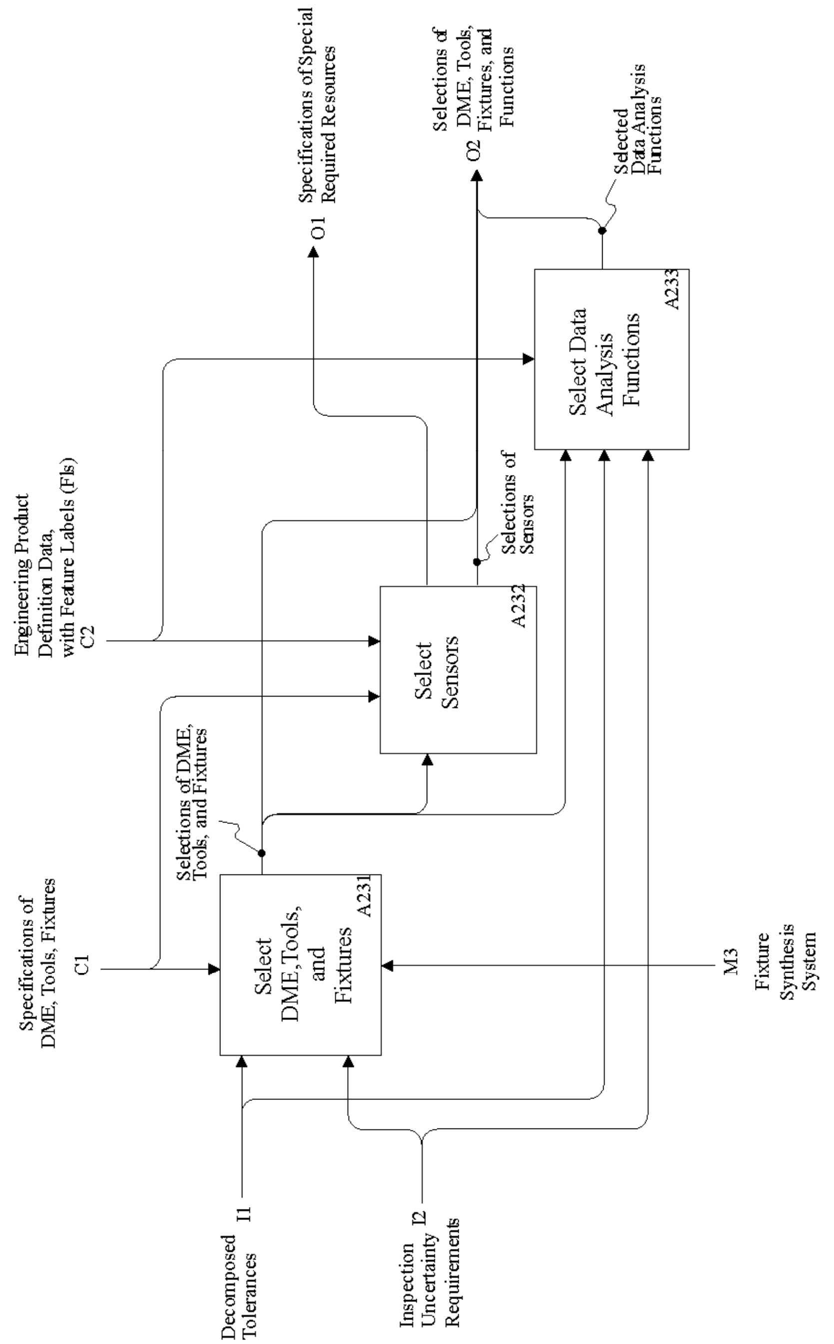


Figure F.6 - A23 Select DME, Tools, Fixtures and Functions

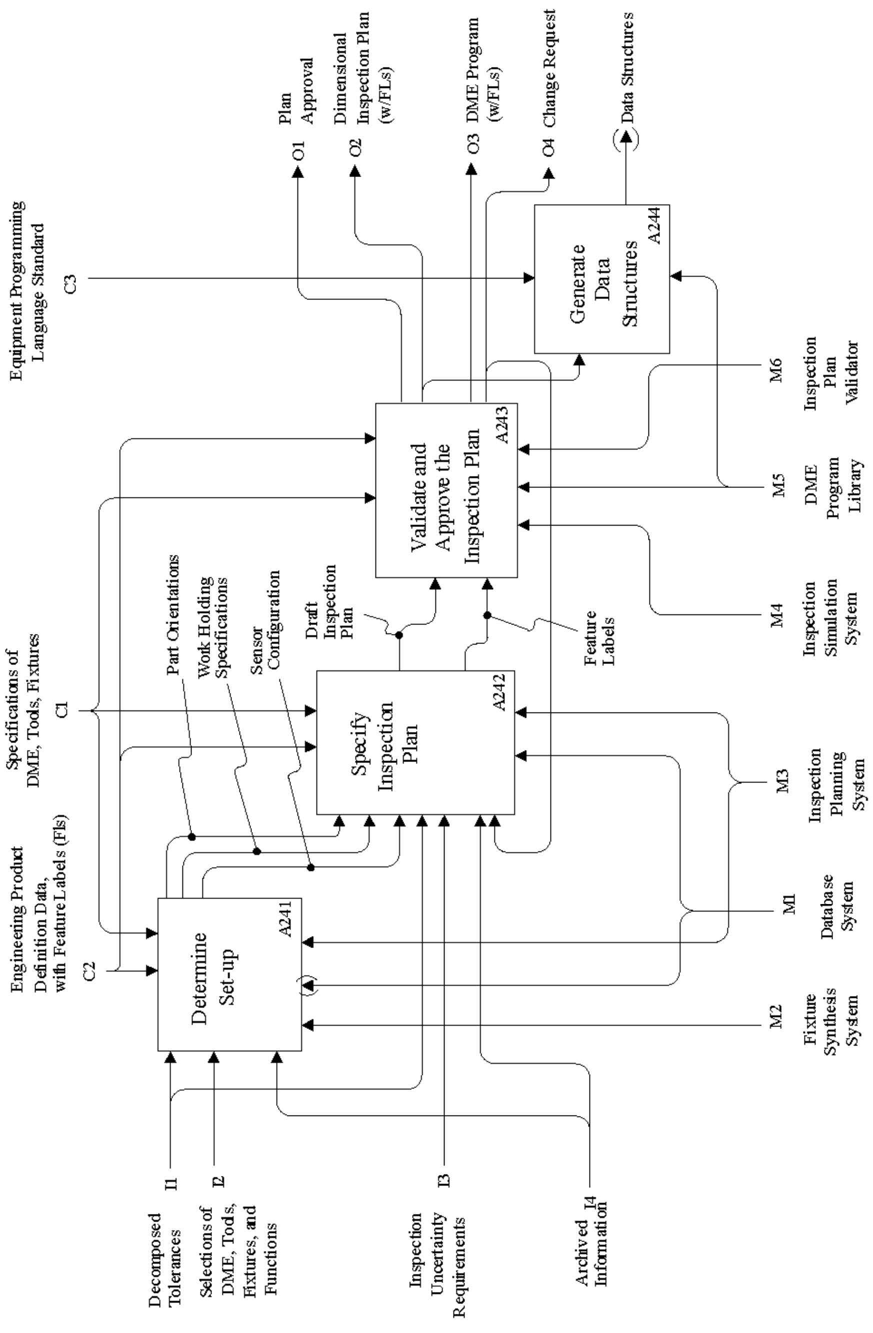


Figure F.7 - A24 Develop Inspection Process Plan

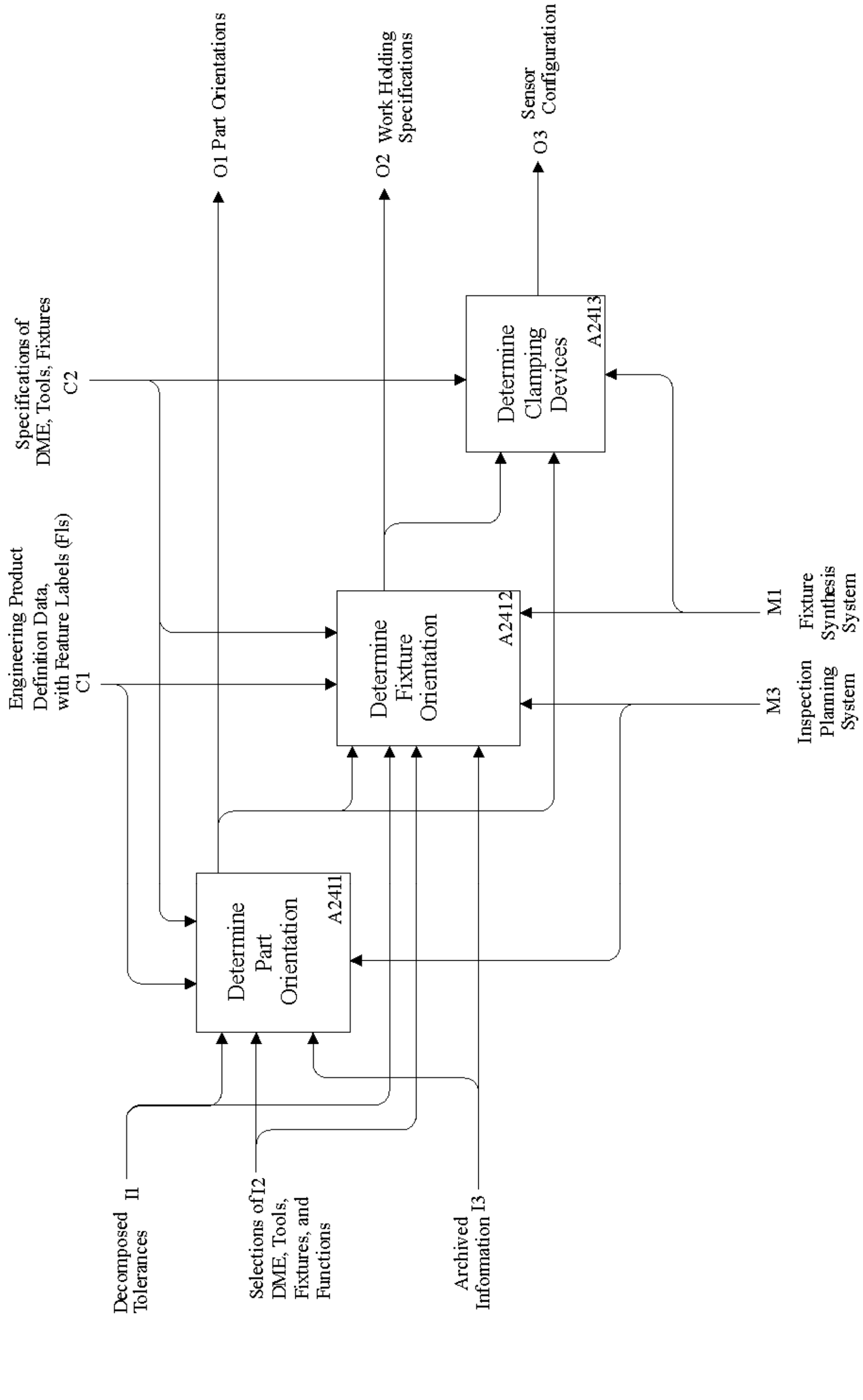


Figure F.8 - A241 Determine Setup

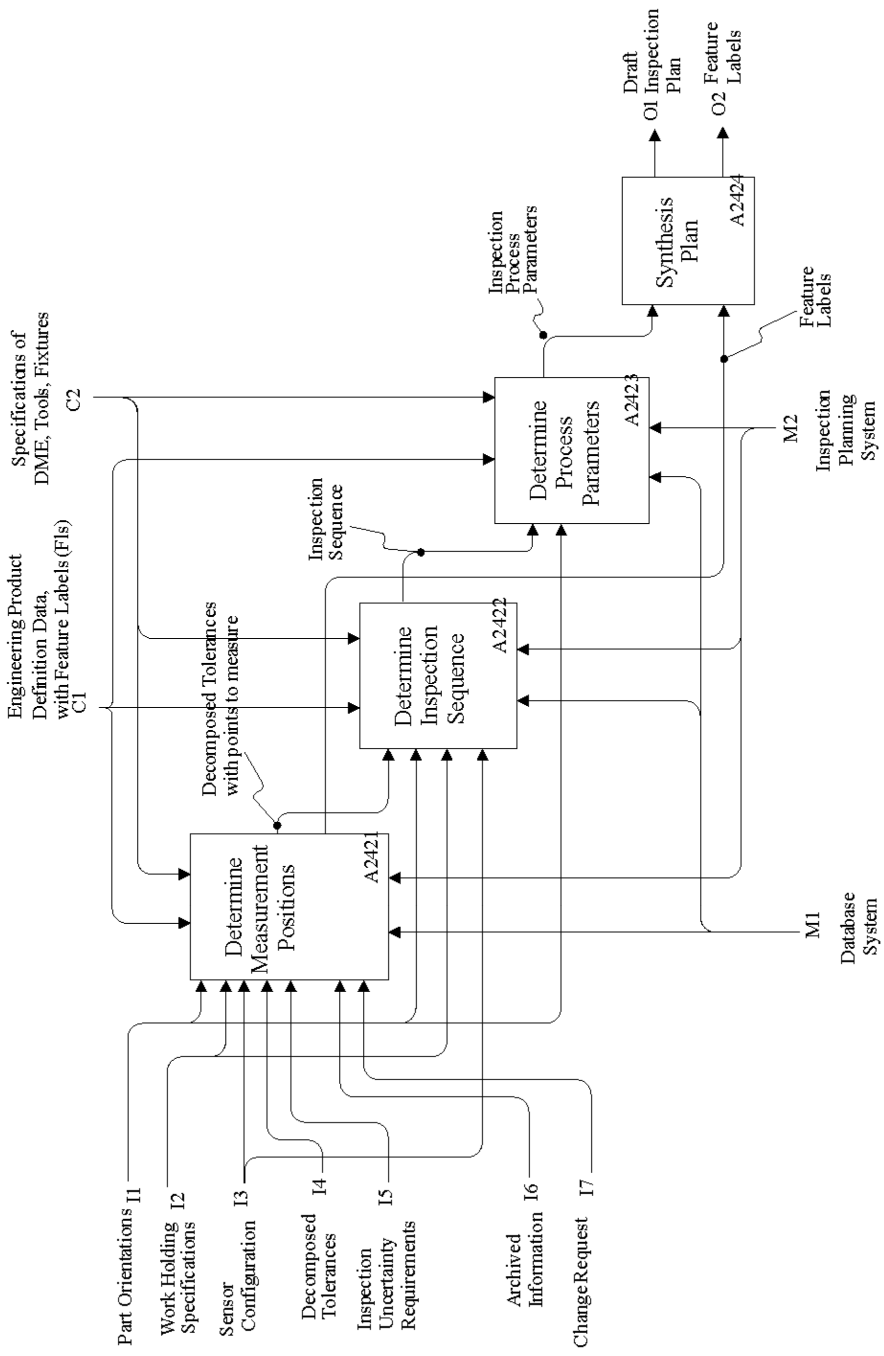


Figure F.9 - A242 Specify Inspection Plan

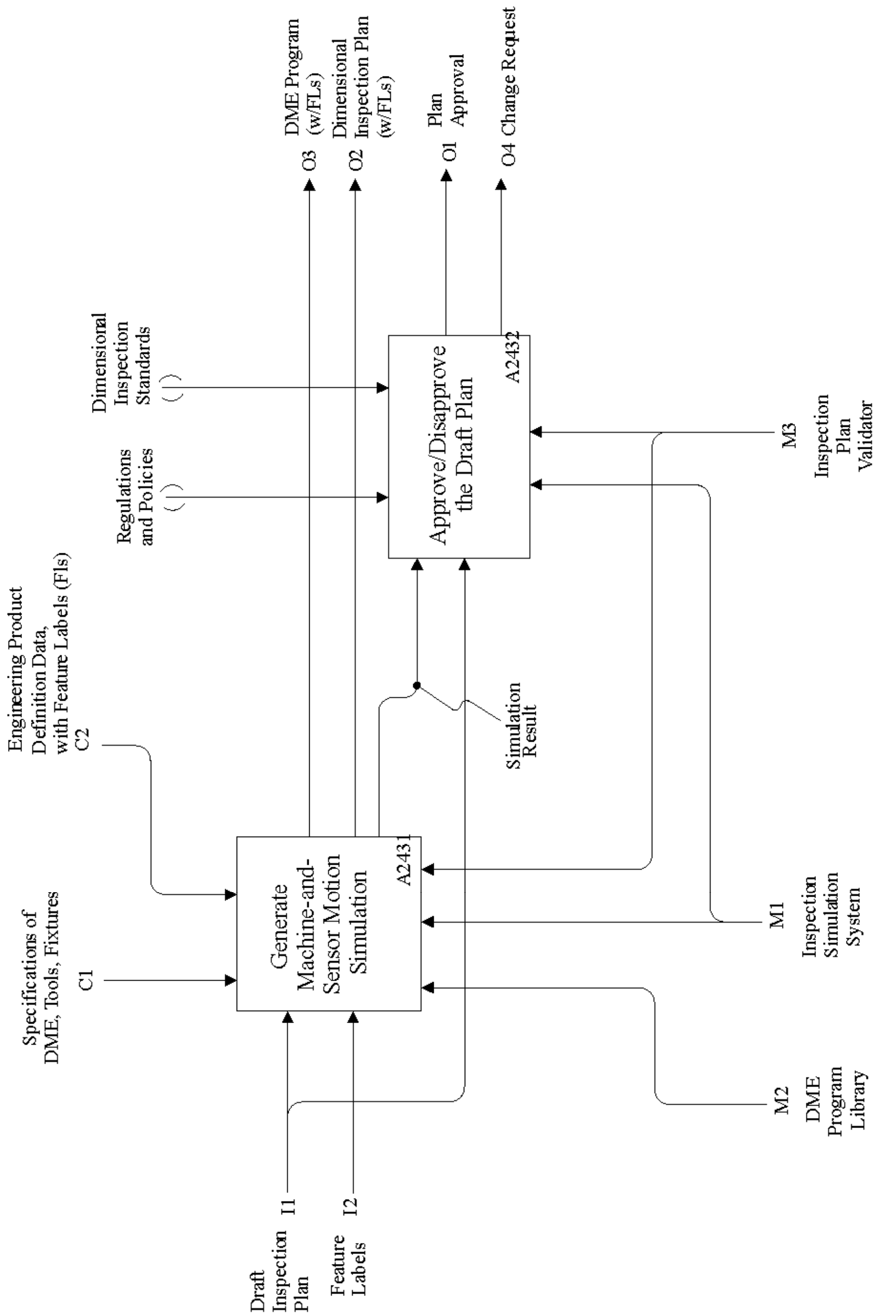


Figure F.10 - A243 Validate and Approve the Inspection Plan

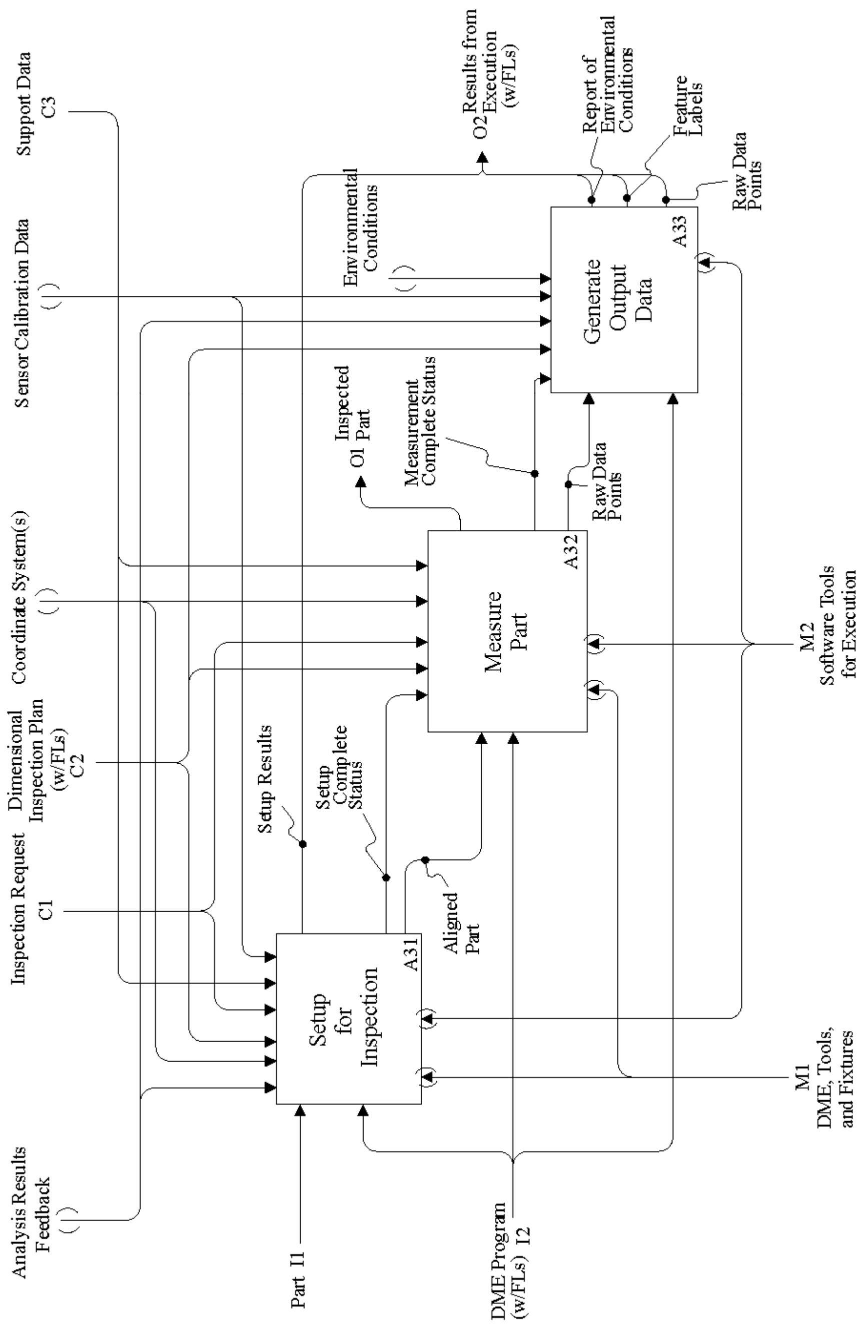


Figure F.11 - A3 Execute Inspection

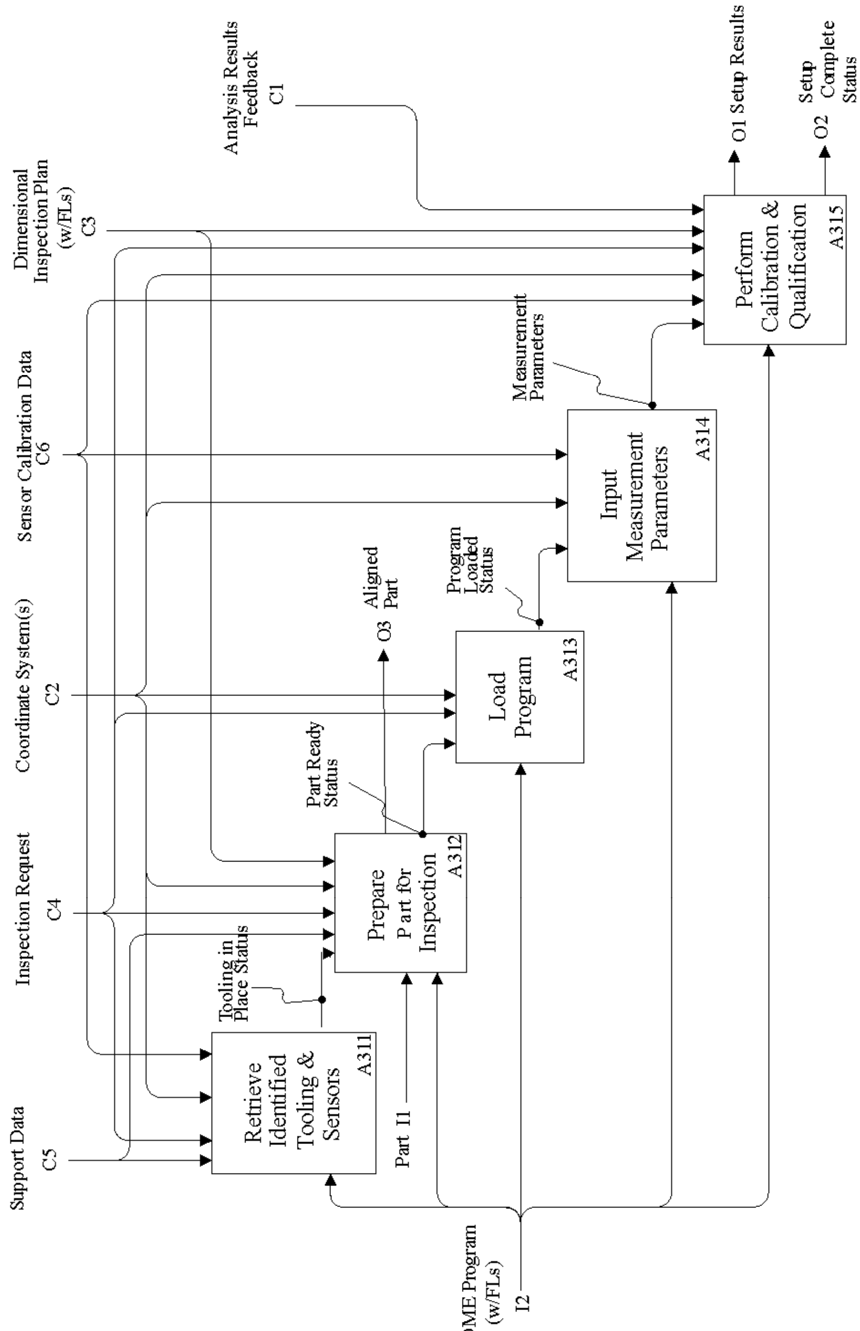


Figure F.12 - A31 Setup for Inspection

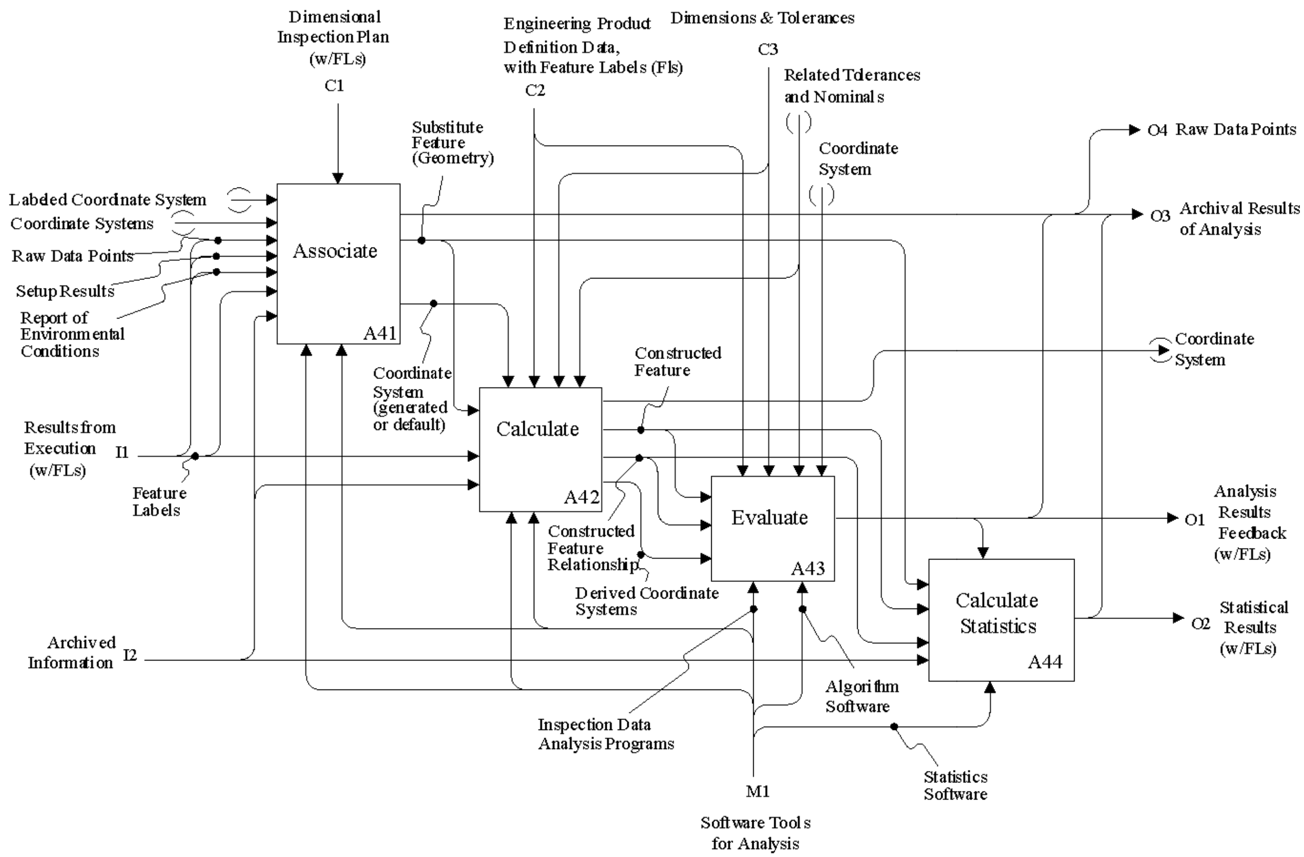


Figure F.13 - A4 Perform Analysis of Inspection

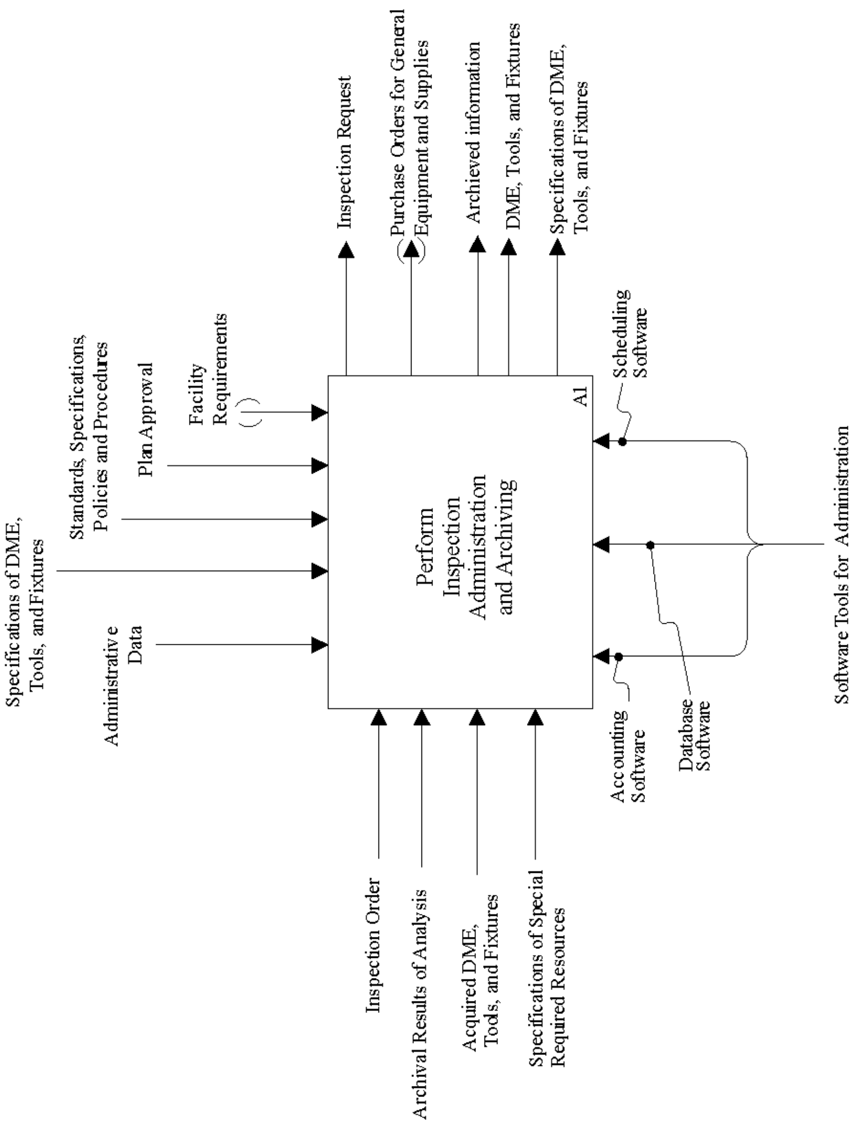


Figure F.14 - A1 Perform Inspection Administration and Archiving

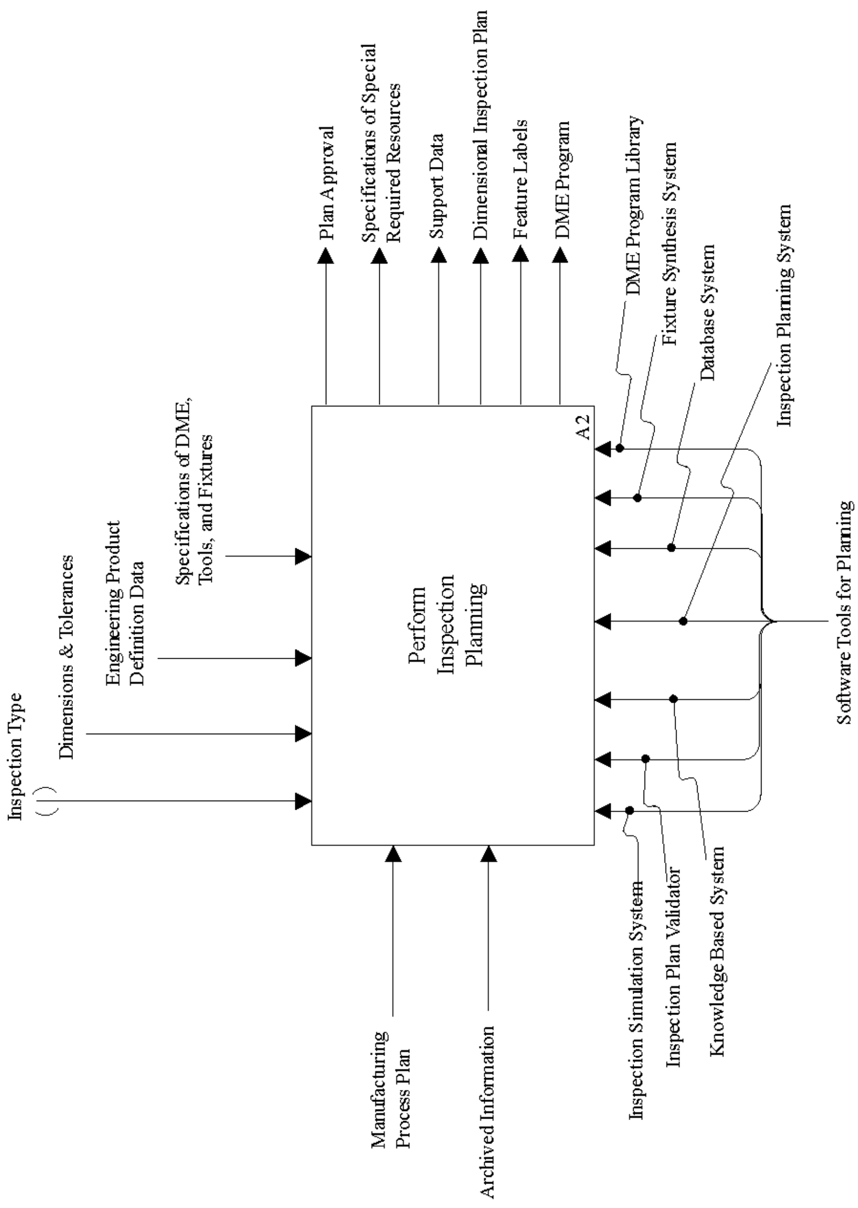


Figure F.15 - A2 Perform Inspection Planning

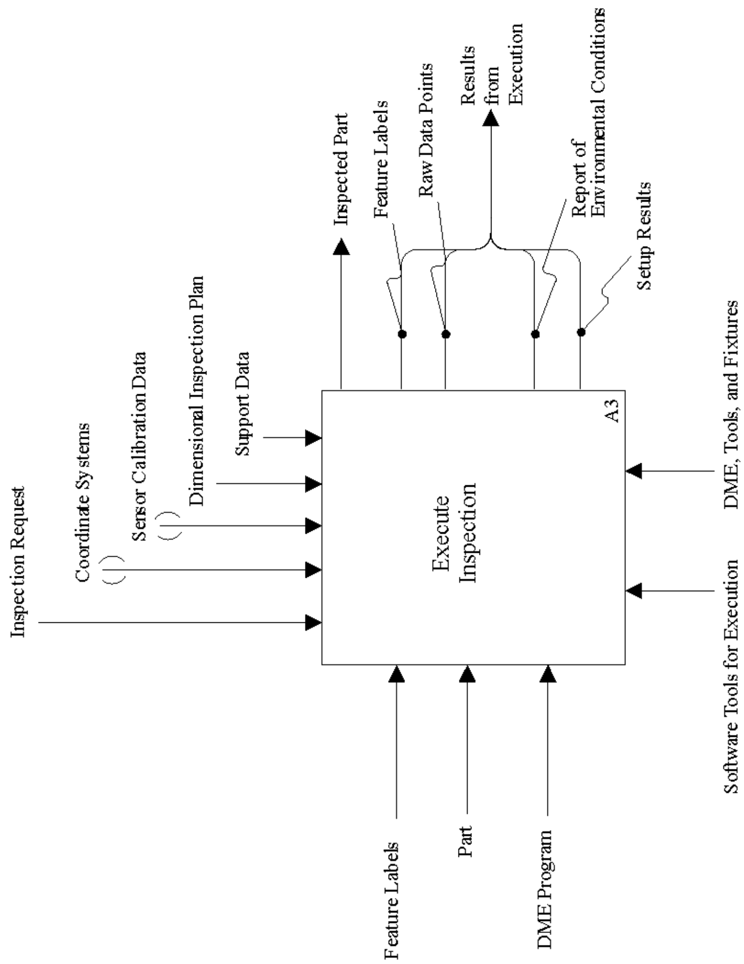


Figure F.16 - A3 Execute Inspection

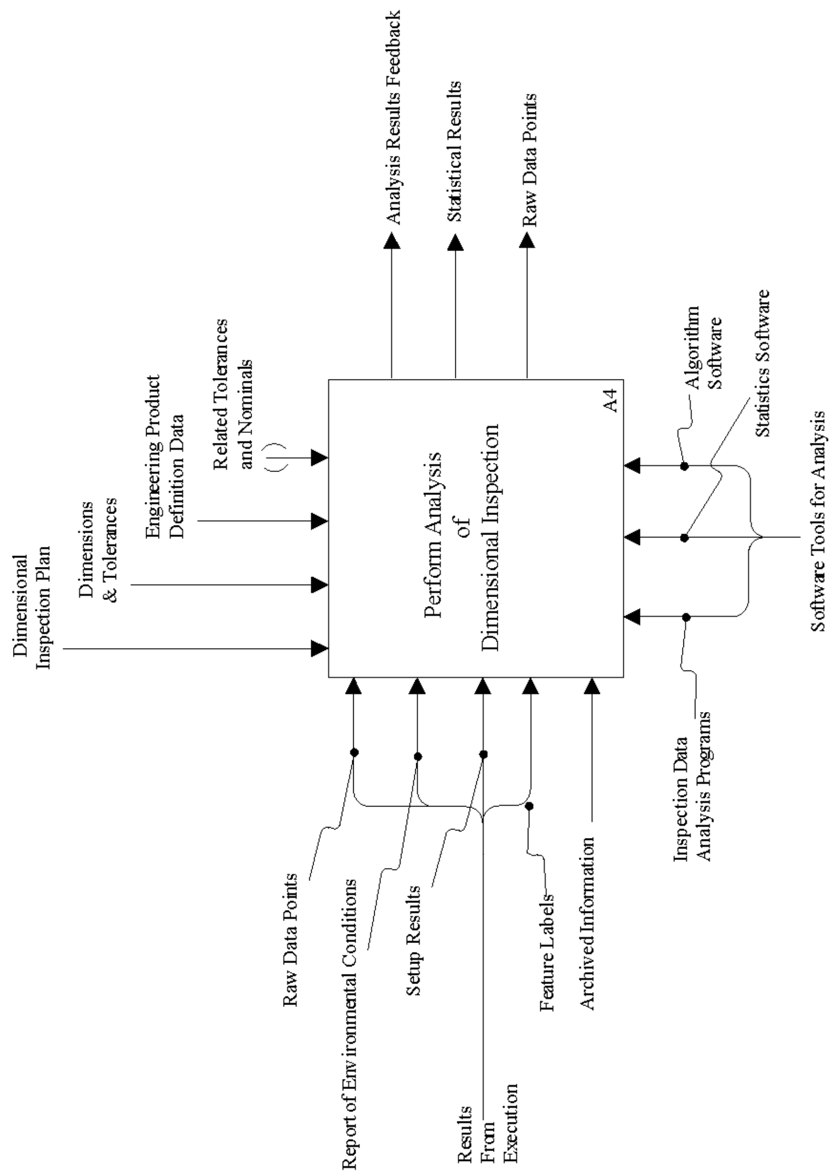


Figure F.17 - A4 Perform Analysis of Dimensional Inspection

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.34. The application reference model is a graphical representation of the structure and constraints of the application objects specified in clause 4. The graphical form of the application reference model is presented in the EXPRESS-G. The application reference model is independent from any implementation method.

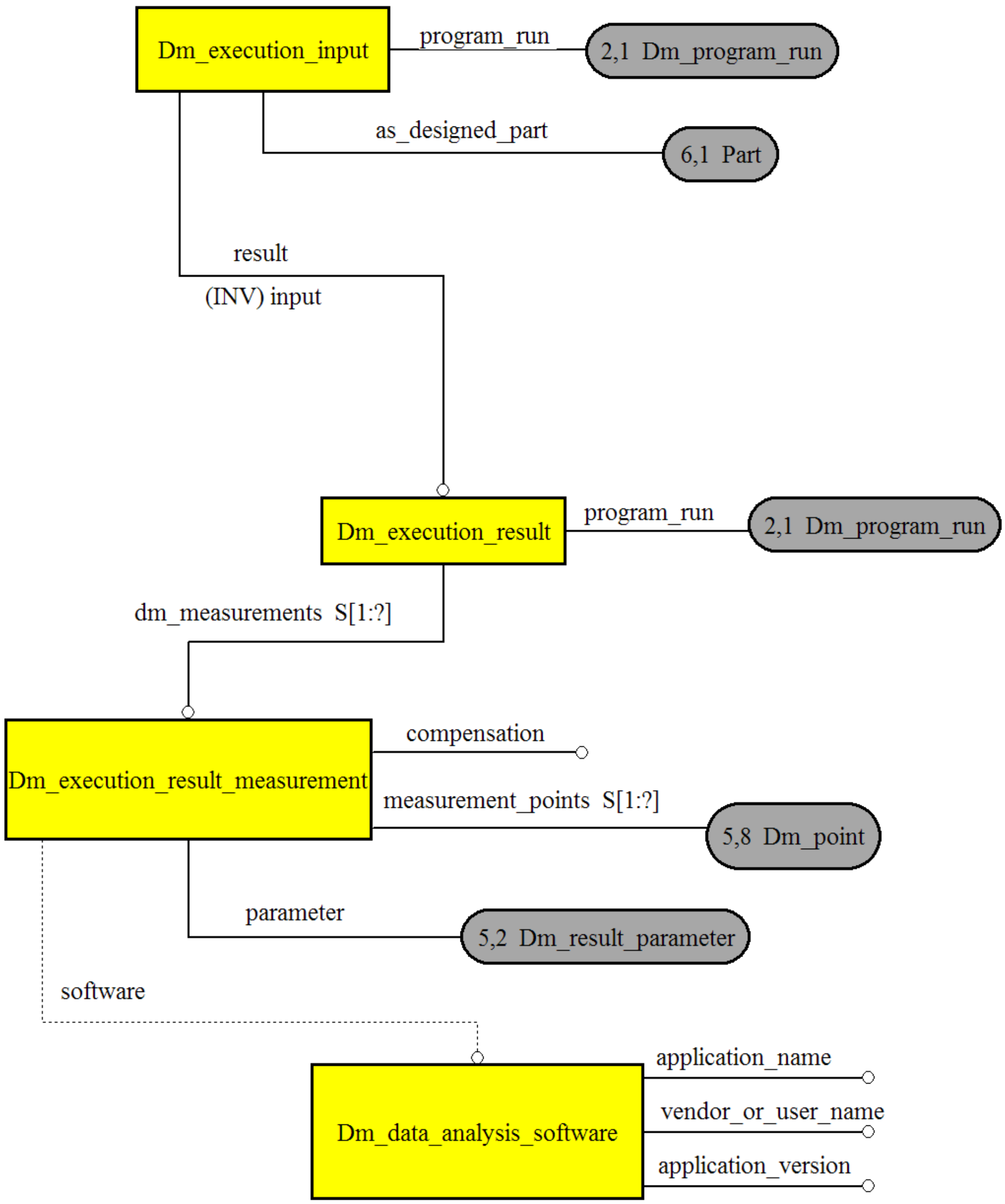


Figure G.1 — ARM EXPRESS-G diagram 1 of 34

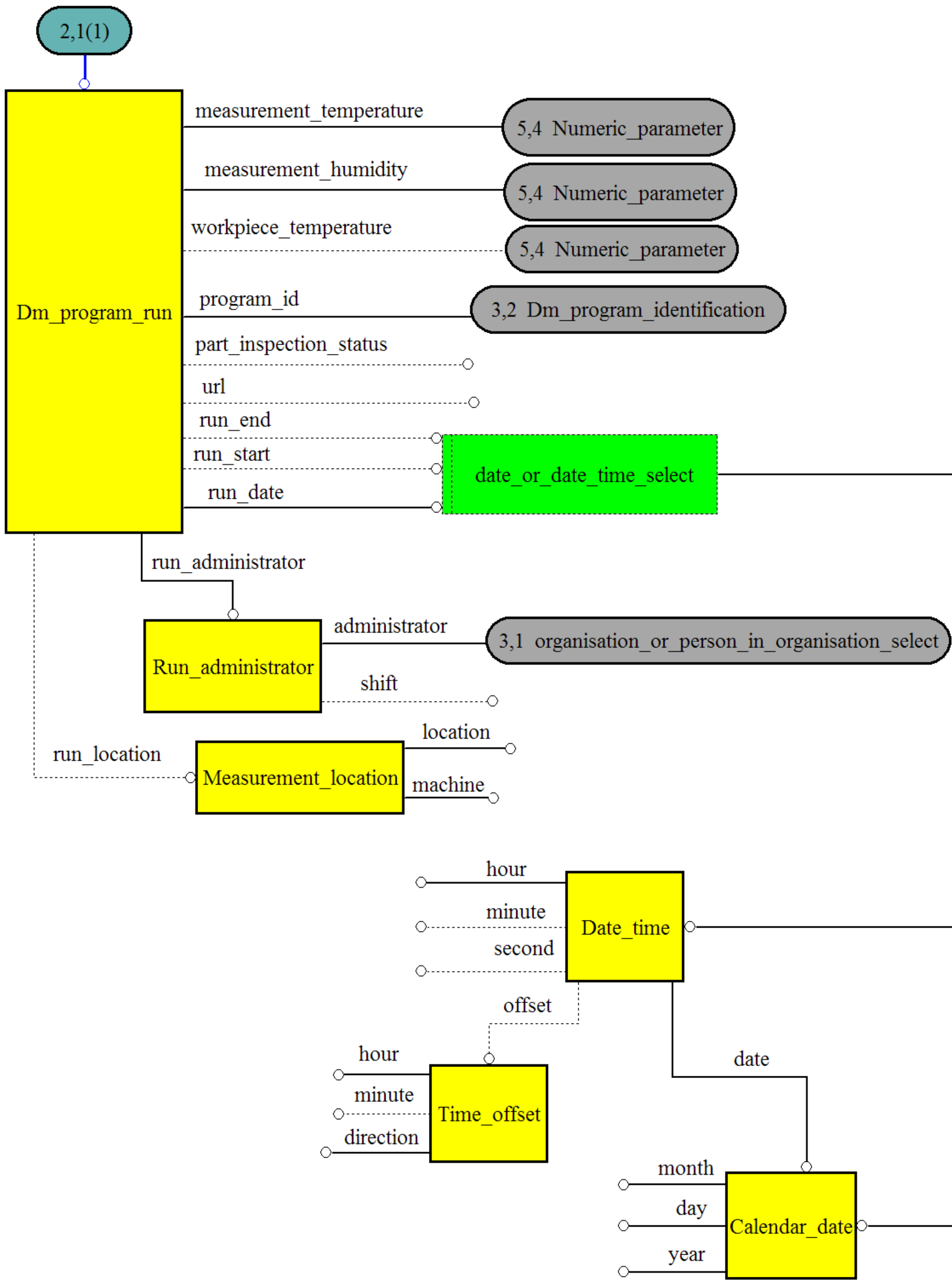


Figure G.2 — ARM EXPRESS-G diagram 2 of 34

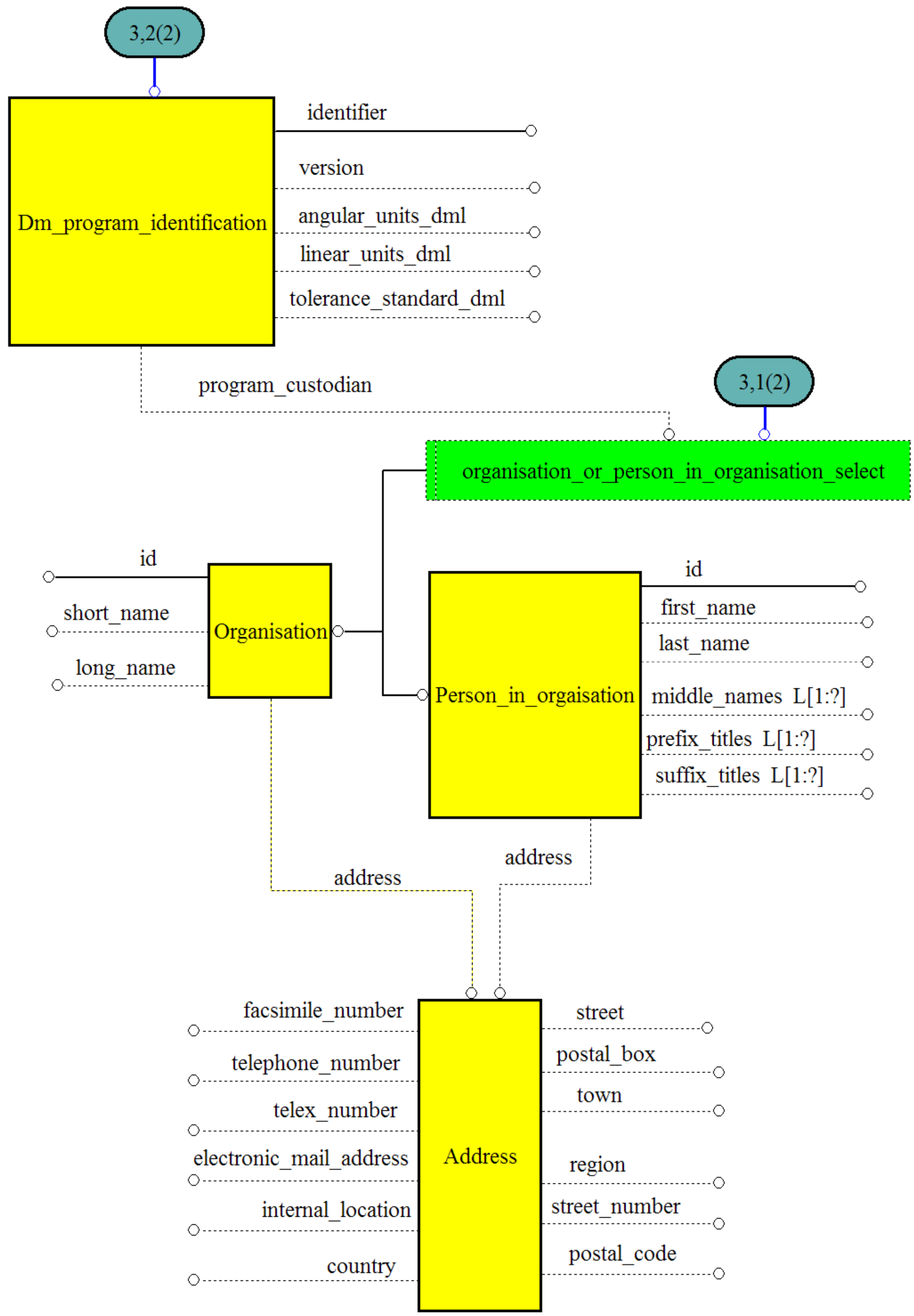


Figure G.3 — ARM EXPRESS-G diagram 3 of 34

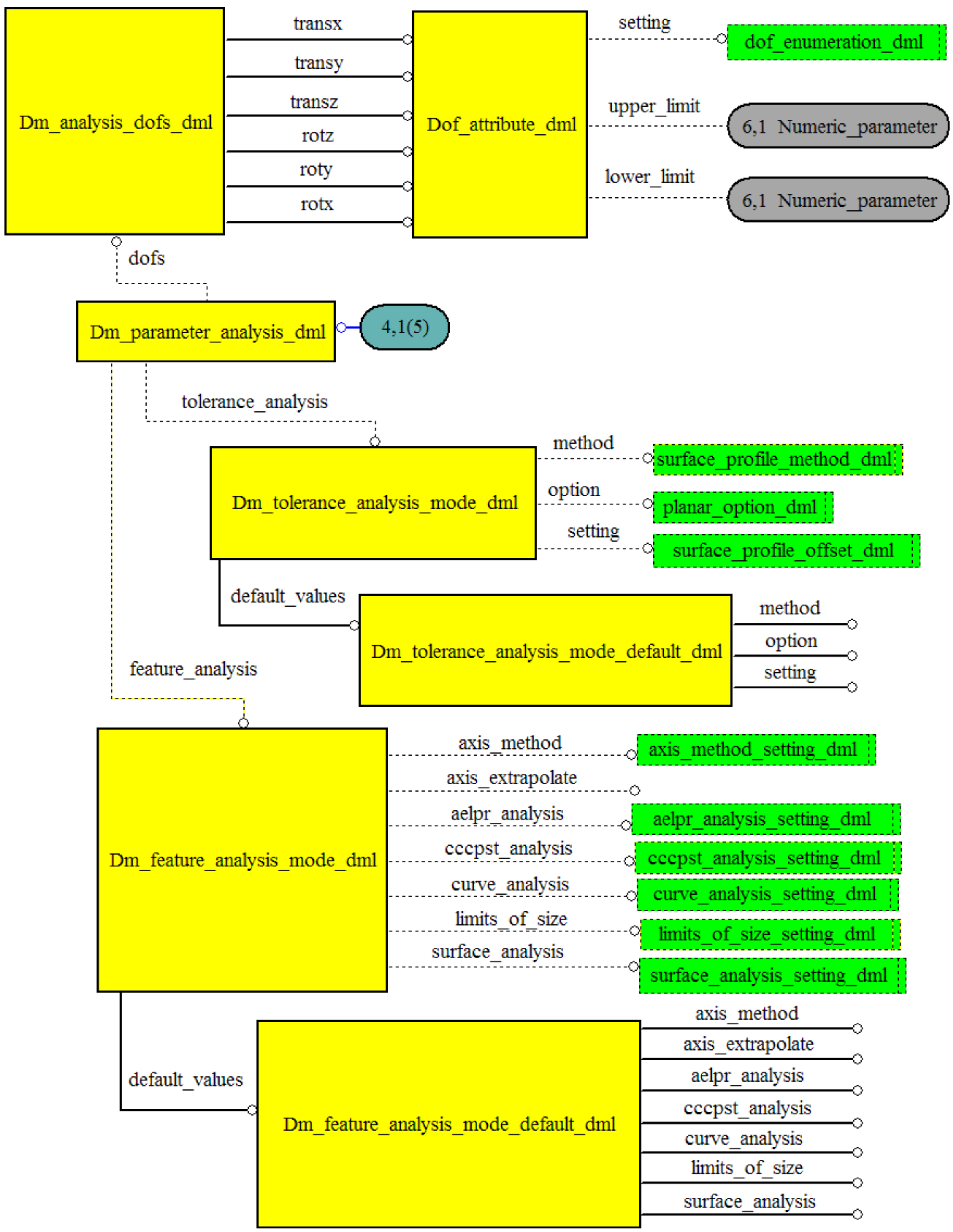


Figure G.4 — ARM EXPRESS-G diagram 4 of 34

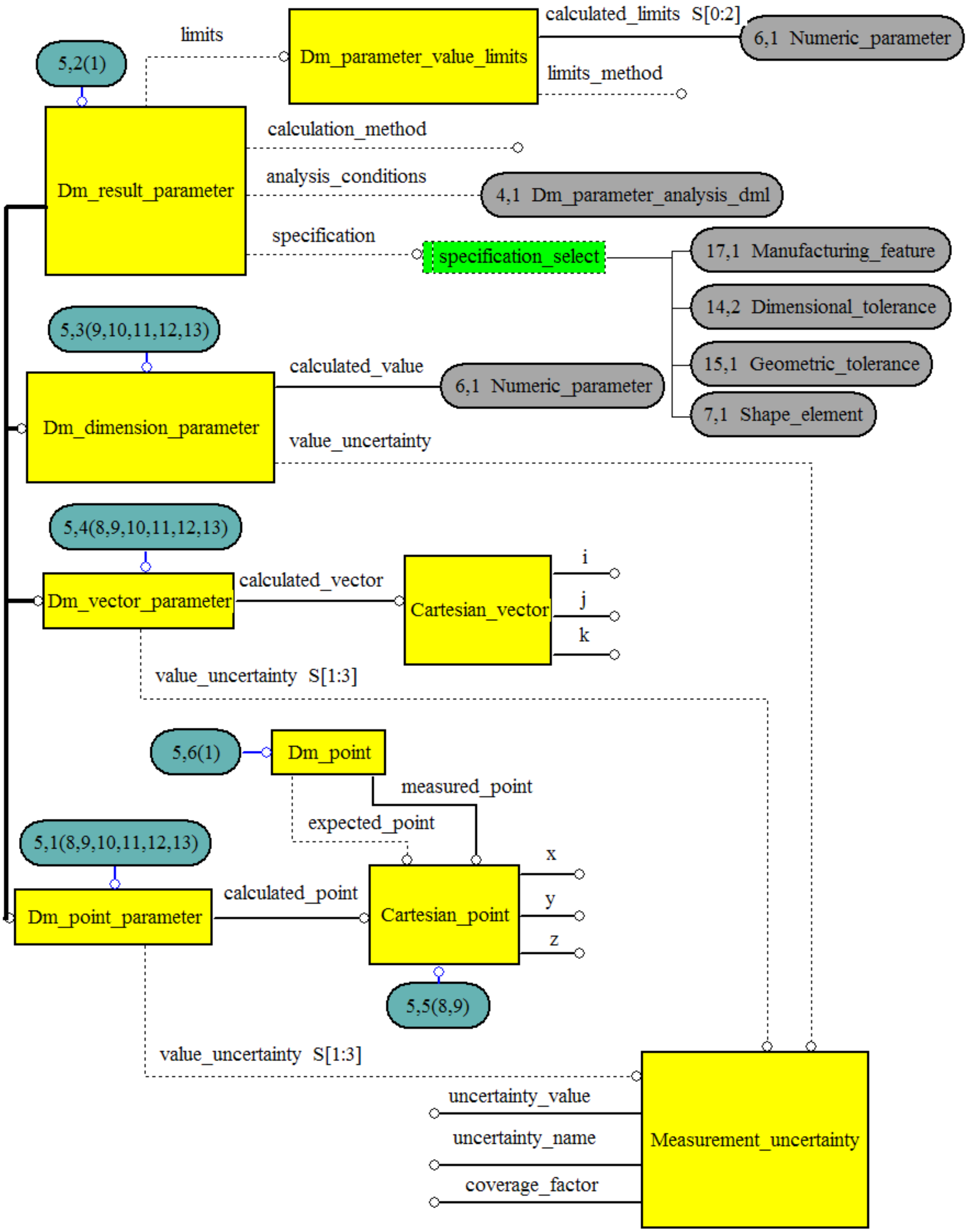


Figure G.5 — ARM EXPRESS-G diagram 5 of 34

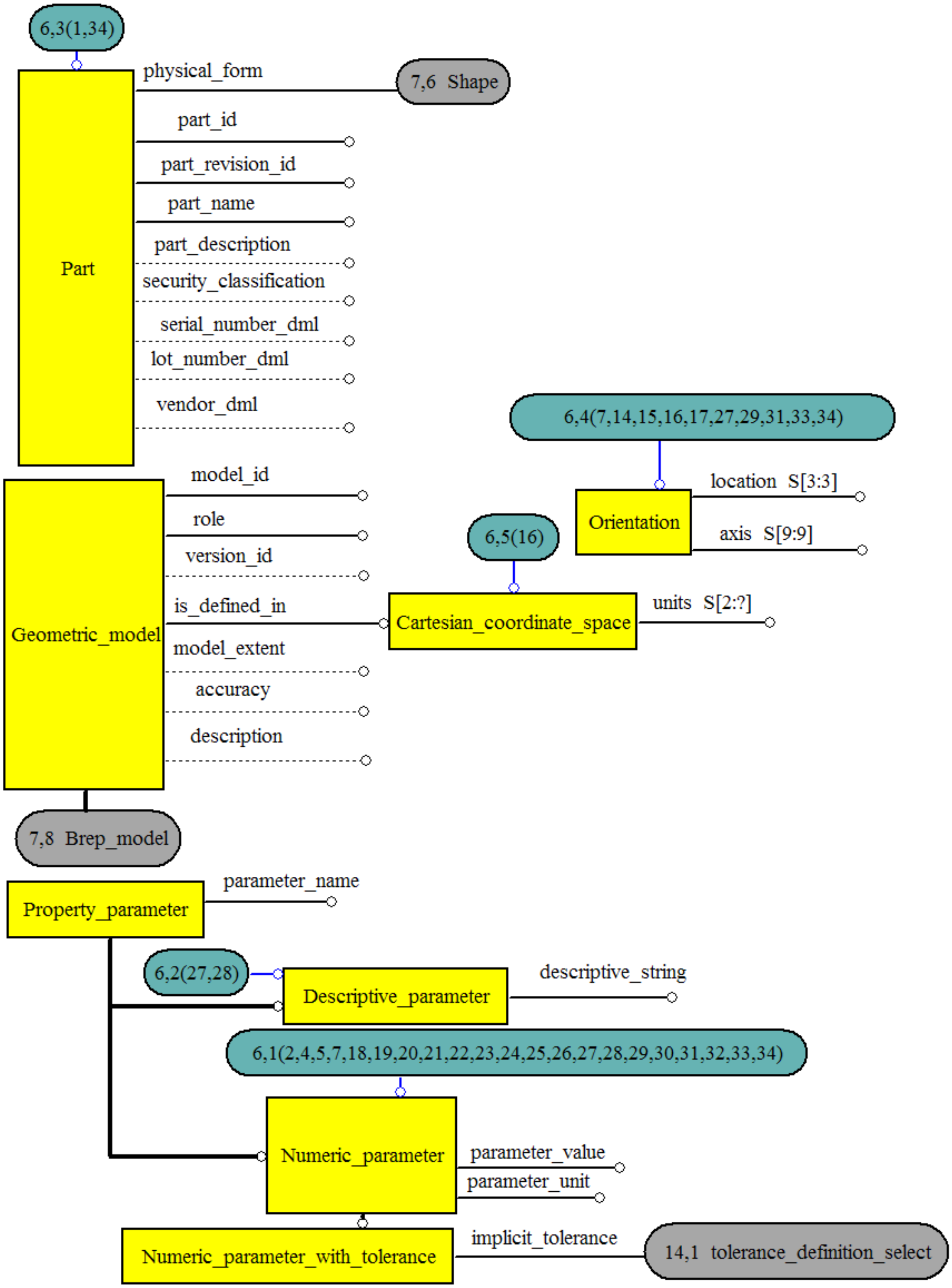


Figure G.6 — ARM EXPRESS-G diagram 6 of 34

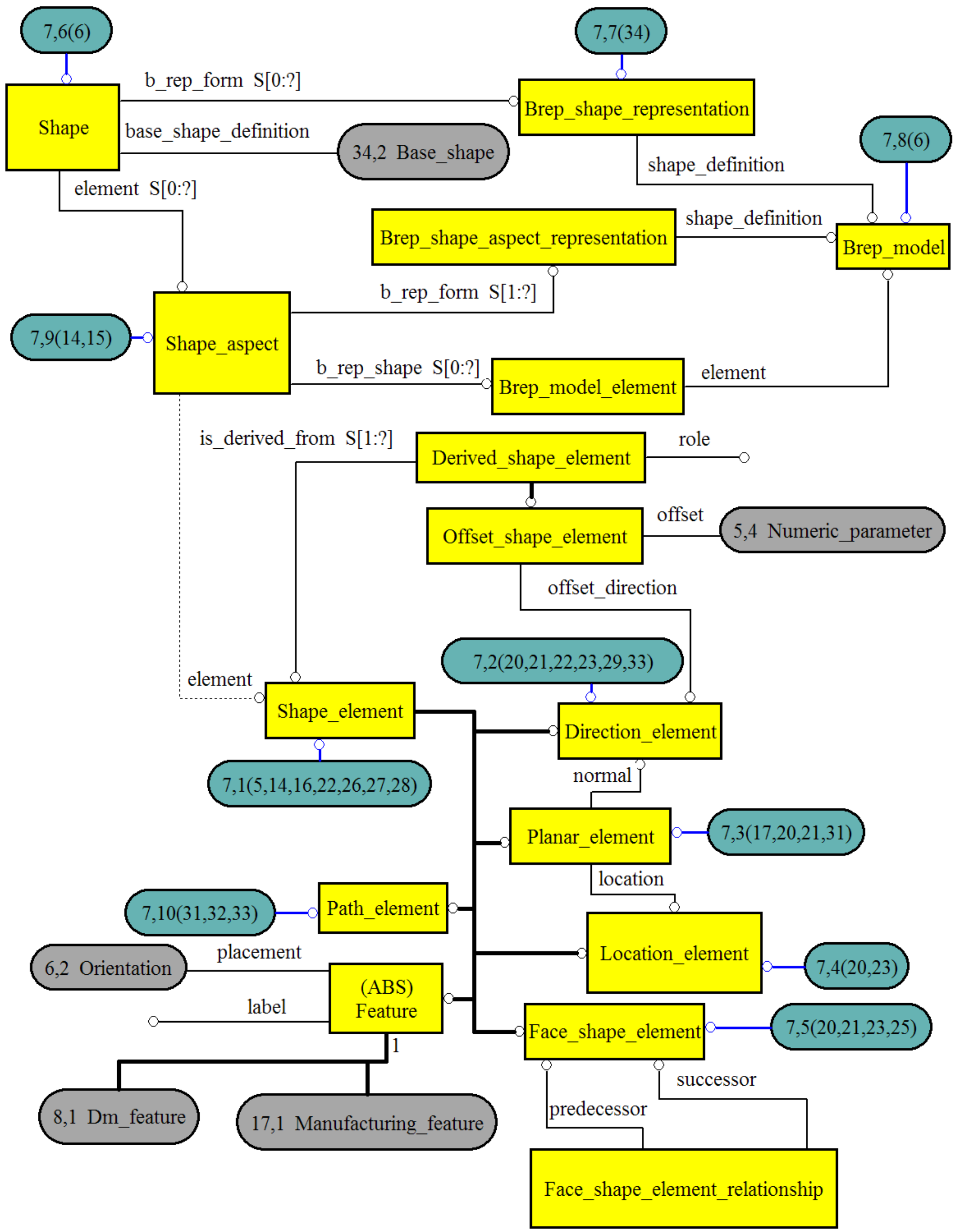


Figure G.7 — ARM EXPRESS-G diagram 7 of 34

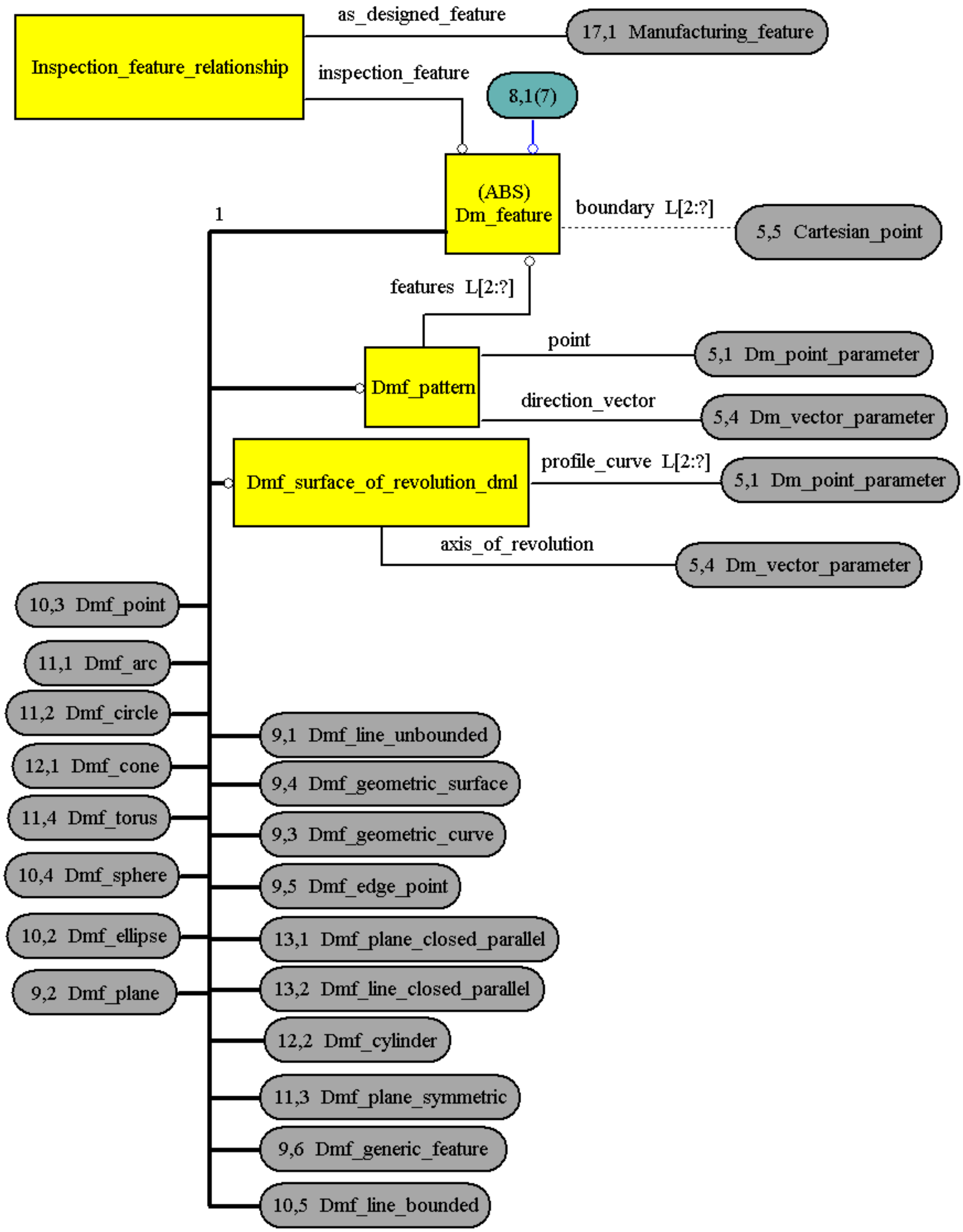


Figure G.8 — ARM EXPRESS-G diagram 8 of 34

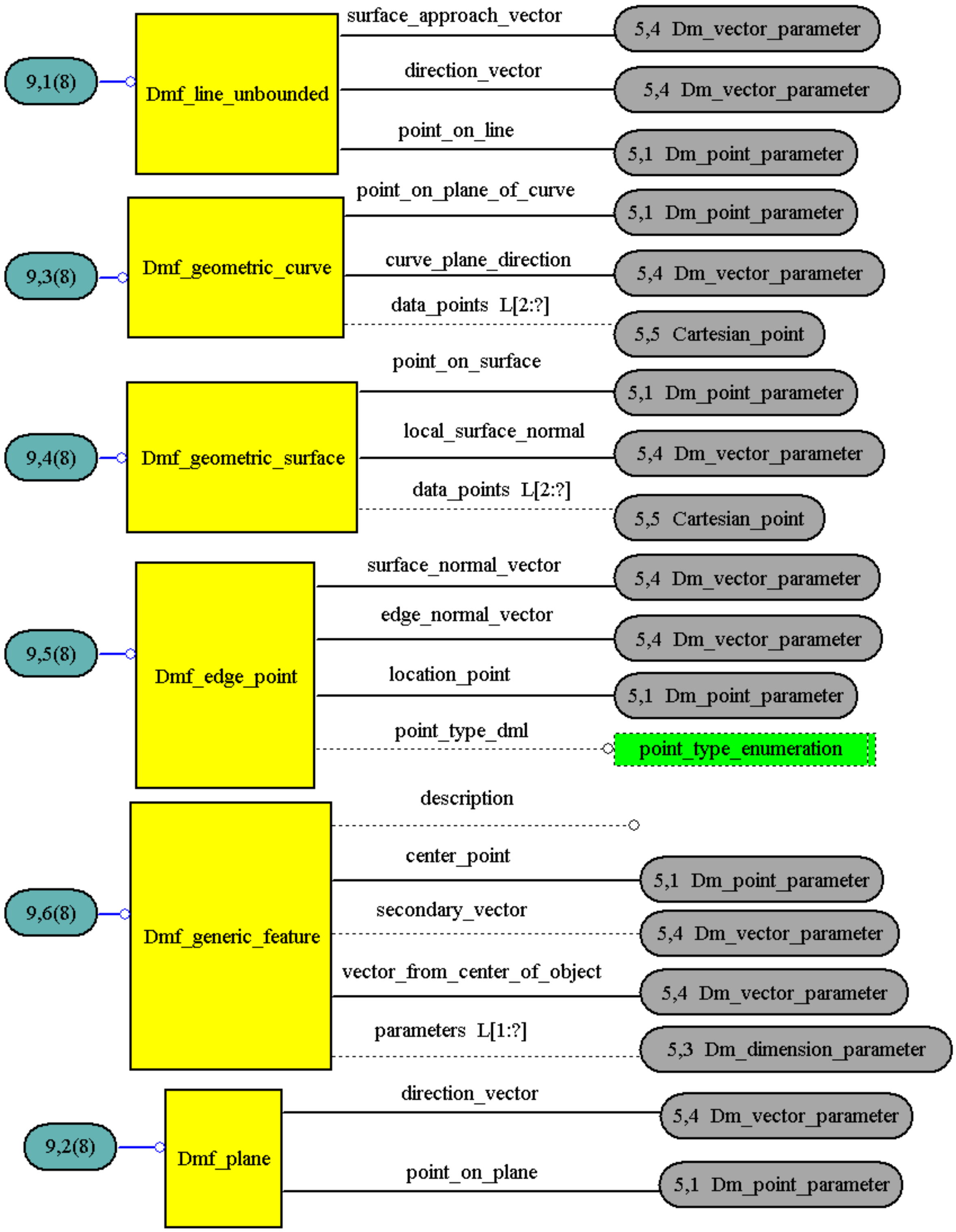


Figure G.9 — ARM EXPRESS-G diagram 9 of 34

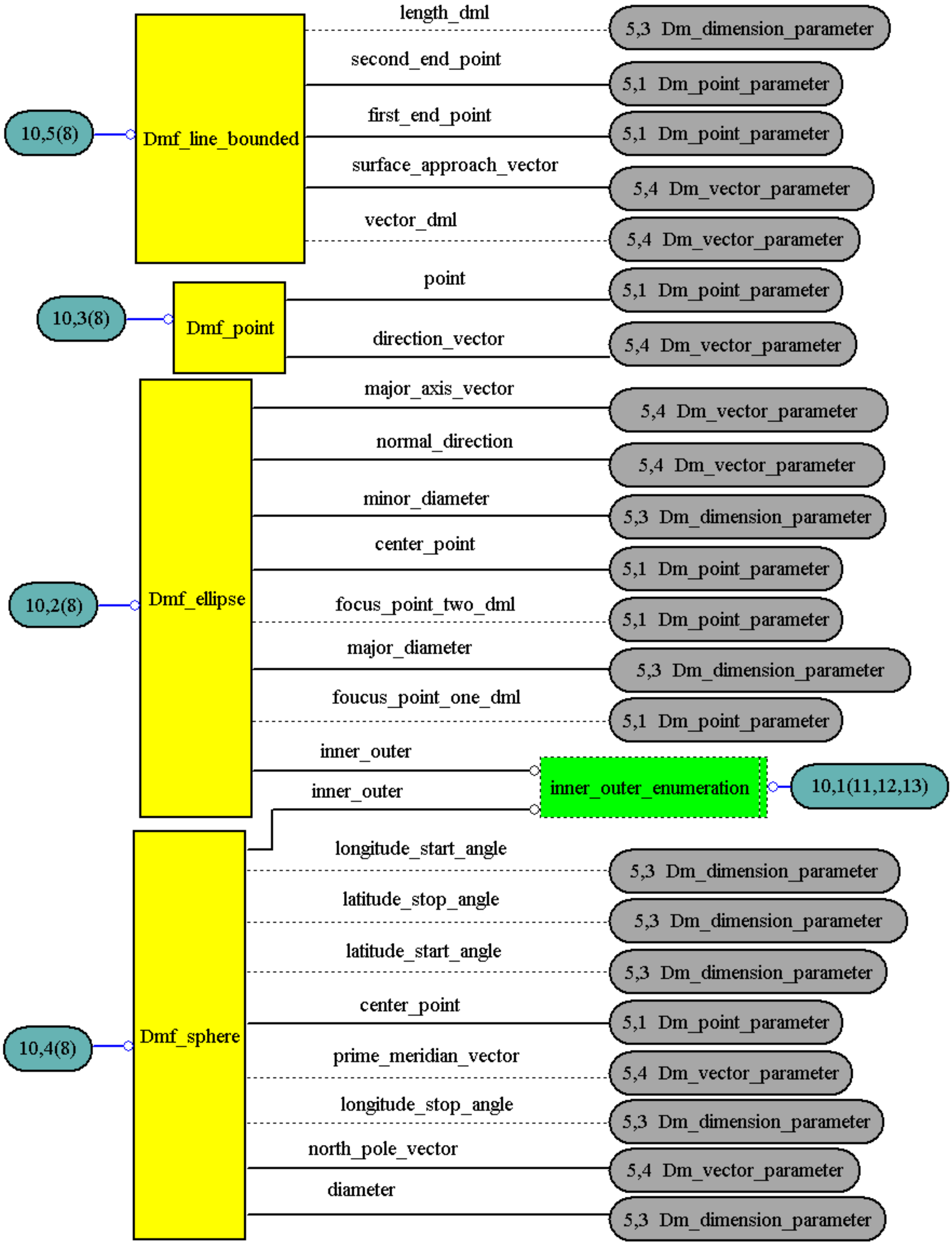


Figure G.10 — ARM EXPRESS-G diagram 10 of 34

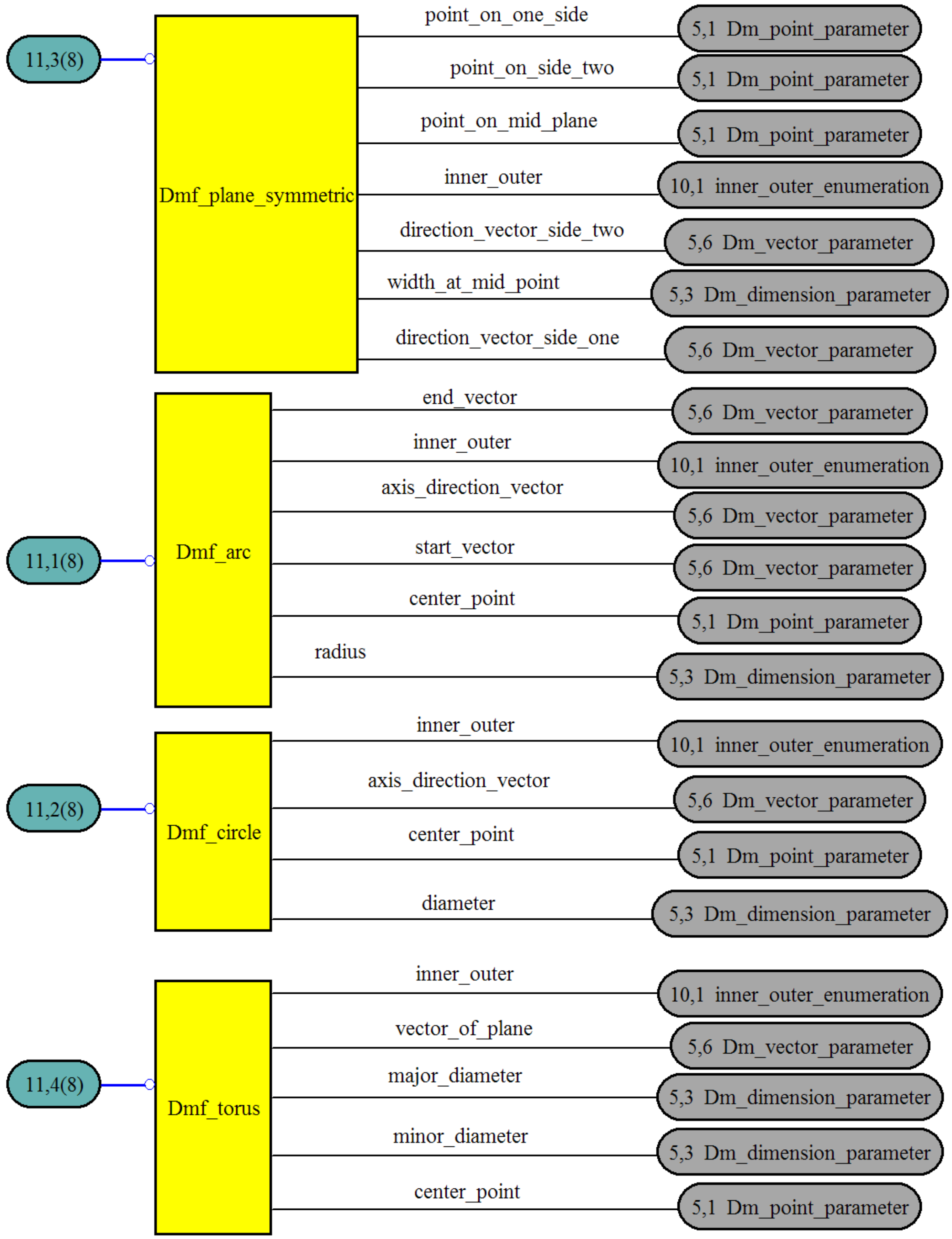


Figure G.11 — ARM EXPRESS-G diagram 11 of 34

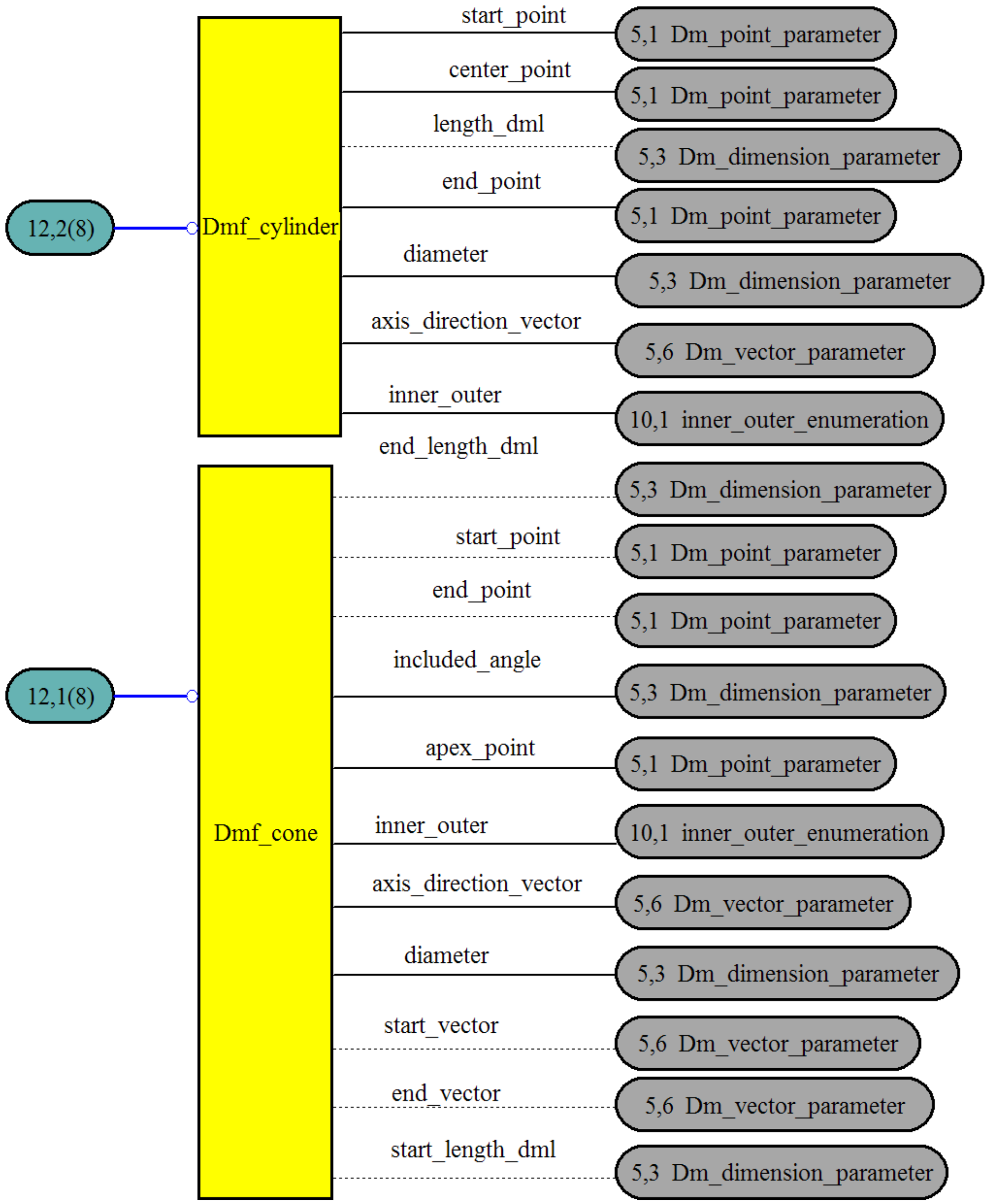


Figure G.12 — ARM EXPRESS-G diagram 12 of 34

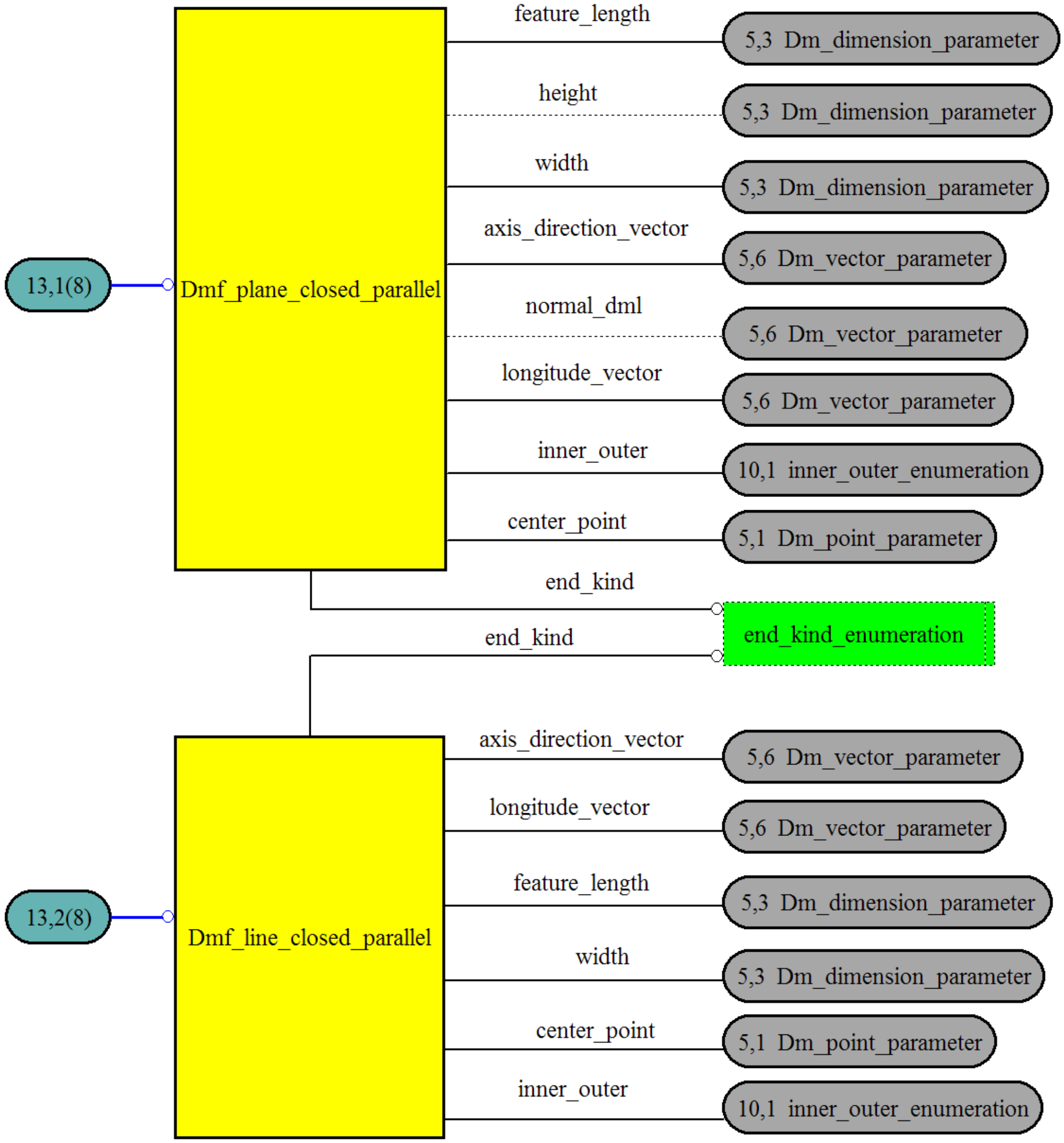


Figure G.13 — ARM EXPRESS-G diagram 13 of 34

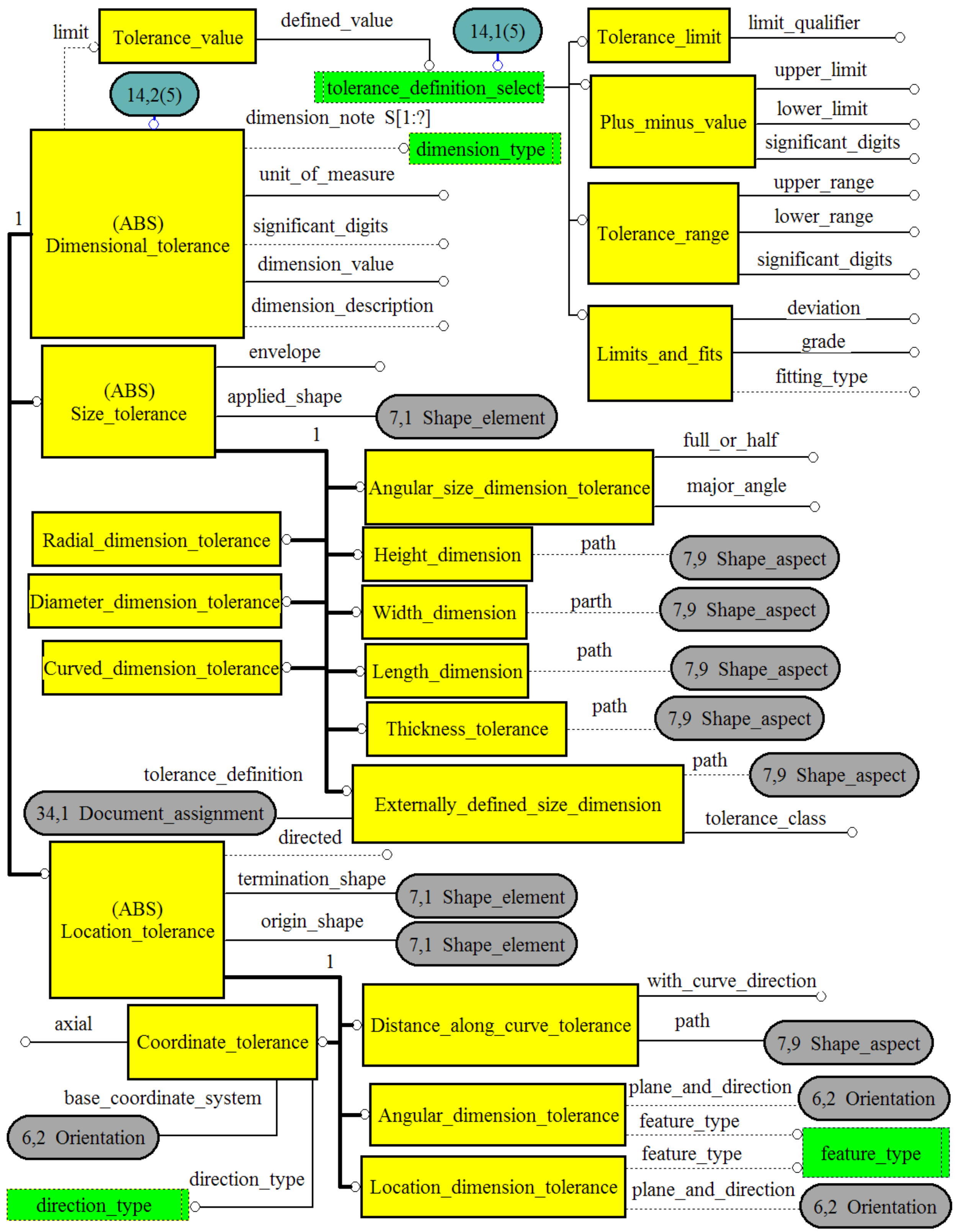


Figure G.14 — ARM EXPRESS-G diagram 14 of 34

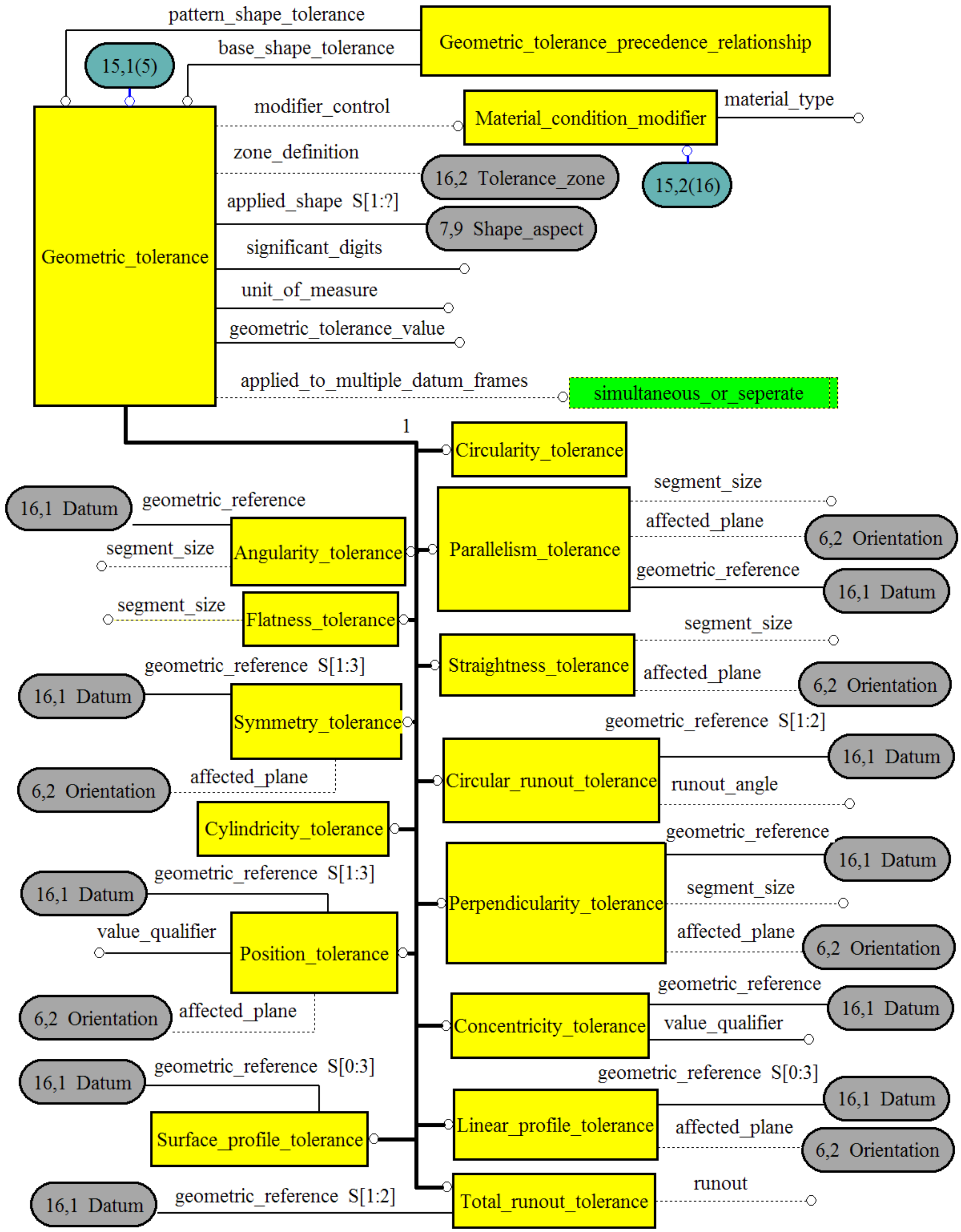


Figure G.15 — ARM EXPRESS-G diagram 15 of 34

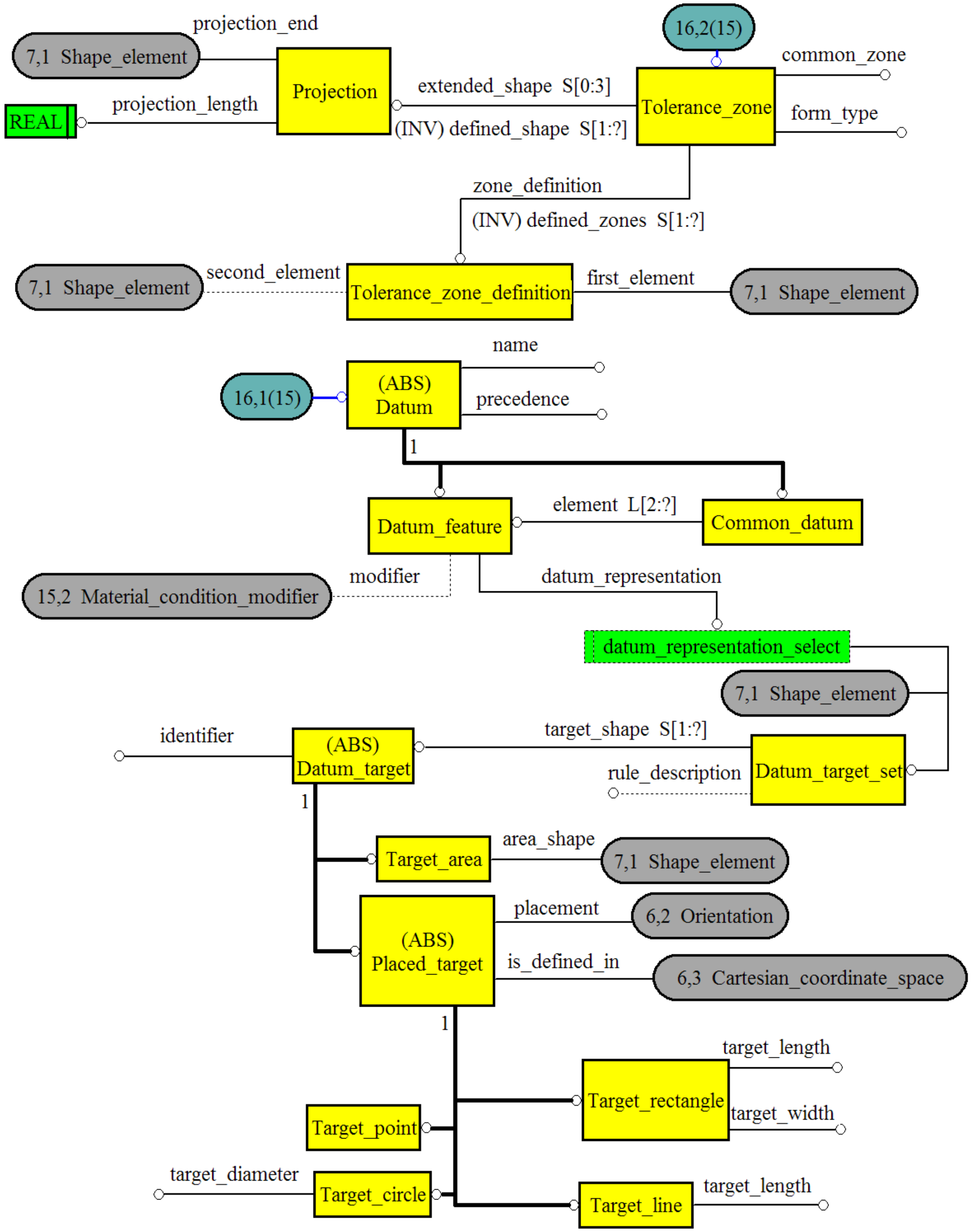


Figure G.16 — ARM EXPRESS-G diagram 16 of 34

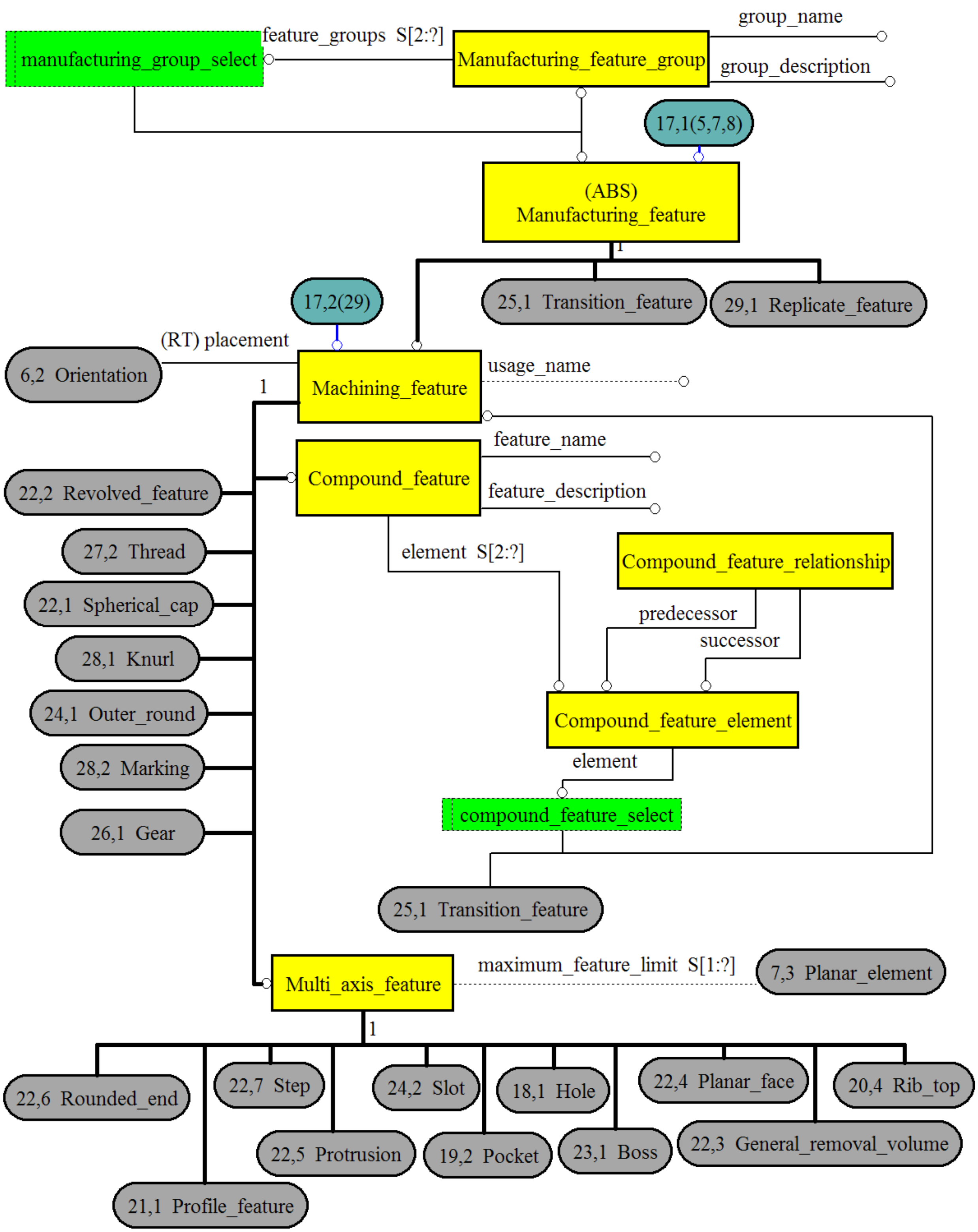


Figure G.17 — ARM EXPRESS-G diagram 17 of 34

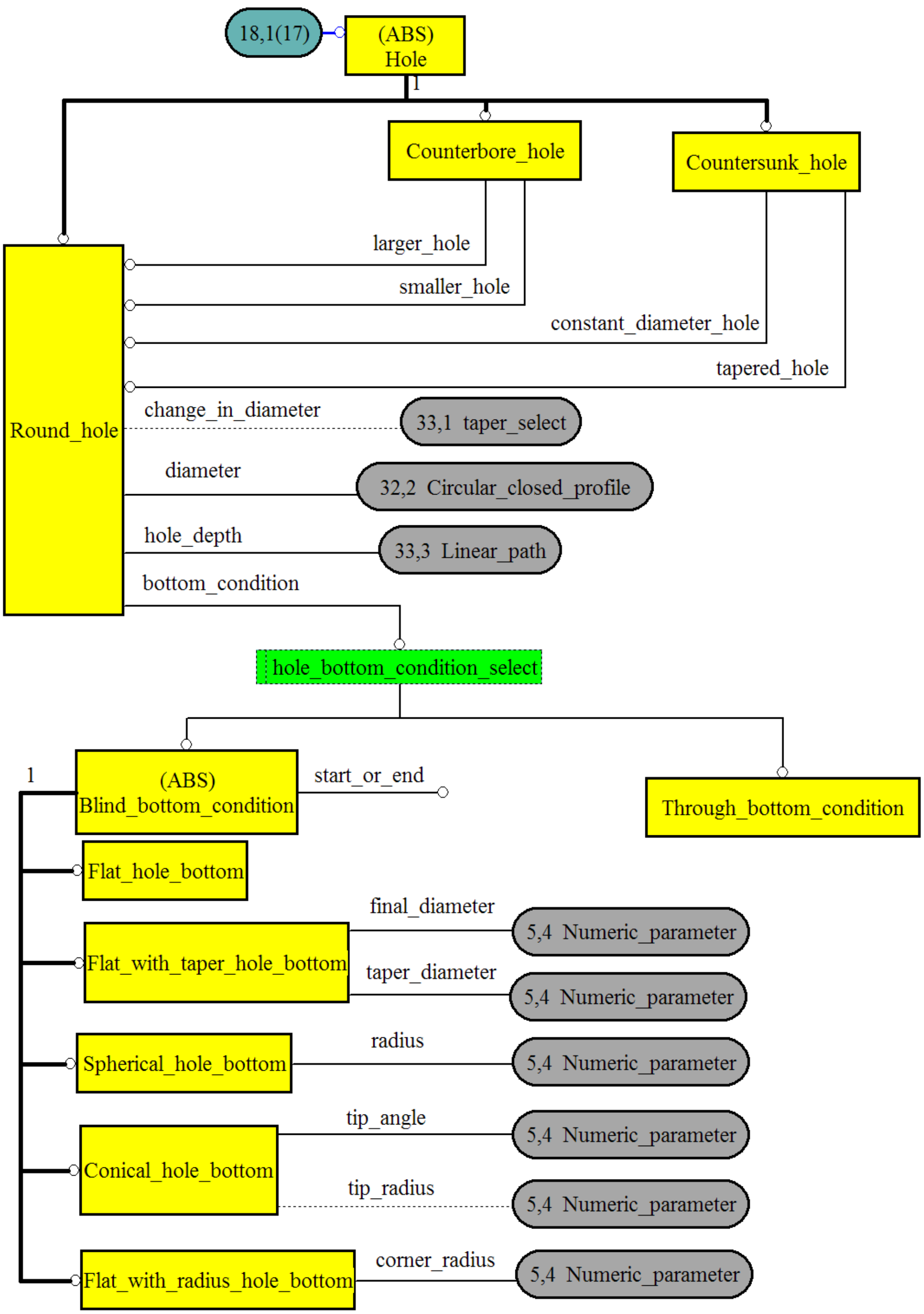


Figure G.18 — ARM EXPRESS-G diagram 18 of 34

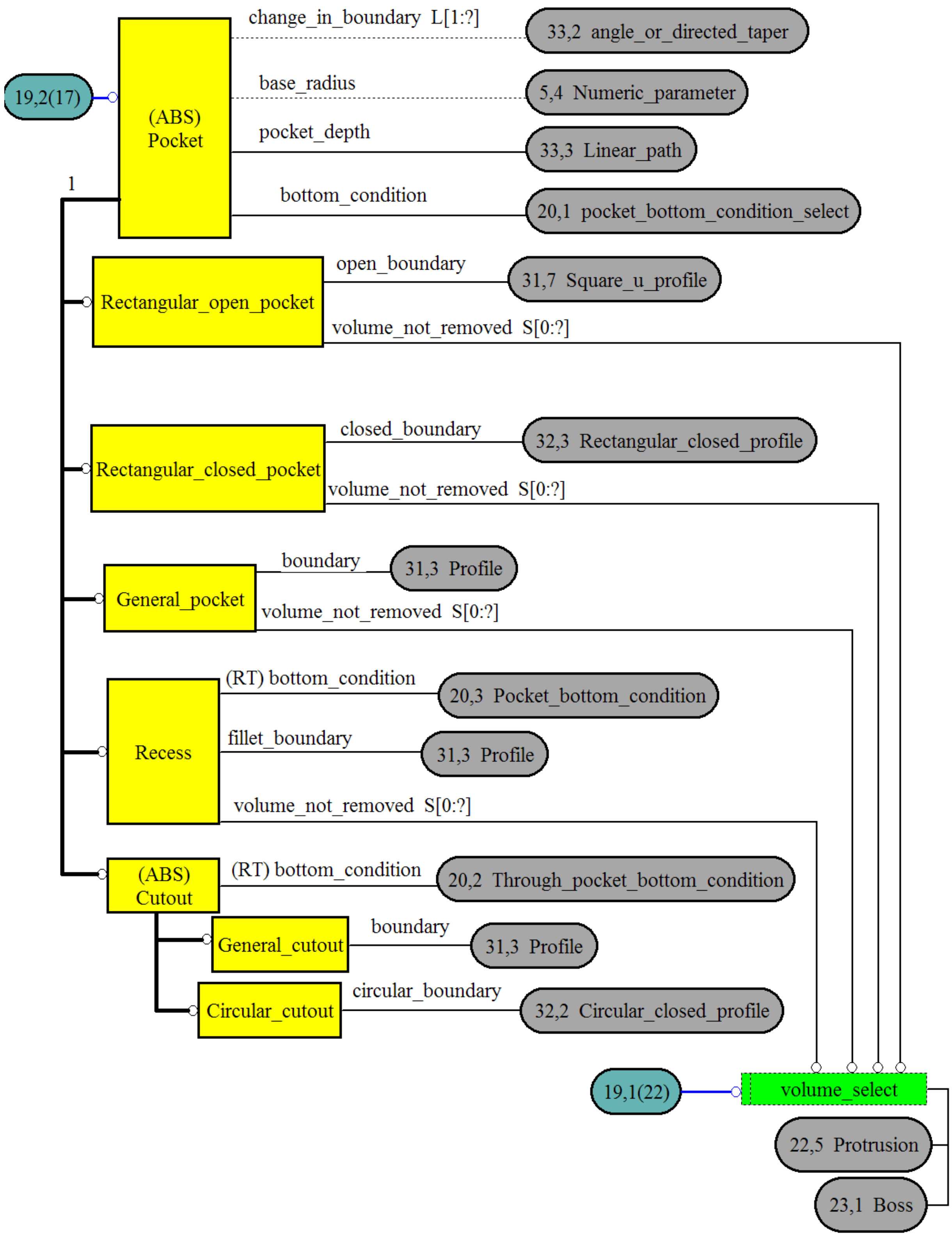


Figure G.19 — ARM EXPRESS-G diagram 19 of 34

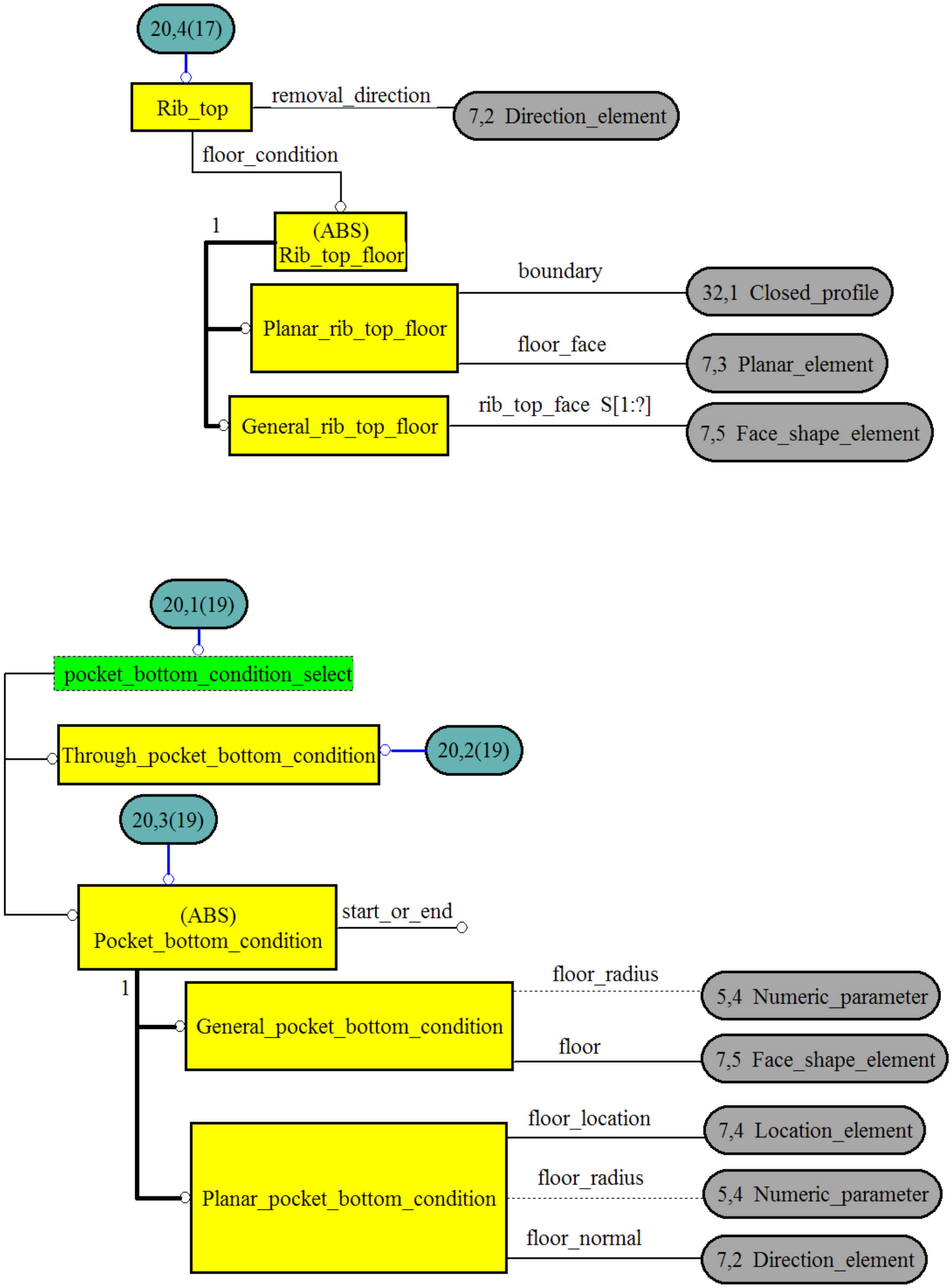


Figure G.20 — ARM EXPRESS-G diagram 20 of 34

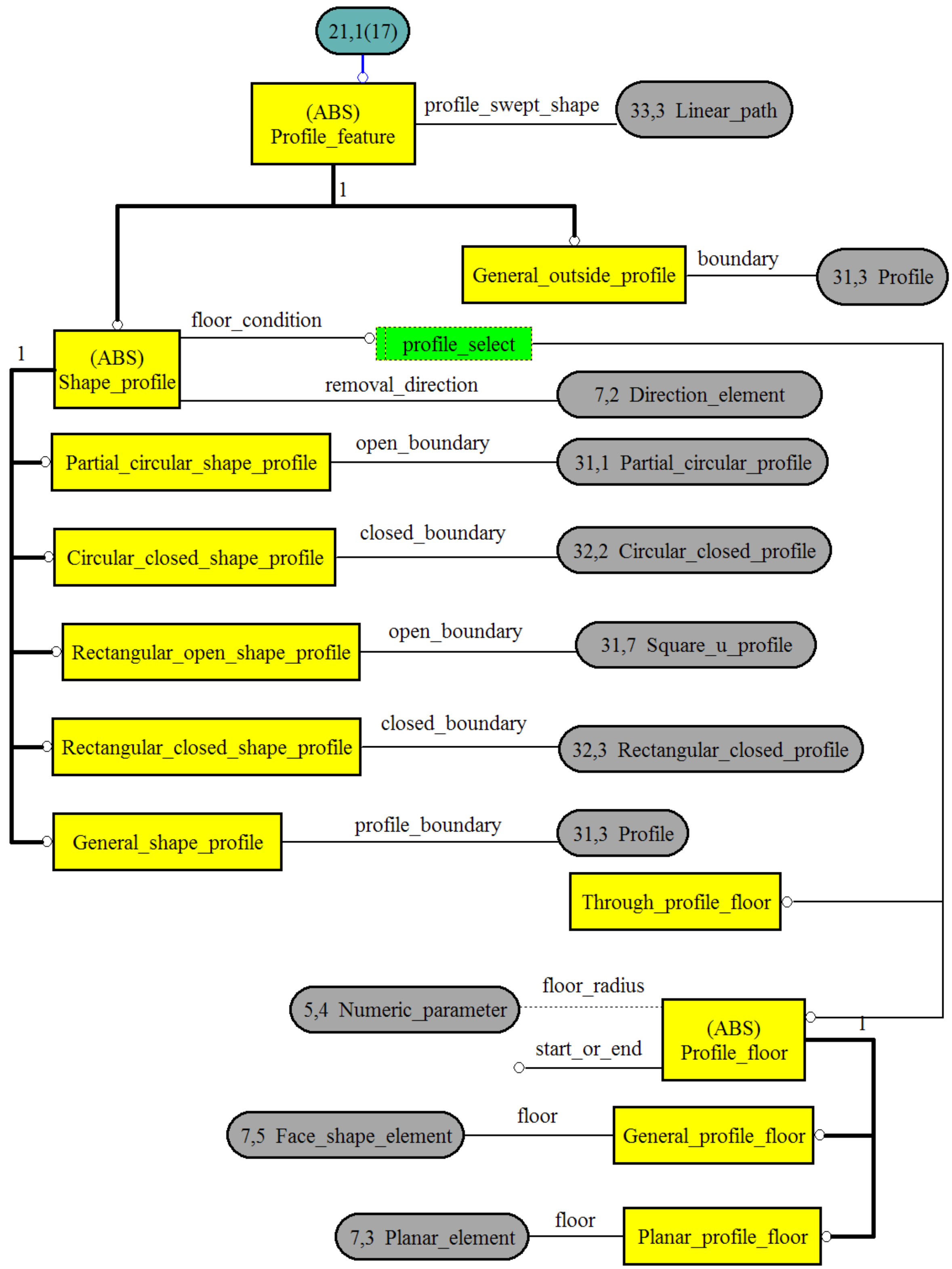


Figure G.21 — ARM EXPRESS-G diagram 21 of 34

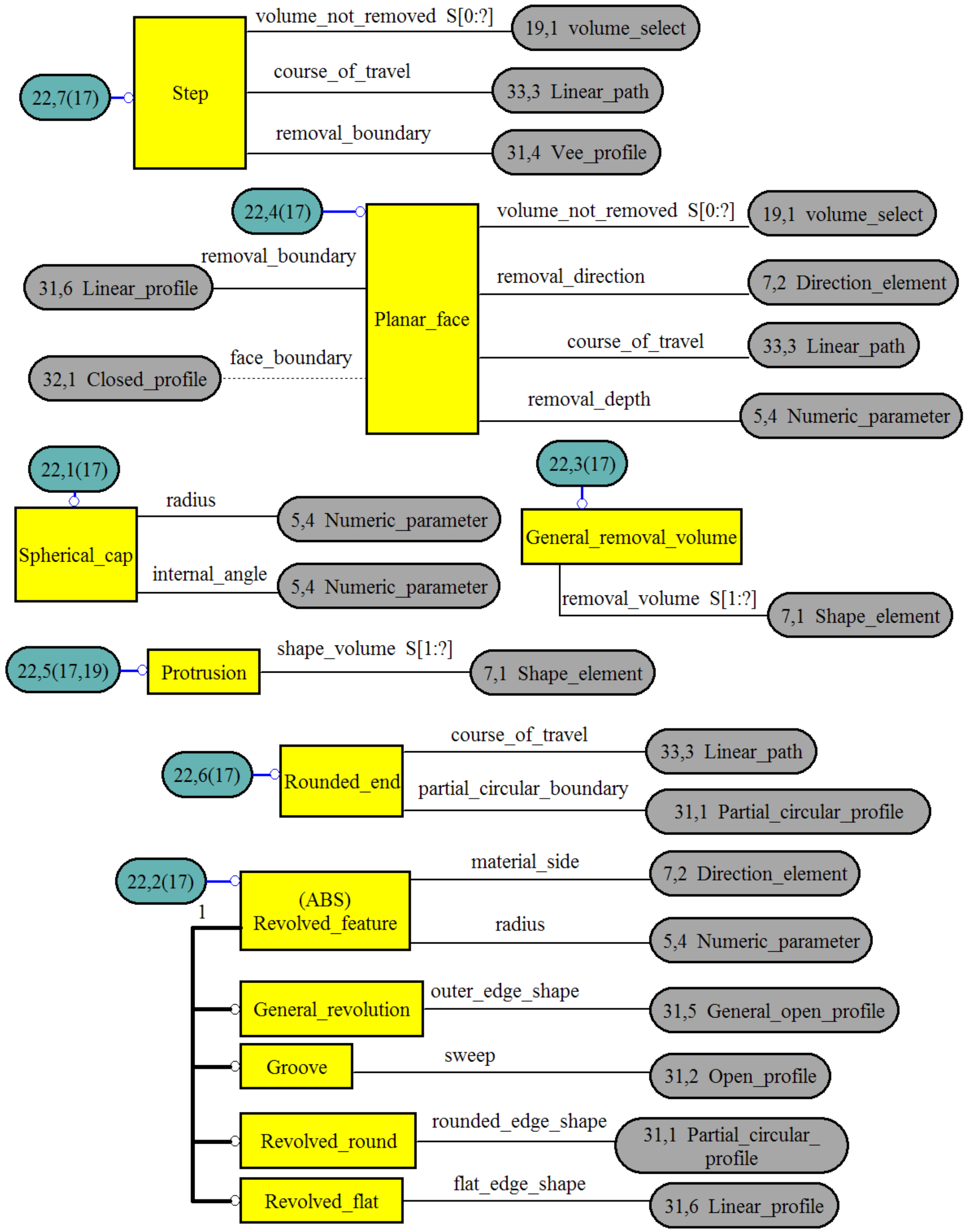


Figure G.22 — ARM EXPRESS-G diagram 22 of 34

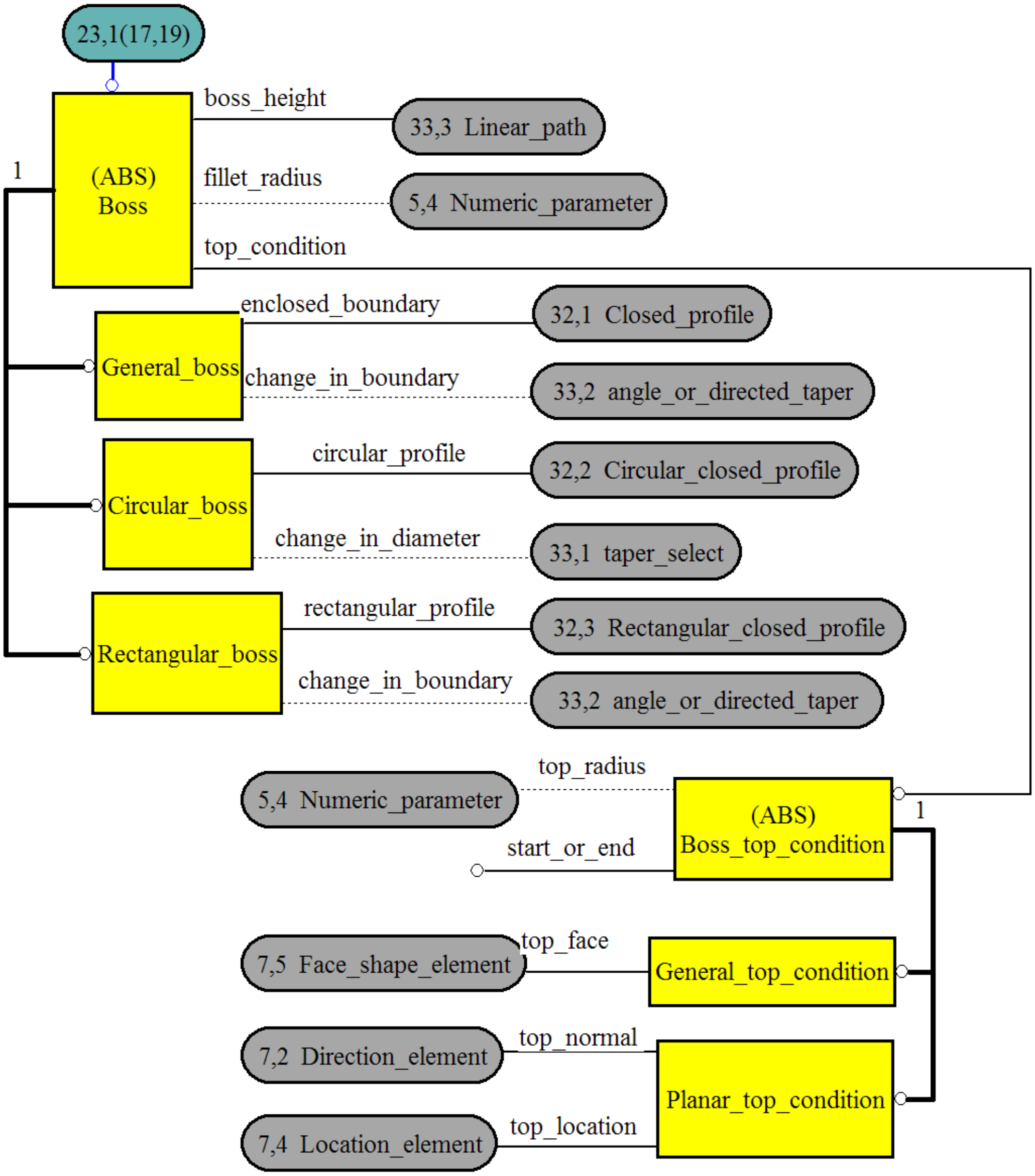


Figure G.23 — ARM EXPRESS-G diagram 23 of 34

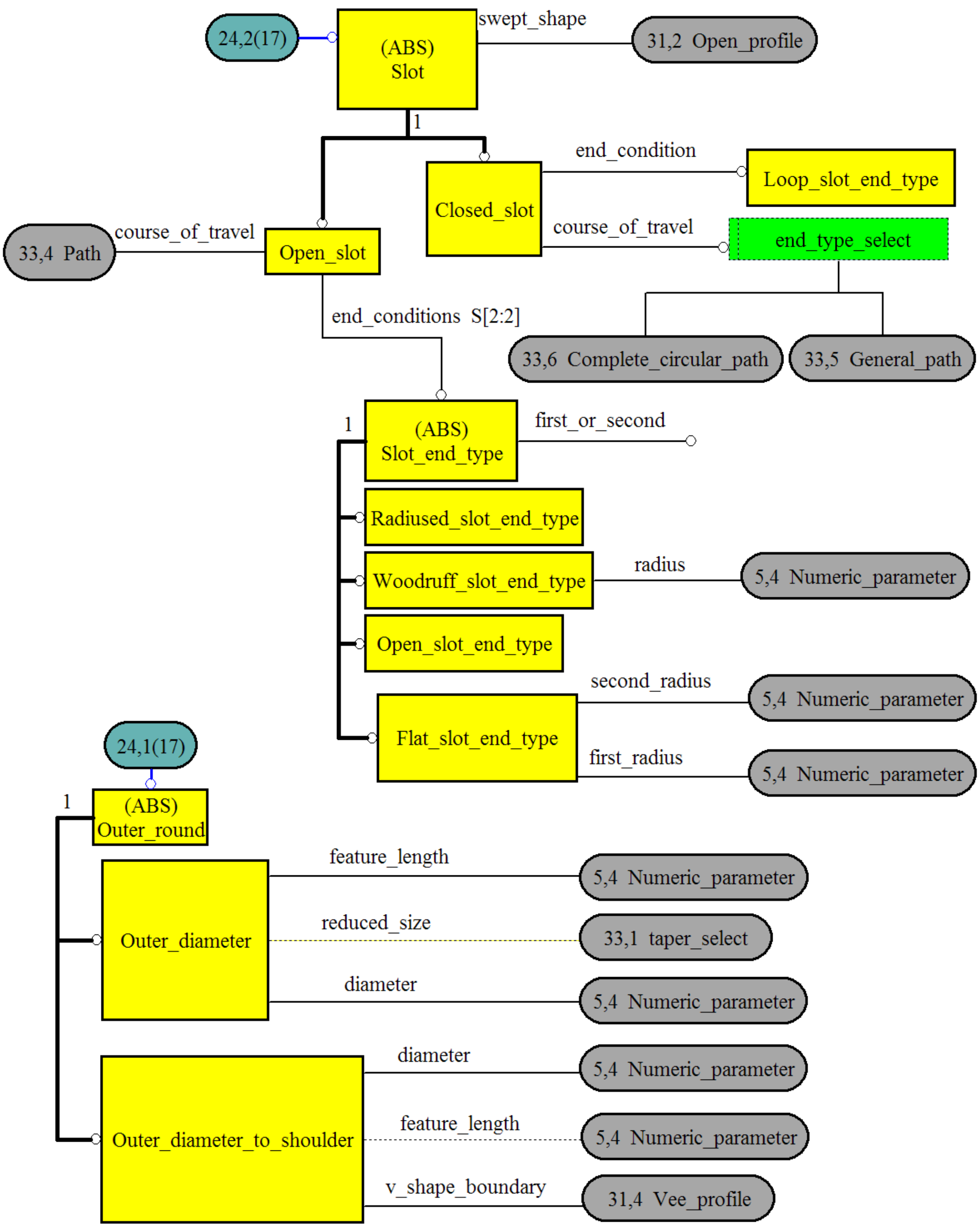


Figure G.24 — ARM EXPRESS-G diagram 24 of 34

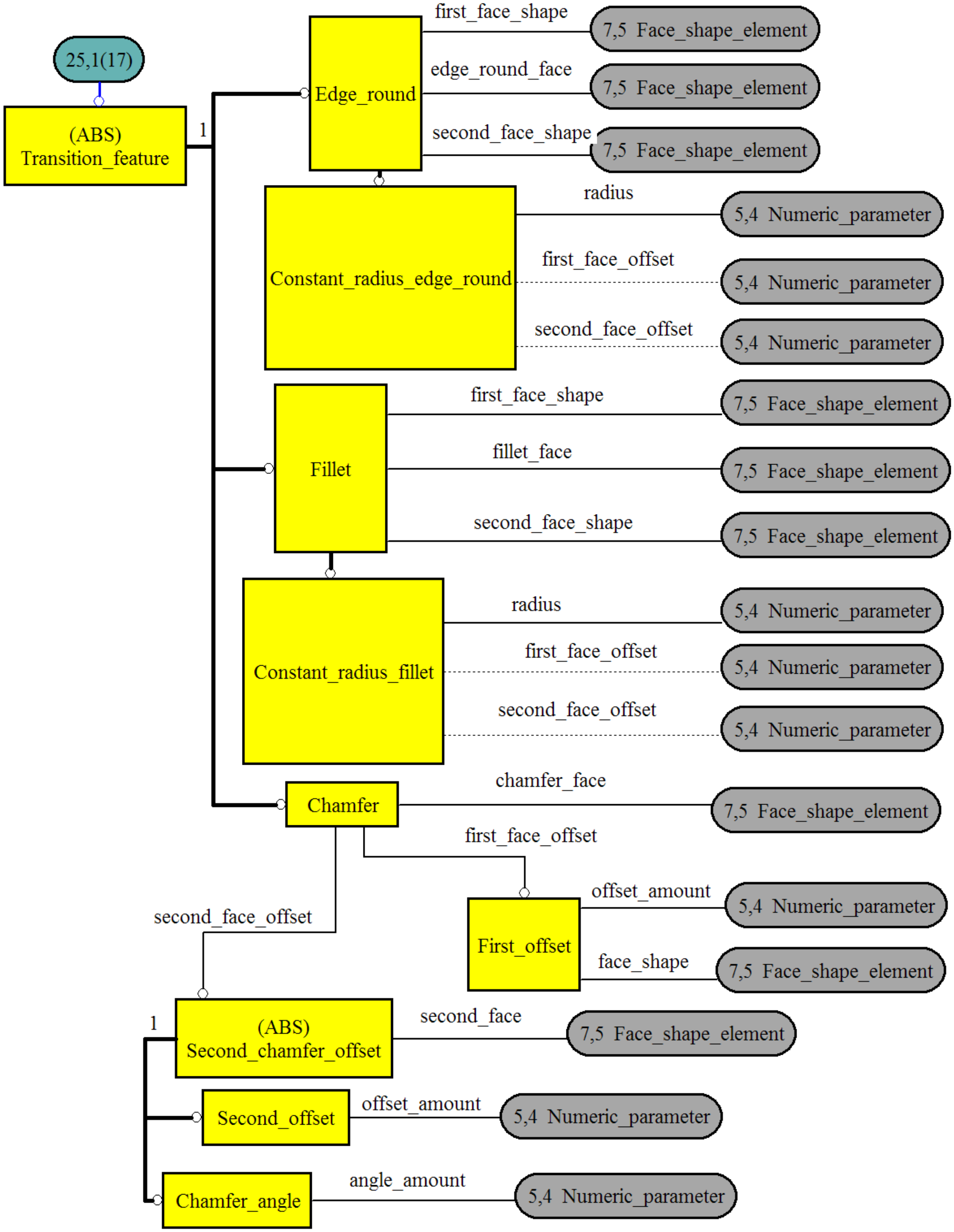


Figure G.25 — ARM EXPRESS-G diagram 25 of 34

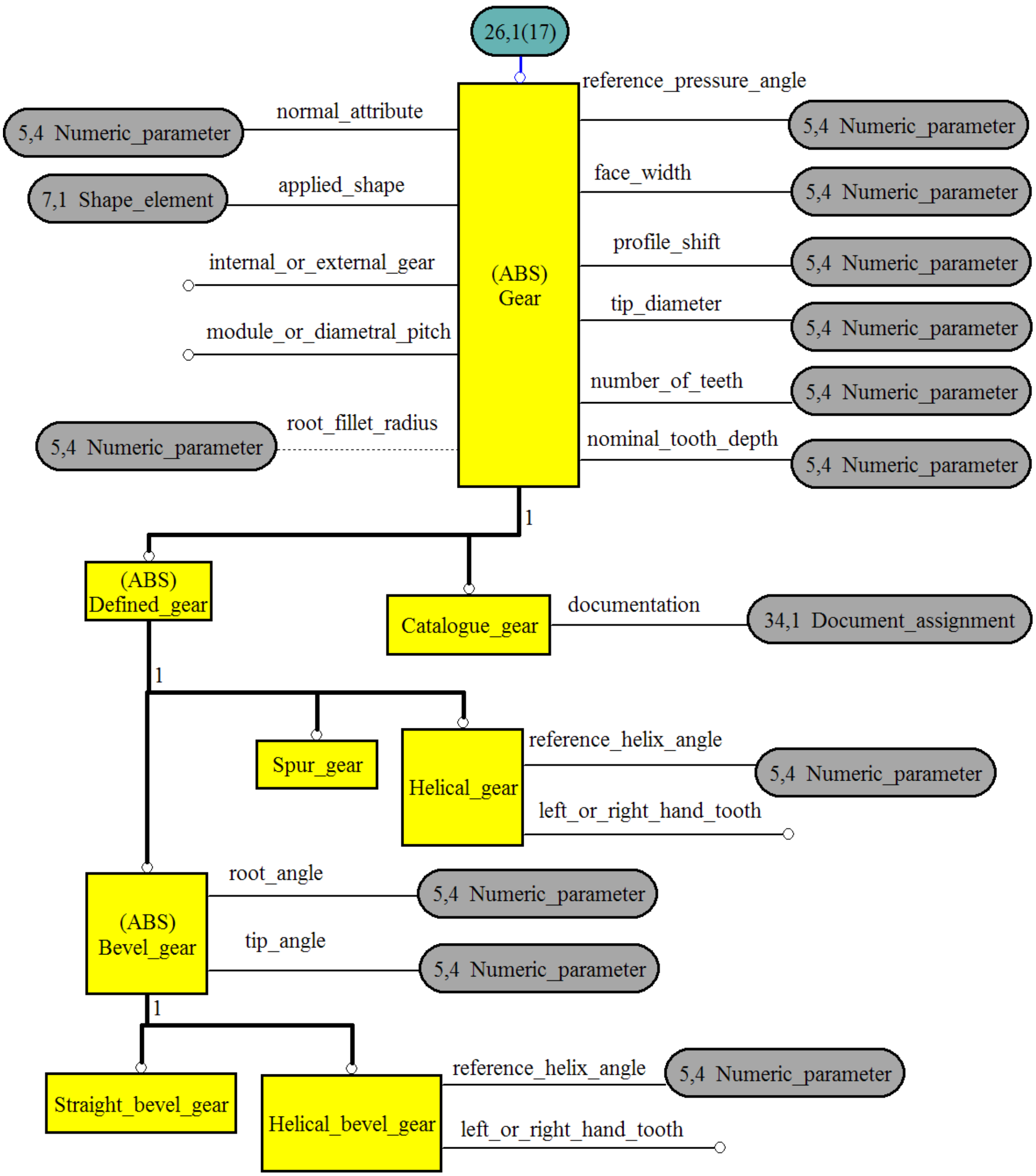


Figure G.26 — ARM EXPRESS-G diagram 26 of 34

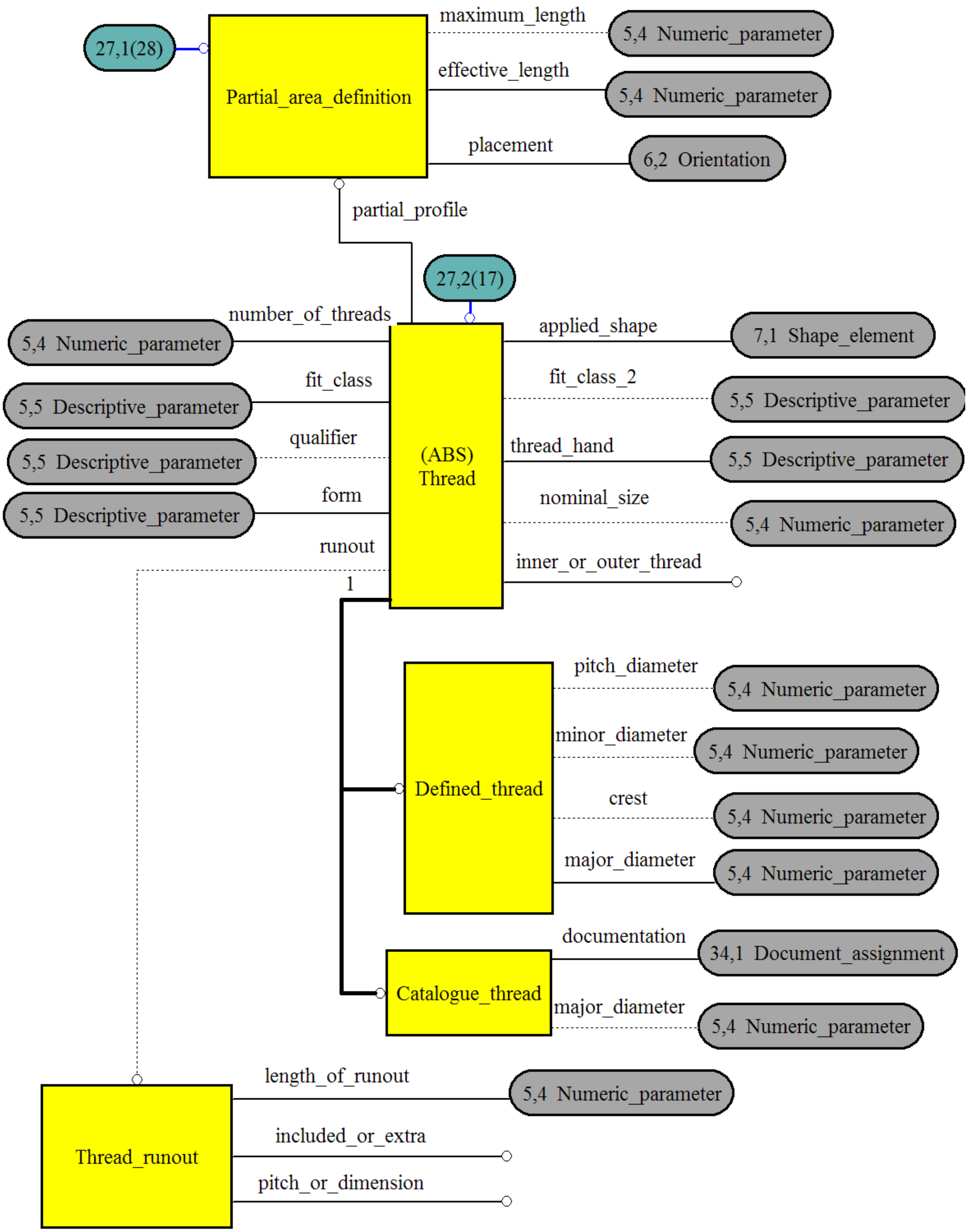


Figure G.27 — ARM EXPRESS-G diagram 27 of 34

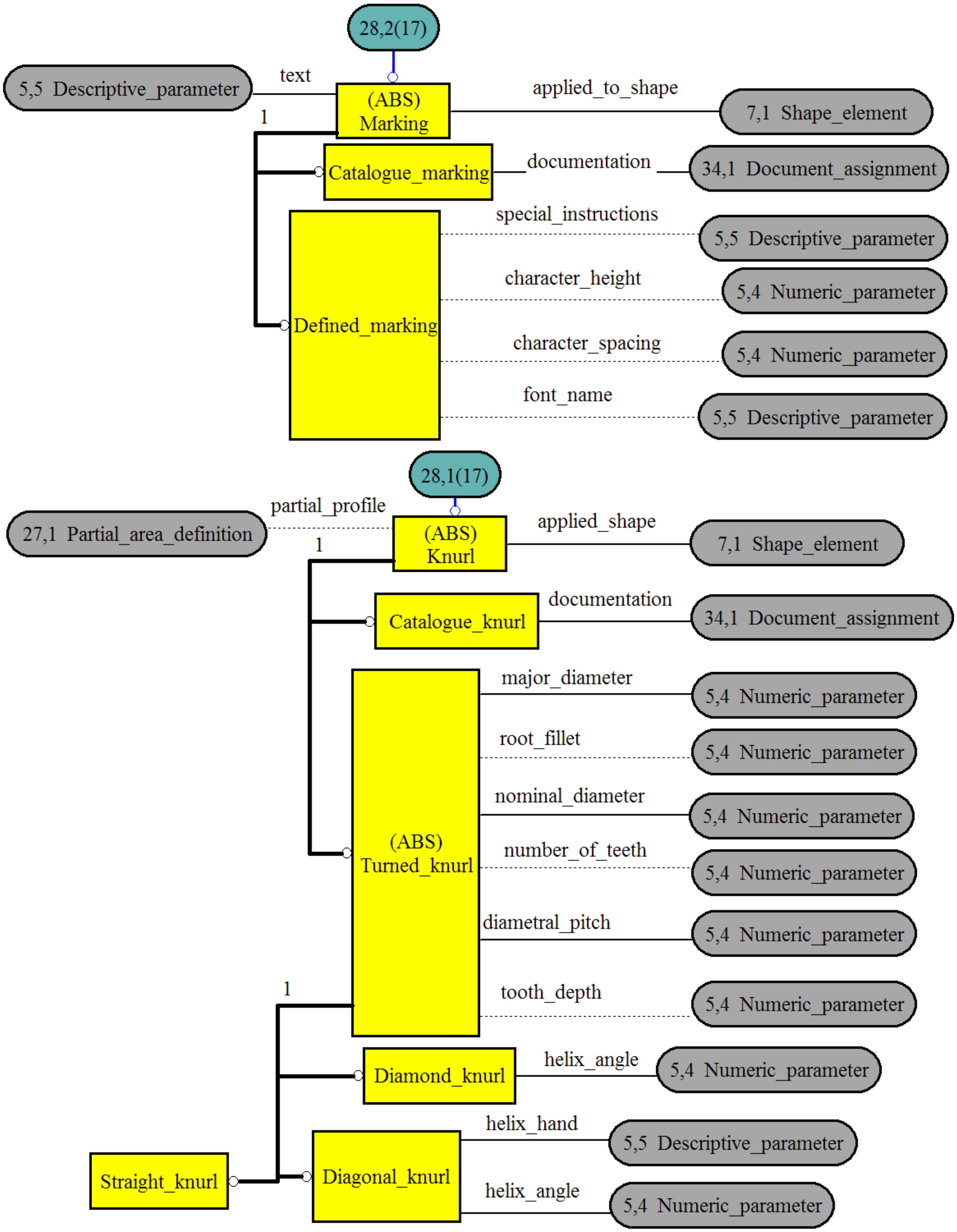


Figure G.28 — ARM EXPRESS-G diagram 28 of 34

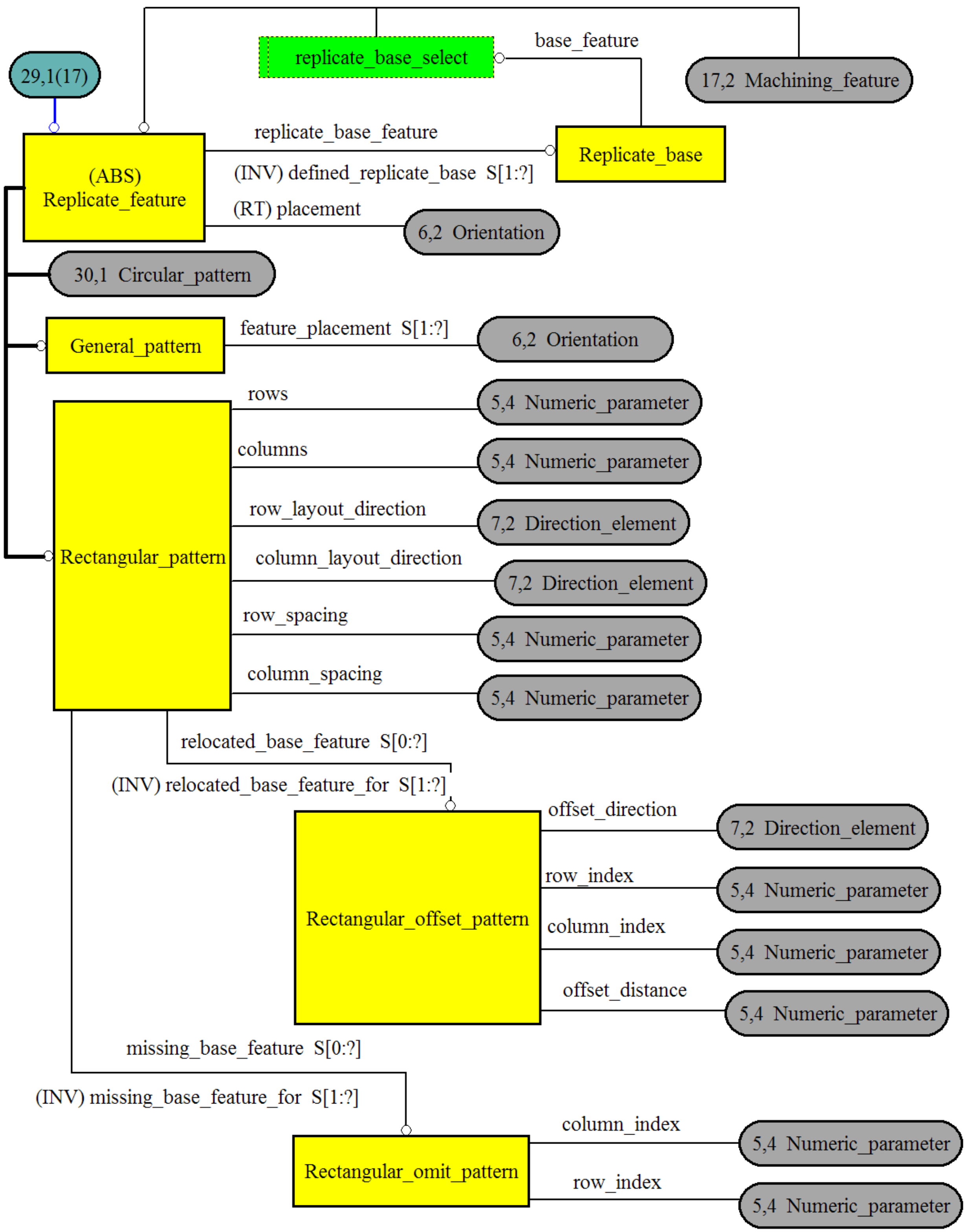


Figure G.29 — ARM EXPRESS-G diagram 29 of 34

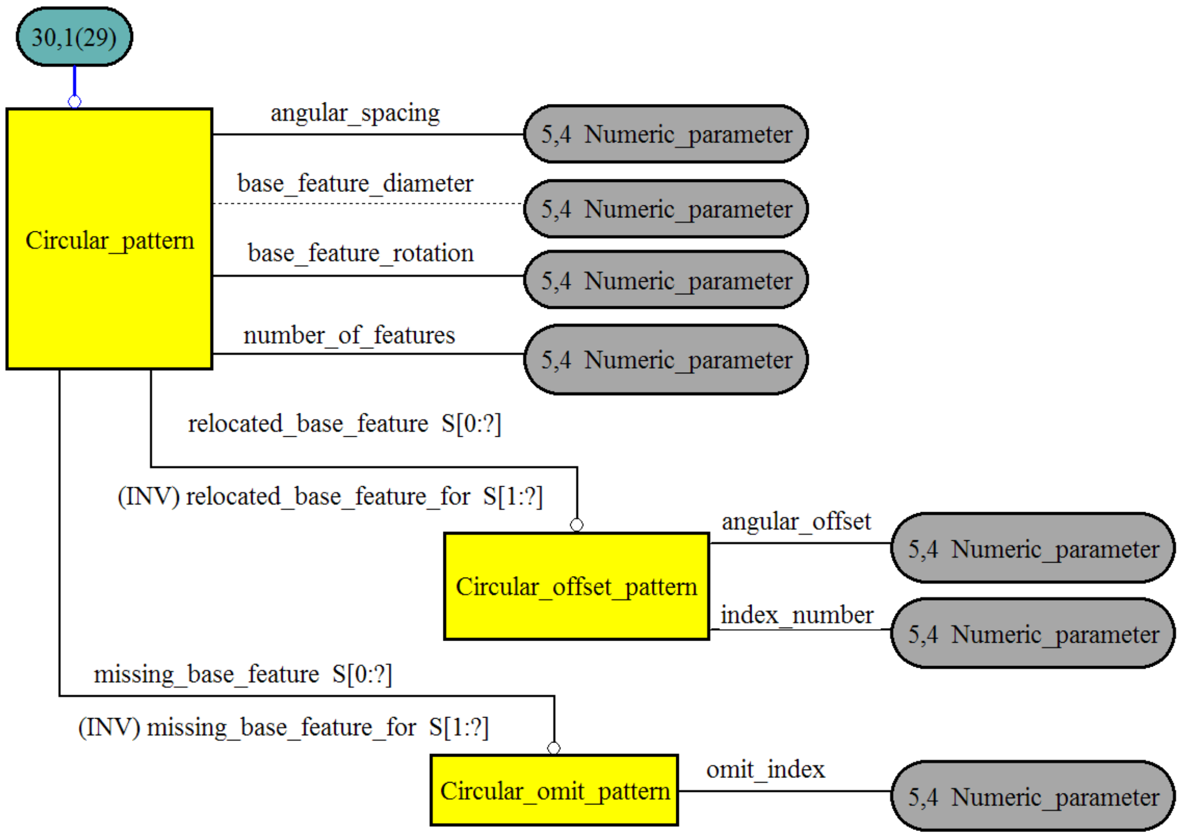


Figure G.30 — ARM EXPRESS-G diagram 30 of 34

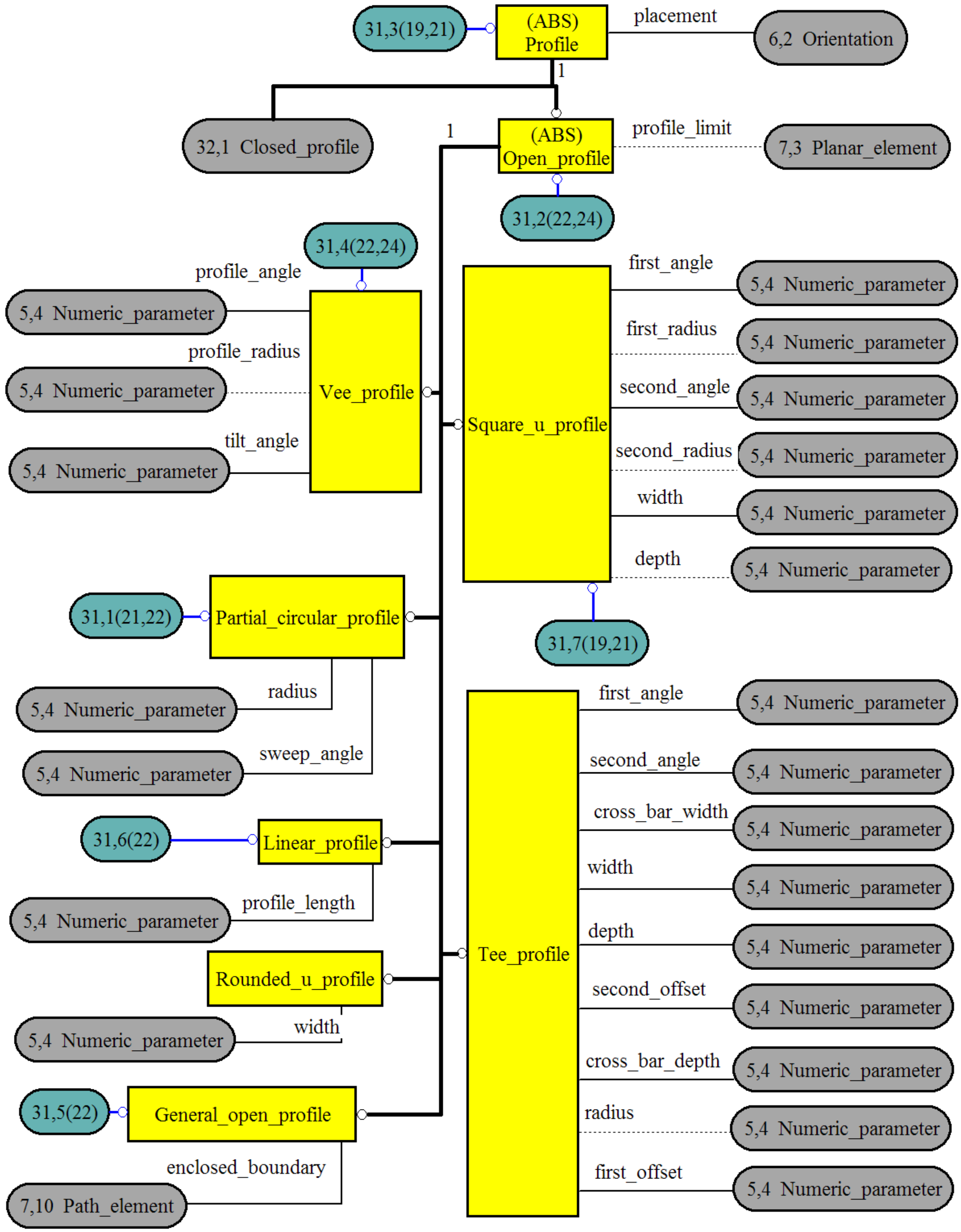


Figure G.31 — ARM EXPRESS-G diagram 31 of 34

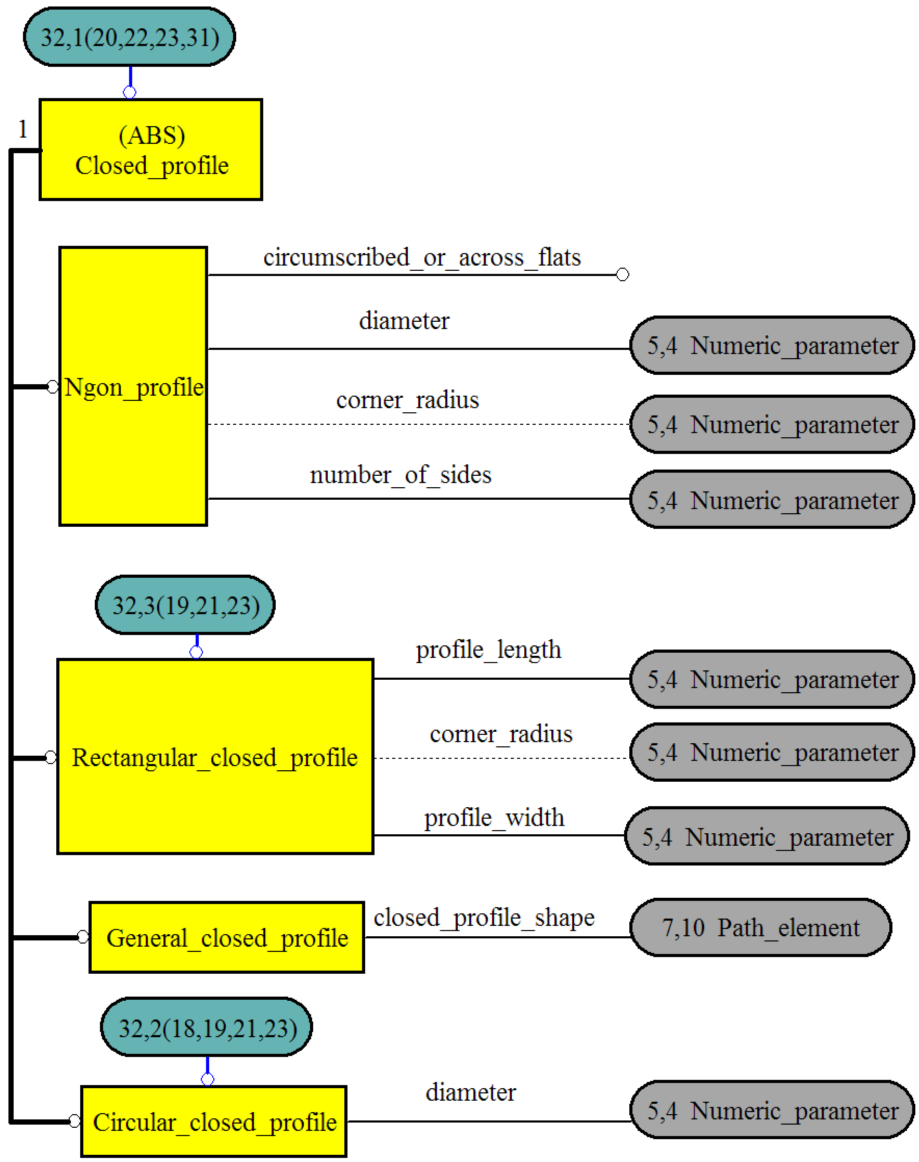


Figure G.32 — ARM EXPRESS-G diagram 32 of 34

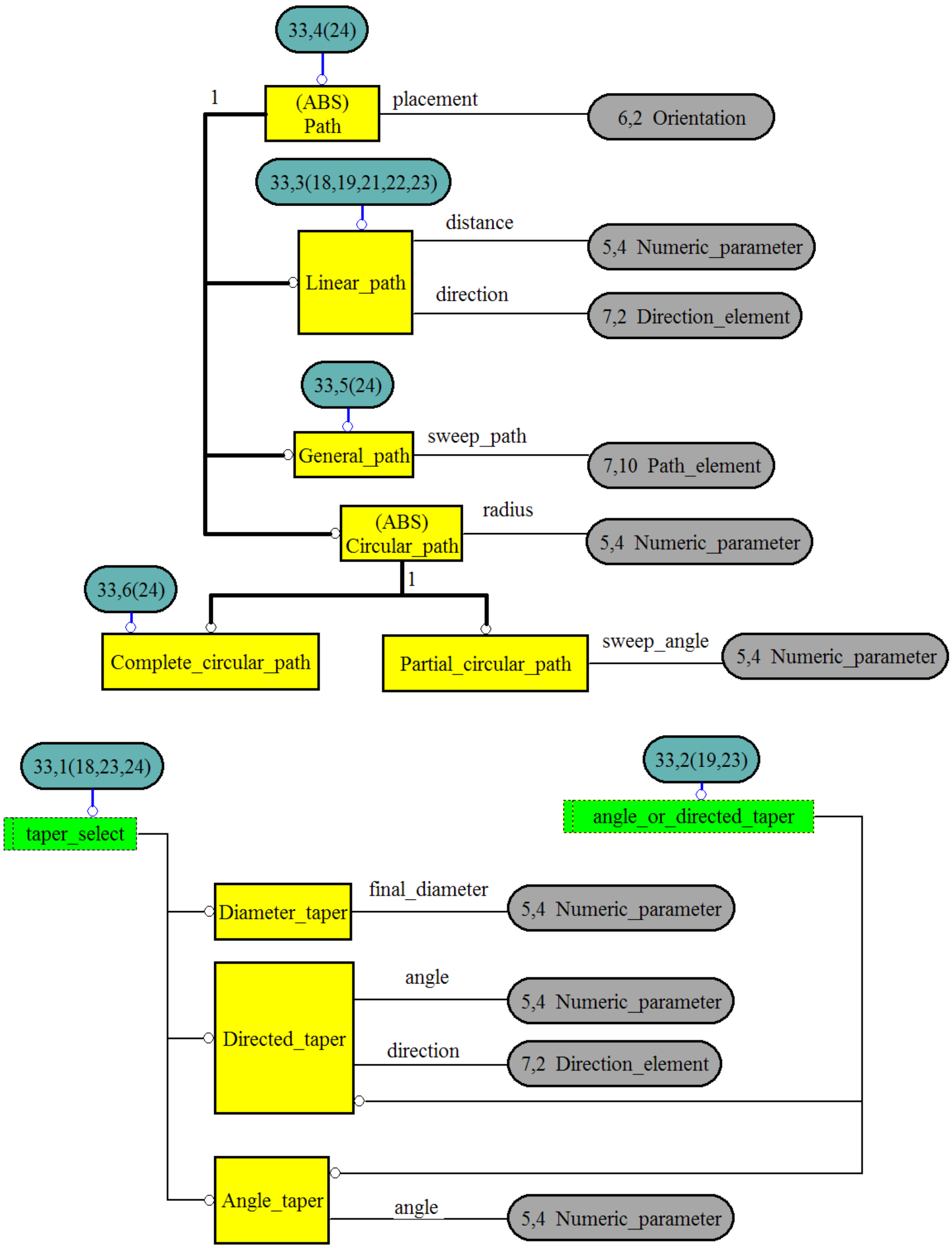


Figure G.33 — ARM EXPRESS-G diagram 33 of 34

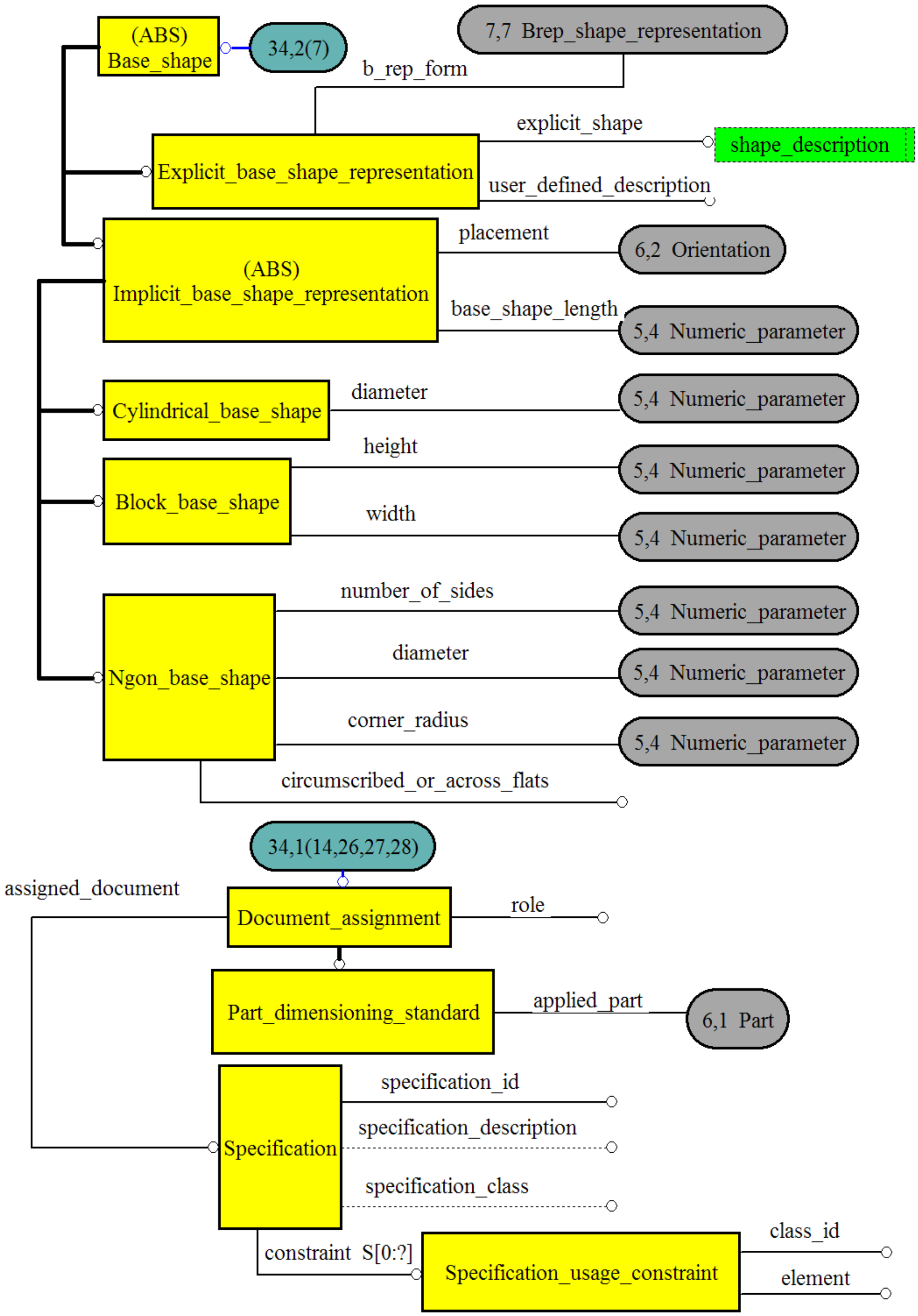


Figure G.34 — ARM EXPRESS-G diagram 34 of 34

Annex H

(informative)

AIM EXPRESS-G

Figure H.1 through H.29 correspond to the AIM EXPRESS annotated listing given in annex A. The figures use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex A of ISO 10303–11:2004.

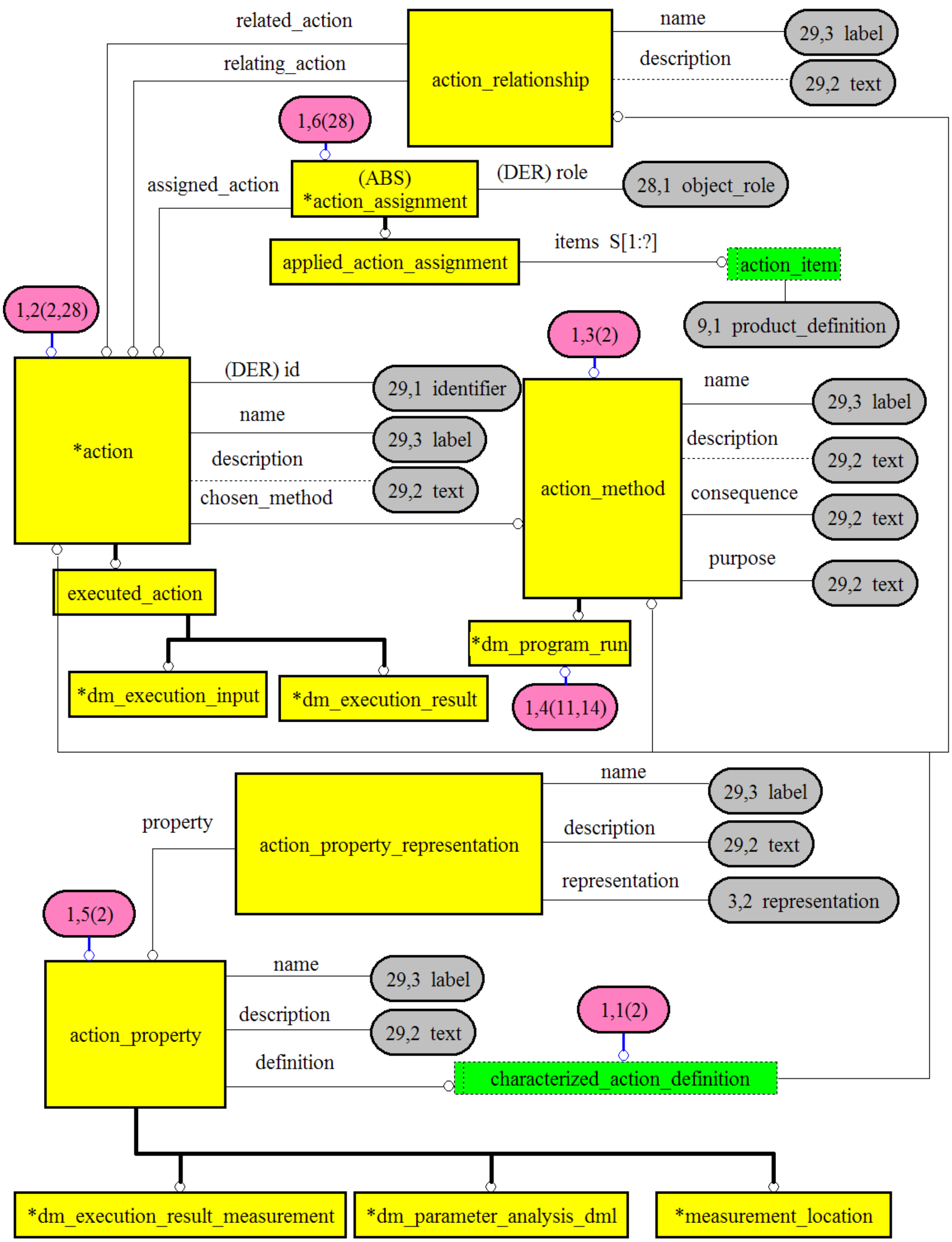


Figure H.1 — AIM EXPRESS-G diagram 1 of 29

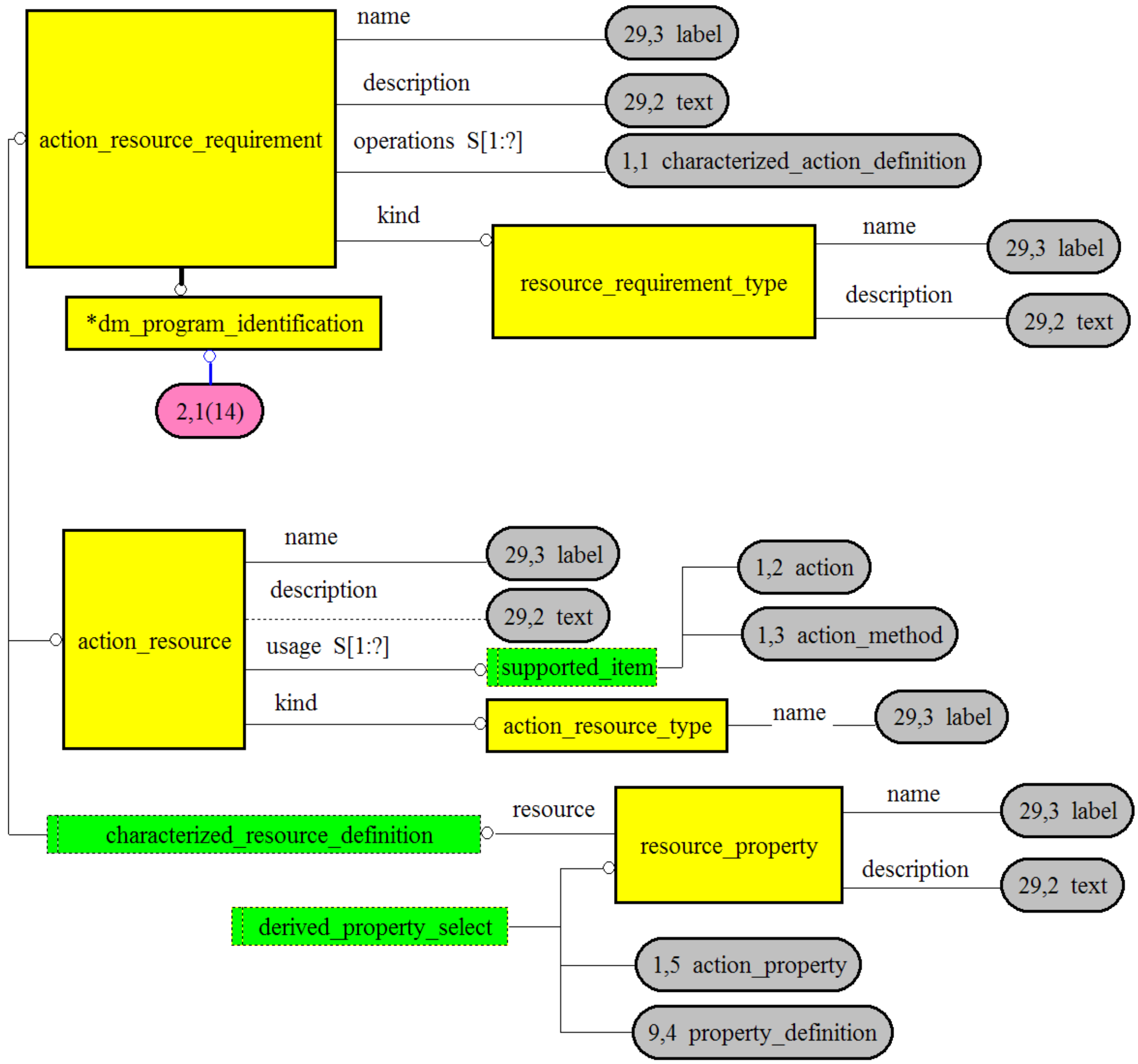


Figure H.2 — AIM EXPRESS-G diagram 2 of 29

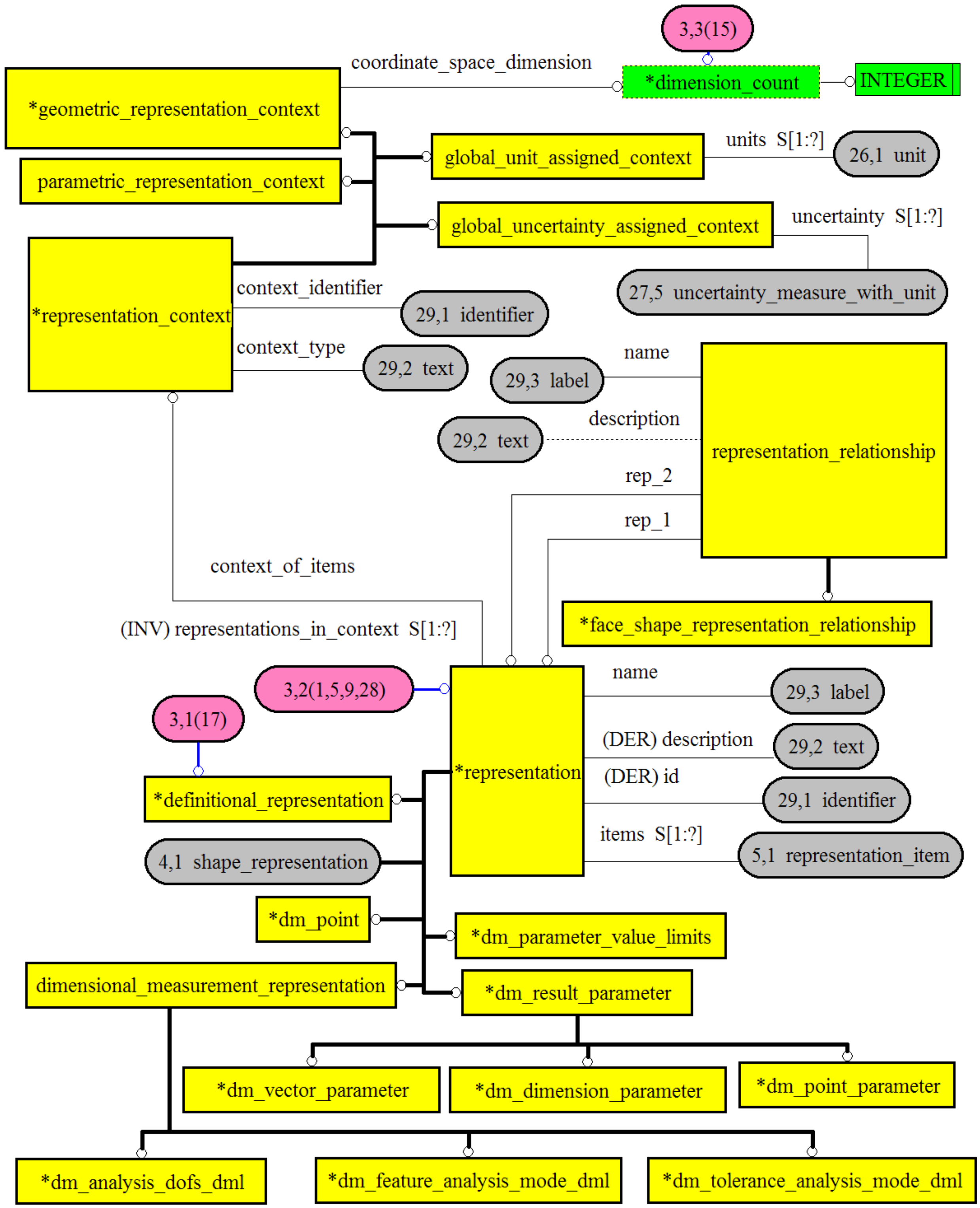


Figure H.3 — AIM EXPRESS-G diagram 3 of 29

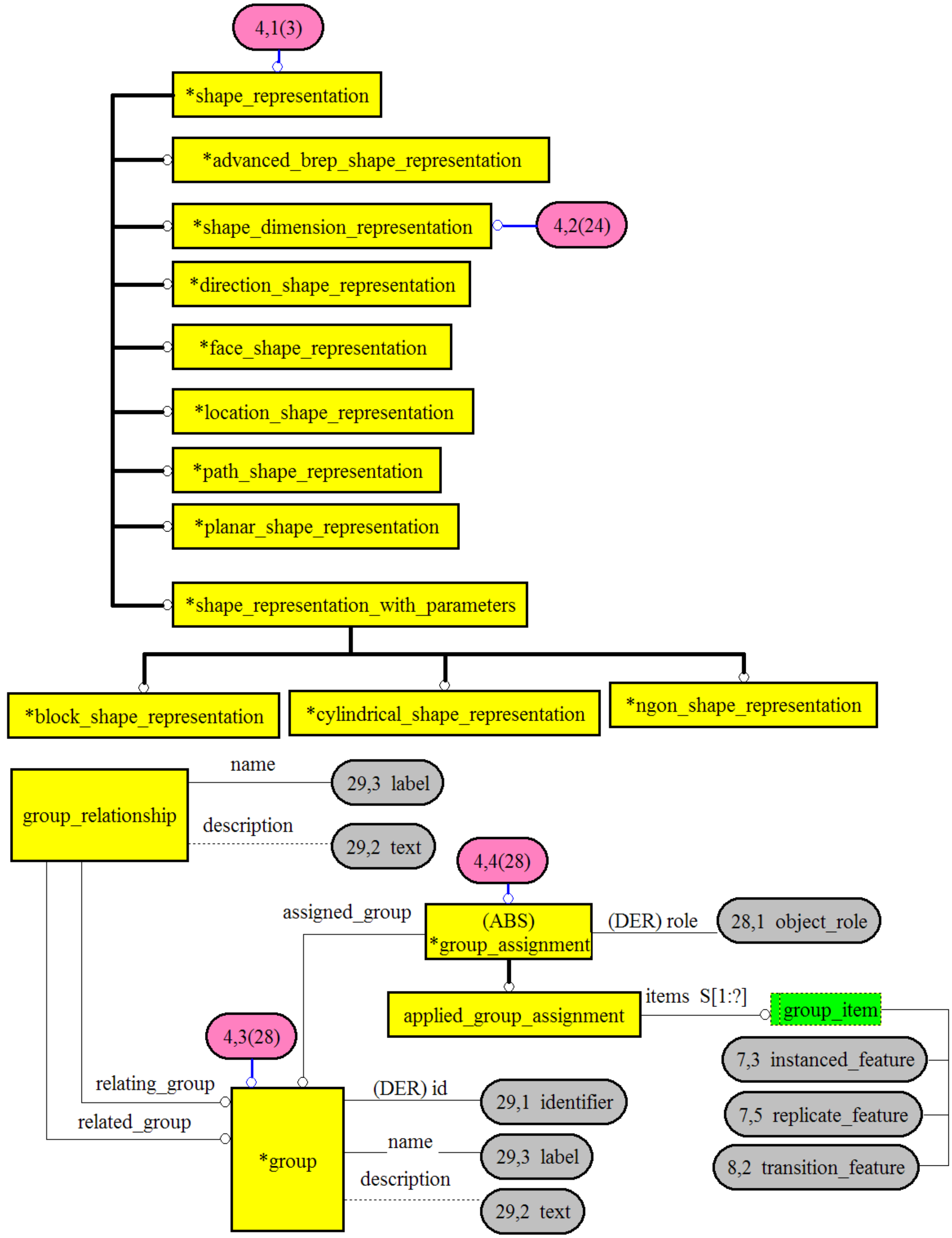


Figure H.4 — AIM EXPRESS-G diagram 4 of 29

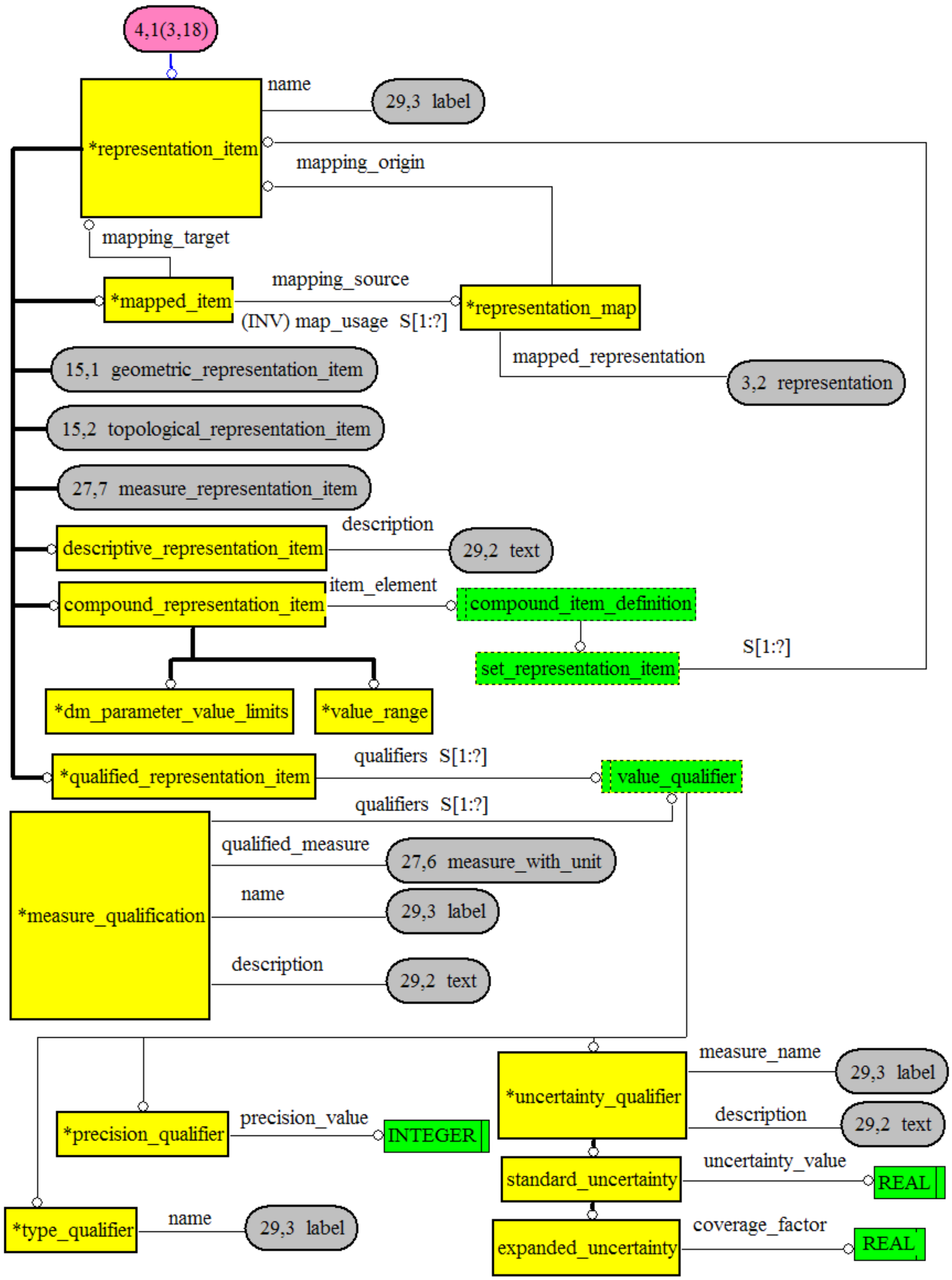


Figure H.5 — AIM EXPRESS-G diagram 5 of 29

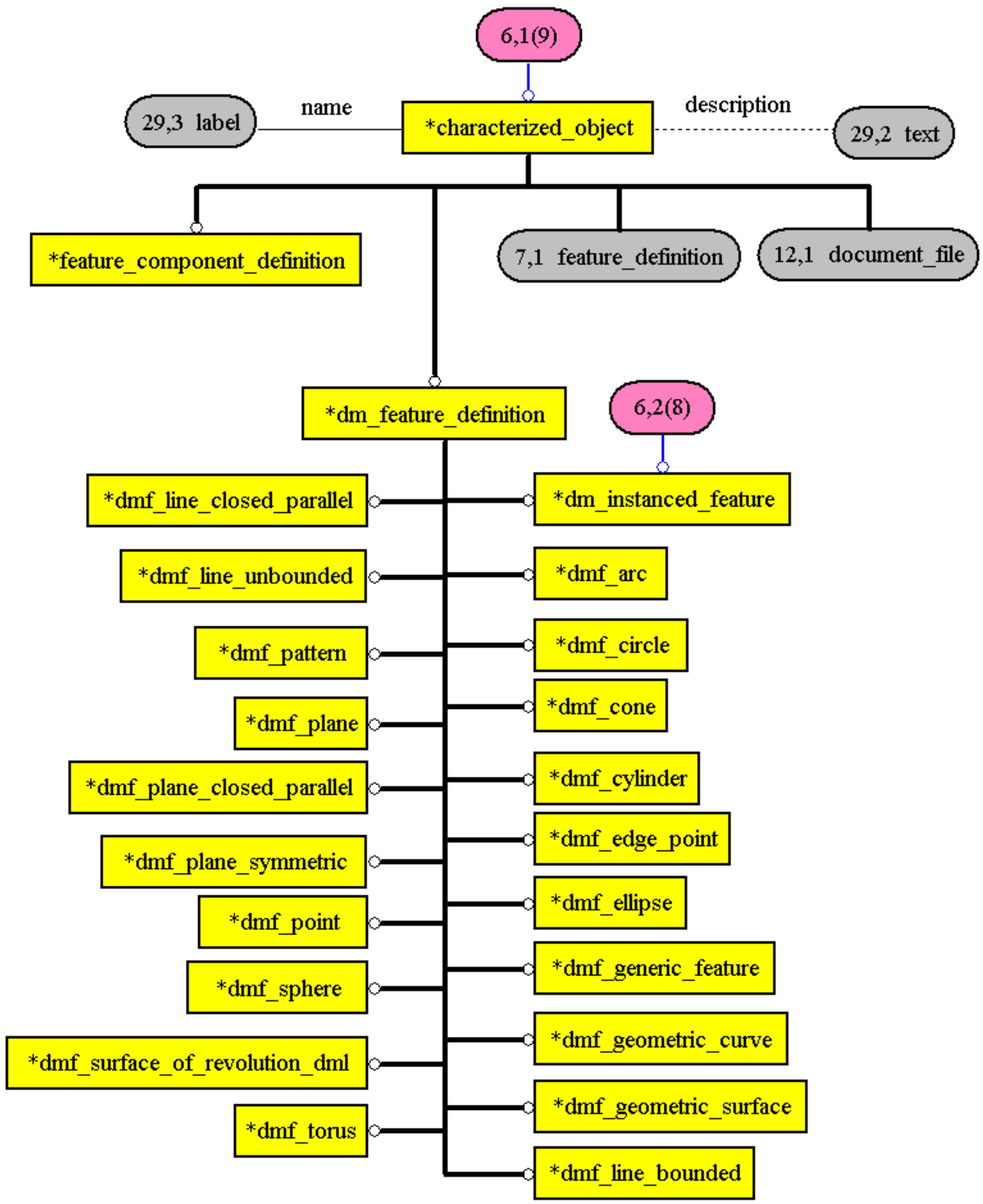


Figure H.6 — AIM EXPRESS-G diagram 6 of 29

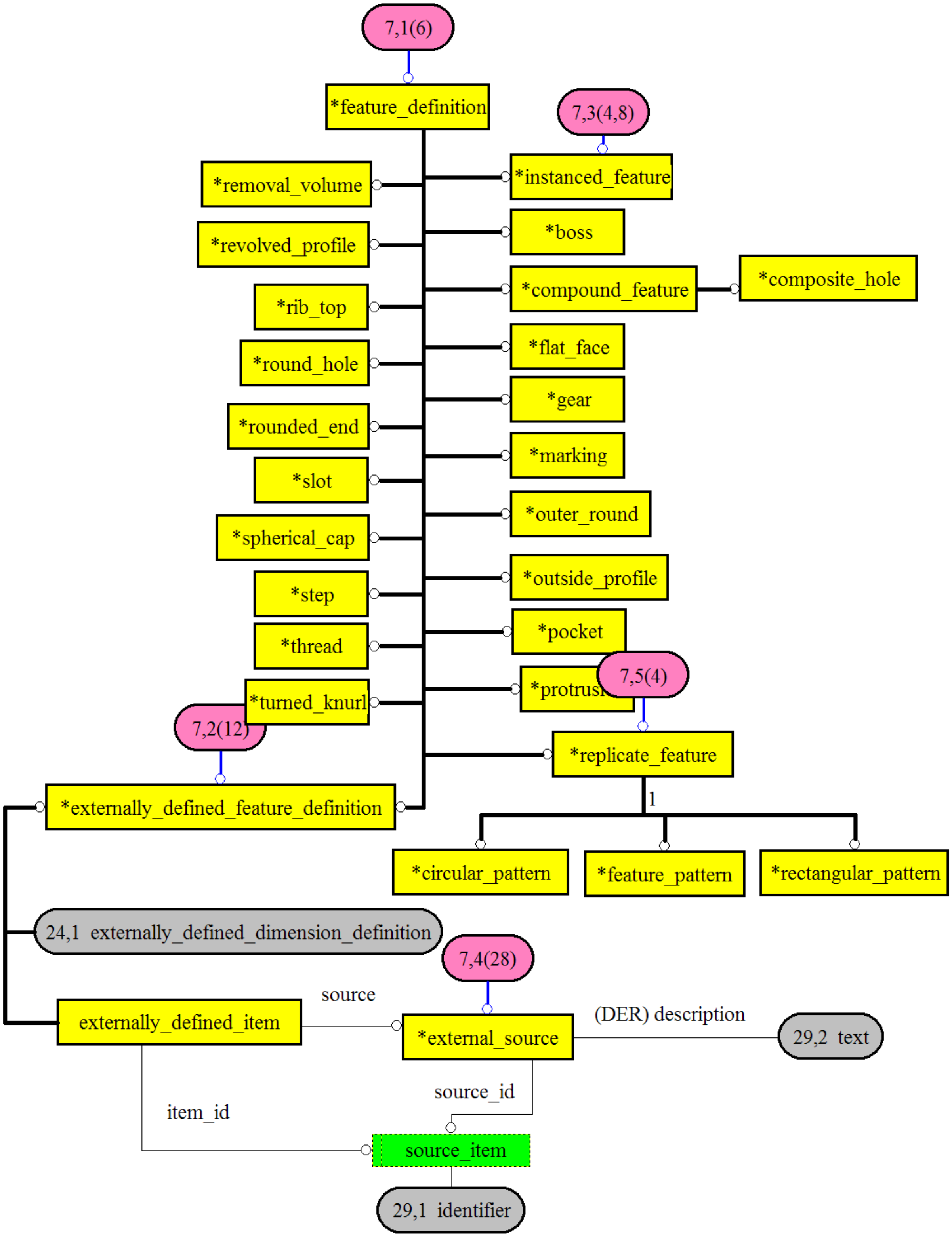


Figure H.7 — AIM EXPRESS-G diagram 7 of 29

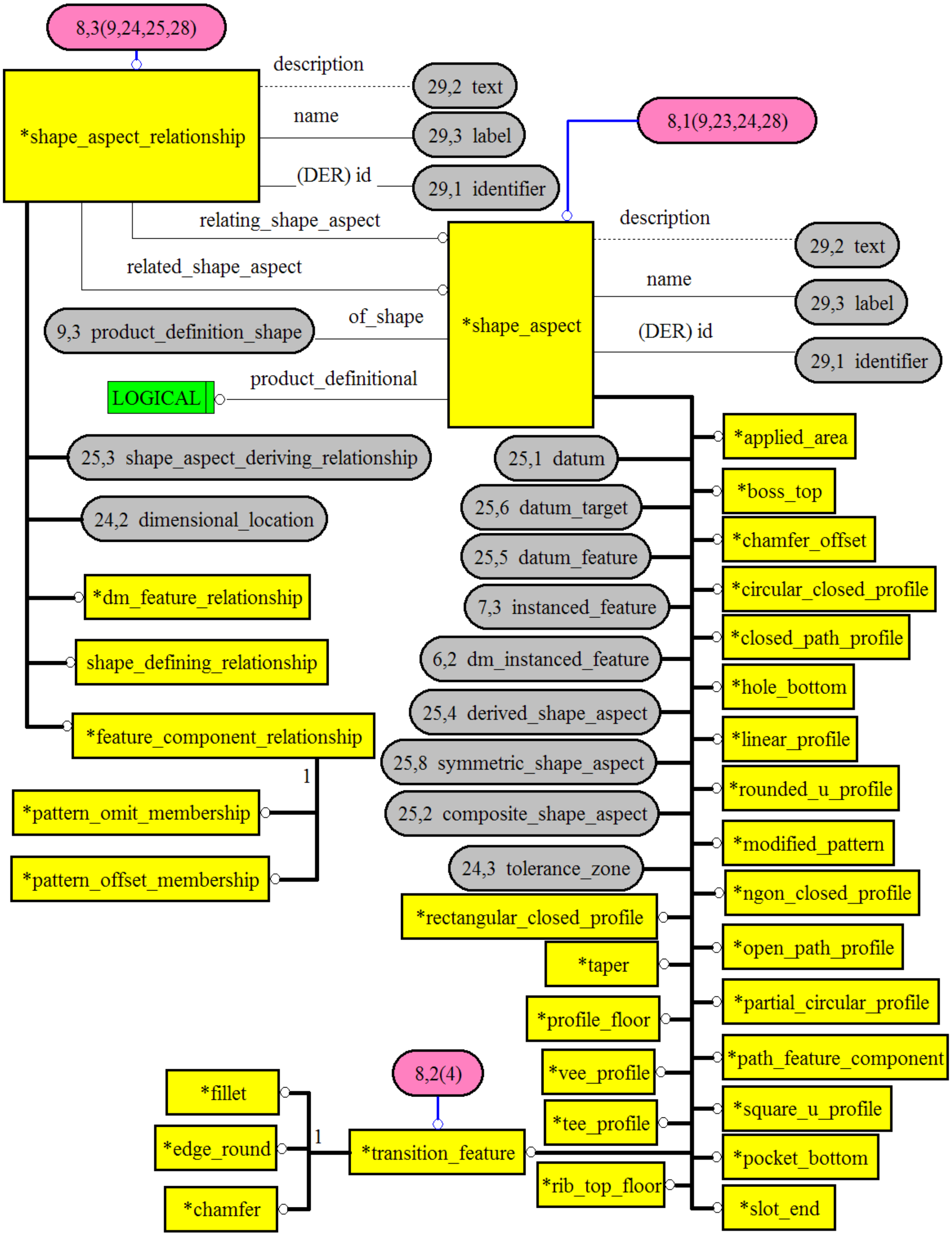


Figure H.8 — AIM EXPRESS-G diagram 8 of 29

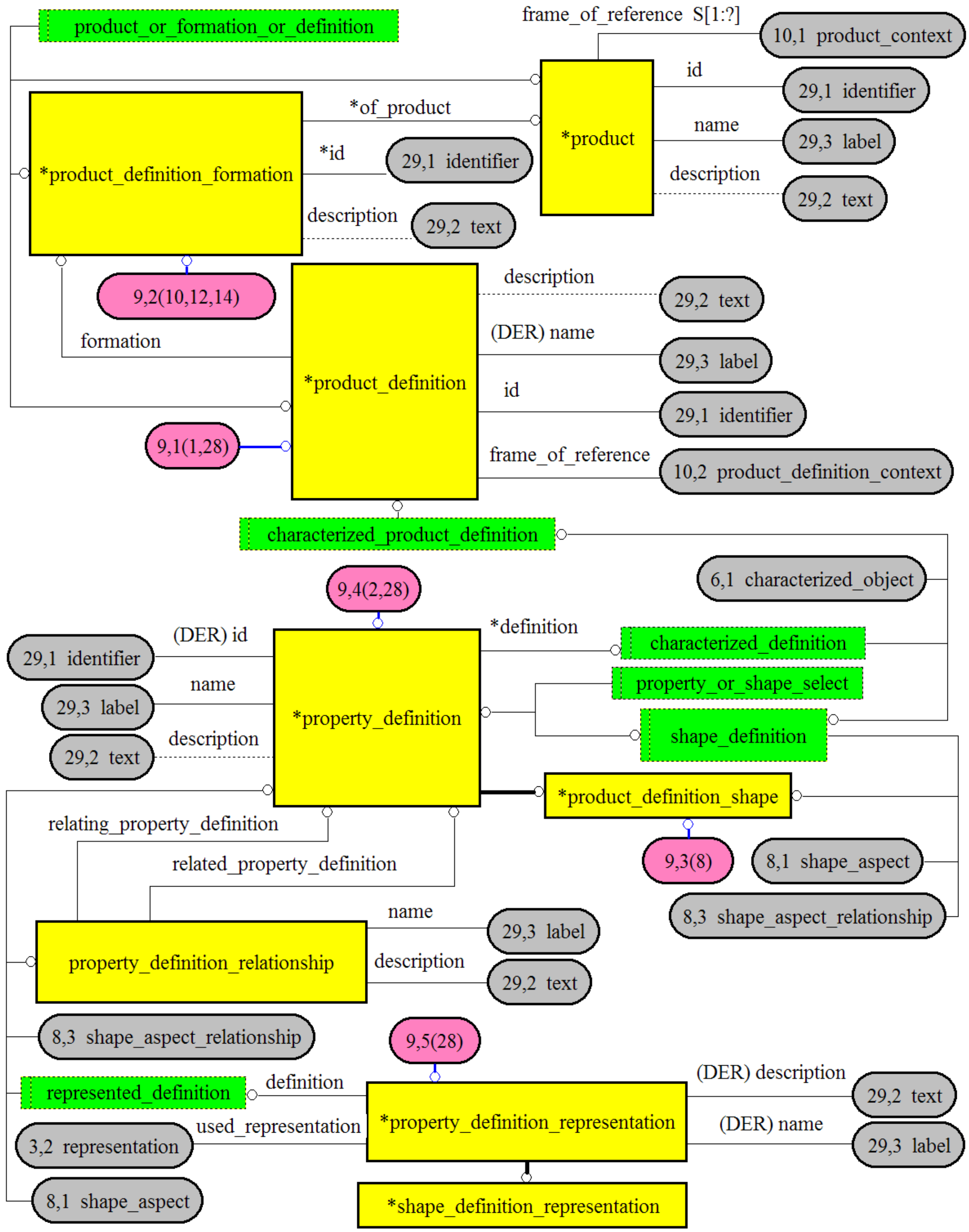


Figure H.9 — AIM EXPRESS-G diagram 9 of 29

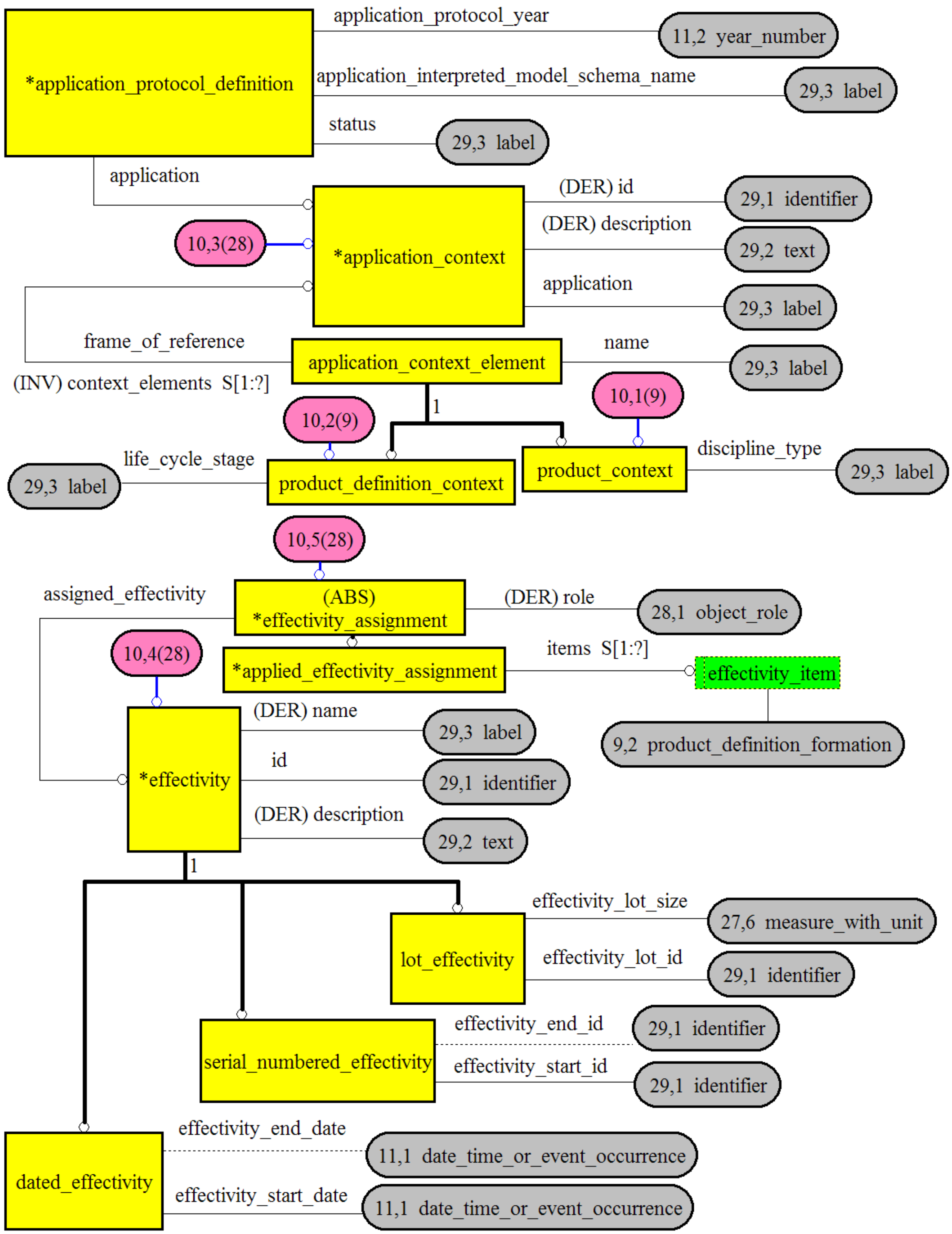


Figure H.10 — AIM EXPRESS-G diagram 10 of 29

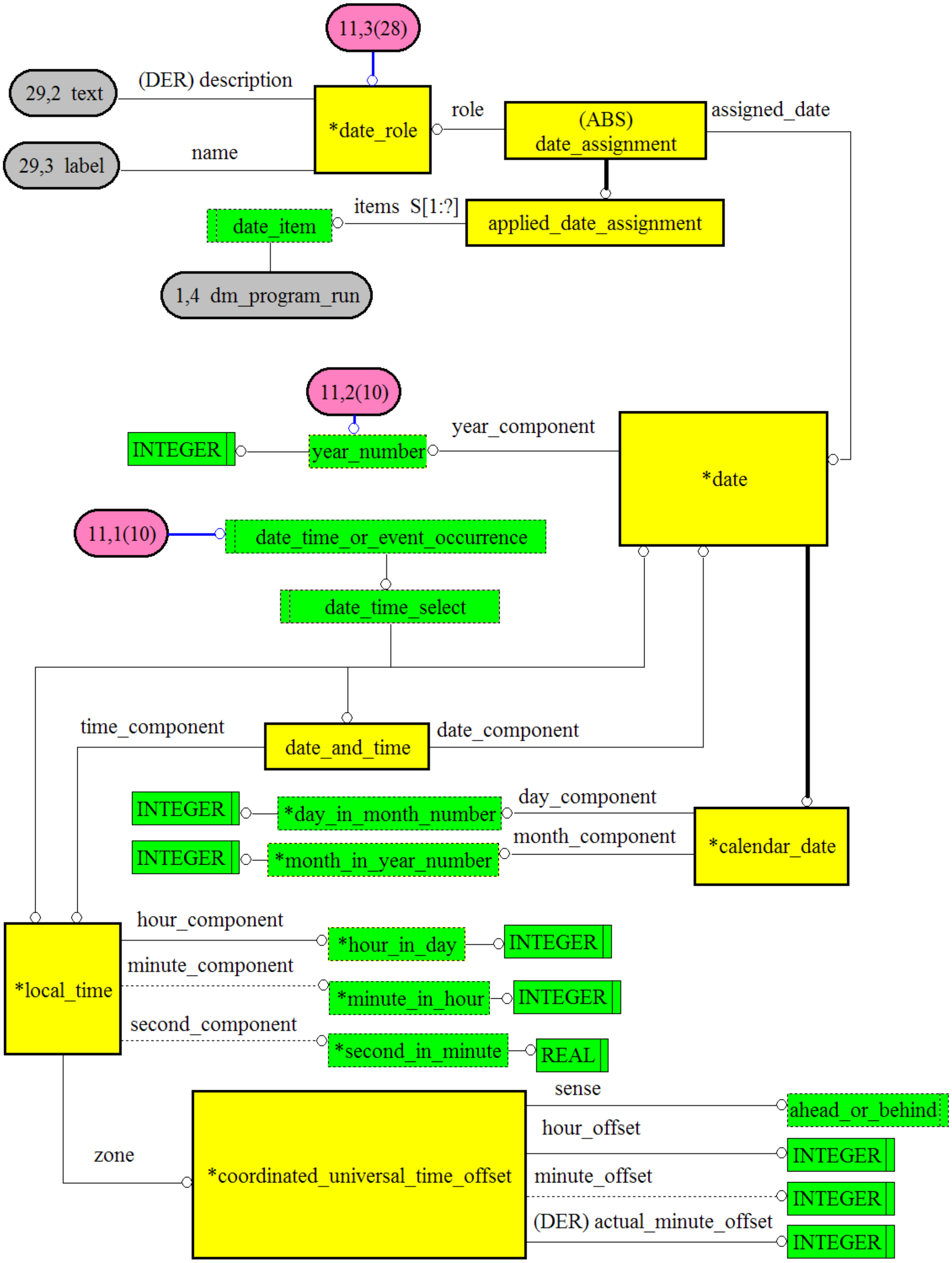


Figure H-11 — AIM EXPRESS-G diagram 11 of 29

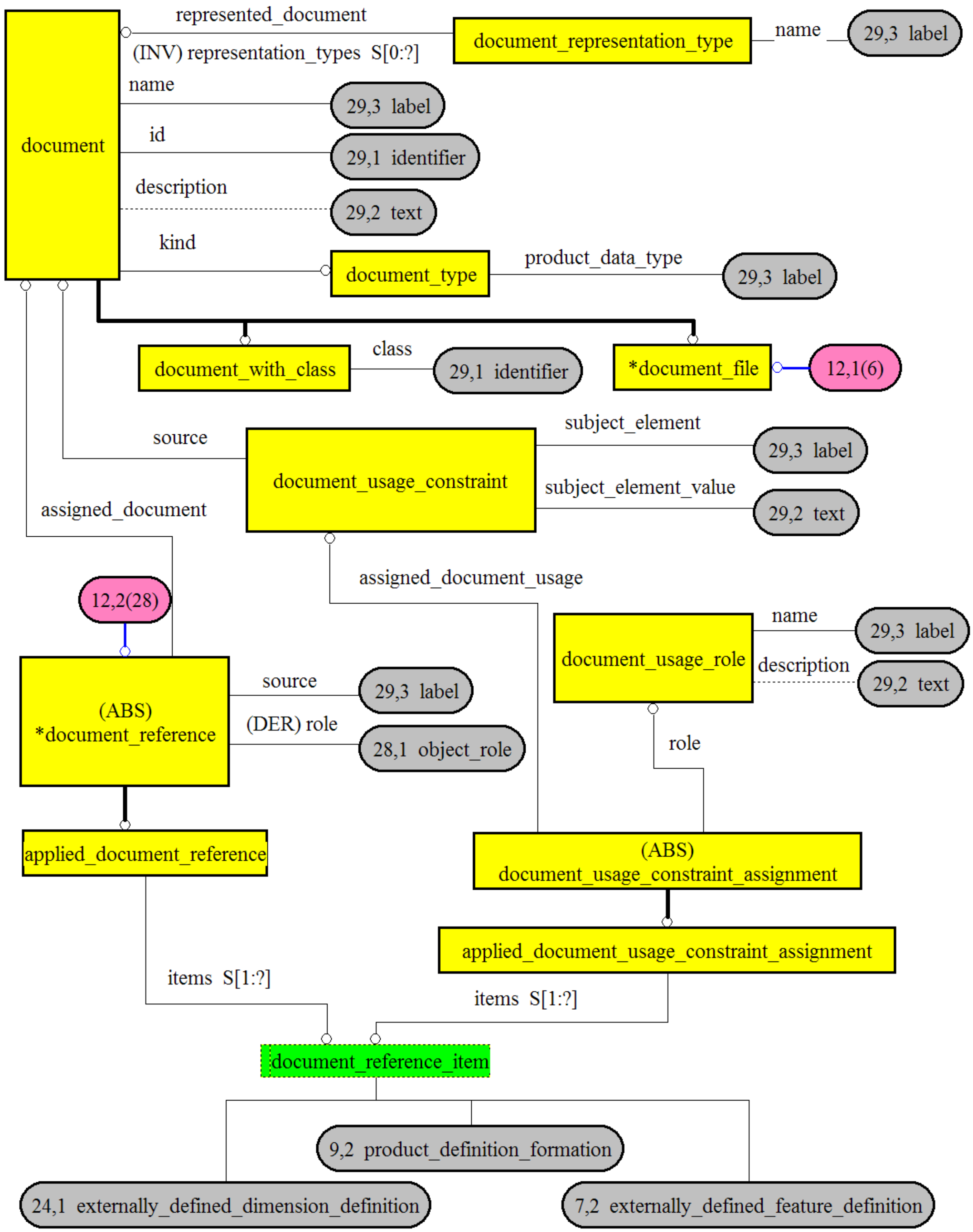


Figure H.12 — AIM EXPRESS-G diagram 12 of 29

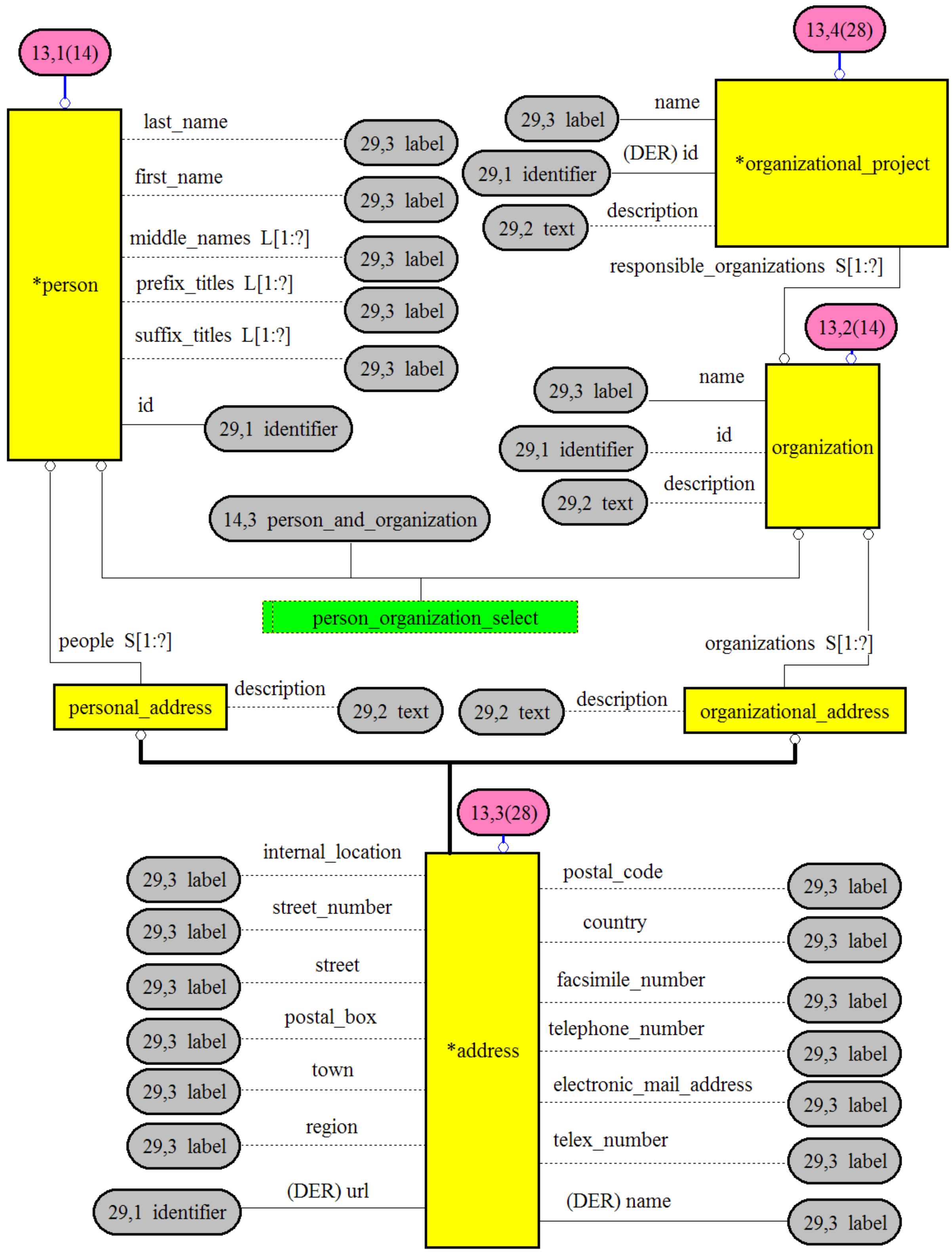


Figure H.13 — AIM EXPRESS-G diagram 13 of 29

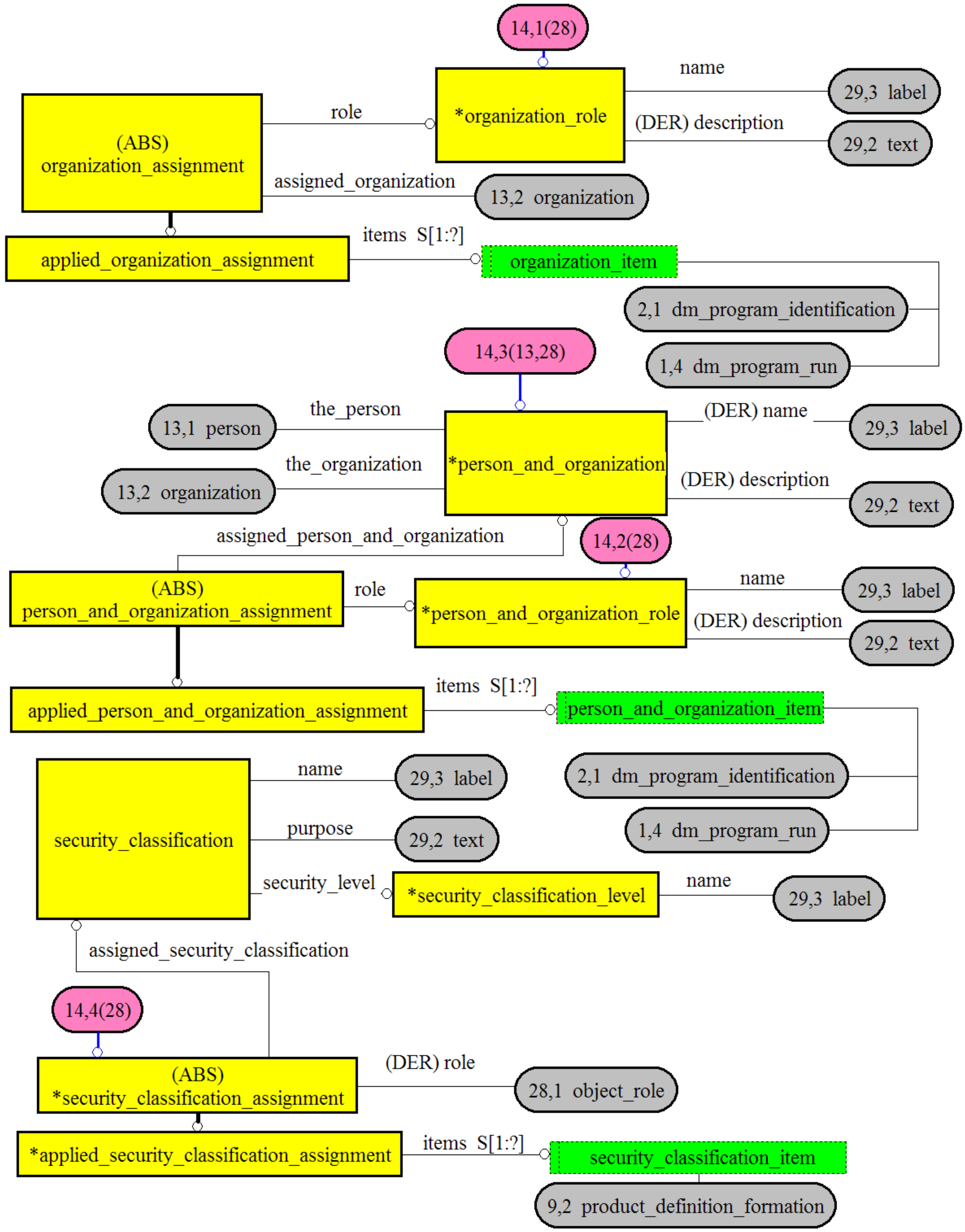


Figure H.14 — AIM EXPRESS-G diagram 14 of 29

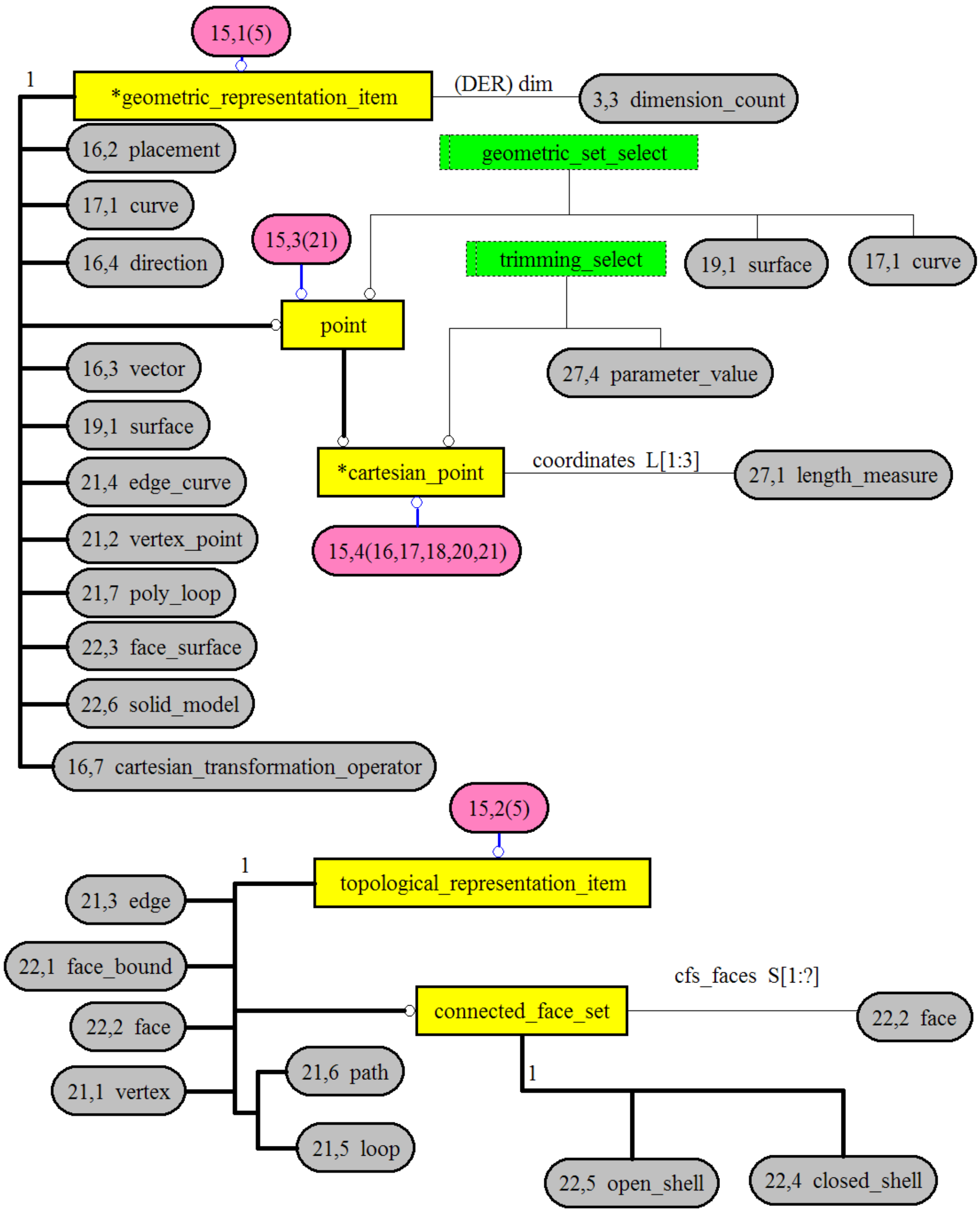


Figure H.15 — AIM EXPRESS-G diagram 15 of 29

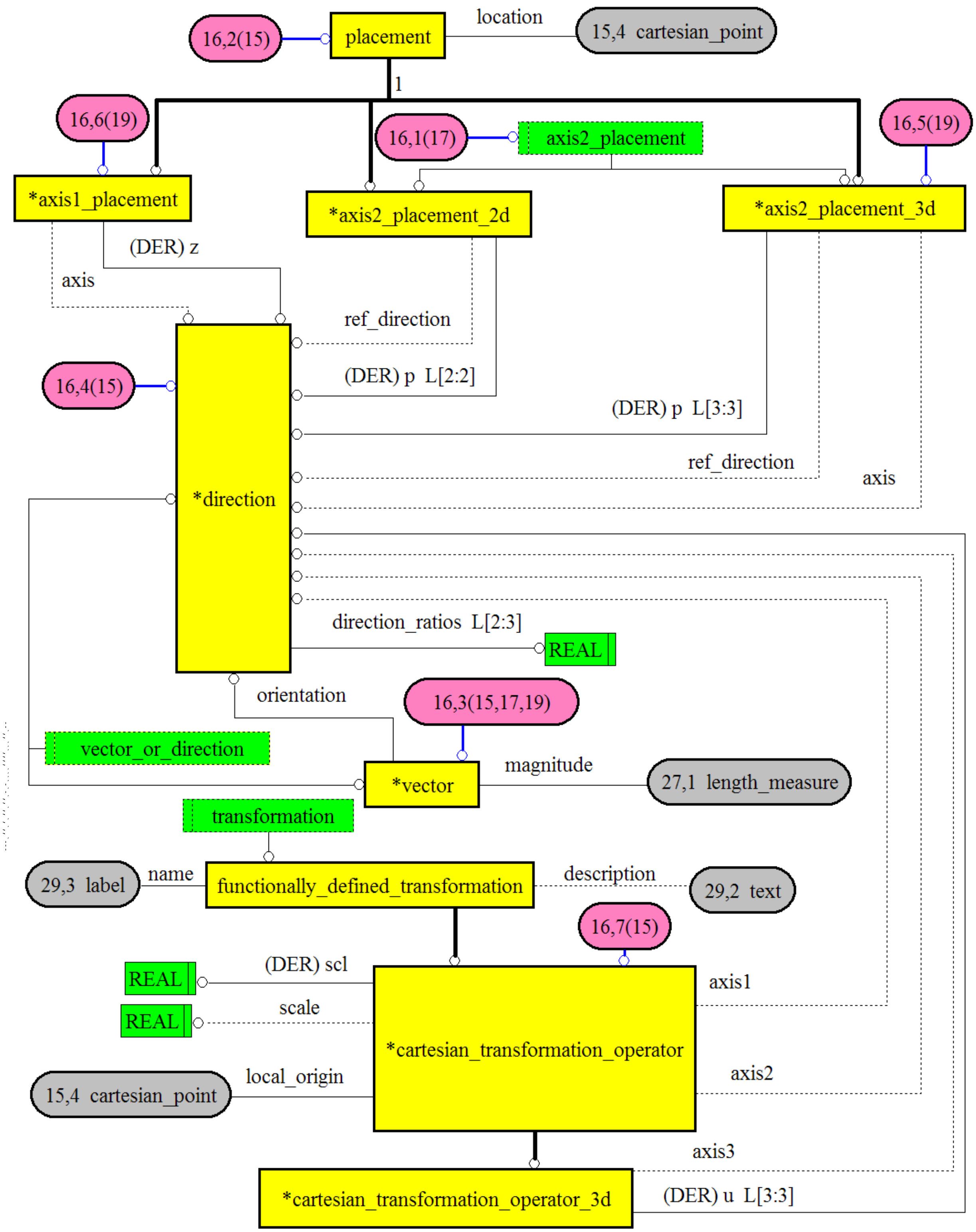


Figure H.16 — AIM EXPRESS-G diagram 16 of 29

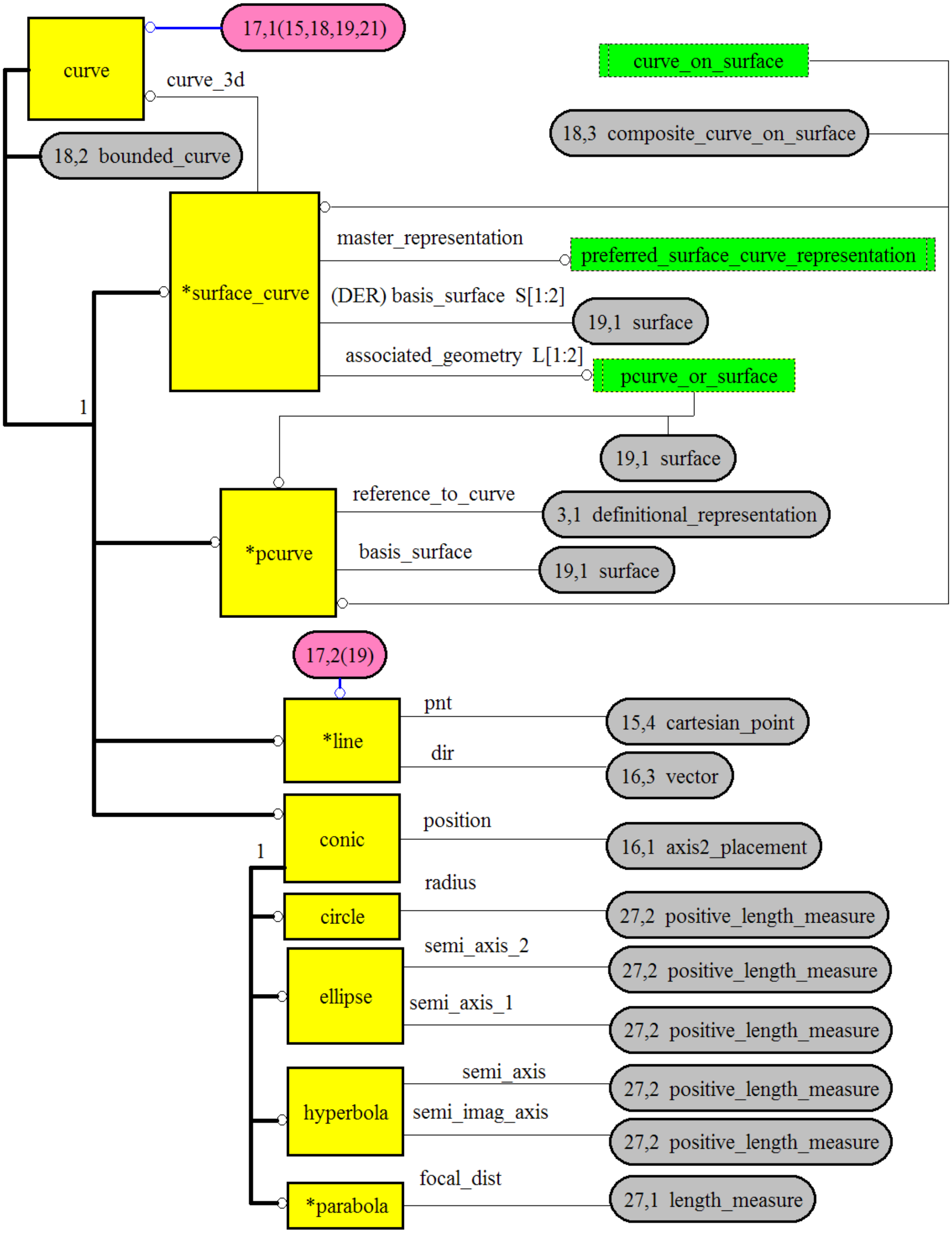


Figure H.17 — AIM EXPRESS-G diagram 17 of 29

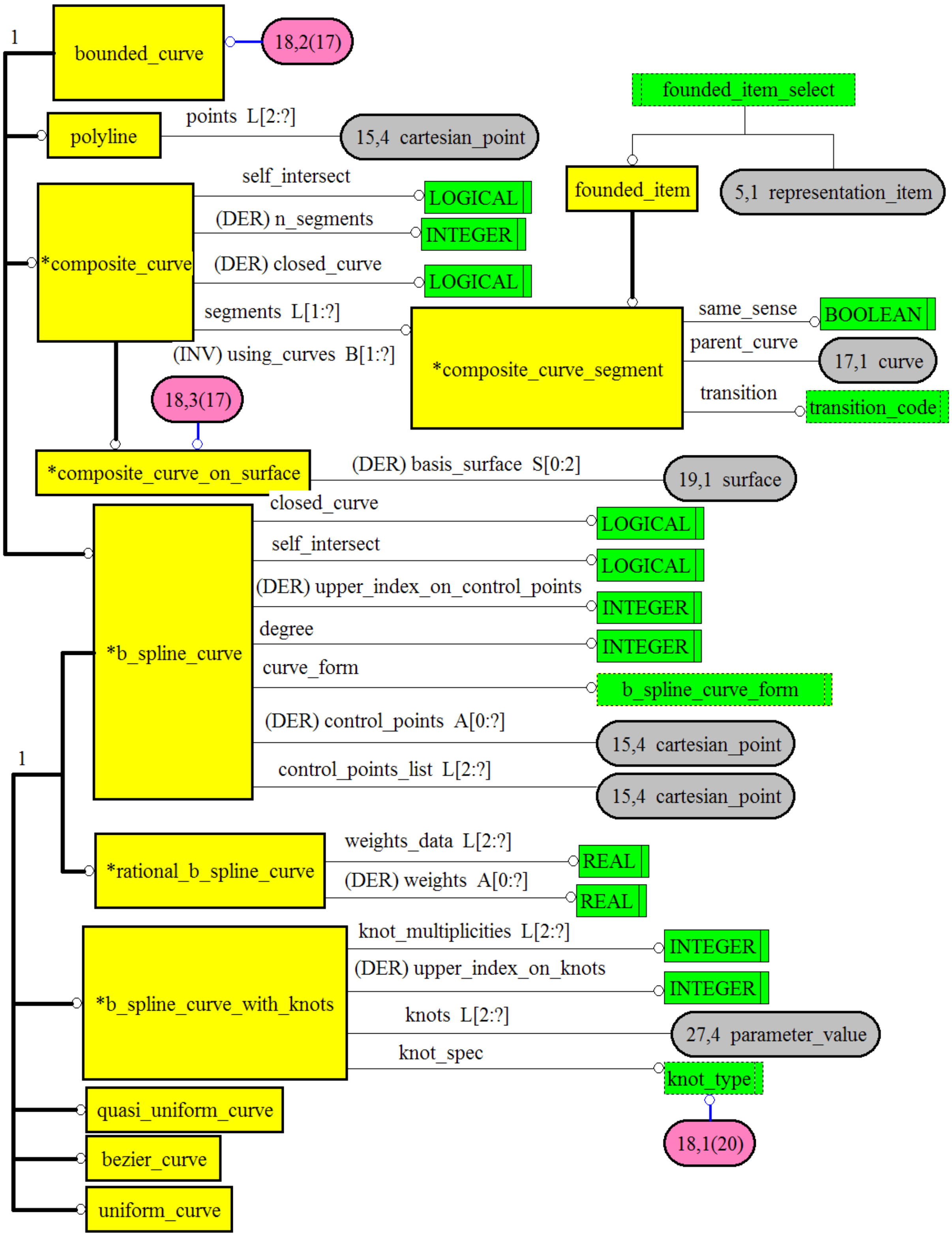


Figure H.18 — AIM EXPRESS-G diagram 18 of 29

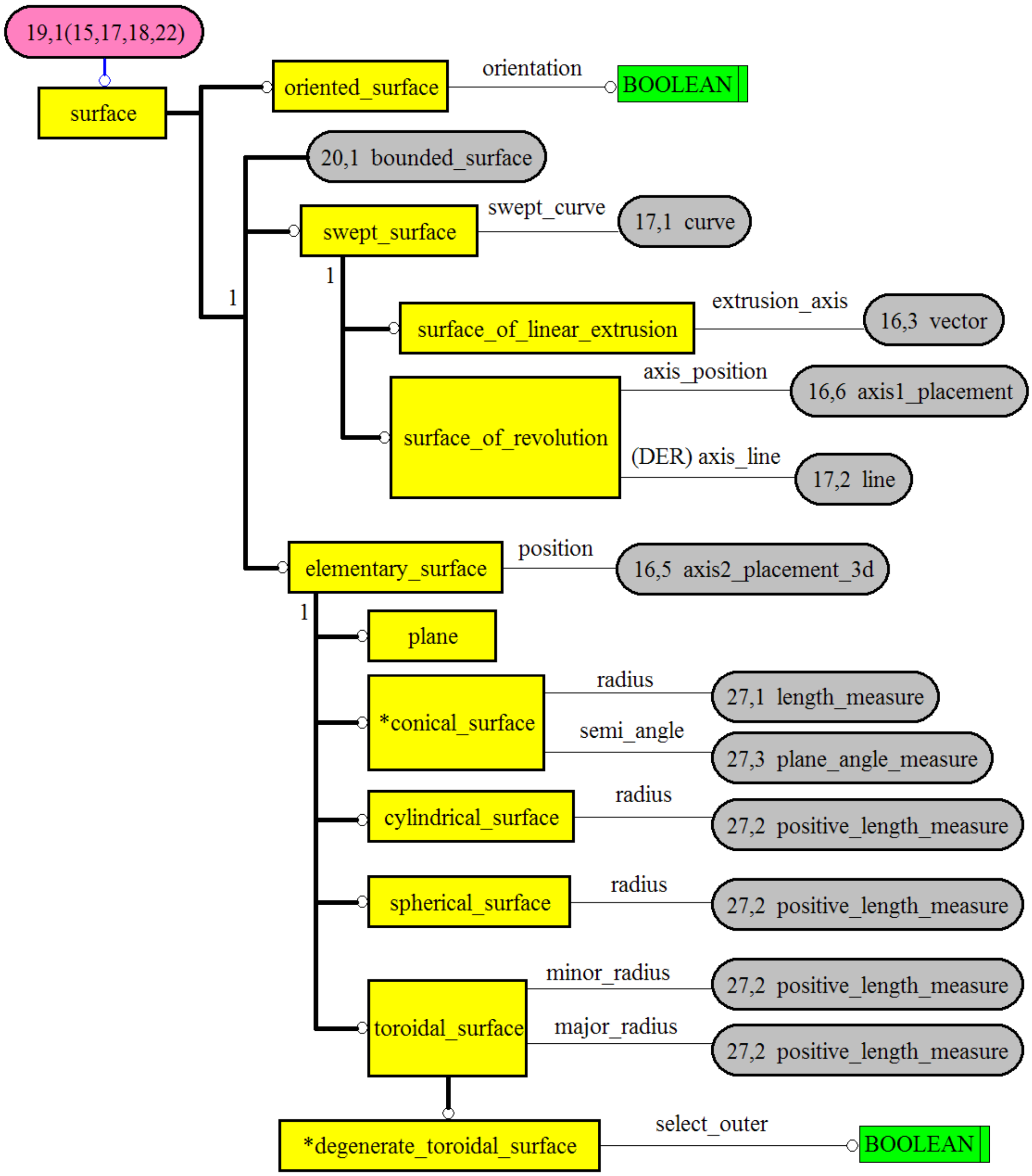


Figure H.19 — AIM EXPRESS-G diagram 19 of 29

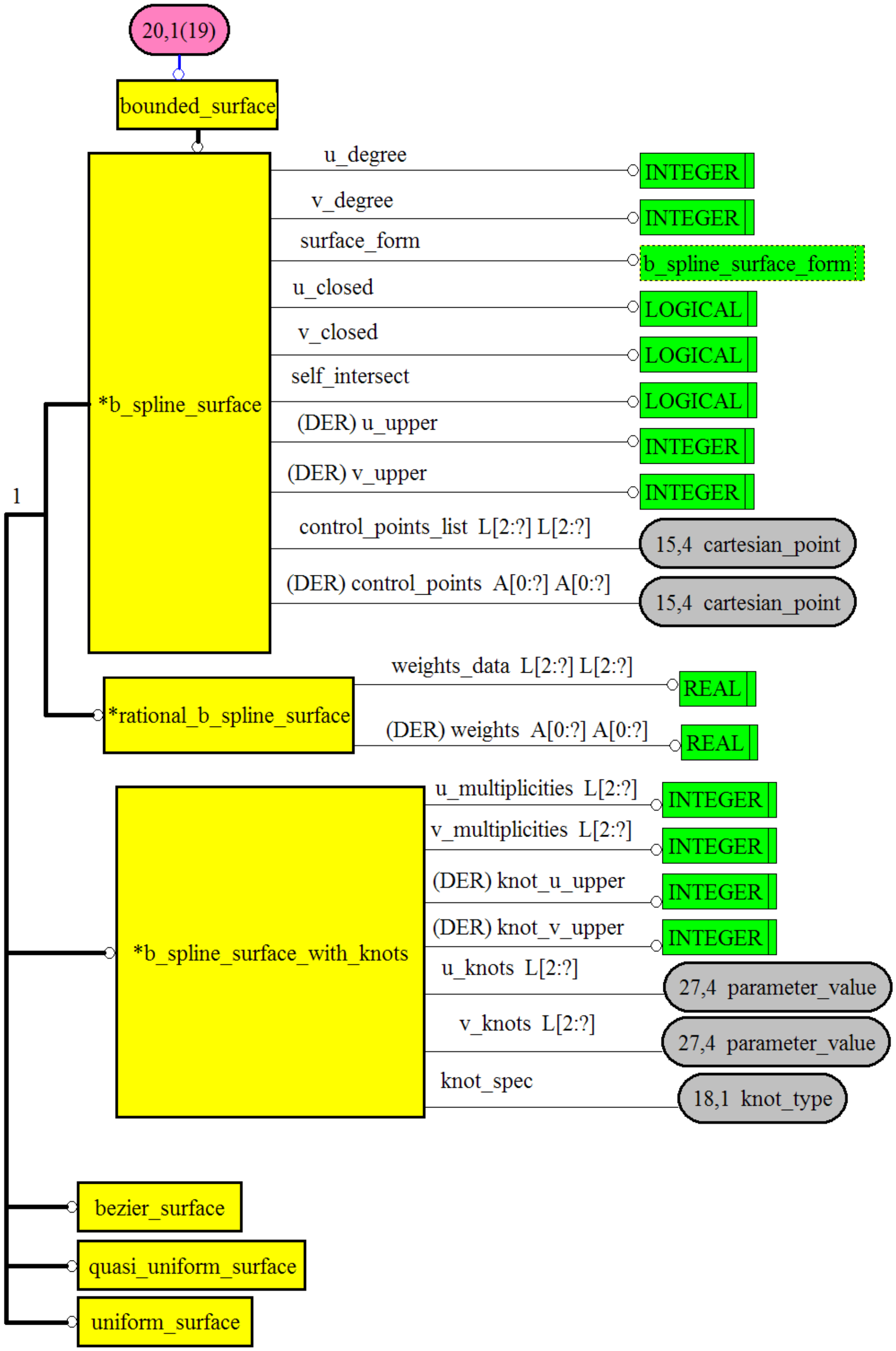


Figure H.20 — AIM EXPRESS-G diagram 20 of 29

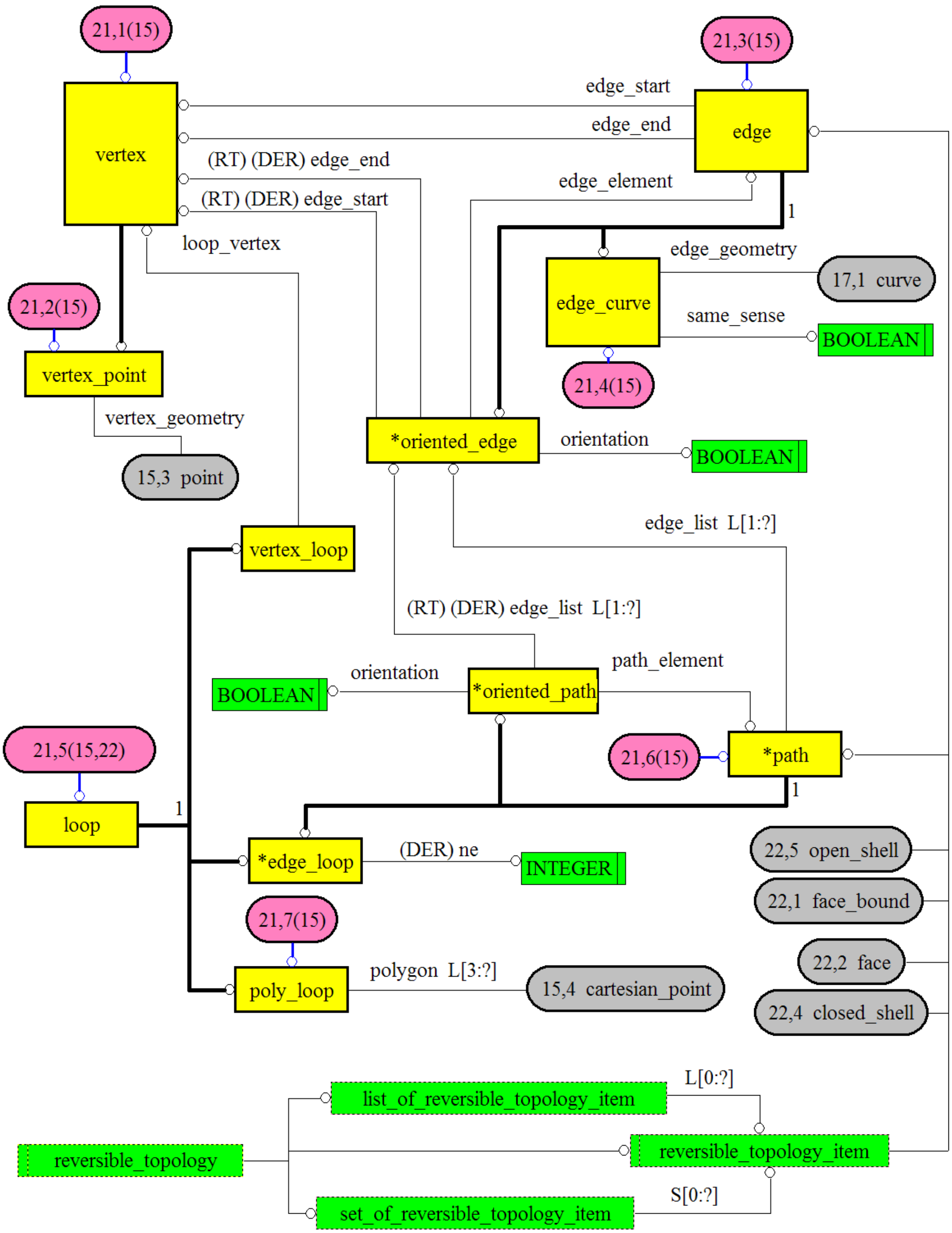


Figure H.21 — AIM EXPRESS-G diagram 21 of 29

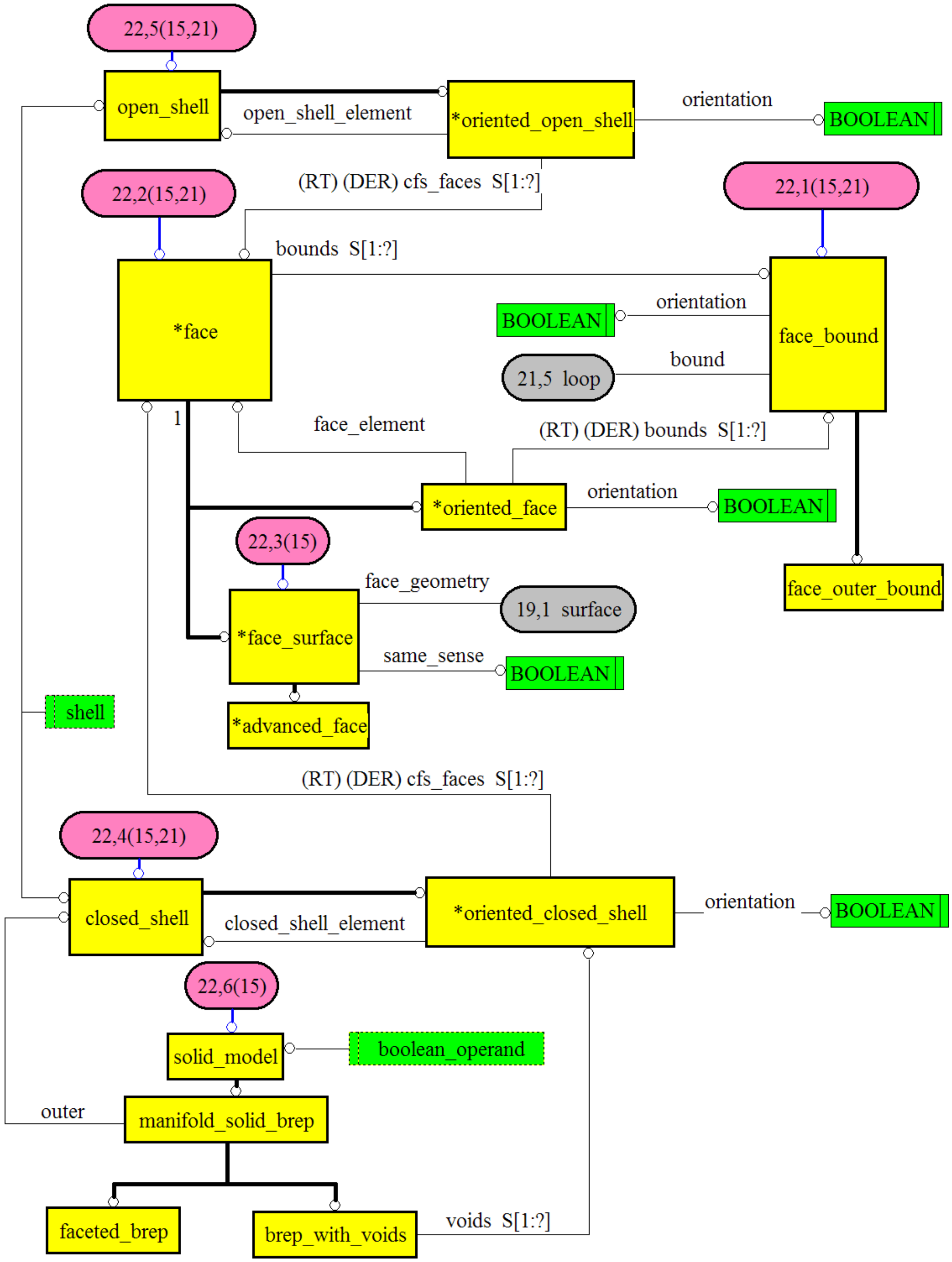


Figure H.22 — AIM EXPRESS-G diagram 22 of 29

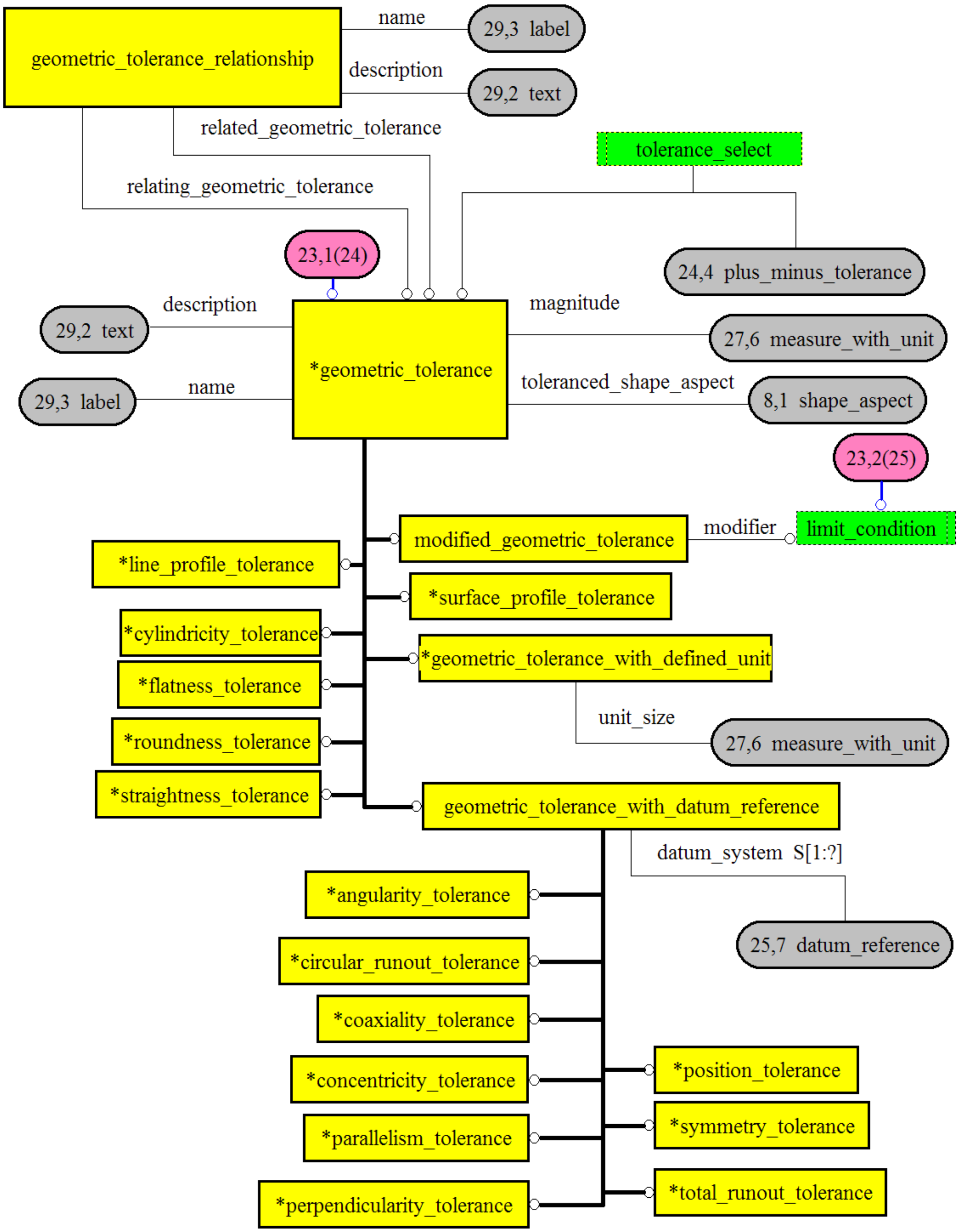


Figure H.23 — AIM EXPRESS-G diagram 23 of 29

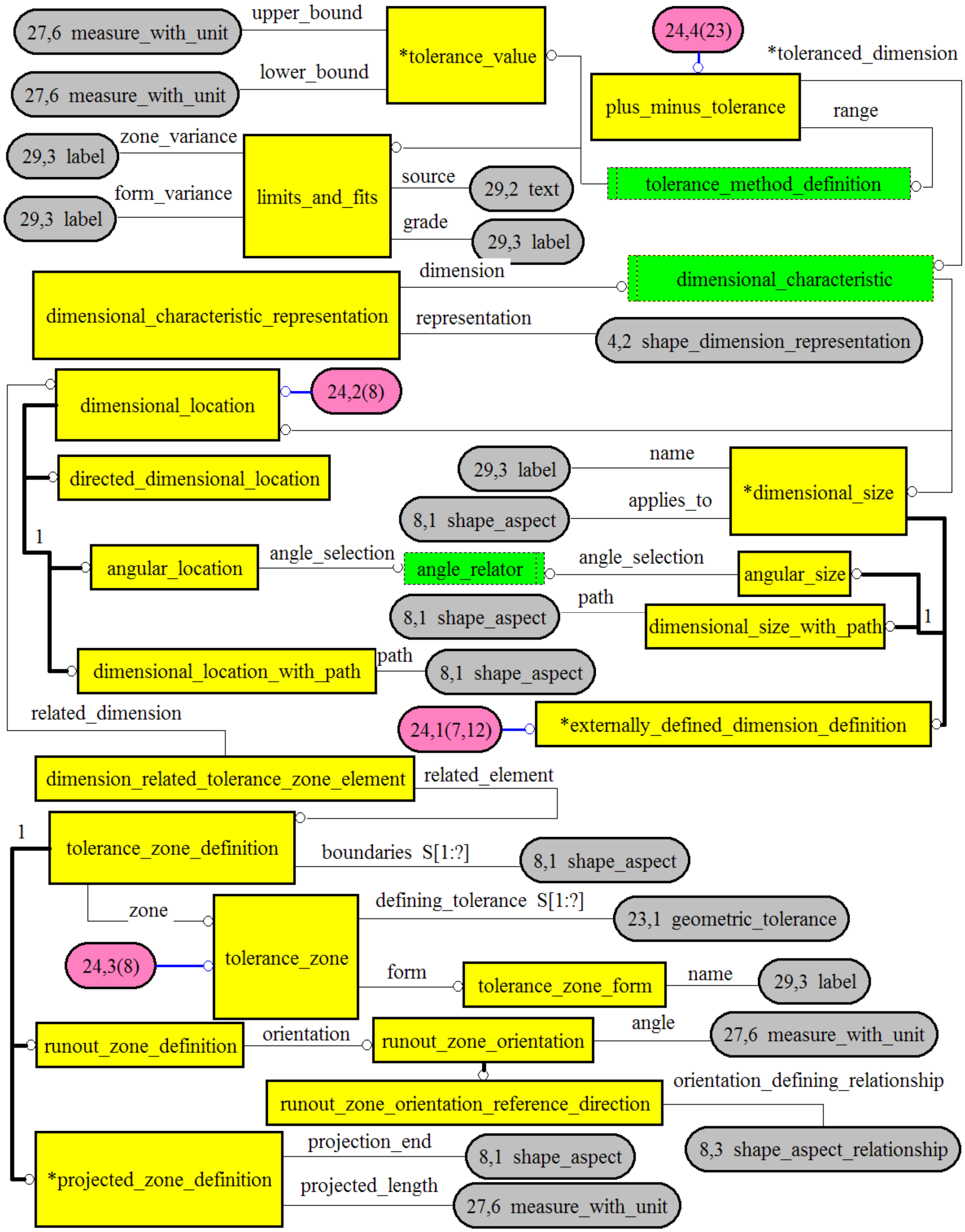


Figure H.24 — AIM EXPRESS-G diagram 24 of 29

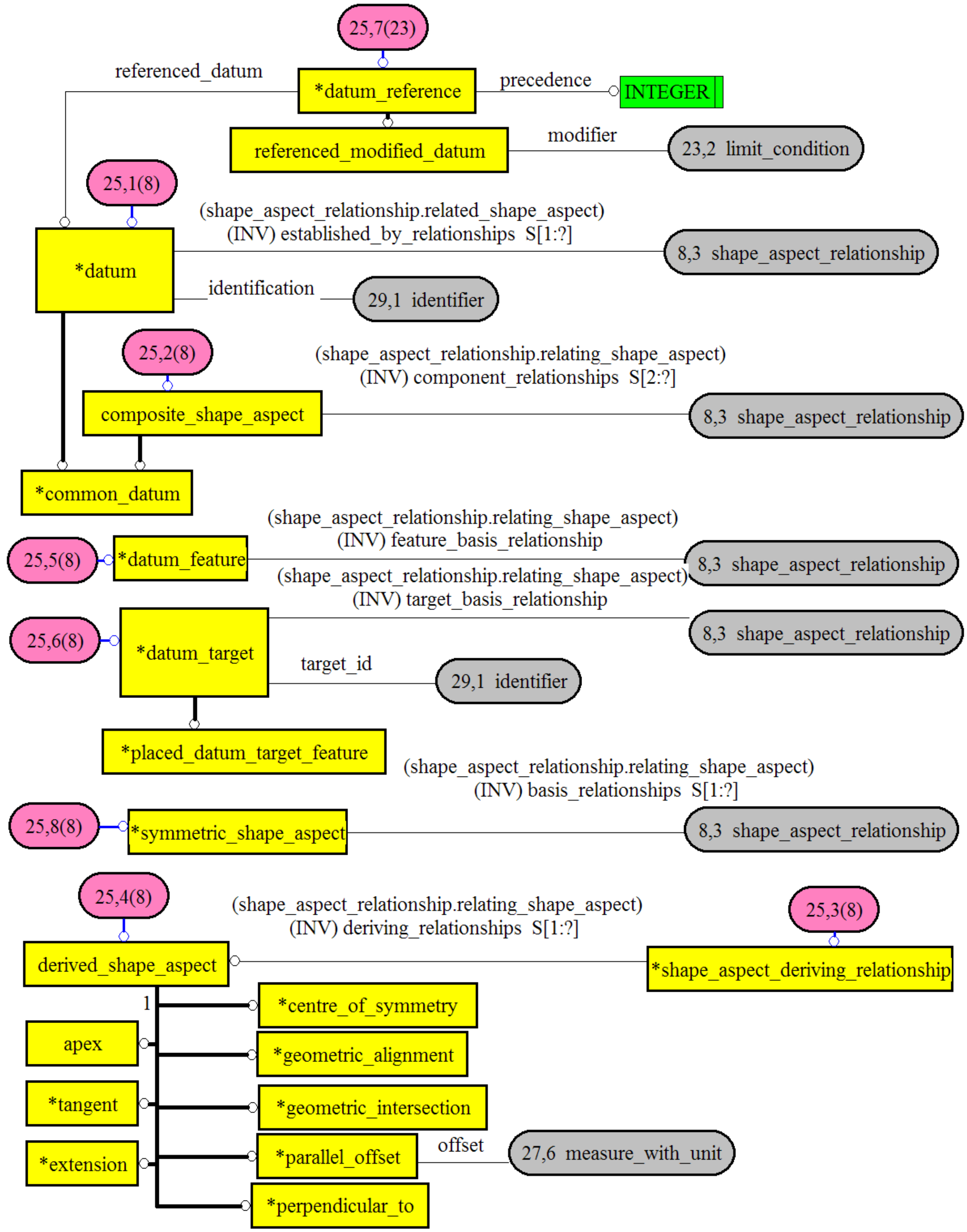


Figure H.25 — AIM EXPRESS-G diagram 25 of 29

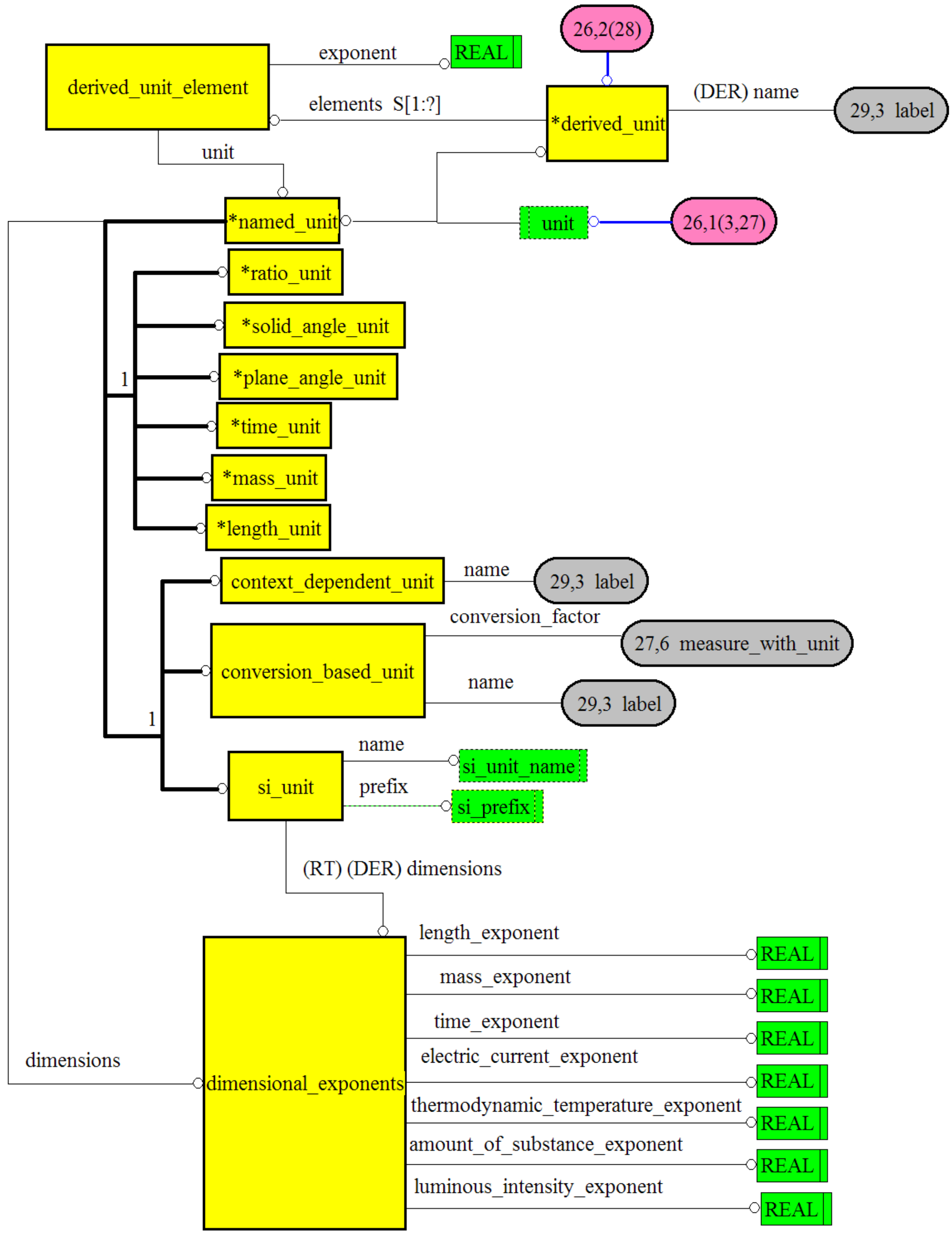


Figure H.26 — AIM EXPRESS-G diagram 26 of 29

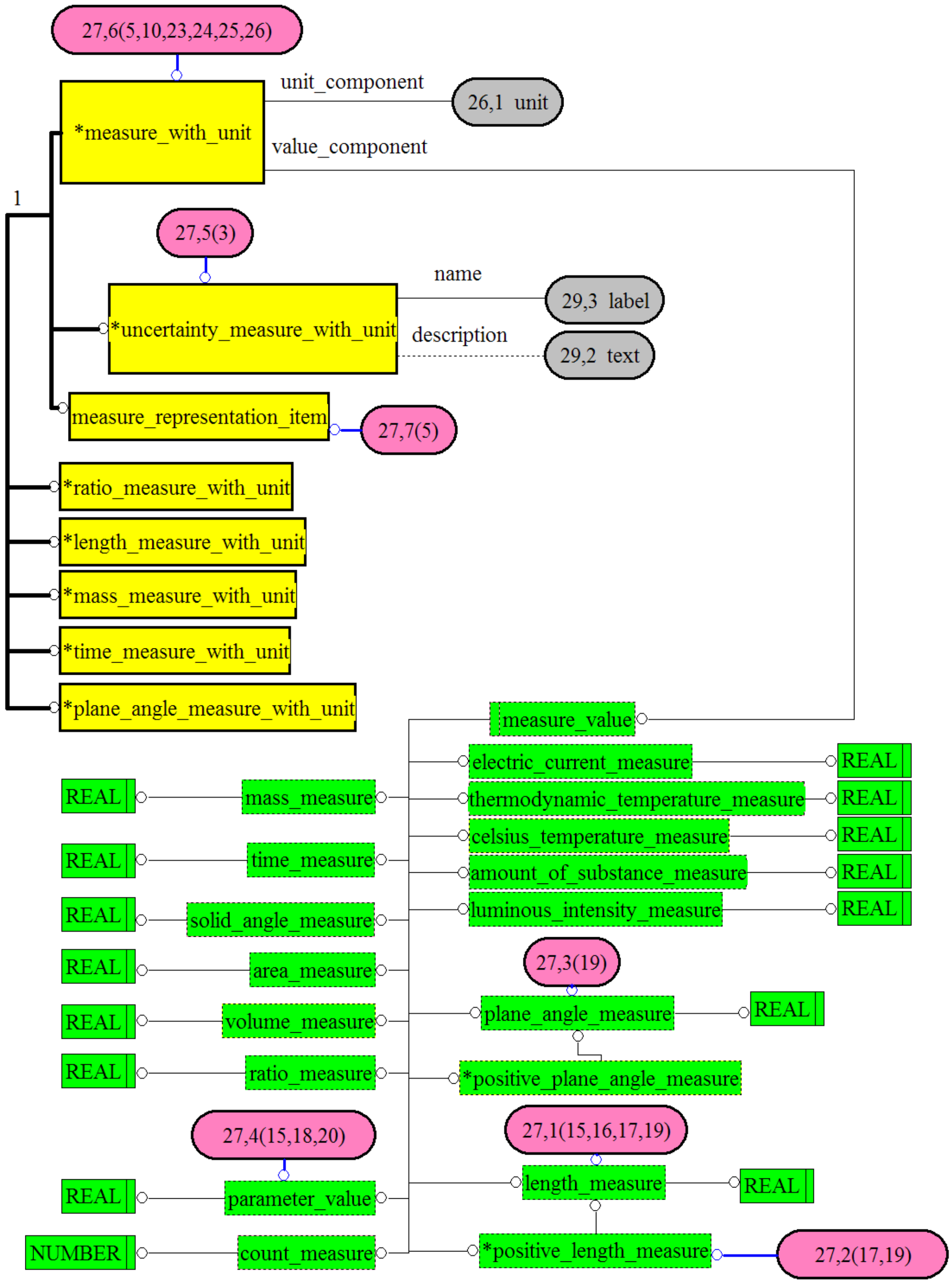


Figure H.27 — AIM EXPRESS-G diagram 27 of 29

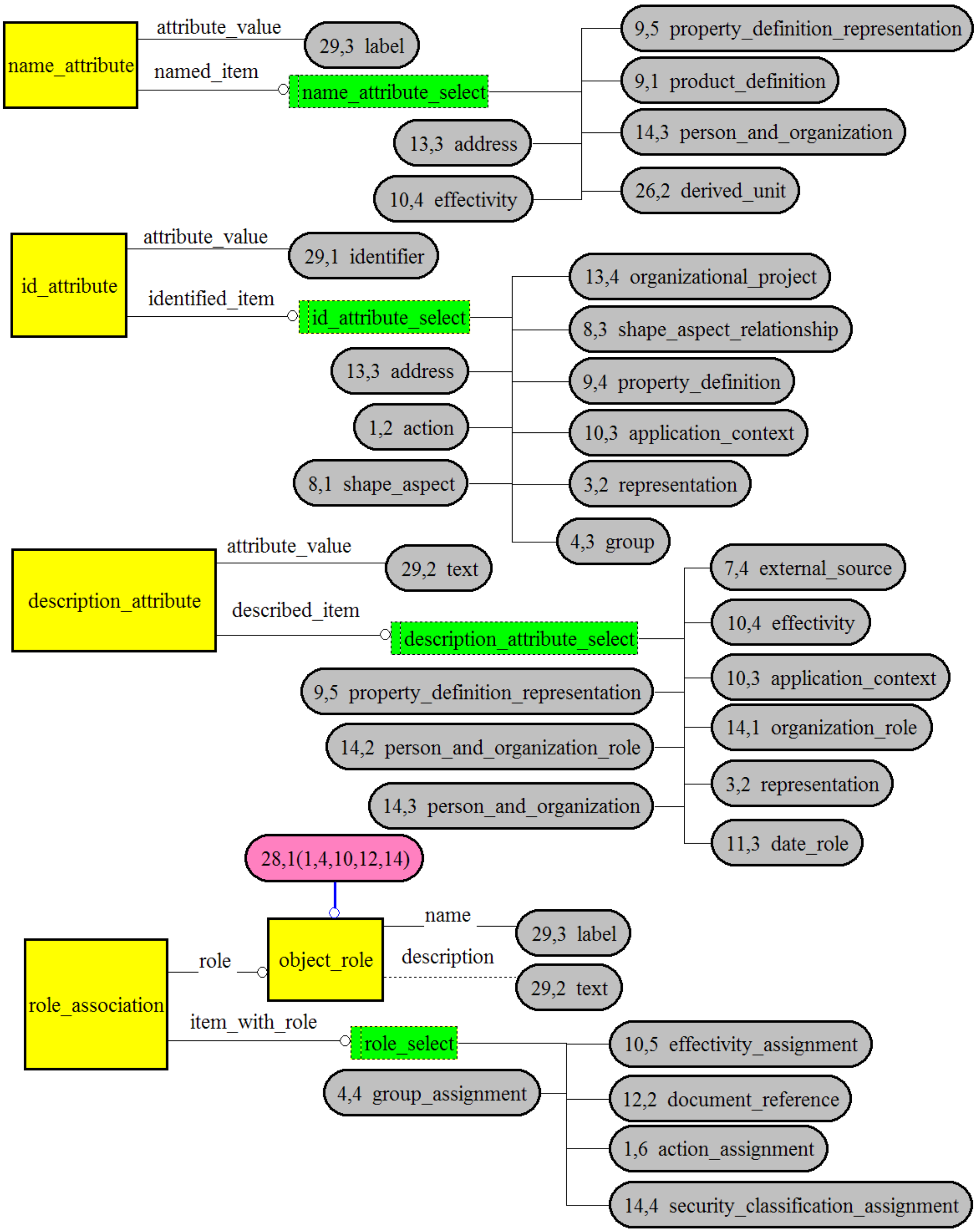


Figure H.28 — AIM EXPRESS-G diagram 28 of 29

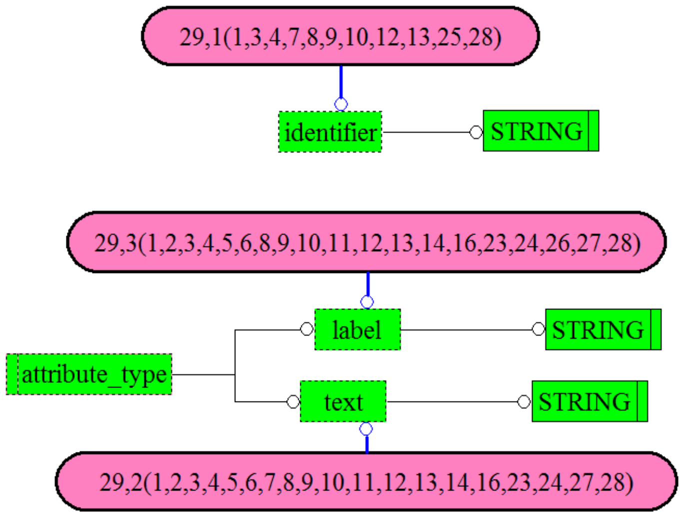


Figure H.29 — AIM EXPRESS-G diagram 29 of 29

Annex I

(informative)

Computer interpretable listings

This annex provides a listing of each EXPRESS schema specified in this part of ISO 10303 without comments or other explanatory text. These listings are available in computer-interpretable form and can be found at the following URLs:

Short names: http://www.tc184-sc4.org/Short_Names

EXPRESS: http://www.tc184-sc4.org/EXPRESS

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@tc184-sc4.org.

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex J

(informative)

Application protocol usage guide

J.1 Usage guidlines and test case definitions

The usage guide is documentation that describes the guidelines describing the usage scenarios, and/or example test cases. This documentation will be develop for the draft international standard of this part of ISO 10303.

J.2 Purpose

This annex contains ISO 10303–21 listings showing how to create data instances that meet the requirements documented in the Mapping Tables for each of the objects in the ARM. Implementors can use these listings as an aid to understanding the representation of each ARM object, and as a set of example instances that can be copied by applications that want to create data sets for this part of ISO 10303.

J.3 Syntax conventions

Each data set is described as a ISO 10303–21 exchange file containing a sequence of AIM instances with comments. The comments shows the implementor what AIM instances must be created to represent an ARM concept. The comments give lists of instance numbers used for a particular path. The instances and arrangement is formally defined by the mapping tables in clause 5.1.

The AIM data part contains instances that meet the requirements described by the mapping table. This listing shows the implementor how the AIM instances for each ARM concept must be linked and what constant values must be set for those instances. When a particular text value is required by the mapping table, it is underlined.

J.4 Rationale for the selection of example parts

Geometry – This part of ISO 10303 uses the same geometry as ISO 10303-240, ISO 10303-238 and ISO 10303-224. Geometry has been thoroughly validated both from an ARM and AIM perspective. Therefore The a test part will be used as an example that AIM geometry is validated.

Dimensional and Geometric tolerances - This part of ISO 10303 uses the same tolerances as ISO 10303-238, ISO 10303-240 and ISO 10303-224. Tolerances have been thoroughly validated both from an ARM and AIM perspective. Therefore The a test part will be used as an example that AIM tolerance is validated.

Machining Features - This part of ISO 10303 uses the same features as ISO 10303-240, 10303-238 and ISO 10303-224. Features have been thoroughly validated both from an ARM and AIM perspective. Therefore The a test part will be used as an example that AIM features is validated.

ISO 10303-224 overlap - There are several UoFs that are in this part of ISO 10303 that are common with ISO 10303-224:2006 design_exception, feature_definition_item, feature_profile, library_reference, manufacturing_feature, manufacturing_part_properties, manufacturing_process_control_documentation, measurement_limitations, part_model, requisitions, shape_representation_for_casting_and_machining. Since these concepts have be thoroughly tested in ISO 10303-224 , the part will serve as an example to valide these concepts on the AIM schema.

Dimensional inspection specific data. A data exchange of this part of ISO 10303 will require a dm_program_run and supporting inspection result data. This test case tests the dm_program_run and dm_executed_result and associated data.

This is a hand populated file with made up values. This test does not validate the accuracy of the value, but the structure of the data and validation of entity rules and schema global rules. The intent is to validate that the schema of this part of ISO 10303 schema supports the data of this part of ISO 10303.

Conclusion: The schema of this part of ISO 10303 has been validated; the entities and rules successfully passed a validation test. The schema of this part of ISO 10303 is accurate and ready for prototype implementations.

J.5 Part example

This test case was generated by hand population. This test does not validate the accuracy of the value, but the structure of the data and validation of entity rules and schema global rules. The intent is to validate that the schema of this part of ISO 10303 supports the data of this part of ISO 10303. This part was successfully tested against the schema in this document.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('AP224 File'),'2;1');
FILE_NAME('92824_800172_g.224','1997-08-01T11:07:18-05:00',('RPTS
Operator'),('RPTS'),'RPTS  MP  6.0','PTC  Pro/ENGINEER  Version  18.0','RPTS
Operator');
FILE_SCHEMA(('DIMENSIONAL_INSPECTION_SCHEMA'));
ENDSEC;
DATA;
#5=DIMENSIONAL_EXPONENTS(1.0,0.0,0.0,0.0,0.0,0.0,0.0);
#6=DIMENSIONAL_EXPONENTS(0.0,0.0,0.0,0.0,0.0,0.0,0.0);
#8=DIMENSIONAL_EXPONENTS(0.0,0.0,1.0,0.0,0.0,0.0,0.0);
#10=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));

#11=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(25.399999999999999),#10);
#15=(CONVERSION_BASED_UNIT('INCH',#11)LENGTH_UNIT()NAMED_UNIT(#5));
#19=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#20=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.017453292519943),#19)
;
#24=(CONVERSION_BASED_UNIT('DEGREE',#20)NAMED_UNIT(#6)PLANE_ANGLE_UNIT());
#28=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
#29=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.000001000000000),#15,'clos
ure','maximum  model  space  distance  between  geometric  entities  at  asserted
connectivities');
```

```
#34=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((
#29))GLOBAL_UNIT_ASSIGNED_CONTEXT((#15,#24,#28))REPRESENTATION_CONTEXT('ID1',
'3D'));

#36=APPLICATION_CONTEXT('dimensional inspection schema');
#37=APPLICATION_PROTOCOL_DEFINITION('Draft                         International
Standard','dimensional inspection schema',1996,#36);
#38=PRODUCT_CONTEXT('dimensional inspection schema',#36,'Mechanical');
#39=PRODUCT('','','',(#38));
#40=PRODUCT_DEFINITION_FORMATION('','',#39);
#41=PRODUCT_DEFINITION_CONTEXT('part           definition',#36,'manufacturing
planning');
#42=PRODUCT_DEFINITION('','product design',#40,#41);
#43=PRODUCT_DEFINITION_SHAPE('product shape','shape for product',#42);
#44=CARTESIAN_POINT('',(-0.031000000000000,-0.183398473276088,0.0));
#45=CARTESIAN_POINT('',(-0.031000000000000,0.183398473276088,5.847965E-34));
#46=VERTEX_POINT('',#44);
#47=VERTEX_POINT('',#45);
#48=DIRECTION('',(0.0,1.0,0.0));
#49=VECTOR('',#48,0.366796946552176);
#50=LINE('',#44,#49);
#51=EDGE_CURVE('',#46,#47,#50,.T.);
#52=ORIENTED_EDGE('',*,*,#51,.T.);
#53=CARTESIAN_POINT('',(-0.186000000000000,0.0,0.0));
#54=VERTEX_POINT('',#53);
#55=CARTESIAN_POINT('',(0.0,0.0,0.0));
#56=DIRECTION('',(0.0,0.0,1.000000000000000));
#57=DIRECTION('',(-0.166666666666667,0.986013297183269,0.0));
#58=AXIS2_PLACEMENT_3D('',#55,#56,#57);
#59=CIRCLE('',#58,0.186000000000000);
#60=EDGE_CURVE('',#47,#54,#59,.T.);
#61=ORIENTED_EDGE('',*,*,#60,.T.);
#62=CARTESIAN_POINT('',(0.0,0.0,0.0));
#63=DIRECTION('',(0.0,0.0,1.0));
#64=DIRECTION('',(-1.0,0.0,0.0));
#65=AXIS2_PLACEMENT_3D('',#62,#63,#64);
#66=CIRCLE('',#65,0.186000000000000);
#67=EDGE_CURVE('',#54,#46,#66,.T.);
#68=ORIENTED_EDGE('',*,*,#67,.T.);
#69=EDGE_LOOP('',(#52,#61,#68));
#70=FACE_BOUND('',#69,.F.);
#71=CARTESIAN_POINT('',(0.0,0.0,0.0));
#72=DIRECTION('',(0.0,0.0,1.0));
#73=DIRECTION('',(1.0,0.0,0.0));
#74=AXIS2_PLACEMENT_3D('',#71,#72,#73);
#75=PLANE('',#74);
#76=ADVANCED_FACE('',(#70),#75,.F.);
#77=SHAPE_ASPECT('featured shape: face 1','',#43,.T.);
#78=PROPERTY_DEFINITION('','',#77);
#79=FACE_SHAPE_REPRESENTATION('',(#76),#34);
#80=SHAPE_DEFINITION_REPRESENTATION(#78,#79);
#81=CARTESIAN_POINT('',(0.031000000000000,0.183398473276088,0.0));
#82=CARTESIAN_POINT('',(0.031000000000000,-0.183398473276088,0.0));
#83=VERTEX_POINT('',#81);
#84=VERTEX_POINT('',#82);
#85=DIRECTION('',(0.0,1.0,0.0));
```



```
#86=VECTOR('',#85,0.366796946552176);
#87=LINE('',#82,#86);
#88=EDGE_CURVE('',#83,#84,#87,.F.);
#89=ORIENTED_EDGE('',*,*,#88,.T.);
#90=CARTESIAN_POINT('',(0.186000000000000,0.0,0.0));
#91=VERTEX_POINT('',#90);
#92=CARTESIAN_POINT('',(0.0,0.0,0.0));
#93=DIRECTION('',(0.0,0.0,1.000000000000000));
#94=DIRECTION('',(0.166666666666667,-0.986013297183269,0.0));
#95=AXIS2_PLACEMENT_3D('',#92,#93,#94);
#96=CIRCLE('',#95,0.186000000000000);
#97=EDGE_CURVE('',#84,#91,#96,.T.);
#98=ORIENTED_EDGE('',*,*,#97,.T.);
#99=CARTESIAN_POINT('',(0.0,0.0,0.0));
#100=DIRECTION('',(0.0,0.0,1.0));
#101=DIRECTION('',(1.0,0.0,0.0));
#102=AXIS2_PLACEMENT_3D('',#99,#100,#101);
#103=CIRCLE('',#102,0.186000000000000);
#104=EDGE_CURVE('',#91,#83,#103,.T.);
#105=ORIENTED_EDGE('',*,*,#104,.T.);
#106=EDGE_LOOP('',(#89,#98,#105));
#107=FACE_BOUND('',#106,.F.);
#108=ADVANCED_FACE('',(#107),#75,.F.);
#109=SHAPE_ASPECT('featured shape: face 2','',#43,.T.);
#110=PROPERTY_DEFINITION('','',#109);
#111=FACE_SHAPE_REPRESENTATION('',(#108),#34);
#112=SHAPE_DEFINITION_REPRESENTATION(#110,#111);
#113=CARTESIAN_POINT('',(0.0,0.090166604983954,4.310000000000000));
#114=CARTESIAN_POINT('',(0.0,-0.090166604983954,4.310000000000000));
#115=VERTEX_POINT('',#113);
#116=VERTEX_POINT('',#114);
#117=CARTESIAN_POINT('',(0.0,0.0,4.310000000000000));
#118=DIRECTION('',(0.0,0.0,1.0));
#119=DIRECTION('',(0.0,-1.0,0.0));
#120=AXIS2_PLACEMENT_3D('',#117,#118,#119);
#121=CIRCLE('',#120,0.090166604983954);
#122=EDGE_CURVE('',#115,#116,#121,.F.);
#123=ORIENTED_EDGE('',*,*,#122,.T.);
#124=CARTESIAN_POINT('',(0.0,0.0,4.310000000000000));
#125=DIRECTION('',(0.0,0.0,1.0));
#126=DIRECTION('',(0.0,1.0,0.0));
#127=AXIS2_PLACEMENT_3D('',#124,#125,#126);
#128=CIRCLE('',#127,0.090166604983954);
#129=EDGE_CURVE('',#116,#115,#128,.F.);
#130=ORIENTED_EDGE('',*,*,#129,.T.);
#131=EDGE_LOOP('',(#123,#130));
#132=FACE_BOUND('',#131,.F.);
#133=CARTESIAN_POINT('',(0.0,0.0,4.310000000000000));
#134=DIRECTION('',(0.0,0.0,1.0));
#135=DIRECTION('',(1.0,0.0,0.0));
#136=AXIS2_PLACEMENT_3D('',#133,#134,#135);
#137=PLANE('',#136);
#138=ADVANCED_FACE('',(#132),#137,.T.);
#139=SHAPE_ASPECT('featured shape: face 3','',#43,.T.);
#140=PROPERTY_DEFINITION('','',#139);
#141=FACE_SHAPE_REPRESENTATION('',(#138),#34);
```

```
#142=SHAPE_DEFINITION_REPRESENTATION(#140,#141);
#143=CARTESIAN_POINT('',(0.175184902317523,0.062500000000000,2.58000000000000
0));
#144=CARTESIAN_POINT('',(0.186000000000000,0.0,2.642500000000000));
#145=VERTEX_POINT('',#143);
#146=VERTEX_POINT('',#144);
#147=CARTESIAN_POINT('',(0.186000000000000,0.0,2.642500000000000));
#148=CARTESIAN_POINT('',(0.186000000000000,0.005471621857183,2.64250000000000
0));
#149=CARTESIAN_POINT('',(0.185536257371889,0.016107401839008,2.64114590848819
1));
#150=CARTESIAN_POINT('',(0.183444961303938,0.031895399078690,2.63461599318274
4));
#151=CARTESIAN_POINT('',(0.180647351963460,0.044822005907028,2.62456383972068
9));
#152=CARTESIAN_POINT('',(0.177772939484603,0.054932562783282,2.61133932574037
7));
#153=CARTESIAN_POINT('',(0.175676556620784,0.061146675813468,2.59594226183308
3));
#154=CARTESIAN_POINT('',(0.175184902317523,0.062500000000000,2.58542285607518
0));
#155=CARTESIAN_POINT('',(0.175184902317523,0.062500000000000,2.58000000000000
0));
#156=B_SPLINE_CURVE_WITH_KNOTS('',3,(#147,#148,#149,#150,#151,#152,#153,#154,
#155),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#157=EDGE_CURVE('',#145,#146,#156,.F.);
#158=ORIENTED_EDGE('',*,*,#157,.T.);
#159=CARTESIAN_POINT('',(0.186000000000000,0.0,3.800000000000000));
#160=VERTEX_POINT('',#159);
#161=DIRECTION('',(0.0,0.0,1.0));
#162=VECTOR('',#161,1.157500000000000);
#163=LINE('',#144,#162);
#164=EDGE_CURVE('',#146,#160,#163,.T.);
#165=ORIENTED_EDGE('',*,*,#164,.T.);
#166=CARTESIAN_POINT('',(1.138922E-17,0.186000000000000,3.800000000000000));
#167=VERTEX_POINT('',#166);
#168=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#169=DIRECTION('',(0.0,0.0,1.0));
#170=DIRECTION('',(1.0,0.0,0.0));
#171=AXIS2_PLACEMENT_3D('',#168,#169,#170);
#172=CIRCLE('',#171,0.186000000000000);
#173=EDGE_CURVE('',#160,#167,#172,.T.);
#174=ORIENTED_EDGE('',*,*,#173,.T.);
#175=CARTESIAN_POINT('',(-0.186000000000000,-5.982216E-
17,3.800000000000000));
#176=VERTEX_POINT('',#175);
#177=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#178=DIRECTION('',(0.0,0.0,1.0));
#179=DIRECTION('',(6.123234E-17,1.0,0.0));
#180=AXIS2_PLACEMENT_3D('',#177,#178,#179);
#181=CIRCLE('',#180,0.186000000000000);
#182=EDGE_CURVE('',#167,#176,#181,.T.);
#183=ORIENTED_EDGE('',*,*,#182,.T.);
#184=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,2.642500000000000));
```



```
#185=VERTEX_POINT('',#184);
#186=DIRECTION('',(0.0,0.0,1.0));
#187=VECTOR('',#186,1.157500000000000);
#188=LINE('',#184,#187);
#189=EDGE_CURVE('',#176,#185,#188,.F.);
#190=ORIENTED_EDGE('',*,*,#189,.T.);
#191=CARTESIAN_POINT('',(-
0.175184902317523,0.062500000000000,2.580000000000001));
#192=VERTEX_POINT('',#191);
#193=CARTESIAN_POINT('',(-
0.175184902317523,0.062500000000000,2.580000000000001));
#194=CARTESIAN_POINT('',(-
0.175184902317523,0.062500000000000,2.585426709005403));
#195=CARTESIAN_POINT('',(-
0.175677115146707,0.061145342990551,2.595951978028653));
#196=CARTESIAN_POINT('',(-
0.177776266176458,0.054921858431442,2.611358030685044));
#197=CARTESIAN_POINT('',(-
0.180650079241388,0.044811772210325,2.624574571398843));
#198=CARTESIAN_POINT('',(-
0.183449265169815,0.031871520572190,2.634630402017820));
#199=CARTESIAN_POINT('',(-
0.185537370272068,0.016090248458853,2.641148978803535));
#200=CARTESIAN_POINT('',(-
0.186000000000000,0.005464543947207,2.642500000000000));
#201=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,2.642500000000000));
#202=B_SPLINE_CURVE_WITH_KNOTS('',3,(#193,#194,#195,#196,#197,#198,#199,#200,
#201),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#203=EDGE_CURVE('',#185,#192,#202,.F.);
#204=ORIENTED_EDGE('',*,*,#203,.T.);
#205=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,2.517500000000000));
#206=VERTEX_POINT('',#205);
#207=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,2.517500000000000));
#208=CARTESIAN_POINT('',(-
0.186000000000000,0.005458851337490,2.517500000000000));
#209=CARTESIAN_POINT('',(-
0.185536093618712,0.016073651023862,2.518856289865521));
#210=CARTESIAN_POINT('',(-
0.183470250159010,0.031747433690688,2.525300488417541));
#211=CARTESIAN_POINT('',(-
0.180661972444195,0.044771773357708,2.535375964999431));
#212=CARTESIAN_POINT('',(-
0.177781605611539,0.054904708977402,2.548606047863121));
#213=CARTESIAN_POINT('',(-
0.175674738176842,0.061152720479089,2.564064031685822));
#214=CARTESIAN_POINT('',(-
0.175184902317523,0.062500000000000,2.574581881011206));
#215=CARTESIAN_POINT('',(-
0.175184902317523,0.062500000000000,2.580000000000001));
#216=B_SPLINE_CURVE_WITH_KNOTS('',3,(#207,#208,#209,#210,#211,#212,#213,#214,
#215),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#217=EDGE_CURVE('',#192,#206,#216,.F.);
```

```
#218=ORIENTED_EDGE('',*,*,#217,.T.);
#219=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,0.344000000000001));
#220=VERTEX_POINT('',#219);
#221=DIRECTION('',(0.0,0.0,1.0));
#222=VECTOR('',#221,2.173499999999999);
#223=LINE('',#219,#222);
#224=EDGE_CURVE('',#206,#220,#223,.F.);
#225=ORIENTED_EDGE('',*,*,#224,.T.);
#226=CARTESIAN_POINT('',(-
0.160499221181911,0.094000000000000,0.250000000000000));
#227=VERTEX_POINT('',#226);
#228=CARTESIAN_POINT('',(-
0.160499221181911,0.094000000000000,0.250000000000000));
#229=CARTESIAN_POINT('',(-
0.160499221181911,0.094000000000000,0.256766446956972));
#230=CARTESIAN_POINT('',(-
0.161327432070906,0.092615564162586,0.270119577772784));
#231=CARTESIAN_POINT('',(-
0.165192932924788,0.085677048896482,0.290595468399896));
#232=CARTESIAN_POINT('',(-
0.170610275358273,0.074562472136122,0.308562918733388));
#233=CARTESIAN_POINT('',(-
0.176618325764372,0.059201828025679,0.324055274565888));
#234=CARTESIAN_POINT('',(-
0.181743103977182,0.041251906320261,0.335390883863212));
#235=CARTESIAN_POINT('',(-
0.185283487333834,0.020349156037856,0.342600715969244));
#236=CARTESIAN_POINT('',(-
0.186000000000000,0.006811219284266,0.344000000000000));
#237=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,0.344000000000000));
#238=B_SPLINE_CURVE_WITH_KNOTS('',3,(#228,#229,#230,#231,#232,#233,#234,#235,
#236,#237),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#239=EDGE_CURVE('',#220,#227,#238,.F.);
#240=ORIENTED_EDGE('',*,*,#239,.T.);
#241=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,0.156000000000001));
#242=VERTEX_POINT('',#241);
#243=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,0.156000000000001));
#244=CARTESIAN_POINT('',(-
0.186000000000000,0.006800366988429,0.156000000000001));
#245=CARTESIAN_POINT('',(-
0.185284237508996,0.020320470184615,0.157398064439927));
#246=CARTESIAN_POINT('',(-
0.181763787252156,0.041157668374866,0.164566419180857));
#247=CARTESIAN_POINT('',(-
0.176635182649387,0.059160281094706,0.175901786861670));
#248=CARTESIAN_POINT('',(-
0.170601634807429,0.074584180536545,0.191460403648173));
#249=CARTESIAN_POINT('',(-
0.165181827617091,0.085698349461980,0.209438767375007));
#250=CARTESIAN_POINT('',(-
0.161310523171716,0.092644858063427,0.230008460101478));
#251=CARTESIAN_POINT('',(-
0.160499221181911,0.094000000000000,0.243288192214063));
```



```
#252=CARTESIAN_POINT('',(-
0.160499221181911,0.094000000000000,0.250000000000000));
#253=B_SPLINE_CURVE_WITH_KNOTS('',3,(#243,#244,#245,#246,#247,#248,#249,#250,
#251,#252),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#254=EDGE_CURVE('',#227,#242,#253,.F.);
#255=ORIENTED_EDGE('',*,*,#254,.T.);
#256=DIRECTION('',(0.0,0.0,1.0));
#257=VECTOR('',#256,0.156000000000000);
#258=LINE('',#53,#257);
#259=EDGE_CURVE('',#242,#54,#258,.F.);
#260=ORIENTED_EDGE('',*,*,#259,.T.);
#261=ORIENTED_EDGE('',*,*,#60,.F.);
#262=CARTESIAN_POINT('',(-
0.031000000000000,0.183398473276088,0.617500000000000));
#263=VERTEX_POINT('',#262);
#264=DIRECTION('',(-6.742245E-17,0.0,1.0));
#265=VECTOR('',#264,0.617500000000000);
#266=LINE('',#45,#265);
#267=EDGE_CURVE('',#47,#263,#266,.T.);
#268=ORIENTED_EDGE('',*,*,#267,.T.);
#269=CARTESIAN_POINT('',(-
0.018500000000000,0.185077686391418,0.630000000000000));
#270=VERTEX_POINT('',#269);
#271=CARTESIAN_POINT('',(-
0.018500000000000,0.185077686391418,0.630000000000000));
#272=CARTESIAN_POINT('',(-
0.020222806907318,0.184905478015536,0.630000000000000));
#273=CARTESIAN_POINT('',(-
0.023380441464034,0.184543239000241,0.629340957979689));
#274=CARTESIAN_POINT('',(-
0.027542467419886,0.183961116646317,0.626594726403956));
#275=CARTESIAN_POINT('',(-
0.030285994332659,0.183520653386331,0.622509121211844));
#276=CARTESIAN_POINT('',(-
0.031000000000000,0.183398473276088,0.619276684284376));
#277=CARTESIAN_POINT('',(-
0.031000000000000,0.183398473276088,0.617500000000000));
#278=B_SPLINE_CURVE_WITH_KNOTS('',3,(#271,#272,#273,#274,#275,#276,#277),.HYP
ERBOLIC_ARC.,.F.,.F.,(4,1,1,1,4),(0.0,0.250000000000000,0.500000000000000,0.7
50000000000000,1.0),.PIECEWISE_BEZIER_KNOTS.);
#279=EDGE_CURVE('',#263,#270,#278,.F.);
#280=ORIENTED_EDGE('',*,*,#279,.T.);
#281=CARTESIAN_POINT('',(0.018500000000000,0.185077686391418,0.63000000000000
0));
#282=VERTEX_POINT('',#281);
#283=CARTESIAN_POINT('',(0.0,0.0,0.630000000000000));
#284=DIRECTION('',(0.0,0.0,-1.0));
#285=DIRECTION('',(-0.099462365591398,0.995041324685042,0.0));
#286=AXIS2_PLACEMENT_3D('',#283,#284,#285);
#287=CIRCLE('',#286,0.186000000000000);
#288=EDGE_CURVE('',#270,#282,#287,.T.);
#289=ORIENTED_EDGE('',*,*,#288,.T.);
#290=CARTESIAN_POINT('',(0.031000000000000,0.183398473276088,0.61750000000000
0));
```

```
#291=VERTEX_POINT('',#290);
#292=CARTESIAN_POINT('',(0.031000000000000,0.183398473276088,0.61750000000000
0));
#293=CARTESIAN_POINT('',(0.031000000000000,0.183398473276088,0.61920375881085
9));
#294=CARTESIAN_POINT('',(0.030354567108582,0.183509179238234,0.62233725817082
5));
#295=CARTESIAN_POINT('',(0.027638547132190,0.183946283585950,0.62650045564978
7));
#296=CARTESIAN_POINT('',(0.023535786372608,0.184523602834953,0.62927917584640
3));
#297=CARTESIAN_POINT('',(0.020288448038623,0.184898916657693,0.63000000000000
0));
#298=CARTESIAN_POINT('',(0.018500000000000,0.185077686391418,0.63000000000000
0));
#299=B_SPLINE_CURVE_WITH_KNOTS('',3,(#292,#293,#294,#295,#296,#297,#298),.HYP
ERBOLIC_ARC.,.F.,.F.,(4,1,1,1,4),(0.0,0.250000000000000,0.500000000000000,0.7
50000000000000,1.0),.PIECEWISE_BEZIER_KNOTS.);
#300=EDGE_CURVE('',#282,#291,#299,.F.);
#301=ORIENTED_EDGE('',*,*,#300,.T.);
#302=DIRECTION('',(0.0,0.0,1.0));
#303=VECTOR('',#302,0.617500000000000);
#304=LINE('',#81,#303);
#305=EDGE_CURVE('',#291,#83,#304,.F.);
#306=ORIENTED_EDGE('',*,*,#305,.T.);
#307=ORIENTED_EDGE('',*,*,#104,.F.);
#308=CARTESIAN_POINT('',(0.186000000000000,0.0,0.156000000000000));
#309=VERTEX_POINT('',#308);
#310=DIRECTION('',(0.0,0.0,1.0));
#311=VECTOR('',#310,0.156000000000000);
#312=LINE('',#90,#311);
#313=EDGE_CURVE('',#91,#309,#312,.T.);
#314=ORIENTED_EDGE('',*,*,#313,.T.);
#315=CARTESIAN_POINT('',(0.160499221181911,0.094000000000000,0.25000000000000
0));
#316=VERTEX_POINT('',#315);
#317=CARTESIAN_POINT('',(0.160499221181911,0.094000000000000,0.25000000000000
0));
#318=CARTESIAN_POINT('',(0.160499221181911,0.094000000000000,0.24329136870794
2));
#319=CARTESIAN_POINT('',(0.161309755247970,0.092646107527992,0.23001745040141
5));
#320=CARTESIAN_POINT('',(0.165177934867762,0.085705567809687,0.20945550416081
0));
#321=CARTESIAN_POINT('',(0.170594327073985,0.074600077742975,0.19148197206178
3));
#322=CARTESIAN_POINT('',(0.176625064117123,0.059189393219094,0.17592576770399
5));
#323=CARTESIAN_POINT('',(0.181755664328797,0.041193271352137,0.16458337327453
6));
#324=CARTESIAN_POINT('',(0.185281410572974,0.020351551813128,0.15740359268013
0));
#325=CARTESIAN_POINT('',(0.186000000000000,0.006812818398979,0.15600000000000
0));
#326=CARTESIAN_POINT('',(0.186000000000000,0.0,0.156000000000000));
```



```
#327=B_SPLINE_CURVE_WITH_KNOTS('',3,(#317,#318,#319,#320,#321,#322,#323,#324,
#325,#326),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#328=EDGE_CURVE('',#309,#316,#327,.F.);
#329=ORIENTED_EDGE('',*,*,#328,.T.);
#330=CARTESIAN_POINT('',(0.186000000000000,0.0,0.344000000000000));
#331=VERTEX_POINT('',#330);
#332=CARTESIAN_POINT('',(0.186000000000000,0.0,0.344000000000000));
#333=CARTESIAN_POINT('',(0.186000000000000,0.006823032072908,0.34400000000000
0));
#334=CARTESIAN_POINT('',(0.185280833739913,0.020378752353648,0.34259554757765
5));
#335=CARTESIAN_POINT('',(0.181735072423795,0.041287079595952,0.33537407462788
6));
#336=CARTESIAN_POINT('',(0.176608347816713,0.059230733088108,0.32403177810720
3));
#337=CARTESIAN_POINT('',(0.170601713482259,0.074581303037190,0.30853777493501
9));
#338=CARTESIAN_POINT('',(0.165188657251781,0.085684909394755,0.29057713245411
6));
#339=CARTESIAN_POINT('',(0.161326264069271,0.092617528844491,0.27010754428205
9));
#340=CARTESIAN_POINT('',(0.160499221181911,0.094000000000000,0.25676207178187
0));
#341=CARTESIAN_POINT('',(0.160499221181911,0.094000000000000,0.25000000000000
0));
#342=B_SPLINE_CURVE_WITH_KNOTS('',3,(#332,#333,#334,#335,#336,#337,#338,#339,
#340,#341),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#343=EDGE_CURVE('',#316,#331,#342,.F.);
#344=ORIENTED_EDGE('',*,*,#343,.T.);
#345=CARTESIAN_POINT('',(0.186000000000000,0.0,2.517500000000000));
#346=VERTEX_POINT('',#345);
#347=DIRECTION('',(0.0,0.0,1.0));
#348=VECTOR('',#347,2.173500000000000);
#349=LINE('',#330,#348);
#350=EDGE_CURVE('',#331,#346,#349,.T.);
#351=ORIENTED_EDGE('',*,*,#350,.T.);
#352=CARTESIAN_POINT('',(0.175184902317523,0.062500000000000,2.58000000000000
0));
#353=CARTESIAN_POINT('',(0.175184902317523,0.062500000000000,2.57458465168521
7));
#354=CARTESIAN_POINT('',(0.175674414168389,0.061153355224084,2.56407098903970
9));
#355=CARTESIAN_POINT('',(0.177778707786166,0.054914219798778,2.54862220454415
8));
#356=CARTESIAN_POINT('',(0.180660130799249,0.044778201243767,2.53538329137297
4));
#357=CARTESIAN_POINT('',(0.183465542301875,0.031773272894993,2.52531636727892
4));
#358=CARTESIAN_POINT('',(0.185534991382704,0.016091351499283,2.51885926065962
5));
#359=CARTESIAN_POINT('',(0.186000000000000,0.005466293856892,2.51750000000000
0));
```

```
#360=CARTESIAN_POINT('',(0.186000000000000,0.0,2.517500000000000));
#361=B_SPLINE_CURVE_WITH_KNOTS('',3,(#352,#353,#354,#355,#356,#357,#358,#359,
#360),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#362=EDGE_CURVE('',#346,#145,#361,.F.);
#363=ORIENTED_EDGE('',*,*,#362,.T.);
#364=EDGE_LOOP('',(#158,#165,#174,#183,#190,#204,#218,#225,#240,#255,#260,#26
1,#268,#280,#289,#301,#306,#307,#314,#329,#344,#351,#363));
#365=FACE_BOUND('',#364,.F.);
#366=CARTESIAN_POINT('',(0.0,0.0,0.0));
#367=DIRECTION('',(0.0,0.0,1.0));
#368=DIRECTION('',(1.0,0.0,0.0));
#369=AXIS2_PLACEMENT_3D('',#366,#367,#368);
#370=CYLINDRICAL_SURFACE('',#369,0.186000000000000);
#371=ADVANCED_FACE('',(#365),#370,.T.);
#372=SHAPE_ASPECT('featured shape: face 4','',#43,.T.);
#373=PROPERTY_DEFINITION('','',#372);
#374=FACE_SHAPE_REPRESENTATION('',(#371),#34);
#375=SHAPE_DEFINITION_REPRESENTATION(#373,#374);
#376=CARTESIAN_POINT('',(-0.175184902317523,-
0.062500000000000,2.580000000000000));
#377=VERTEX_POINT('',#376);
#378=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,2.642500000000000));
#379=CARTESIAN_POINT('',(-0.186000000000000,-
0.005466293856892,2.642500000000000));
#380=CARTESIAN_POINT('',(-0.185534991382705,-
0.016091351499283,2.641140739340373));
#381=CARTESIAN_POINT('',(-0.183465542301875,-
0.031773272894993,2.634683632721078));
#382=CARTESIAN_POINT('',(-0.180660130799249,-
0.044778201243767,2.624616708627026));
#383=CARTESIAN_POINT('',(-0.177778707786166,-
0.054914219798778,2.611377795455842));
#384=CARTESIAN_POINT('',(-0.175674414168389,-
0.061153355224084,2.595929010960289));
#385=CARTESIAN_POINT('',(-0.175184902317523,-
0.062500000000000,2.585415348314783));
#386=CARTESIAN_POINT('',(-0.175184902317523,-
0.062500000000000,2.580000000000000));
#387=B_SPLINE_CURVE_WITH_KNOTS('',3,(#378,#379,#380,#381,#382,#383,#384,#385,
#386),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#388=EDGE_CURVE('',#377,#185,#387,.F.);
#389=ORIENTED_EDGE('',*,*,#388,.T.);
#390=ORIENTED_EDGE('',*,*,#189,.F.);
#391=CARTESIAN_POINT('',(-3.416765E-17,-
0.186000000000000,3.800000000000000));
#392=VERTEX_POINT('',#391);
#393=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#394=DIRECTION('',(0.0,0.0,1.0));
#395=DIRECTION('',(-1.0,1.224647E-16,0.0));
#396=AXIS2_PLACEMENT_3D('',#393,#394,#395);
#397=CIRCLE('',#396,0.186000000000000);
#398=EDGE_CURVE('',#176,#392,#397,.T.);
```



```
#399=ORIENTED_EDGE('',*,*,#398,.T.);
#400=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#401=DIRECTION('',(0.0,0.0,1.0));
#402=DIRECTION('',(-1.836970E-16,-1.0,0.0));
#403=AXIS2_PLACEMENT_3D('',#400,#401,#402);
#404=CIRCLE('',#403,0.186000000000000);
#405=EDGE_CURVE('',#392,#160,#404,.T.);
#406=ORIENTED_EDGE('',*,*,#405,.T.);
#407=ORIENTED_EDGE('',*,*,#164,.F.);
#408=CARTESIAN_POINT('',(0.175184902317523,-
0.062500000000000,2.580000000000000));
#409=VERTEX_POINT('',#408);
#410=CARTESIAN_POINT('',(0.175184902317523,-
0.062500000000000,2.580000000000000));
#411=CARTESIAN_POINT('',(0.175184902317523,-
0.062500000000000,2.585418118988794));
#412=CARTESIAN_POINT('',(0.175674738176842,-
0.061152720479089,2.595935968314176));
#413=CARTESIAN_POINT('',(0.177781605611539,-
0.054904708977402,2.611393952136881));
#414=CARTESIAN_POINT('',(0.180661972444195,-
0.044771773357707,2.624624035000569));
#415=CARTESIAN_POINT('',(0.183470250159010,-
0.031747433690688,2.634699511582460));
#416=CARTESIAN_POINT('',(0.185536093618712,-
0.016073651023862,2.641143710134478));
#417=CARTESIAN_POINT('',(0.186000000000000,-
0.005458851337490,2.642500000000000));
#418=CARTESIAN_POINT('',(0.186000000000000,-4.555686E-17,2.642500000000000));
#419=B_SPLINE_CURVE_WITH_KNOTS('',3,(#410,#411,#412,#413,#414,#415,#416,#417,
#418),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#420=EDGE_CURVE('',#146,#409,#419,.F.);
#421=ORIENTED_EDGE('',*,*,#420,.T.);
#422=CARTESIAN_POINT('',(0.186000000000000,-4.555686E-17,2.517500000000000));
#423=CARTESIAN_POINT('',(0.186000000000000,-
0.005464543947207,2.517500000000000));
#424=CARTESIAN_POINT('',(0.185537370272068,-
0.016090248458853,2.518851021196464));
#425=CARTESIAN_POINT('',(0.183449265169815,-
0.031871520572190,2.525369597982180));
#426=CARTESIAN_POINT('',(0.180650079241389,-
0.044811772210324,2.535425428601157));
#427=CARTESIAN_POINT('',(0.177776266176458,-
0.054921858431443,2.548641969314958));
#428=CARTESIAN_POINT('',(0.175677115146707,-
0.061145342990551,2.564048021971347));
#429=CARTESIAN_POINT('',(0.175184902317523,-
0.062500000000000,2.574573290994597));
#430=CARTESIAN_POINT('',(0.175184902317523,-
0.062500000000000,2.580000000000000));
#431=B_SPLINE_CURVE_WITH_KNOTS('',3,(#422,#423,#424,#425,#426,#427,#428,#429,
#430),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
```

```
#432=EDGE_CURVE('',#409,#346,#431,.F.);
#433=ORIENTED_EDGE('',*,*,#432,.T.);
#434=ORIENTED_EDGE('',*,*,#350,.F.);
#435=CARTESIAN_POINT('',(0.160499221181911,-
0.094000000000000,0.250000000000000));
#436=VERTEX_POINT('',#435);
#437=CARTESIAN_POINT('',(0.160499221181911,-
0.094000000000000,0.250000000000000));
#438=CARTESIAN_POINT('',(0.160499221181911,-
0.094000000000000,0.256711807785937));
#439=CARTESIAN_POINT('',(0.161310523171716,-
0.092644858063427,0.269991539898522));
#440=CARTESIAN_POINT('',(0.165181827617091,-
0.085698349461980,0.290561232624993));
#441=CARTESIAN_POINT('',(0.170601634807429,-
0.074584180536545,0.308539596351827));
#442=CARTESIAN_POINT('',(0.176635182649387,-
0.059160281094706,0.324098213138330));
#443=CARTESIAN_POINT('',(0.181763787252156,-
0.041157668374865,0.335433580819143));
#444=CARTESIAN_POINT('',(0.185284237508996,-
0.020320470184614,0.342601935560072));
#445=CARTESIAN_POINT('',(0.186000000000000,-
0.006800366988429,0.344000000000000));
#446=CARTESIAN_POINT('',(0.186000000000000,-4.555686E-17,0.344000000000000));
#447=B_SPLINE_CURVE_WITH_KNOTS('',3,(#437,#438,#439,#440,#441,#442,#443,#444,
#445,#446),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#448=EDGE_CURVE('',#331,#436,#447,.F.);
#449=ORIENTED_EDGE('',*,*,#448,.T.);
#450=CARTESIAN_POINT('',(0.186000000000000,-4.555686E-17,0.156000000000000));
#451=CARTESIAN_POINT('',(0.186000000000000,-
0.006811219284266,0.156000000000000));
#452=CARTESIAN_POINT('',(0.185283487333834,-
0.020349156037856,0.157399284030756));
#453=CARTESIAN_POINT('',(0.181743103977182,-
0.041251906320261,0.164609116136788));
#454=CARTESIAN_POINT('',(0.176618325764372,-
0.059201828025679,0.175944725434112));
#455=CARTESIAN_POINT('',(0.170610275358273,-
0.074562472136122,0.191437081266612));
#456=CARTESIAN_POINT('',(0.165192932924788,-
0.085677048896483,0.209404531600104));
#457=CARTESIAN_POINT('',(0.161327432070906,-
0.092615564162586,0.229880422227216));
#458=CARTESIAN_POINT('',(0.160499221181911,-
0.094000000000000,0.243233553043028));
#459=CARTESIAN_POINT('',(0.160499221181911,-
0.094000000000000,0.250000000000000));
#460=B_SPLINE_CURVE_WITH_KNOTS('',3,(#450,#451,#452,#453,#454,#455,#456,#457,
#458,#459),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#461=EDGE_CURVE('',#436,#309,#460,.F.);
#462=ORIENTED_EDGE('',*,*,#461,.T.);
```



```
#463=ORIENTED_EDGE('',*,*,#313,.F.);
#464=ORIENTED_EDGE('',*,*,#97,.F.);
#465=CARTESIAN_POINT('',(0.031000000000000,-
0.183398473276088,0.617500000000000));
#466=VERTEX_POINT('',#465);
#467=DIRECTION('',(0.0,0.0,1.0));
#468=VECTOR('',#467,0.617500000000000);
#469=LINE('',#82,#468);
#470=EDGE_CURVE('',#84,#466,#469,.T.);
#471=ORIENTED_EDGE('',*,*,#470,.T.);
#472=CARTESIAN_POINT('',(0.018500000000000,-
0.185077686391418,0.630000000000000));
#473=VERTEX_POINT('',#472);
#474=CARTESIAN_POINT('',(0.018500000000000,-
0.185077686391418,0.630000000000000));
#475=CARTESIAN_POINT('',(0.020288448038622,-
0.184898916657693,0.630000000000000));
#476=CARTESIAN_POINT('',(0.023535786372608,-
0.184523602834953,0.629279175846403));
#477=CARTESIAN_POINT('',(0.027638547132190,-
0.183946283585950,0.626500455649787));
#478=CARTESIAN_POINT('',(0.030354567108582,-
0.183509179238234,0.622337258170825));
#479=CARTESIAN_POINT('',(0.031000000000000,-
0.183398473276088,0.619203758810859));
#480=CARTESIAN_POINT('',(0.031000000000000,-
0.183398473276088,0.617500000000000));
#481=B_SPLINE_CURVE_WITH_KNOTS('',3,(#474,#475,#476,#477,#478,#479,#480),.HYP
ERBOLIC_ARC.,.F.,.F.,(4,1,1,1,4),(0.0,0.250000000000000,0.500000000000000,0.7
50000000000000,1.0),.PIECEWISE_BEZIER_KNOTS.);
#482=EDGE_CURVE('',#466,#473,#481,.F.);
#483=ORIENTED_EDGE('',*,*,#482,.T.);
#484=CARTESIAN_POINT('',(-0.018500000000000,-
0.185077686391418,0.630000000000000));
#485=VERTEX_POINT('',#484);
#486=CARTESIAN_POINT('',(0.0,0.0,0.630000000000000));
#487=DIRECTION('',(0.0,0.0,1.0));
#488=DIRECTION('',(-0.099462365591398,-0.995041324685042,0.0));
#489=AXIS2_PLACEMENT_3D('',#486,#487,#488);
#490=CIRCLE('',#489,0.186000000000000);
#491=EDGE_CURVE('',#473,#485,#490,.F.);
#492=ORIENTED_EDGE('',*,*,#491,.T.);
#493=CARTESIAN_POINT('',(-0.031000000000000,-
0.183398473276088,0.617500000000000));
#494=VERTEX_POINT('',#493);
#495=CARTESIAN_POINT('',(-0.031000000000000,-
0.183398473276088,0.617500000000000));
#496=CARTESIAN_POINT('',(-0.031000000000000,-
0.183398473276088,0.619276684284376));
#497=CARTESIAN_POINT('',(-0.030285994332659,-
0.183520653386331,0.622509121211843));
#498=CARTESIAN_POINT('',(-0.027542467419886,-
0.183961116646316,0.626594726403956));
#499=CARTESIAN_POINT('',(-0.023380441464035,-
0.184543239000241,0.629340957979689));
```

```
#500=CARTESIAN_POINT('',(-0.020222806907318,-
0.184905478015536,0.630000000000000));
#501=CARTESIAN_POINT('',(-0.018500000000000,-
0.185077686391418,0.630000000000000));
#502=B_SPLINE_CURVE_WITH_KNOTS('',3,(#495,#496,#497,#498,#499,#500,#501),.HYP
ERBOLIC_ARC.,.F.,.F.,(4,1,1,1,4),(0.0,0.250000000000000,0.500000000000000,0.7
50000000000000,1.0),.PIECEWISE_BEZIER_KNOTS.);
#503=EDGE_CURVE('',#485,#494,#502,.F.);
#504=ORIENTED_EDGE('',*,*,#503,.T.);
#505=DIRECTION('',(0.0,0.0,1.0));
#506=VECTOR('',#505,0.617500000000000);
#507=LINE('',#44,#506);
#508=EDGE_CURVE('',#494,#46,#507,.F.);
#509=ORIENTED_EDGE('',*,*,#508,.T.);
#510=ORIENTED_EDGE('',*,*,#67,.F.);
#511=ORIENTED_EDGE('',*,*,#259,.F.);
#512=CARTESIAN_POINT('',(-0.160499221181911,-
0.094000000000000,0.250000000000000));
#513=VERTEX_POINT('',#512);
#514=CARTESIAN_POINT('',(-0.160499221181911,-
0.094000000000000,0.250000000000000));
#515=CARTESIAN_POINT('',(-0.160499221181911,-
0.094000000000000,0.243237928218130));
#516=CARTESIAN_POINT('',(-0.161326264069271,-
0.092617528844491,0.229892455717941));
#517=CARTESIAN_POINT('',(-0.165188657251781,-
0.085684909394755,0.209422867545884));
#518=CARTESIAN_POINT('',(-0.170601713482259,-
0.074581303037190,0.191462225064981));
#519=CARTESIAN_POINT('',(-0.176608347816713,-
0.059230733088108,0.175968221892797));
#520=CARTESIAN_POINT('',(-0.181735072423795,-
0.041287079595952,0.164625925372114));
#521=CARTESIAN_POINT('',(-0.185280833739912,-
0.020378752353648,0.157404452422345));
#522=CARTESIAN_POINT('',(-0.186000000000000,-
0.006823032072908,0.156000000000000));
#523=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,0.156000000000000));
#524=B_SPLINE_CURVE_WITH_KNOTS('',3,(#514,#515,#516,#517,#518,#519,#520,#521,
#522,#523),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#525=EDGE_CURVE('',#242,#513,#524,.F.);
#526=ORIENTED_EDGE('',*,*,#525,.T.);
#527=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,0.344000000000001));
#528=CARTESIAN_POINT('',(-0.186000000000000,-
0.006812818398979,0.344000000000001));
#529=CARTESIAN_POINT('',(-0.185281410572974,-
0.020351551813128,0.342596407319869));
#530=CARTESIAN_POINT('',(-0.181755664328797,-
0.041193271352137,0.335416626725464));
#531=CARTESIAN_POINT('',(-0.176625064117123,-
0.059189393219094,0.324074232296005));
#532=CARTESIAN_POINT('',(-0.170594327073985,-
0.074600077742975,0.308518027938217));
```



```
#533=CARTESIAN_POINT('',(-0.165177934867762,-
0.085705567809687,0.290544495839190));
#534=CARTESIAN_POINT('',(-0.161309755247970,-
0.092646107527992,0.269982549598585));
#535=CARTESIAN_POINT('',(-0.160499221181911,-
0.094000000000000,0.256708631292058));
#536=CARTESIAN_POINT('',(-0.160499221181911,-
0.094000000000000,0.250000000000000));
#537=B_SPLINE_CURVE_WITH_KNOTS('',3,(#527,#528,#529,#530,#531,#532,#533,#534,
#535,#536),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#538=EDGE_CURVE('',#513,#220,#537,.F.);
#539=ORIENTED_EDGE('',*,*,#538,.T.);
#540=ORIENTED_EDGE('',*,*,#224,.F.);
#541=CARTESIAN_POINT('',(-0.175184902317523,-
0.062500000000000,2.580000000000000));
#542=CARTESIAN_POINT('',(-0.175184902317523,-
0.062500000000000,2.574577143924821));
#543=CARTESIAN_POINT('',(-0.175676556620784,-
0.061146675813468,2.564057738166918));
#544=CARTESIAN_POINT('',(-0.177772939484603,-
0.054932562783283,2.548660674259625));
#545=CARTESIAN_POINT('',(-0.180647351963460,-
0.044822005907028,2.535436160279311));
#546=CARTESIAN_POINT('',(-0.183444961303938,-
0.031895399078691,2.525384006817258));
#547=CARTESIAN_POINT('',(-0.185536257371889,-
0.016107401839008,2.518854091511808));
#548=CARTESIAN_POINT('',(-0.186000000000000,-
0.005471621857183,2.517500000000000));
#549=CARTESIAN_POINT('',(-0.186000000000000,2.277843E-17,2.517500000000000));
#550=B_SPLINE_CURVE_WITH_KNOTS('',3,(#541,#542,#543,#544,#545,#546,#547,#548,
#549),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#551=EDGE_CURVE('',#206,#377,#550,.F.);
#552=ORIENTED_EDGE('',*,*,#551,.T.);
#553=EDGE_LOOP('',(#389,#390,#399,#406,#407,#421,#433,#434,#449,#462,#463,#46
4,#471,#483,#492,#504,#509,#510,#511,#526,#539,#540,#552));
#554=FACE_BOUND('',#553,.F.);
#555=CARTESIAN_POINT('',(0.0,0.0,0.0));
#556=DIRECTION('',(0.0,0.0,1.0));
#557=DIRECTION('',(1.0,0.0,0.0));
#558=AXIS2_PLACEMENT_3D('',#555,#556,#557);
#559=CYLINDRICAL_SURFACE('',#558,0.186000000000000);
#560=ADVANCED_FACE('',(#554),#559,.T.);
#561=SHAPE_ASPECT('','',#43,.T.);
#562=PROPERTY_DEFINITION('','',#561);
#563=FACE_SHAPE_REPRESENTATION('',(#560),#34);
#564=SHAPE_DEFINITION_REPRESENTATION(#562,#563);
#565=ORIENTED_EDGE('',*,*,#51,.F.);
#566=ORIENTED_EDGE('',*,*,#508,.F.);
#567=DIRECTION('',(0.0,-1.0,0.0));
#568=VECTOR('',#567,0.366796946552176);
#569=LINE('',#262,#568);
```

```
#570=EDGE_CURVE('',#494,#263,#569,.F.);
#571=ORIENTED_EDGE('',*,*,#570,.T.);
#572=ORIENTED_EDGE('',*,*,#267,.F.);
#573=EDGE_LOOP('',(#565,#566,#571,#572));
#574=FACE_BOUND('',#573,.F.);
#575=CARTESIAN_POINT('',(-
0.031000000000000,0.094000000000000,0.250000000000000));
#576=CARTESIAN_POINT('',(-0.031000000000000,-
0.094000000000000,0.250000000000000));
#577=VERTEX_POINT('',#575);
#578=VERTEX_POINT('',#576);
#579=CARTESIAN_POINT('',(-0.031000000000000,0.0,0.250000000000000));
#580=DIRECTION('',(-1.0,0.0,0.0));
#581=DIRECTION('',(0.0,-1.0,0.0));
#582=AXIS2_PLACEMENT_3D('',#579,#580,#581);
#583=CIRCLE('',#582,0.094000000000000);
#584=EDGE_CURVE('',#577,#578,#583,.F.);
#585=ORIENTED_EDGE('',*,*,#584,.T.);
#586=CARTESIAN_POINT('',(-0.031000000000000,0.0,0.250000000000000));
#587=DIRECTION('',(-1.0,0.0,0.0));
#588=DIRECTION('',(0.0,1.0,0.0));
#589=AXIS2_PLACEMENT_3D('',#586,#587,#588);
#590=CIRCLE('',#589,0.094000000000000);
#591=EDGE_CURVE('',#578,#577,#590,.F.);
#592=ORIENTED_EDGE('',*,*,#591,.T.);
#593=EDGE_LOOP('',(#585,#592));
#594=FACE_BOUND('',#593,.F.);
#595=CARTESIAN_POINT('',(-0.031000000000000,0.186000000000000,0.0));
#596=DIRECTION('',(-1.0,0.0,0.0));
#597=DIRECTION('',(0.0,0.0,1.0));
#598=AXIS2_PLACEMENT_3D('',#595,#596,#597);
#599=PLANE('',#598);
#600=ADVANCED_FACE('',(#574,#594),#599,.F.);
#601=SHAPE_ASPECT('featured shape: face 6','',#43,.T.);
#602=PROPERTY_DEFINITION('','',#601);
#603=FACE_SHAPE_REPRESENTATION('',(#600),#34);
#604=SHAPE_DEFINITION_REPRESENTATION(#602,#603);
#605=ORIENTED_EDGE('',*,*,#279,.F.);
#606=ORIENTED_EDGE('',*,*,#570,.F.);
#607=ORIENTED_EDGE('',*,*,#503,.F.);
#608=DIRECTION('',(0.0,-1.0,0.0));
#609=VECTOR('',#608,0.370155372782836);
#610=LINE('',#269,#609);
#611=EDGE_CURVE('',#485,#270,#610,.F.);
#612=ORIENTED_EDGE('',*,*,#611,.T.);
#613=EDGE_LOOP('',(#605,#606,#607,#612));
#614=FACE_BOUND('',#613,.F.);
#615=CARTESIAN_POINT('',(-
0.018500000000000,0.186000000000000,0.617500000000000));
#616=DIRECTION('',(0.0,1.0,0.0));
#617=DIRECTION('',(-1.0,0.0,0.0));
#618=AXIS2_PLACEMENT_3D('',#615,#616,#617);
#619=CYLINDRICAL_SURFACE('',#618,0.012500000000000);
#620=ADVANCED_FACE('',(#614),#619,.F.);
#621=SHAPE_ASPECT('featured shape: face 7','',#43,.T.);
#622=PROPERTY_DEFINITION('','',#621);
```



```
#623=FACE_SHAPE_REPRESENTATION('',(#620),#34);
#624=SHAPE_DEFINITION_REPRESENTATION(#622,#623);
#625=ORIENTED_EDGE('',*,*,#288,.F.);
#626=ORIENTED_EDGE('',*,*,#611,.F.);
#627=ORIENTED_EDGE('',*,*,#491,.F.);
#628=DIRECTION('',(0.0,-1.0,0.0));
#629=VECTOR('',#628,0.370155372782836);
#630=LINE('',#281,#629);
#631=EDGE_CURVE('',#473,#282,#630,.F.);
#632=ORIENTED_EDGE('',*,*,#631,.T.);
#633=EDGE_LOOP('',(#625,#626,#627,#632));
#634=FACE_BOUND('',#633,.F.);
#635=CARTESIAN_POINT('',(-
0.018500000000000,0.186000000000000,0.630000000000000));
#636=DIRECTION('',(0.0,0.0,1.0));
#637=DIRECTION('',(1.0,0.0,0.0));
#638=AXIS2_PLACEMENT_3D('',#635,#636,#637);
#639=PLANE('',#638);
#640=ADVANCED_FACE('',(#634),#639,.F.);
#641=SHAPE_ASPECT('featured shape: face 8','',#43,.T.);
#642=PROPERTY_DEFINITION('','',#641);
#643=FACE_SHAPE_REPRESENTATION('',(#640),#34);
#644=SHAPE_DEFINITION_REPRESENTATION(#642,#643);
#645=ORIENTED_EDGE('',*,*,#300,.F.);
#646=ORIENTED_EDGE('',*,*,#631,.F.);
#647=ORIENTED_EDGE('',*,*,#482,.F.);
#648=DIRECTION('',(0.0,-1.0,0.0));
#649=VECTOR('',#648,0.366796946552176);
#650=LINE('',#290,#649);
#651=EDGE_CURVE('',#466,#291,#650,.F.);
#652=ORIENTED_EDGE('',*,*,#651,.T.);
#653=EDGE_LOOP('',(#645,#646,#647,#652));
#654=FACE_BOUND('',#653,.F.);
#655=CARTESIAN_POINT('',(0.018500000000000,0.186000000000000,0.61750000000000
0));
#656=DIRECTION('',(0.0,1.0,0.0));
#657=DIRECTION('',(-1.0,0.0,0.0));
#658=AXIS2_PLACEMENT_3D('',#655,#656,#657);
#659=CYLINDRICAL_SURFACE('',#658,0.012500000000000);
#660=ADVANCED_FACE('',(#654),#659,.F.);
#661=SHAPE_ASPECT('featured shape: face 9','',#43,.T.);
#662=PROPERTY_DEFINITION('','',#661);
#663=FACE_SHAPE_REPRESENTATION('',(#660),#34);
#664=SHAPE_DEFINITION_REPRESENTATION(#662,#663);
#665=ORIENTED_EDGE('',*,*,#88,.F.);
#666=ORIENTED_EDGE('',*,*,#305,.F.);
#667=ORIENTED_EDGE('',*,*,#651,.F.);
#668=ORIENTED_EDGE('',*,*,#470,.F.);
#669=EDGE_LOOP('',(#665,#666,#667,#668));
#670=FACE_BOUND('',#669,.F.);
#671=CARTESIAN_POINT('',(0.031000000000000,-
0.094000000000000,0.250000000000000));
#672=CARTESIAN_POINT('',(0.031000000000000,0.094000000000000,0.25000000000000
0));
#673=VERTEX_POINT('',#671);
#674=VERTEX_POINT('',#672);
```

```
#675=CARTESIAN_POINT('',(0.031000000000000,0.0,0.250000000000000));
#676=DIRECTION('',(-1.0,0.0,0.0));
#677=DIRECTION('',(0.0,-1.0,0.0));
#678=AXIS2_PLACEMENT_3D('',#675,#676,#677);
#679=CIRCLE('',#678,0.094000000000000);
#680=EDGE_CURVE('',#673,#674,#679,.T.);
#681=ORIENTED_EDGE('',*,*,#680,.T.);
#682=CARTESIAN_POINT('',(0.031000000000000,0.0,0.250000000000000));
#683=DIRECTION('',(-1.0,0.0,0.0));
#684=DIRECTION('',(0.0,1.0,0.0));
#685=AXIS2_PLACEMENT_3D('',#682,#683,#684);
#686=CIRCLE('',#685,0.094000000000000);
#687=EDGE_CURVE('',#674,#673,#686,.T.);
#688=ORIENTED_EDGE('',*,*,#687,.T.);
#689=EDGE_LOOP('',(#681,#688));
#690=FACE_BOUND('',#689,.F.);
#691=CARTESIAN_POINT('',(0.031000000000000,0.186000000000000,0.61750000000000
0));
#692=DIRECTION('',(1.0,0.0,0.0));
#693=DIRECTION('',(0.0,0.0,-1.0));
#694=AXIS2_PLACEMENT_3D('',#691,#692,#693);
#695=PLANE('',#694);
#696=ADVANCED_FACE('',(#670,#690),#695,.F.);
#697=SHAPE_ASPECT('featured shape: face 10','',#43,.T.);
#698=PROPERTY_DEFINITION('','',#697);
#699=FACE_SHAPE_REPRESENTATION('',(#696),#34);
#700=SHAPE_DEFINITION_REPRESENTATION(#698,#699);
#701=ORIENTED_EDGE('',*,*,#343,.F.);
#702=DIRECTION('',(-1.0,0.0,0.0));
#703=VECTOR('',#702,0.129499221181911);
#704=LINE('',#315,#703);
#705=EDGE_CURVE('',#316,#674,#704,.T.);
#706=ORIENTED_EDGE('',*,*,#705,.T.);
#707=ORIENTED_EDGE('',*,*,#680,.F.);
#708=DIRECTION('',(-1.0,0.0,0.0));
#709=VECTOR('',#708,0.129499221181911);
#710=LINE('',#435,#709);
#711=EDGE_CURVE('',#673,#436,#710,.F.);
#712=ORIENTED_EDGE('',*,*,#711,.T.);
#713=ORIENTED_EDGE('',*,*,#448,.F.);
#714=EDGE_LOOP('',(#701,#706,#707,#712,#713));
#715=FACE_BOUND('',#714,.F.);
#716=CARTESIAN_POINT('',(0.186000000000000,0.0,0.250000000000000));
#717=DIRECTION('',(-1.0,0.0,0.0));
#718=DIRECTION('',(0.0,-1.0,0.0));
#719=AXIS2_PLACEMENT_3D('',#716,#717,#718);
#720=CYLINDRICAL_SURFACE('',#719,0.094000000000000);
#721=ADVANCED_FACE('',(#715),#720,.F.);
#722=SHAPE_ASPECT('featured shape: face 11','',#43,.T.);
#723=PROPERTY_DEFINITION('','',#722);
#724=FACE_SHAPE_REPRESENTATION('',(#721),#34);
#725=SHAPE_DEFINITION_REPRESENTATION(#723,#724);
#726=ORIENTED_EDGE('',*,*,#239,.F.);
#727=ORIENTED_EDGE('',*,*,#538,.F.);
#728=DIRECTION('',(-1.0,0.0,0.0));
#729=VECTOR('',#728,0.129499221181911);
```



```
#730=LINE('',#576,#729);
#731=EDGE_CURVE('',#513,#578,#730,.F.);
#732=ORIENTED_EDGE('',*,*,#731,.T.);
#733=ORIENTED_EDGE('',*,*,#584,.F.);
#734=DIRECTION('',(-1.0,0.0,0.0));
#735=VECTOR('',#734,0.129499221181911);
#736=LINE('',#575,#735);
#737=EDGE_CURVE('',#577,#227,#736,.T.);
#738=ORIENTED_EDGE('',*,*,#737,.T.);
#739=EDGE_LOOP('',(#726,#727,#732,#733,#738));
#740=FACE_BOUND('',#739,.F.);
#741=ADVANCED_FACE('',(#740),#720,.F.);
#742=SHAPE_ASPECT('featured shape: face 12','',#43,.T.);
#743=PROPERTY_DEFINITION('','',#742);
#744=FACE_SHAPE_REPRESENTATION('',(#741),#34);
#745=SHAPE_DEFINITION_REPRESENTATION(#743,#744);
#746=ORIENTED_EDGE('',*,*,#254,.F.);
#747=ORIENTED_EDGE('',*,*,#737,.F.);
#748=ORIENTED_EDGE('',*,*,#591,.F.);
#749=ORIENTED_EDGE('',*,*,#731,.F.);
#750=ORIENTED_EDGE('',*,*,#525,.F.);
#751=EDGE_LOOP('',(#746,#747,#748,#749,#750));
#752=FACE_BOUND('',#751,.F.);
#753=CARTESIAN_POINT('',(0.186000000000000,0.0,0.250000000000000));
#754=DIRECTION('',(-1.0,0.0,0.0));
#755=DIRECTION('',(0.0,-1.0,0.0));
#756=AXIS2_PLACEMENT_3D('',#753,#754,#755);
#757=CYLINDRICAL_SURFACE('',#756,0.094000000000000);
#758=ADVANCED_FACE('',(#752),#757,.F.);
#759=SHAPE_ASPECT('featured shape: face 13','',#43,.T.);
#760=PROPERTY_DEFINITION('','',#759);
#761=FACE_SHAPE_REPRESENTATION('',(#758),#34);
#762=SHAPE_DEFINITION_REPRESENTATION(#760,#761);
#763=ORIENTED_EDGE('',*,*,#328,.F.);
#764=ORIENTED_EDGE('',*,*,#461,.F.);
#765=ORIENTED_EDGE('',*,*,#711,.F.);
#766=ORIENTED_EDGE('',*,*,#687,.F.);
#767=ORIENTED_EDGE('',*,*,#705,.F.);
#768=EDGE_LOOP('',(#763,#764,#765,#766,#767));
#769=FACE_BOUND('',#768,.F.);
#770=ADVANCED_FACE('',(#769),#757,.F.);
#771=SHAPE_ASPECT('featured shape: face 14','',#43,.T.);
#772=PROPERTY_DEFINITION('','',#771);
#773=FACE_SHAPE_REPRESENTATION('',(#770),#34);
#774=SHAPE_DEFINITION_REPRESENTATION(#772,#773);
#775=CARTESIAN_POINT('',(-7.654042E-18,0.125000000000000,3.940000000000000));
#776=CARTESIAN_POINT('',(-7.654042E-18,-
0.125000000000000,3.940000000000000));
#777=VERTEX_POINT('',#775);
#778=VERTEX_POINT('',#776);
#779=CARTESIAN_POINT('',(0.0,0.0,3.940000000000000));
#780=DIRECTION('',(0.0,0.0,1.0));
#781=DIRECTION('',(0.0,-1.0,0.0));
#782=AXIS2_PLACEMENT_3D('',#779,#780,#781);
#783=CIRCLE('',#782,0.125000000000000);
#784=EDGE_CURVE('',#777,#778,#783,.F.);
```

```
#785=ORIENTED_EDGE('',*,*,#784,.T.);
#786=CARTESIAN_POINT('',(0.0,-0.125000000000000,4.179999999999999));
#787=VERTEX_POINT('',#786);
#788=DIRECTION('',(0.0,0.0,-1.0));
#789=VECTOR('',#788,0.240000000000001);
#790=LINE('',#786,#789);
#791=EDGE_CURVE('',#778,#787,#790,.F.);
#792=ORIENTED_EDGE('',*,*,#791,.T.);
#793=CARTESIAN_POINT('',(-1.530808E-17,0.125000000000000,4.180000000000000));
#794=VERTEX_POINT('',#793);
#795=CARTESIAN_POINT('',(0.0,0.0,4.180000000000000));
#796=DIRECTION('',(0.0,0.0,1.0));
#797=DIRECTION('',(0.0,-1.0,0.0));
#798=AXIS2_PLACEMENT_3D('',#795,#796,#797);
#799=CIRCLE('',#798,0.125000000000000);
#800=EDGE_CURVE('',#787,#794,#799,.T.);
#801=ORIENTED_EDGE('',*,*,#800,.T.);
#802=DIRECTION('',(0.0,0.0,-1.0));
#803=VECTOR('',#802,0.240000000000000);
#804=LINE('',#793,#803);
#805=EDGE_CURVE('',#794,#777,#804,.T.);
#806=ORIENTED_EDGE('',*,*,#805,.T.);
#807=EDGE_LOOP('',(#785,#792,#801,#806));
#808=FACE_BOUND('',#807,.F.);
#809=CARTESIAN_POINT('',(0.0,0.0,3.921499999999999));
#810=DIRECTION('',(0.0,0.0,1.0));
#811=DIRECTION('',(0.0,-1.0,0.0));
#812=AXIS2_PLACEMENT_3D('',#809,#810,#811);
#813=CYLINDRICAL_SURFACE('',#812,0.125000000000000);
#814=ADVANCED_FACE('',(#808),#813,.T.);
#815=SHAPE_ASPECT('featured shape: face 15','',#43,.T.);
#816=PROPERTY_DEFINITION('','',#815);
#817=FACE_SHAPE_REPRESENTATION('',(#814),#34);
#818=SHAPE_DEFINITION_REPRESENTATION(#816,#817);
#819=CARTESIAN_POINT('',(0.0,0.0,3.939999999999999));
#820=DIRECTION('',(0.0,0.0,-1.0));
#821=DIRECTION('',(0.0,-1.0,0.0));
#822=AXIS2_PLACEMENT_3D('',#819,#820,#821);
#823=CIRCLE('',#822,0.125000000000000);
#824=EDGE_CURVE('',#778,#777,#823,.T.);
#825=ORIENTED_EDGE('',*,*,#824,.T.);
#826=ORIENTED_EDGE('',*,*,#805,.F.);
#827=CARTESIAN_POINT('',(0.0,0.0,4.180000000000000));
#828=DIRECTION('',(0.0,0.0,1.0));
#829=DIRECTION('',(-1.224647E-16,1.0,0.0));
#830=AXIS2_PLACEMENT_3D('',#827,#828,#829);
#831=CIRCLE('',#830,0.125000000000000);
#832=EDGE_CURVE('',#794,#787,#831,.T.);
#833=ORIENTED_EDGE('',*,*,#832,.T.);
#834=ORIENTED_EDGE('',*,*,#791,.F.);
#835=EDGE_LOOP('',(#825,#826,#833,#834));
#836=FACE_BOUND('',#835,.F.);
#837=CARTESIAN_POINT('',(0.0,0.0,3.921499999999999));
#838=DIRECTION('',(0.0,0.0,1.0));
#839=DIRECTION('',(0.0,-1.0,0.0));
#840=AXIS2_PLACEMENT_3D('',#837,#838,#839);
```



```
#841=CYLINDRICAL_SURFACE('',#840,0.125000000000000);
#842=ADVANCED_FACE('',(#836),#841,.T.);
#843=SHAPE_ASPECT('featured shape: face 16','',#43,.T.);
#844=PROPERTY_DEFINITION('','',#843);
#845=FACE_SHAPE_REPRESENTATION('',(#842),#34);
#846=SHAPE_DEFINITION_REPRESENTATION(#844,#845);
#847=ORIENTED_EDGE('',*,*,#122,.F.);
#848=DIRECTION('',(-3.169619E-17,0.258819045102521,-0.965925826289068));
#849=VECTOR('',#848,0.134585903453311);
#850=LINE('',#113,#849);
#851=EDGE_CURVE('',#115,#794,#850,.T.);
#852=ORIENTED_EDGE('',*,*,#851,.T.);
#853=ORIENTED_EDGE('',*,*,#800,.F.);
#854=DIRECTION('',(0.0,-0.258819045102521,-0.965925826289068));
#855=VECTOR('',#854,0.134585903453311);
#856=LINE('',#114,#855);
#857=EDGE_CURVE('',#787,#116,#856,.F.);
#858=ORIENTED_EDGE('',*,*,#857,.T.);
#859=EDGE_LOOP('',(#847,#852,#853,#858));
#860=FACE_BOUND('',#859,.F.);
#861=CARTESIAN_POINT('',(0.0,0.0,4.180000000000000));
#862=DIRECTION('',(0.0,0.0,-1.0));
#863=DIRECTION('',(0.0,1.0,0.0));
#864=AXIS2_PLACEMENT_3D('',#861,#862,#863);
#865=CONICAL_SURFACE('',#864,0.125000000000000,15.000000000000009);
#866=ADVANCED_FACE('',(#860),#865,.T.);
#867=SHAPE_ASPECT('featured shape: face 17','',#43,.T.);
#868=PROPERTY_DEFINITION('','',#867);
#869=FACE_SHAPE_REPRESENTATION('',(#866),#34);
#870=SHAPE_DEFINITION_REPRESENTATION(#868,#869);
#871=ORIENTED_EDGE('',*,*,#129,.F.);
#872=ORIENTED_EDGE('',*,*,#857,.F.);
#873=ORIENTED_EDGE('',*,*,#832,.F.);
#874=ORIENTED_EDGE('',*,*,#851,.F.);
#875=EDGE_LOOP('',(#871,#872,#873,#874));
#876=FACE_BOUND('',#875,.F.);
#877=CARTESIAN_POINT('',(0.0,0.0,4.180000000000000));
#878=DIRECTION('',(0.0,0.0,-1.0));
#879=DIRECTION('',(0.0,1.0,0.0));
#880=AXIS2_PLACEMENT_3D('',#877,#878,#879);
#881=CONICAL_SURFACE('',#880,0.125000000000000,15.000000000000009);
#882=ADVANCED_FACE('',(#876),#881,.T.);
#883=SHAPE_ASPECT('featured shape: face 18','',#43,.T.);
#884=PROPERTY_DEFINITION('','',#883);
#885=FACE_SHAPE_REPRESENTATION('',(#882),#34);
#886=SHAPE_DEFINITION_REPRESENTATION(#884,#885);
#887=ORIENTED_EDGE('',*,*,#173,.F.);
#888=ORIENTED_EDGE('',*,*,#405,.F.);
#889=DIRECTION('',(4.891782E-17,0.399444320513099,0.916757456915311));
#890=VECTOR('',#889,0.152712147519443);
#891=LINE('',#391,#890);
#892=EDGE_CURVE('',#392,#778,#891,.T.);
#893=ORIENTED_EDGE('',*,*,#892,.T.);
#894=ORIENTED_EDGE('',*,*,#784,.F.);
#895=DIRECTION('',(0.0,-0.399444320513099,0.916757456915311));
#896=VECTOR('',#895,0.152712147519443);
```



```
#897=LINE('',#166,#896);
#898=EDGE_CURVE('',#777,#167,#897,.F.);
#899=ORIENTED_EDGE('',*,*,#898,.T.);
#900=EDGE_LOOP('',(#887,#888,#893,#894,#899));
#901=FACE_BOUND('',#900,.F.);
#902=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#903=DIRECTION('',(0.0,0.0,-1.0));
#904=DIRECTION('',(0.0,1.0,0.0));
#905=AXIS2_PLACEMENT_3D('',#902,#903,#904);
#906=CONICAL_SURFACE('',#905,0.186000000000000,23.543444867436634);
#907=ADVANCED_FACE('',(#901),#906,.T.);
#908=SHAPE_ASPECT('featured shape: face 19','',#43,.T.);
#909=PROPERTY_DEFINITION('','',#908);
#910=FACE_SHAPE_REPRESENTATION('',(#907),#34);
#911=SHAPE_DEFINITION_REPRESENTATION(#909,#910);
#912=ORIENTED_EDGE('',*,*,#182,.F.);
#913=ORIENTED_EDGE('',*,*,#898,.F.);
#914=ORIENTED_EDGE('',*,*,#824,.F.);
#915=ORIENTED_EDGE('',*,*,#892,.F.);
#916=ORIENTED_EDGE('',*,*,#398,.F.);
#917=EDGE_LOOP('',(#912,#913,#914,#915,#916));
#918=FACE_BOUND('',#917,.F.);
#919=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#920=DIRECTION('',(0.0,0.0,-1.0));
#921=DIRECTION('',(0.0,1.0,0.0));
#922=AXIS2_PLACEMENT_3D('',#919,#920,#921);
#923=CONICAL_SURFACE('',#922,0.186000000000000,23.543444867436634);
#924=ADVANCED_FACE('',(#918),#923,.T.);
#925=SHAPE_ASPECT('featured shape: face 20','',#43,.T.);
#926=PROPERTY_DEFINITION('','',#925);
#927=FACE_SHAPE_REPRESENTATION('',(#924),#34);
#928=SHAPE_DEFINITION_REPRESENTATION(#926,#927);
#929=ORIENTED_EDGE('',*,*,#157,.F.);
#930=DIRECTION('',(-1.0,0.0,0.0));
#931=VECTOR('',#930,0.350369804635046);
#932=LINE('',#143,#931);
#933=EDGE_CURVE('',#145,#192,#932,.T.);
#934=ORIENTED_EDGE('',*,*,#933,.T.);
#935=ORIENTED_EDGE('',*,*,#203,.F.);
#936=ORIENTED_EDGE('',*,*,#388,.F.);
#937=DIRECTION('',(-1.0,0.0,0.0));
#938=VECTOR('',#937,0.350369804635046);
#939=LINE('',#408,#938);
#940=EDGE_CURVE('',#377,#409,#939,.F.);
#941=ORIENTED_EDGE('',*,*,#940,.T.);
#942=ORIENTED_EDGE('',*,*,#420,.F.);
#943=EDGE_LOOP('',(#929,#934,#935,#936,#941,#942));
#944=FACE_BOUND('',#943,.F.);
#945=CARTESIAN_POINT('',(0.186000000000000,0.0,2.580000000000000));
#946=DIRECTION('',(-1.0,0.0,0.0));
#947=DIRECTION('',(0.0,-1.0,0.0));
#948=AXIS2_PLACEMENT_3D('',#945,#946,#947);
#949=CYLINDRICAL_SURFACE('',#948,0.062500000000000);
#950=ADVANCED_FACE('',(#944),#949,.F.);
#951=SHAPE_ASPECT('featured shape: face 21','',#43,.T.);
#952=PROPERTY_DEFINITION('','',#951);
```



```
#953=FACE_SHAPE_REPRESENTATION('',(#950),#34);
#954=SHAPE_DEFINITION_REPRESENTATION(#952,#953);
#955=ORIENTED_EDGE('',*,*,#217,.F.);
#956=ORIENTED_EDGE('',*,*,#933,.F.);
#957=ORIENTED_EDGE('',*,*,#362,.F.);
#958=ORIENTED_EDGE('',*,*,#432,.F.);
#959=ORIENTED_EDGE('',*,*,#940,.F.);
#960=ORIENTED_EDGE('',*,*,#551,.F.);
#961=EDGE_LOOP('',(#955,#956,#957,#958,#959,#960));
#962=FACE_BOUND('',#961,.F.);
#963=CARTESIAN_POINT('',(0.186000000000000,0.0,2.580000000000000));
#964=DIRECTION('',(-1.0,0.0,0.0));
#965=DIRECTION('',(0.0,-1.0,0.0));
#966=AXIS2_PLACEMENT_3D('',#963,#964,#965);
#967=CYLINDRICAL_SURFACE('',#966,0.062500000000000);
#968=ADVANCED_FACE('',(#962),#967,.F.);
#969=SHAPE_ASPECT('featured shape: face 22','',#43,.T.);
#970=PROPERTY_DEFINITION('','',#969);
#971=FACE_SHAPE_REPRESENTATION('',(#968),#34);
#972=SHAPE_DEFINITION_REPRESENTATION(#970,#971);
#973=CLOSED_SHELL('',(#76,#108,#138,#371,#560,#600,#620,#640,#660,#696,#721,#
741,#758,#770,#814,#842,#866,#882,#907,#924,#950,#968));
#974=MANIFOLD_SOLID_BREP('',#973);
#975=ADVANCED_BREP_SHAPE_REPRESENTATION('',(#974),#34);
#976=SHAPE_DEFINITION_REPRESENTATION(#43,#975);
#977=CARTESIAN_POINT('',(0.0,0.0,0.0));
#978=DIRECTION('',(0.0,1.0,0.0));
#979=DIRECTION('',(0.0,0.0,1.0));
#980=AXIS2_PLACEMENT_3D('DTM1',#977,#978,#979);
#981=DATUM_FEATURE('','',#43,.T.);
#982=PROPERTY_DEFINITION('','',#981);
#983=SHAPE_REPRESENTATION('',(#980),#34);
#984=SHAPE_DEFINITION_REPRESENTATION(#982,#983);
#985=DATUM('DTM1','',#43,.F.,'DTM1');
#986=SHAPE_ASPECT_RELATIONSHIP('','',#981,#985);
#987=DATUM_REFERENCE(1,#985);
#988=CARTESIAN_POINT('',(0.0,0.0,0.0));
#989=DIRECTION('',(-1.0,0.0,0.0));
#990=DIRECTION('',(0.0,0.0,1.0));
#991=AXIS2_PLACEMENT_3D('DTM2',#988,#989,#990);
#992=DATUM_FEATURE('','',#43,.T.);
#993=PROPERTY_DEFINITION('','',#992);
#994=SHAPE_REPRESENTATION('',(#991),#34);
#995=SHAPE_DEFINITION_REPRESENTATION(#993,#994);
#996=DATUM('DTM2','',#43,.F.,'DTM2');
#997=SHAPE_ASPECT_RELATIONSHIP('','',#992,#996);
#998=DATUM_REFERENCE(1,#996);
#999=CARTESIAN_POINT('',(0.186000000000000,0.0,2.155000000000000));
#1000=DIRECTION('',(-1.0,0.0,0.0));
#1001=DIRECTION('',(0.0,-1.0,0.0));
#1002=AXIS2_PLACEMENT_3D('DTM3',#999,#1000,#1001);
#1003=DATUM_FEATURE('','',#43,.T.);
#1004=PROPERTY_DEFINITION('','',#1003);
#1005=SHAPE_REPRESENTATION('',(#1002),#34);
#1006=SHAPE_DEFINITION_REPRESENTATION(#1004,#1005);
#1007=DATUM('DTM3','',#43,.F.,'DTM3');
```

```
#1008=SHAPE_ASPECT_RELATIONSHIP('','',#1003,#1007);
#1009=DATUM_REFERENCE(1,#1007);
#1010=CARTESIAN_POINT('',(0.0,0.186000000000000,2.155000000000000));
#1011=DIRECTION('',(0.0,1.0,0.0));
#1012=DIRECTION('',(1.0,0.0,0.0));
#1013=AXIS2_PLACEMENT_3D('DTM4',#1010,#1011,#1012);
#1014=DATUM_FEATURE('','',#43,.T.);
#1015=PROPERTY_DEFINITION('','',#1014);
#1016=SHAPE_REPRESENTATION('',(#1013),#34);
#1017=SHAPE_DEFINITION_REPRESENTATION(#1015,#1016);
#1018=DATUM('DTM4','',#43,.F.,'DTM4');
#1019=SHAPE_ASPECT_RELATIONSHIP('','',#1014,#1018);
#1020=DATUM_REFERENCE(1,#1018);
#1021=CARTESIAN_POINT('',(0.186000000000000,0.0,0.250000000000000));
#1022=DIRECTION('',(0.0,0.0,-1.0));
#1023=DIRECTION('',(-1.0,0.0,0.0));
#1024=AXIS2_PLACEMENT_3D('DTM5',#1021,#1022,#1023);
#1025=DATUM_FEATURE('','',#43,.T.);
#1026=PROPERTY_DEFINITION('','',#1025);
#1027=SHAPE_REPRESENTATION('',(#1024),#34);
#1028=SHAPE_DEFINITION_REPRESENTATION(#1026,#1027);
#1029=DATUM('DTM5','',#43,.T.,'DTM5');
#1030=SHAPE_ASPECT_RELATIONSHIP('','',#1025,#1029);
#1031=DATUM_REFERENCE(1,#1029);
#1032=SHAPE_ASPECT('','',#43,.T.);

#1061=(CHARACTERIZED_OBJECT('','')
       FEATURE_DEFINITION()
       INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.)
       SLOT());
#1062=PROPERTY_DEFINITION('','',#1061);
#1063=PRODUCT_DEFINITION_SHAPE('','',#1061);
#1064=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#1065=DIRECTION('',(0.0,1.0,0.0));
#1066=DIRECTION('',(1.0,0.0,0.0));
#1067=AXIS2_PLACEMENT_3D('orientation',#1064,#1065,#1066);

#1077=SHAPE_REPRESENTATION('',(#600,#620,#640,#660,#696),#34);
#1078=SHAPE_DEFINITION_REPRESENTATION(#1062,#1077);
#1079=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1067),#34);
#1080=SHAPE_DEFINITION_REPRESENTATION(#1062,#1079);
#1081=FEATURE_COMPONENT_DEFINITION('','');
#1082=PRODUCT_DEFINITION_SHAPE('','',#1081);

#1083=SQUARE_U_PROFILE('','swept shape',#1082,.F.);
#1084=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#1085=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#1086=PRECISION_QUALIFIER(4);
#1092=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.062000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1084
,#1085,#1086))REPRESENTATION_ITEM('width'));
#1097=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
```



```
#1102=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#1103=STANDARD_UNCERTAINTY('upper limit','plus',0.007500000000000);
#1104=STANDARD_UNCERTAINTY('lower limit','minus',-0.007500000000000);
#1105=PRECISION_QUALIFIER(4);
#1111=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.012500000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1103
,#1104,#1105))REPRESENTATION_ITEM('first radius'));
#1112=STANDARD_UNCERTAINTY('upper limit','plus',0.007500000000000);
#1113=STANDARD_UNCERTAINTY('lower limit','minus',-0.007500000000000);
#1114=PRECISION_QUALIFIER(4);
#1120=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.012500000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1112
,#1113,#1114))REPRESENTATION_ITEM('second radius'));
#1121=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1067,#1092,#1097,#1102,#1111,
#1120),#34);
#1122=PROPERTY_DEFINITION('','',#1083);
#1123=SHAPE_DEFINITION_REPRESENTATION(#1122,#1121);
#1124=SHAPE_ASPECT('','swept shape occurrence',#1063,.T.);
#1125=SHAPE_DEFINING_RELATIONSHIP('','profile usage',#1083,#1124);
#1126=FEATURE_COMPONENT_DEFINITION('','');
#1127=PRODUCT_DEFINITION_SHAPE('','',#1126);

#1128=PATH_FEATURE_COMPONENT('linear path','linear',#1127,.F.);
#1133=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.372000000000000),#15)REPRESENTATION_ITEM('distance'));
#1134=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1067,#1133),#34);
#1135=PROPERTY_DEFINITION('','',#1128);
#1136=SHAPE_DEFINITION_REPRESENTATION(#1135,#1134);
#1137=DIRECTION_SHAPE_REPRESENTATION('',(#1065),#34);
#1138=PROPERTY_DEFINITION_REPRESENTATION(#1135,#1137);
#1139=SHAPE_ASPECT('','course of travel occurrence',#1063,.T.);
#1140=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#1128,#1139);
#1141=FEATURE_COMPONENT_DEFINITION('','');
#1142=PRODUCT_DEFINITION_SHAPE('','',#1141);
#1143=SLOT_END('','open',#1142,.F.);
#1144=FEATURE_COMPONENT_DEFINITION('','');
#1145=PRODUCT_DEFINITION_SHAPE('','',#1144);
#1146=SLOT_END('','open',#1145,.F.);
#1147=SHAPE_ASPECT('','end condition occurrence',#1063,.T.);
#1148=FEATURE_COMPONENT_RELATIONSHIP('course of travel start','slot end
usage',#1143,#1147);
#1149=SHAPE_ASPECT('','end condition occurrence',#1063,.T.);
#1150=FEATURE_COMPONENT_RELATIONSHIP('course of travel end','slot end
usage',#1146,#1149);

#1157=(CHARACTERIZED_OBJECT('','')
       FEATURE_DEFINITION()
       INSTANCED_FEATURE()
       ROUND_HOLE()
       SHAPE_ASPECT('','',#43,.T.));
#1158=PROPERTY_DEFINITION('','',#1157);
#1159=PRODUCT_DEFINITION_SHAPE('','',#1157);
#1160=CARTESIAN_POINT('',(0.186000000000000,1.138922E-17,0.250000000000000));
#1161=DIRECTION('',(-1.0,0.0,0.0));
```

```
#1162=DIRECTION('',(0.0,-1.0,0.0));
#1163=AXIS2_PLACEMENT_3D('orientation',#1160,#1161,#1162);
#1164=SHAPE_REPRESENTATION('',(#721,#741,#758,#770),#34);
#1165=SHAPE_DEFINITION_REPRESENTATION(#1158,#1164);
#1166=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1163),#34);
#1167=SHAPE_DEFINITION_REPRESENTATION(#1158,#1166);
#1168=FEATURE_COMPONENT_DEFINITION('','');
#1169=PRODUCT_DEFINITION_SHAPE('','',#1168);
#1170=CIRCULAR_CLOSED_PROFILE('circular profile','',#1169,.F.);
#1171=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#1172=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#1173=PRECISION_QUALIFIER(4);
#1179=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.188000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1171
,#1172,#1173))REPRESENTATION_ITEM('diameter'));
#1180=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1163,#1179),#34);
#1181=PROPERTY_DEFINITION('','',#1170);
#1182=SHAPE_DEFINITION_REPRESENTATION(#1181,#1180);
#1183=FEATURE_COMPONENT_DEFINITION('','');
#1184=PRODUCT_DEFINITION_SHAPE('','',#1183);
#1185=PATH_FEATURE_COMPONENT('linear path','linear',#1184,.F.);
#1190=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.372000000000000),#15)REPRESENTATION_ITEM('distance'));
#1191=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1163,#1190),#34);
#1192=PROPERTY_DEFINITION('','',#1185);
#1193=SHAPE_DEFINITION_REPRESENTATION(#1192,#1191);
#1194=DIRECTION_SHAPE_REPRESENTATION('',(#1161),#34);
#1195=PROPERTY_DEFINITION_REPRESENTATION(#1192,#1194);
#1196=SHAPE_ASPECT('','diameter occurrence',#1159,.T.);
#1197=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#1170,#1196);
#1198=SHAPE_ASPECT('','hole depth occurrence',#1159,.T.);
#1199=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#1185,#1198);

#1200=FEATURE_COMPONENT_DEFINITION('','');
#1201=PRODUCT_DEFINITION_SHAPE('','',#1200);
#1202=HOLE_BOTTOM('bottom condition','through',#1201,.F.);
#1203=SHAPE_ASPECT('','bottom condition occurrence',#1159,.T.);
#1204=FEATURE_COMPONENT_RELATIONSHIP('','hole bottom usage',#1202,#1203);
#1205=CARTESIAN_POINT('',(0.186000000000000,1.387779E-17,0.250000000000000));
#1206=DIRECTION('',(-1.0,0.0,0.0));
#1207=AXIS1_PLACEMENT('',#1205,#1206);
#1208=SHAPE_ASPECT('','',#43,.F.);
#1209=PROPERTY_DEFINITION('','',#1208);
#1210=SHAPE_REPRESENTATION('',(#1207),#34);
#1211=SHAPE_DEFINITION_REPRESENTATION(#1209,#1210);
#1212=DIMENSIONAL_LOCATION('','',#77,#1208);
#1213=PRECISION_QUALIFIER(4);
#1219=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.250000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1213
))REPRESENTATION_ITEM(''));
#1220=SHAPE_DIMENSION_REPRESENTATION('',(#1219),#34);
#1221=PRECISION_QUALIFIER(4);
#1222=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.030000000000000),#15);
#1223=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.030000000000000),#15);
#1224=MEASURE_QUALIFICATION('','',#1222,(#1221));
```



```
#1225=MEASURE_QUALIFICATION('','',#1223,(#1221));
#1226=TOLERANCE_VALUE(#1223,#1222);
#1227=PLUS_MINUS_TOLERANCE(#1226,#1212);
#1228=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#1212,#1220);
#1235=(CHARACTERIZED_OBJECT('','outer                diameter                to
shoulder')FEATURE_DEFINITION()INSTANCED_FEATURE()OUTER_ROUND()SHAPE_ASPECT(''
,'outer diameter to shoulder',#43,.T.));
#1236=PROPERTY_DEFINITION('','',#1235);
#1237=PRODUCT_DEFINITION_SHAPE('','',#1235);
#1238=CARTESIAN_POINT('',(0.0,-3.235562E-32,3.940000000000000));
#1239=DIRECTION('',(0.0,0.0,1.0));
#1240=DIRECTION('',(0.0,-1.0,0.0));
#1241=AXIS2_PLACEMENT_3D('orientation',#1238,#1239,#1240);
#1242=STANDARD_UNCERTAINTY('upper limit','plus',0.030000000000000);
#1243=STANDARD_UNCERTAINTY('lower limit','minus',-0.030000000000000);
#1244=PRECISION_QUALIFIER(4);
#1250=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.250000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1242
,#1243,#1244))REPRESENTATION_ITEM('diameter'));
#1251=SHAPE_REPRESENTATION('',(#814,#842),#34);
#1252=SHAPE_DEFINITION_REPRESENTATION(#1236,#1251);
#1253=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1241,#1250),#34);
#1254=SHAPE_DEFINITION_REPRESENTATION(#1236,#1253);
#1255=CARTESIAN_POINT('',(0.0,-0.125000000000000,3.940000000000000));
#1256=DIRECTION('',(1.0,0.0,0.0));
#1257=DIRECTION('',(0.0,0.0,1.0));
#1258=AXIS2_PLACEMENT_3D('orientation',#1255,#1256,#1257);
#1259=FEATURE_COMPONENT_DEFINITION('','');
#1260=PRODUCT_DEFINITION_SHAPE('','',#1259);
#1261=VEE_PROFILE('vee profile','v-shape',#1260,.F.);
#1266=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#1271=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.0
),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#1276=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.0),#15)REPRESENTATION_ITEM('profile radius'));
#1277=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1258,#1266,#1271,#1276),#34);
#1278=PROPERTY_DEFINITION('','',#1261);
#1279=SHAPE_DEFINITION_REPRESENTATION(#1278,#1277);
#1280=SHAPE_ASPECT('','v-shape boundary occurrence',#1237,.T.);
#1281=SHAPE_DEFINING_RELATIONSHIP('vee                    boundary','profile
usage',#1261,#1280);
#1282=CARTESIAN_POINT('',(0.0,-3.235562E-32,3.940000000000000));
#1283=DIRECTION('',(0.0,0.0,1.0));
#1284=DIRECTION('',(0.0,-1.0,0.0));
#1285=AXIS2_PLACEMENT_3D('',#1282,#1283,#1284);
#1286=SHAPE_ASPECT('','',#43,.F.);
#1287=PROPERTY_DEFINITION('','',#1286);
#1288=SHAPE_REPRESENTATION('',(#1285),#34);
#1289=SHAPE_DEFINITION_REPRESENTATION(#1287,#1288);
#1290=DIMENSIONAL_LOCATION('','',#1286,#1032);
#1291=PRECISION_QUALIFIER(4);
#1297=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(3.690000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1291
))REPRESENTATION_ITEM(''));
#1298=SHAPE_DIMENSION_REPRESENTATION('',(#1297),#34);
```

```
#1299=PRECISION_QUALIFIER(4);
#1300=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.010000000000000),#15);
#1301=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.010000000000000),#15);
#1302=MEASURE_QUALIFICATION('','',#1300,(#1299));
#1303=MEASURE_QUALIFICATION('','',#1301,(#1299));
#1304=TOLERANCE_VALUE(#1301,#1300);
#1305=PLUS_MINUS_TOLERANCE(#1304,#1290);
#1306=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#1290,#1298);
#1313=(CHARACTERIZED_OBJECT('','outer
diameter')FEATURE_DEFINITION()INSTANCED_FEATURE()OUTER_ROUND()SHAPE_ASPECT(''
,'outer diameter',#43,.T.));
#1314=PROPERTY_DEFINITION('','',#1313);
#1315=PRODUCT_DEFINITION_SHAPE('','',#1313);
#1316=CARTESIAN_POINT('',(0.0,-3.235562E-32,3.799999999999999));
#1317=DIRECTION('',(0.0,0.0,1.0));
#1318=DIRECTION('',(0.0,-1.0,0.0));
#1319=AXIS2_PLACEMENT_3D('orientation',#1316,#1317,#1318);
#1320=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#1321=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#1322=PRECISION_QUALIFIER(4);
#1328=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.372000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1320
,#1321,#1322))REPRESENTATION_ITEM('diameter'));
#1329=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#1330=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#1331=PRECISION_QUALIFIER(4);
#1337=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.140000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1329
,#1330,#1331))REPRESENTATION_ITEM('length'));
#1338=SHAPE_REPRESENTATION('',(#907,#924),#34);
#1339=SHAPE_DEFINITION_REPRESENTATION(#1314,#1338);
#1340=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1319,#1328,#1337),#34);
#1341=SHAPE_DEFINITION_REPRESENTATION(#1314,#1340);
#1342=FEATURE_COMPONENT_DEFINITION('','');
#1343=PRODUCT_DEFINITION_SHAPE('','',#1342);
#1344=TAPER('','diameter taper',#1343,.F.);
#1345=STANDARD_UNCERTAINTY('upper limit','plus',0.030000000000000);
#1346=STANDARD_UNCERTAINTY('lower limit','minus',-0.030000000000000);
#1347=PRECISION_QUALIFIER(4);
#1353=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.250000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1345
,#1346,#1347))REPRESENTATION_ITEM('final diameter'));
#1354=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1353),#34);
#1355=PROPERTY_DEFINITION('','',#1344);
#1356=SHAPE_DEFINITION_REPRESENTATION(#1355,#1354);
#1357=SHAPE_ASPECT('','reduced size occurrence',#1315,.T.);
#1358=FEATURE_COMPONENT_RELATIONSHIP('reduced                     size','taper
usage',#1344,#1357);

/*  Round_hole  */

#1365=(CHARACTERIZED_OBJECT('','')
       FEATURE_DEFINITION()
       INSTANCED_FEATURE()
       ROUND_HOLE()
       SHAPE_ASPECT('','',#43,.T.));
```



```
#1366=PROPERTY_DEFINITION('','',#1365);
#1367=PRODUCT_DEFINITION_SHAPE('','',#1365);
#1368=CARTESIAN_POINT('',(0.186000000000000,1.138922E-17,2.580000000000000));
#1369=DIRECTION('',(-1.0,0.0,0.0));
#1370=DIRECTION('',(0.0,-1.0,0.0));
#1371=AXIS2_PLACEMENT_3D('orientation',#1368,#1369,#1370);
#1372=SHAPE_REPRESENTATION('',(#950,#968),#34);
#1373=SHAPE_DEFINITION_REPRESENTATION(#1366,#1372);
#1374=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371),#34);
#1375=SHAPE_DEFINITION_REPRESENTATION(#1366,#1374);

/*  CIRCULAR_CLOSED_PROFILE FOR HOLE  */

#1376=FEATURE_COMPONENT_DEFINITION('','');
#1377=PRODUCT_DEFINITION_SHAPE('','',#1376);
#1378=CIRCULAR_CLOSED_PROFILE('circular profile','',#1377,.F.);
#1379=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#1380=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#1381=PRECISION_QUALIFIER(4);
#1387=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.125000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1379,#1380,#1381))
       REPRESENTATION_ITEM('diameter'));
#1388=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371,#1387),#34);
#1389=PROPERTY_DEFINITION('','',#1378);
#1390=SHAPE_DEFINITION_REPRESENTATION(#1389,#1388);

/*  LINEAR_PATH FOR HOLE  */

#1391=FEATURE_COMPONENT_DEFINITION('','');
#1392=PRODUCT_DEFINITION_SHAPE('','',#1391);
#1393=PATH_FEATURE_COMPONENT('linear path','linear',#1392,.F.);
#1398=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.372000000000000),#15)
       REPRESENTATION_ITEM('distance'));
#1399=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371,#1398),#34);
#1400=PROPERTY_DEFINITION('','',#1393);
#1401=SHAPE_DEFINITION_REPRESENTATION(#1400,#1399);
#1402=DIRECTION_SHAPE_REPRESENTATION('',(#1369),#34);
#1403=PROPERTY_DEFINITION_REPRESENTATION(#1400,#1402);
#1404=SHAPE_ASPECT('','diameter occurrence',#1367,.T.);
#1405=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#1378,#1404);
#1406=SHAPE_ASPECT('','hole depth occurrence',#1367,.T.);
#1407=SHAPE_DEFINING_RELATIONSHIP('hole   depth','path   feature   component
usage',#1393,#1406);

/* HOLE_BOTTOM FOR HOLE */

#1408=FEATURE_COMPONENT_DEFINITION('','');
#1409=PRODUCT_DEFINITION_SHAPE('','',#1408);
#1410=HOLE_BOTTOM('bottom condition','through',#1409,.F.);
#1411=SHAPE_ASPECT('','bottom condition occurrence',#1367,.T.);
#1412=FEATURE_COMPONENT_RELATIONSHIP('','hole bottom usage',#1410,#1411);
#1413=CARTESIAN_POINT('',(0.186000000000000,1.387779E-17,2.580000000000000));
```

```
#1414=DIRECTION('',(-1.0,0.0,0.0));
#1415=AXIS1_PLACEMENT('',#1413,#1414);
#1416=SHAPE_ASPECT('','',#43,.F.);
#1417=PROPERTY_DEFINITION('','',#1416);
#1418=SHAPE_REPRESENTATION('',(#1415),#34);
#1419=SHAPE_DEFINITION_REPRESENTATION(#1417,#1418);
#1420=DIMENSIONAL_LOCATION('','',#1032,#1416);
#1421=PRECISION_QUALIFIER(4);
#1427=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(2.330000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1421))
       REPRESENTATION_ITEM(''));
#1428=SHAPE_DIMENSION_REPRESENTATION('',(#1427),#34);
#1429=PRECISION_QUALIFIER(4);
#1430=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.030000000000000),#15);
#1431=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.030000000000000),#15);
#1432=MEASURE_QUALIFICATION('','',#1430,(#1429));
#1433=MEASURE_QUALIFICATION('','',#1431,(#1429));
#1434=TOLERANCE_VALUE(#1431,#1430);
#1435=PLUS_MINUS_TOLERANCE(#1434,#1420);
#1436=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#1420,#1428);

#1437=CHAMFER('','',#43,.T.);
#1438=PROPERTY_DEFINITION('','',#1437);
#1439=FACE_SHAPE_REPRESENTATION('chamfer face',(#866),#34);
#1440=PROPERTY_DEFINITION_REPRESENTATION(#1438,#1439);

#1441=CHAMFER_OFFSET('','first offset',#43,.F.);
#1442=CHAMFER_OFFSET('','second offset',#43,.F.);
#1443=FEATURE_COMPONENT_RELATIONSHIP('','',#1437,#1441);
#1444=FEATURE_COMPONENT_RELATIONSHIP('','',#1437,#1442);
#1445=PROPERTY_DEFINITION('','',#1441);
#1446=PROPERTY_DEFINITION('','',#1442);
#1447=STANDARD_UNCERTAINTY('upper limit','plus',0.030000000000000);
#1448=STANDARD_UNCERTAINTY('lower limit','minus',-0.030000000000000);
#1449=PRECISION_QUALIFIER(4);
#1455=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.130000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1447
,#1448,#1449))REPRESENTATION_ITEM('offset amount'));
#1456=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1455),#34);
#1457=SHAPE_DEFINITION_REPRESENTATION(#1445,#1456);
#1458=FACE_SHAPE_REPRESENTATION('first chamfer face',(#814),#34);
#1459=PROPERTY_DEFINITION_REPRESENTATION(#1445,#1458);
#1460=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#1461=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#1462=PRECISION_QUALIFIER(4);
#1468=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(15.
0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()QUALIFIED_REPRESENTATION_ITEM((#1460,#1
461,#1462))REPRESENTATION_ITEM('offset angle'));
#1469=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1468),#34);
#1470=SHAPE_DEFINITION_REPRESENTATION(#1446,#1469);
#1471=FACE_SHAPE_REPRESENTATION('second chamfer face',(#138),#34);
#1472=PROPERTY_DEFINITION_REPRESENTATION(#1446,#1471);

#1473=CHAMFER('','',#43,.T.);
```

```
#1474=PROPERTY_DEFINITION('','',#1473);
#1475=FACE_SHAPE_REPRESENTATION('chamfer face',(#882),#34);
#1476=PROPERTY_DEFINITION_REPRESENTATION(#1474,#1475);
#1477=CHAMFER_OFFSET('','first offset',#43,.F.);
#1478=CHAMFER_OFFSET('','second offset',#43,.F.);
#1479=FEATURE_COMPONENT_RELATIONSHIP('','',#1473,#1477);
#1480=FEATURE_COMPONENT_RELATIONSHIP('','',#1473,#1478);
#1481=PROPERTY_DEFINITION('','',#1477);
#1482=PROPERTY_DEFINITION('','',#1478);
#1483=STANDARD_UNCERTAINTY('upper limit','plus',0.030000000000000);
#1484=STANDARD_UNCERTAINTY('lower limit','minus',-0.030000000000000);
#1485=PRECISION_QUALIFIER(4);
#1491=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.130000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1483,#1484,#1485))
       REPRESENTATION_ITEM('offset amount'));
#1492=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1491),#34);
#1493=SHAPE_DEFINITION_REPRESENTATION(#1481,#1492);
#1494=FACE_SHAPE_REPRESENTATION('first chamfer face',(#842),#34);
#1495=PROPERTY_DEFINITION_REPRESENTATION(#1481,#1494);
#1496=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#1497=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#1498=PRECISION_QUALIFIER(4);
#1504=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(15.
0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()QUALIFIED_REPRESENTATION_ITEM((#1496,#1
497,#1498))REPRESENTATION_ITEM('offset angle'));
#1505=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1504),#34);
#1506=SHAPE_DEFINITION_REPRESENTATION(#1482,#1505);
#1507=FACE_SHAPE_REPRESENTATION('second chamfer face',(#138),#34);
#1508=PROPERTY_DEFINITION_REPRESENTATION(#1482,#1507);
#1509=SHAPE_ASPECT_RELATIONSHIP('','',#1473,#1437);
#1510=DESCRIPTIVE_REPRESENTATION_ITEM('note  text','BREAK  ALL  SHARP  EDGES
APPOX .015.');
#1511=REPRESENTATION('note text',(#1510),#34);
#1512=PROPERTY_DEFINITION('part property','part note',#42);
#1513=PROPERTY_DEFINITION('part property','part note',#42);
#1514=PROPERTY_DEFINITION_REPRESENTATION(#1513,#1511);
#1515=PROPERTY_DEFINITION_RELATIONSHIP('','',#1512,#1513);
#1516=DESCRIPTIVE_REPRESENTATION_ITEM('note text','HEAT TREAT PER MIL-H-6875
TO 150,000-160,000 PSI.');
#1517=REPRESENTATION('note text',(#1516),#34);
#1518=PROPERTY_DEFINITION('part property','part note',#42);
#1519=PROPERTY_DEFINITION('part property','part note',#42);
#1520=PROPERTY_DEFINITION_REPRESENTATION(#1519,#1517);
#1521=PROPERTY_DEFINITION_RELATIONSHIP('','',#1518,#1519);


#1538=SECURITY_CLASSIFICATION_LEVEL('unclassified');
#1539=SECURITY_CLASSIFICATION('classification','classify            product
version',#1538);
#1540=APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT(#1539,(#40));
#1560=SERIAL_NUMBERED_EFFECTIVITY('','','');
#1562=APPLIED_EFFECTIVITY_ASSIGNMENT(#1560,(#40));
#1564=LOT_EFFECTIVITY('','',#1565);
#1565=MEASURE_WITH_UNIT(LENGTH_MEASURE(10.0),#10);
```

```
#1566=APPLIED_EFFECTIVITY_ASSIGNMENT(#1564,(#40));

/*  ---------------  dm_execution_input  --------------------*/

#1600=DM_EXECUTION_INPUT('','',#2000);
#1610=ACTION_RELATIONSHIP('','',#1600,#1700);
#1620=APPLIED_ACTION_ASSIGNMENT(#1600,(#42));


/*  ---------------  dm_execution_result --------------------*/

#1700=DM_EXECUTION_RESULT('','',#2000);


/*  ---------------  dm_execution_result_measurment --------------------*/

#1740=DM_EXECUTION_RESULT_MEASUREMENT('TRUE','',#1700);
#1742=ACTION_PROPERTY_REPRESENTATION('','',#1740,#1744);

/*  ---------------  dm_data_analysis_software --------------------*/

#1744=REPRESENTATION('dm data acquisition software',(#1745,#1746,#1747),#34);
#1745=DESCRIPTIVE_REPRESENTATION_ITEM('application          name','calculation
stuff');
#1746=DESCRIPTIVE_REPRESENTATION_ITEM('application      versaion','calculation
stuff');
#1747=DESCRIPTIVE_REPRESENTATION_ITEM('vendor name','calculation stuff');

#1748=ACTION_PROPERTY_REPRESENTATION('','',#1740,#17490);
#1790=ACTION_PROPERTY_REPRESENTATION('','',#1740,#1791);

/*  ---------------  dmf_result_parameter  --------------------*/

#1791=DM_RESULT_PARAMETER('point',(#1794),#34);
#1792=REPRESENTATION_RELATIONSHIP('','',#1791,#14000);

#1794=DESCRIPTIVE_REPRESENTATION_ITEM('calculation        method','calculation
stuff');

#1795=ACTION_PROPERTY_REPRESENTATION('','',#1800,#1791);

#1796=PROPERTY_DEFINITION_REPRESENTATION(#1159,#1791);
#1797=NAME_ATTRIBUTE('specification',#1796);

#1798=(CYLINDRICITY_TOLERANCE()
       GEOMETRIC_TOLERANCE('','',#1799,#1157)
       MODIFIED_GEOMETRIC_TOLERANCE(.LEAST_MATERIAL_CONDITION.));

#1799=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#15);

/*  ---------------  dm_parameter_analysis_dml  --------------------*/

#1800=DM_PARAMETER_ANALYSIS_DML('','',#1801);
#1801=ACTION_METHOD('','','','');
#1802=ACTION_PROPERTY_REPRESENTATION('','',#1800,#1900);
#1804=ACTION_PROPERTY_REPRESENTATION('','',#1800,#1920);
```



```
#1805=ACTION_PROPERTY_REPRESENTATION('','',#1800,#1940);

/*  ---------------  dm_analysis_dofs_dml  --------------------*/

#1900=DM_ANALYSIS_DOFS_DML('',(#1904,#1905,#1906,#1907,#1908,#1909),#34);


#1904=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1914))
       REPRESENTATION_ITEM('transx upper limit'));

#1905=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1915))
       REPRESENTATION_ITEM('transy upper limit'));

#1906=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1916))
       REPRESENTATION_ITEM('transz upper limit'));

#1907=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1916))
       REPRESENTATION_ITEM('rotx lower limit'));

#1908=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1915))
       REPRESENTATION_ITEM('roty lower limit'));

#1909=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#1914))
       REPRESENTATION_ITEM('rotz lower limit'));

/*  ---------------  Dof_attribute_dml  --------------------*/

#1914=TYPE_QUALIFIER('degree of freedom limited');
#1915=TYPE_QUALIFIER('degree of freedom not allowed');
#1916=TYPE_QUALIFIER('degree of freedom allowed');

/*  ---------------  dm_tolerance_analysis_mode_dml  --------------------*/

#1920=DM_TOLERANCE_ANALYSIS_MODE_DML('',(#1921,#1922,#1923,#1924,#1925,#1926)
,#34);

#1921=DESCRIPTIVE_REPRESENTATION_ITEM('feature  tolerance  mode  method','least
square 3d');
```

```
#1922=DESCRIPTIVE_REPRESENTATION_ITEM('feature tolerance mode option','high
point');
#1923=DESCRIPTIVE_REPRESENTATION_ITEM('feature tolerance mode
setting','inner');

/* --------------- dm_tolerance_analysis_mode_default_dml ----------------
----*/

#1924=DESCRIPTIVE_REPRESENTATION_ITEM('feature tolerance mode method
default','standard');
#1925=DESCRIPTIVE_REPRESENTATION_ITEM('feature tolerance mode option
default','best fit');
#1926=DESCRIPTIVE_REPRESENTATION_ITEM('feature tolerance mode setting
default','standard');

/* --------------- dm_feature_analysis_mode_dml --------------------*/

#1940=DM_FEATURE_ANALYSIS_MODE_DML('',(#1941,#1942,#1943,#1944,#1945,#1946,#1
947,

#1948,#1949,#1950,#1951,#1952,#1953,#1954),#34);

#1941=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode axis','cross
section centers');
#1942=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode axis
extrapolate','unknown');
#1943=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode aelpr','minimum
maximum');
#1944=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode ccpst','maximum
size inscribed');
#1945=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode
curve','bspline');
#1946=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode limits','two
point');
#1947=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode
surface','bezier');


/* --------------- dm_feature_analysis_mode_default_dml ------------------
--*/

#1948=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode axis
default','least square');
#1949=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode axis extrapolate
default','TRUE');
#1950=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode aelpr
default','least square');
#1951=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode ccpst
default','least square');
#1952=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode curve
default','least square');
#1953=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode limits
default','functional');
#1954=DESCRIPTIVE_REPRESENTATION_ITEM('feature analysis mode surface
default','least square');
```



```
/*  ---------------  dm_program_run  --------------------*/

#2000=DM_PROGRAM_RUN('','','','');

#2630=ACTION_PROPERTY_REPRESENTATION('','',#2640,#2642);

#2640=ACTION_PROPERTY('','',#2000);
#2642=REPRESENTATION('measurement temperature',(#2692,#2694),#34);
#2684=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#2685=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#2686=PRECISION_QUALIFIER(4);
#2692=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#2684,#2685,#2686))
       REPRESENTATION_ITEM('measurement temperature'));
#2694=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#2684,#2685,#2686))
       REPRESENTATION_ITEM('measurement humidity'));

#2695=APPLIED_DATE_ASSIGNMENT(#2800, #2802, (#2000));
#2696=APPLIED_DATE_ASSIGNMENT(#2813, #2815, (#2000));
#2697=APPLIED_DATE_ASSIGNMENT(#2826, #2828, (#2000));
#2698=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#2986,#2989,(#2900));

 /*  ---------------  Run_administrator  --------------------*/

#2710=ORGANIZATION($,'RPTS','equipment company name');
#2711=ORGANIZATION_ROLE('run administrator');
#2712=APPLIED_ORGANIZATION_ASSIGNMENT(#2710,#2711,(#2000));

/*  ---------------  Measurement_location  --------------------*/

#2720=MEASUREMENT_LOCATION('location  of  measurement  used','machine  used  for
measurement',#2000);

/*  ---------------  Calendar_date  --------------------*/

#2800=CALENDAR_DATE(2003, 27, 6);
#2802=DATE_ROLE('run end');

#2813=CALENDAR_DATE(2003, 27, 6);
#2815=DATE_ROLE('run start');

#2826=CALENDAR_DATE(2003, 27, 6);
#2828=DATE_ROLE('run date');

/*  ---------------  Date_time  --------------------*/

#2903=DATE_AND_TIME(#2800,#2904);

/*  ---------------  Time_offset  --------------------*/

#2904=LOCAL_TIME(12,05,12.0,#2905);
```

```
/*  ---------------  Time_offset  --------------------*/

#2905=COORDINATED_UNIVERSAL_TIME_OFFSET(0,0,.BEHIND.);

/*  ---------------  dm_program_identification  --------------------*/

#2900=DM_PROGRAM_IDENTIFICATION('program   id   stuff','version   number   good
stuff',#2910,(#2000));
#2910=RESOURCE_REQUIREMENT_TYPE('','');

#2912=RESOURCE_PROPERTY('angular units dml','calculation stuff',#2900);
#2914=RESOURCE_PROPERTY('linear units dml','calculation stuff',#2900);
#2916=RESOURCE_PROPERTY('tolerance standard dml','calculation stuff',#2900);

/*  ---------------  Person_in_organization  --------------------*/

#2986=PERSON_AND_ORGANIZATION(#2988,#2990);
#2988=PERSON('1','Slovensky','Len',('W'),('Dr'),(''));
#2989=PERSON_AND_ORGANIZATION_ROLE('program custodian');

/*  ---------------  Organization  --------------------*/

#2990=ORGANIZATION('NGIT','Northrop Grumman','');

/*  ---------------  Address  --------------------*/

#2991=ORGANIZATIONAL_ADDRESS('','1398','Gumbert','','Amelia','Ohio',
   '45102','USA','','513-383-8311','slovensky@scra.org','',(#2990),'');

/*  ---------------  dmf_plane  --------------------*/

#4000=(CHARACTERIZED_OBJECT('','')
       DMF_PLANE()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4002=PRODUCT_DEFINITION_SHAPE('','',#4000);

#4004=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4013),#34);
#4006=PROPERTY_DEFINITION('','',#4000);
#4008=PROPERTY_DEFINITION_REPRESENTATION(#4006,#4004);

#4010=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#4011=DIRECTION('',(0.0,1.0,0.0));
#4012=DIRECTION('',(1.0,0.0,0.0));
#4013=AXIS2_PLACEMENT_3D('orientation',#4010,#4011,#4012);

#4077=SHAPE_REPRESENTATION('boundary',(#4079),#34);
#4078=SHAPE_DEFINITION_REPRESENTATION(#4006,#4077);
#4079=CARTESIAN_POINT(' ',(0.0,0.0,4.180000000000000));

#4080=PROPERTY_DEFINITION_REPRESENTATION(#4006,#14081);
#4081=NAME_ATTRIBUTE('direction vector',#4080);
```



```
#4090=PROPERTY_DEFINITION_REPRESENTATION(#4006,#15091);
#4091=NAME_ATTRIBUTE('point on plane',#4090);

/*  ---------------  dmf_point  --------------------*/

#4100=(CHARACTERIZED_OBJECT('','')
       DMF_POINT()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4102=PRODUCT_DEFINITION_SHAPE('','',#4100);

#4104=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4113),#34);
#4106=PROPERTY_DEFINITION('','',#4100);
#4108=PROPERTY_DEFINITION_REPRESENTATION(#4106,#4104);

#4110=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#4111=DIRECTION('',(0.0,1.0,0.0));
#4112=DIRECTION('',(1.0,0.0,0.0));
#4113=AXIS2_PLACEMENT_3D('orientation',#4110,#4111,#4112);

#4177=SHAPE_REPRESENTATION('boundary',(#4179),#34);
#4178=SHAPE_DEFINITION_REPRESENTATION(#4106,#4177);
#4179=CARTESIAN_POINT(' ',(0.0,0.0,4.180000000000000));

#4180=PROPERTY_DEFINITION_REPRESENTATION(#4106,#14181);
#4161=NAME_ATTRIBUTE('direction vector',#4180);

#4190=PROPERTY_DEFINITION_REPRESENTATION(#4106,#15191);
#4191=NAME_ATTRIBUTE('point',#4190);

/*  ---------------  dmf_line_unbounded  --------------------*/

#4300=(CHARACTERIZED_OBJECT('','')
       DMF_LINE_UNBOUNDED()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4302=PRODUCT_DEFINITION_SHAPE('','',#4300);

#4304=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4313),#34);
#4306=PROPERTY_DEFINITION('','',#4300);
#4308=PROPERTY_DEFINITION_REPRESENTATION(#4306,#4304);

#4310=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#4311=DIRECTION('',(0.0,1.0,0.0));
#4312=DIRECTION('',(1.0,0.0,0.0));
#4313=AXIS2_PLACEMENT_3D('orientation',#4310,#4311,#4312);

#4377=SHAPE_REPRESENTATION('boundary',(#4379),#34);
#4378=SHAPE_DEFINITION_REPRESENTATION(#4306,#4377);
```



```
#4379=CARTESIAN_POINT(' ',(0.0,0.0,4.180000000000000));

#4380=PROPERTY_DEFINITION_REPRESENTATION(#4306,#14181);
#4381=NAME_ATTRIBUTE('surface approach vector',#4380);

#4382=PROPERTY_DEFINITION_REPRESENTATION(#4306,#14201);
#4383=NAME_ATTRIBUTE('direction vector',#4382);

#4390=PROPERTY_DEFINITION_REPRESENTATION(#4306,#15200);
#4391=NAME_ATTRIBUTE('point on line',#4390);

/*  ---------------  dmf_geometric_curve  --------------------*/

#4400=(CHARACTERIZED_OBJECT('','')
       DMF_GEOMETRIC_CURVE()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4402=PRODUCT_DEFINITION_SHAPE('','',#4400);

#4404=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4413),#34);
#4406=PROPERTY_DEFINITION('','',#4400);
#4408=PROPERTY_DEFINITION_REPRESENTATION(#4406,#4404);

#4410=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#4411=DIRECTION('',(0.0,1.0,0.0));
#4412=DIRECTION('',(1.0,0.0,0.0));
#4413=AXIS2_PLACEMENT_3D('orientation',#4410,#4411,#4412);

#4477=SHAPE_REPRESENTATION('boundary',(#4479),#34);
#4478=SHAPE_DEFINITION_REPRESENTATION(#4406,#4477);
#4479=CARTESIAN_POINT(' ',(0.0,0.0,4.180000000000000));

#4482=PROPERTY_DEFINITION_REPRESENTATION(#4406,#14201);
#4483=NAME_ATTRIBUTE('curve plane direction',#4482);

#4490=PROPERTY_DEFINITION_REPRESENTATION(#4406,#15200);
#4491=NAME_ATTRIBUTE('point on plane of curve',#4490);

#4492=SHAPE_DEFINITION_REPRESENTATION(#4406,#4493);
#4493=SHAPE_REPRESENTATION('',(#379,#380,#381,#382,#383,#384,#385,#386),#34);
#4495=NAME_ATTRIBUTE('data point',#4492);

/*  ---------------  dmf_geometric_surface  --------------------*/

#4500=(CHARACTERIZED_OBJECT('','')
       DMF_GEOMETRIC_SURFACE()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4502=PRODUCT_DEFINITION_SHAPE('','',#4500);

#4504=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4513),#34);
```



```
#4506=PROPERTY_DEFINITION('','',#4500);
#4508=PROPERTY_DEFINITION_REPRESENTATION(#4506,#4504);

#4510=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#4511=DIRECTION('',(0.0,1.0,0.0));
#4512=DIRECTION('',(1.0,0.0,0.0));
#4513=AXIS2_PLACEMENT_3D('orientation',#4510,#4511,#4512);

#4577=SHAPE_REPRESENTATION('boundary',(#4579),#34);
#4578=SHAPE_DEFINITION_REPRESENTATION(#4506,#4577);
#4579=CARTESIAN_POINT(' ',(0.0,0.0,4.180000000000000));

#4582=PROPERTY_DEFINITION_REPRESENTATION(#4506,#14201);
#4583=NAME_ATTRIBUTE('local surface normal',#4582);

#4590=PROPERTY_DEFINITION_REPRESENTATION(#4506,#15200);
#4591=NAME_ATTRIBUTE('point on surface',#4590);

#4592=SHAPE_DEFINITION_REPRESENTATION(#4506,#4593);
#4593=SHAPE_REPRESENTATION('',(#379,#380,#381,#382,#383,#384,#385,#386),#34);
#4595=NAME_ATTRIBUTE('data point',#4592);

/*  ---------------  dmf_edge_point  --------------------*/

#4600=(CHARACTERIZED_OBJECT('','')
       DMF_EDGE_POINT()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4602=PRODUCT_DEFINITION_SHAPE('','',#4600);

#4604=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4613,#4616),#34);
#4606=PROPERTY_DEFINITION('','',#4600);
#4608=PROPERTY_DEFINITION_REPRESENTATION(#4606,#4604);

#4610=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#4611=DIRECTION('',(0.0,1.0,0.0));
#4612=DIRECTION('',(1.0,0.0,0.0));
#4613=AXIS2_PLACEMENT_3D('orientation',#4610,#4611,#4612);

#4616=DESCRIPTIVE_REPRESENTATION_ITEM('point type dml','t edge');

#4677=SHAPE_REPRESENTATION('boundary',(#4679),#34);
#4678=SHAPE_DEFINITION_REPRESENTATION(#4606,#4677);
#4679=CARTESIAN_POINT(' ',(0.0,0.0,4.180000000000000));

#4680=PROPERTY_DEFINITION_REPRESENTATION(#4606,#14201);
#4681=NAME_ATTRIBUTE('edge normal vector',#4680);

#4682=PROPERTY_DEFINITION_REPRESENTATION(#4606,#14201);
#4683=NAME_ATTRIBUTE('surface normal vector',#4682);

#4690=PROPERTY_DEFINITION_REPRESENTATION(#4606,#15200);
```



```
#4691=NAME_ATTRIBUTE('location point',#4690);

/*  ---------------  dmf_generic_feature  --------------------*/

#4700=(CHARACTERIZED_OBJECT('','')
       DMF_GENERIC_FEATURE()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4702=PRODUCT_DEFINITION_SHAPE('','',#4700);

#4704=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4713,#4716),#34);
#4706=PROPERTY_DEFINITION('','',#4700);
#4708=PROPERTY_DEFINITION_REPRESENTATION(#4706,#4704);

#4710=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#4711=DIRECTION('',(0.0,1.0,0.0));
#4712=DIRECTION('',(1.0,0.0,0.0));
#4713=AXIS2_PLACEMENT_3D('orientation',#4710,#4711,#4712);

#4716=DESCRIPTIVE_REPRESENTATION_ITEM('description','calculation stuff');

#4777=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#4778=SHAPE_DEFINITION_REPRESENTATION(#4706,#4777);

#4780=PROPERTY_DEFINITION_REPRESENTATION(#4706,#14201);
#4781=NAME_ATTRIBUTE('vector from center of object',#4780);

#4782=PROPERTY_DEFINITION_REPRESENTATION(#4706,#14201);
#4783=NAME_ATTRIBUTE('secondary vector',#4782);

#4790=PROPERTY_DEFINITION_REPRESENTATION(#4706,#15200);
#4791=NAME_ATTRIBUTE('center point',#4790);

#4792=PROPERTY_DEFINITION_REPRESENTATION(#4706,#16000);
#4793=NAME_ATTRIBUTE('parameter',#4792);

/*  ---------------  dmf_line_bounded  --------------------*/

#4800=(CHARACTERIZED_OBJECT('','')
       DMF_LINE_BOUNDED()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4802=PRODUCT_DEFINITION_SHAPE('','',#4800);

#4804=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4813),#34);
#4806=PROPERTY_DEFINITION('','',#4800);
#4808=PROPERTY_DEFINITION_REPRESENTATION(#4806,#4804);

#4810=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
```



```
#4811=DIRECTION('',(0.0,1.0,0.0));
#4812=DIRECTION('',(1.0,0.0,0.0));
#4813=AXIS2_PLACEMENT_3D('orientation',#4810,#4811,#4812);

#4877=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#4878=SHAPE_DEFINITION_REPRESENTATION(#4806,#4877);

#4880=PROPERTY_DEFINITION_REPRESENTATION(#4806,#14201);
#4881=NAME_ATTRIBUTE('vector dml',#4880);

#4882=PROPERTY_DEFINITION_REPRESENTATION(#4806,#14201);
#4883=NAME_ATTRIBUTE('surface approach vector',#4882);

#4890=PROPERTY_DEFINITION_REPRESENTATION(#4806,#15200);
#4891=NAME_ATTRIBUTE('second end point',#4890);

#4892=PROPERTY_DEFINITION_REPRESENTATION(#4806,#15200);
#4893=NAME_ATTRIBUTE('first end point',#4892);

#4894=PROPERTY_DEFINITION_REPRESENTATION(#4806,#16000);
#4895=NAME_ATTRIBUTE('length dml',#4894);

/*  ---------------  dmf_sphere  --------------------*/

#4900=(CHARACTERIZED_OBJECT('','')
       DMF_SPHERE()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#4902=PRODUCT_DEFINITION_SHAPE('','',#4900);

#4904=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#4913,#4916),#34);
#4906=PROPERTY_DEFINITION('','',#4900);
#4908=PROPERTY_DEFINITION_REPRESENTATION(#4906,#4904);

#4910=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#4911=DIRECTION('',(0.0,1.0,0.0));
#4912=DIRECTION('',(1.0,0.0,0.0));
#4913=AXIS2_PLACEMENT_3D('orientation',#4910,#4911,#4912);


#4916=DESCRIPTIVE_REPRESENTATION_ITEM('inner or outer','inner');

#4927=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#4928=SHAPE_DEFINITION_REPRESENTATION(#4906,#4927);

#4930=PROPERTY_DEFINITION_REPRESENTATION(#4906,#14201);
#4931=NAME_ATTRIBUTE('north pole vector',#4930);

#4932=PROPERTY_DEFINITION_REPRESENTATION(#4906,#14201);
#4933=NAME_ATTRIBUTE('prime meridian vector',#4932);
```

```
#4940=PROPERTY_DEFINITION_REPRESENTATION(#4906,#15200);
#4941=NAME_ATTRIBUTE('center point',#4940);

#4944=PROPERTY_DEFINITION_REPRESENTATION(#4906,#16000);
#4945=NAME_ATTRIBUTE('latitude start angle',#4944);

#4946=PROPERTY_DEFINITION_REPRESENTATION(#4906,#16000);
#4947=NAME_ATTRIBUTE('latitude stop angle',#4946);

#4948=PROPERTY_DEFINITION_REPRESENTATION(#4906,#16000);
#4949=NAME_ATTRIBUTE('longitude start angle',#4948);

#4950=PROPERTY_DEFINITION_REPRESENTATION(#4906,#16000);
#4951=NAME_ATTRIBUTE('longitude stop angle',#4950);

#4952=PROPERTY_DEFINITION_REPRESENTATION(#4906,#16000);
#4953=NAME_ATTRIBUTE('diameter',#4952);

/*  ---------------  dmf_ellipse  --------------------*/

#5000=(CHARACTERIZED_OBJECT('','')
       DMF_ELLIPSE()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5002=PRODUCT_DEFINITION_SHAPE('','',#5000);

#5004=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5013,#5016),#34);
#5006=PROPERTY_DEFINITION('','',#5000);
#5008=PROPERTY_DEFINITION_REPRESENTATION(#5006,#5004);

#5010=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5011=DIRECTION('',(0.0,1.0,0.0));
#5012=DIRECTION('',(1.0,0.0,0.0));
#5013=AXIS2_PLACEMENT_3D('orientation',#5010,#5011,#5012);

#5016=DESCRIPTIVE_REPRESENTATION_ITEM('inner or outer','inner');

#5027=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5028=SHAPE_DEFINITION_REPRESENTATION(#5006,#5027);

#5030=PROPERTY_DEFINITION_REPRESENTATION(#5006,#14201);
#5031=NAME_ATTRIBUTE('normal direction',#5030);

#5032=PROPERTY_DEFINITION_REPRESENTATION(#5006,#14201);
#5033=NAME_ATTRIBUTE('major axis vector',#5032);

#5034=PROPERTY_DEFINITION_REPRESENTATION(#5006,#15200);
#5035=NAME_ATTRIBUTE('center point',#5034);

#5036=PROPERTY_DEFINITION_REPRESENTATION(#5006,#15200);
#5037=NAME_ATTRIBUTE('focus point one dml',#5036);
```



```
#5038=PROPERTY_DEFINITION_REPRESENTATION(#5006,#15200);
#5039=NAME_ATTRIBUTE('focus point two dml',#5038);

#5040=PROPERTY_DEFINITION_REPRESENTATION(#5006,#16000);
#5041=NAME_ATTRIBUTE('major diameter',#5040);

#5044=PROPERTY_DEFINITION_REPRESENTATION(#5006,#16000);
#5045=NAME_ATTRIBUTE('minor diameter',#5044);


/*  ---------------  dmf_plane_symmetric  --------------------*/

#5100=(CHARACTERIZED_OBJECT('','')
       DMF_PLANE_SYMMETRIC()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5102=PRODUCT_DEFINITION_SHAPE('','',#5100);

#5104=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5113,#5116),#34);
#5106=PROPERTY_DEFINITION('','',#5100);
#5108=PROPERTY_DEFINITION_REPRESENTATION(#5106,#5104);

#5110=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5111=DIRECTION('',(0.0,1.0,0.0));
#5112=DIRECTION('',(1.0,0.0,0.0));
#5113=AXIS2_PLACEMENT_3D('orientation',#5110,#5111,#5112);

#5116=DESCRIPTIVE_REPRESENTATION_ITEM('inner or outer','inner');

#5127=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5128=SHAPE_DEFINITION_REPRESENTATION(#5106,#5127);


#5130=PROPERTY_DEFINITION_REPRESENTATION(#5106,#14201);
#5131=NAME_ATTRIBUTE('direction vector side one',#5130);

#5132=PROPERTY_DEFINITION_REPRESENTATION(#5106,#14201);
#5133=NAME_ATTRIBUTE('direction vector side two',#5132);

#5134=PROPERTY_DEFINITION_REPRESENTATION(#5106,#15200);
#5135=NAME_ATTRIBUTE('point on one side',#5134);

#5136=PROPERTY_DEFINITION_REPRESENTATION(#5106,#15200);
#5137=NAME_ATTRIBUTE('point on side two',#5136);

#5138=PROPERTY_DEFINITION_REPRESENTATION(#5106,#15200);
#5139=NAME_ATTRIBUTE('point on mid plane',#5138);

#5140=PROPERTY_DEFINITION_REPRESENTATION(#5106,#16000);
#5141=NAME_ATTRIBUTE('width at mid point',#5140);

/*  ---------------  dmf_arc  --------------------*/
```

```
#5200=(CHARACTERIZED_OBJECT('','')
       DMF_ARC()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5202=PRODUCT_DEFINITION_SHAPE('','',#5200);

#5204=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5213,#5216),#34);
#5206=PROPERTY_DEFINITION('','',#5200);
#5208=PROPERTY_DEFINITION_REPRESENTATION(#5206,#5204);

#5210=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5211=DIRECTION('',(0.0,1.0,0.0));
#5212=DIRECTION('',(1.0,0.0,0.0));
#5213=AXIS2_PLACEMENT_3D('orientation',#5210,#5211,#5212);

#5216=DESCRIPTIVE_REPRESENTATION_ITEM('inner or outer','inner');

#5227=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5228=SHAPE_DEFINITION_REPRESENTATION(#5206,#5227);

#5230=PROPERTY_DEFINITION_REPRESENTATION(#5206,#14201);
#5231=NAME_ATTRIBUTE('start vector',#5230);

#5232=PROPERTY_DEFINITION_REPRESENTATION(#5206,#14201);
#5233=NAME_ATTRIBUTE('end vector',#5232);

#5234=PROPERTY_DEFINITION_REPRESENTATION(#5206,#14201);
#5235=NAME_ATTRIBUTE('axis direction vector',#5234);

#5236=PROPERTY_DEFINITION_REPRESENTATION(#5206,#15200);
#5237=NAME_ATTRIBUTE('center point',#5236);

#5240=PROPERTY_DEFINITION_REPRESENTATION(#5206,#16000);
#5241=NAME_ATTRIBUTE('radius',#5240);

/*  ---------------  dmf_circle  --------------------*/

#5300=(CHARACTERIZED_OBJECT('','')
       DMF_CIRCLE()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5302=PRODUCT_DEFINITION_SHAPE('','',#5300);

#5304=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5313,#5316),#34);
#5306=PROPERTY_DEFINITION('','',#5300);
#5308=PROPERTY_DEFINITION_REPRESENTATION(#5306,#5304);

#5310=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
```



```
#5311=DIRECTION('',(0.0,1.0,0.0));
#5312=DIRECTION('',(1.0,0.0,0.0));
#5313=AXIS2_PLACEMENT_3D('orientation',#5310,#5311,#5312);

#5316=DESCRIPTIVE_REPRESENTATION_ITEM('inner outer','inner');

#5327=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5328=SHAPE_DEFINITION_REPRESENTATION(#5306,#5327);

#5334=PROPERTY_DEFINITION_REPRESENTATION(#5306,#14201);
#5335=NAME_ATTRIBUTE('axis direction vector',#5334);

#5336=PROPERTY_DEFINITION_REPRESENTATION(#5306,#15200);
#5337=NAME_ATTRIBUTE('center point',#5336);

#5340=PROPERTY_DEFINITION_REPRESENTATION(#5306,#16000);
#5341=NAME_ATTRIBUTE('diameter',#5340);

/*  ---------------  dmf_torus  --------------------*/

#5400=(CHARACTERIZED_OBJECT('','')
       DMF_TORUS()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5402=PRODUCT_DEFINITION_SHAPE('','',#5400);

#5404=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5413,#5416),#34);
#5406=PROPERTY_DEFINITION('','',#5400);
#5408=PROPERTY_DEFINITION_REPRESENTATION(#5406,#5404);

#5410=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5411=DIRECTION('',(0.0,1.0,0.0));
#5412=DIRECTION('',(1.0,0.0,0.0));
#5413=AXIS2_PLACEMENT_3D('orientation',#5410,#5411,#5412);

#5416=DESCRIPTIVE_REPRESENTATION_ITEM('inner outer','inner');

#5427=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5428=SHAPE_DEFINITION_REPRESENTATION(#5406,#5427);

#5434=PROPERTY_DEFINITION_REPRESENTATION(#5406,#14201);
#5435=NAME_ATTRIBUTE('vector of plane',#5434);

#5436=PROPERTY_DEFINITION_REPRESENTATION(#5406,#15200);
#5437=NAME_ATTRIBUTE('center point',#5436);

#5440=PROPERTY_DEFINITION_REPRESENTATION(#5406,#16000);
#5441=NAME_ATTRIBUTE('major diameter',#5440);

#5442=PROPERTY_DEFINITION_REPRESENTATION(#5406,#16000);
#5443=NAME_ATTRIBUTE('minor diameter',#5442);
```

```
/*  ---------------  dmf_cylinder --------------------*/

#5500=(CHARACTERIZED_OBJECT('','')
       DMF_CYLINDER()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5502=PRODUCT_DEFINITION_SHAPE('','',#5500);

#5504=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5513,#5516),#34);
#5506=PROPERTY_DEFINITION('','',#5500);
#5508=PROPERTY_DEFINITION_REPRESENTATION(#5506,#5504);

#5510=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5511=DIRECTION('',(0.0,1.0,0.0));
#5512=DIRECTION('',(1.0,0.0,0.0));
#5513=AXIS2_PLACEMENT_3D('orientation',#5510,#5511,#5512);

#5516=DESCRIPTIVE_REPRESENTATION_ITEM('inner outer','inner');

#5527=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5528=SHAPE_DEFINITION_REPRESENTATION(#5506,#5527);

#5524=PROPERTY_DEFINITION_REPRESENTATION(#5506,#14201);
#5525=NAME_ATTRIBUTE('axis direction vector',#5524);

#5532=PROPERTY_DEFINITION_REPRESENTATION(#5506,#15200);
#5533=NAME_ATTRIBUTE('center point',#5532);

#5534=PROPERTY_DEFINITION_REPRESENTATION(#5506,#15200);
#5535=NAME_ATTRIBUTE('start point',#5534);

#5536=PROPERTY_DEFINITION_REPRESENTATION(#5506,#15200);
#5537=NAME_ATTRIBUTE('end point',#5536);

#5540=PROPERTY_DEFINITION_REPRESENTATION(#5506,#16000);
#5541=NAME_ATTRIBUTE('diameter',#5540);

#5542=PROPERTY_DEFINITION_REPRESENTATION(#5506,#16000);
#5543=NAME_ATTRIBUTE('length dml',#5542);

/*  ---------------  dmf_cone --------------------*/

#5600=(CHARACTERIZED_OBJECT('','')
       DMF_CONE()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5602=PRODUCT_DEFINITION_SHAPE('','',#5600);

#5604=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5613,#5616),#34);
```



```
#5606=PROPERTY_DEFINITION('','',#5600);
#5608=PROPERTY_DEFINITION_REPRESENTATION(#5606,#5604);

#5610=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5611=DIRECTION('',(0.0,1.0,0.0));
#5612=DIRECTION('',(1.0,0.0,0.0));
#5613=AXIS2_PLACEMENT_3D('orientation',#5610,#5611,#5612);

#5616=DESCRIPTIVE_REPRESENTATION_ITEM('inner outer','inner');

#5617=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5618=SHAPE_DEFINITION_REPRESENTATION(#5606,#5617);

#5624=PROPERTY_DEFINITION_REPRESENTATION(#5606,#14201);
#5625=NAME_ATTRIBUTE('axis direction vector',#5624);

#5626=PROPERTY_DEFINITION_REPRESENTATION(#5606,#14201);
#5627=NAME_ATTRIBUTE('start vector',#5626);

#5628=PROPERTY_DEFINITION_REPRESENTATION(#5606,#14201);
#5629=NAME_ATTRIBUTE('end vector',#5628);

#5632=PROPERTY_DEFINITION_REPRESENTATION(#5606,#15200);
#5633=NAME_ATTRIBUTE('apex point',#5632);

#5634=PROPERTY_DEFINITION_REPRESENTATION(#5606,#15200);
#5635=NAME_ATTRIBUTE('start point',#5634);

#5636=PROPERTY_DEFINITION_REPRESENTATION(#5606,#15200);
#5637=NAME_ATTRIBUTE('end point',#5636);

#5640=PROPERTY_DEFINITION_REPRESENTATION(#5606,#16000);
#5641=NAME_ATTRIBUTE('diameter',#5640);

#5642=PROPERTY_DEFINITION_REPRESENTATION(#5606,#16000);
#5643=NAME_ATTRIBUTE('start length dml',#5642);

#5644=PROPERTY_DEFINITION_REPRESENTATION(#5606,#16000);
#5645=NAME_ATTRIBUTE('included angle',#5644);

#5646=PROPERTY_DEFINITION_REPRESENTATION(#5606,#16000);
#5647=NAME_ATTRIBUTE('end length dml',#5646);

/*  ---------------  dmf_plane_closed_parallel --------------------*/

#5700=(CHARACTERIZED_OBJECT('','')
       DMF_PLANE_CLOSED_PARALLEL()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5702=PRODUCT_DEFINITION_SHAPE('','',#5700);

#5704=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5713,#5716,#5717),#34);
```

```
#5706=PROPERTY_DEFINITION('','',#5700);
#5708=PROPERTY_DEFINITION_REPRESENTATION(#5706,#5704);

#5710=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5711=DIRECTION('',(0.0,1.0,0.0));
#5712=DIRECTION('',(1.0,0.0,0.0));
#5713=AXIS2_PLACEMENT_3D('orientation',#5710,#5711,#5712);

#5716=DESCRIPTIVE_REPRESENTATION_ITEM('inner or outer','inner');
#5717=DESCRIPTIVE_REPRESENTATION_ITEM('end kind','square');

#5720=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5721=SHAPE_DEFINITION_REPRESENTATION(#5706,#5720);

#5724=PROPERTY_DEFINITION_REPRESENTATION(#5706,#14201);
#5725=NAME_ATTRIBUTE('axis direction vector',#5724);

#5726=PROPERTY_DEFINITION_REPRESENTATION(#5706,#14201);
#5727=NAME_ATTRIBUTE('longitude vector',#5726);

#5728=PROPERTY_DEFINITION_REPRESENTATION(#5706,#14201);
#5729=NAME_ATTRIBUTE('normal dml',#5728);

#5732=PROPERTY_DEFINITION_REPRESENTATION(#5706,#15200);
#5733=NAME_ATTRIBUTE('center point',#5732);

#5740=PROPERTY_DEFINITION_REPRESENTATION(#5706,#16000);
#5741=NAME_ATTRIBUTE('feature length',#5740);

#5742=PROPERTY_DEFINITION_REPRESENTATION(#5706,#16000);
#5743=NAME_ATTRIBUTE('width',#5742);

#5744=PROPERTY_DEFINITION_REPRESENTATION(#5706,#16000);
#5745=NAME_ATTRIBUTE('height',#5744);

/*  ---------------  dmf_line_closed_parallel --------------------*/

#5800=(CHARACTERIZED_OBJECT('','')
       DMF_LINE_CLOSED_PARALLEL()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5802=PRODUCT_DEFINITION_SHAPE('','',#5800);

#5804=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5813,#5816,#5817),#34);
#5806=PROPERTY_DEFINITION('','',#5800);
#5808=PROPERTY_DEFINITION_REPRESENTATION(#5806,#5804);

#5810=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5811=DIRECTION('',(0.0,1.0,0.0));
#5812=DIRECTION('',(1.0,0.0,0.0));
#5813=AXIS2_PLACEMENT_3D('orientation',#5810,#5811,#5812);
```



```
#5816=DESCRIPTIVE_REPRESENTATION_ITEM('inner or outer','inner');
#5817=DESCRIPTIVE_REPRESENTATION_ITEM('end kind','square');

#5820=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5821=SHAPE_DEFINITION_REPRESENTATION(#5806,#5820);

#5824=PROPERTY_DEFINITION_REPRESENTATION(#5806,#14201);
#5825=NAME_ATTRIBUTE('axis direction vector',#5824);

#5826=PROPERTY_DEFINITION_REPRESENTATION(#5806,#14201);
#5827=NAME_ATTRIBUTE('longitude vector',#5826);

#5832=PROPERTY_DEFINITION_REPRESENTATION(#5806,#15200);
#5833=NAME_ATTRIBUTE('center point',#5832);

#5840=PROPERTY_DEFINITION_REPRESENTATION(#5806,#16000);
#5841=NAME_ATTRIBUTE('feature length',#5840);

#5842=PROPERTY_DEFINITION_REPRESENTATION(#5806,#16000);
#5843=NAME_ATTRIBUTE('width',#5842);

/*  ---------------  dmf_pattern --------------------*/

#5900=(CHARACTERIZED_OBJECT('','')
       DMF_PATTERN()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#5902=PRODUCT_DEFINITION_SHAPE('','',#5900);

#5904=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5913),#34);
#5906=PROPERTY_DEFINITION('','',#5900);
#5908=PROPERTY_DEFINITION_REPRESENTATION(#5906,#5904);

#5910=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#5911=DIRECTION('',(0.0,1.0,0.0));
#5912=DIRECTION('',(1.0,0.0,0.0));
#5913=AXIS2_PLACEMENT_3D('orientation',#5910,#5911,#5912);


#5920=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#5921=SHAPE_DEFINITION_REPRESENTATION(#5906,#5920);

#5926=PROPERTY_DEFINITION_REPRESENTATION(#5906,#14201);
#5927=NAME_ATTRIBUTE('direction vector',#5926);

#5932=PROPERTY_DEFINITION_REPRESENTATION(#5906,#15200);
#5933=NAME_ATTRIBUTE('point',#5932);

#5950=DM_FEATURE_RELATIONSHIP('','',#5900,#5700);
#5952=DM_FEATURE_RELATIONSHIP('','',#5900,#5800);
```

```
/*  ---------------  dmf_surface_of_revolution_dml --------------------*/

#6000=(CHARACTERIZED_OBJECT('','')
       DMF_SURFACE_OF_REVOLUTION_DML()
       DM_FEATURE_DEFINITION()
       DM_INSTANCED_FEATURE()
       SHAPE_ASPECT('','',#43,.T.));

#6002=PRODUCT_DEFINITION_SHAPE('','',#6000);

#6004=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6013),#34);
#6006=PROPERTY_DEFINITION('','',#6000);
#6008=PROPERTY_DEFINITION_REPRESENTATION(#6006,#6004);

#6010=CARTESIAN_POINT('',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#6011=DIRECTION('',(0.0,1.0,0.0));
#6012=DIRECTION('',(1.0,0.0,0.0));
#6013=AXIS2_PLACEMENT_3D('orientation',#6010,#6011,#6012);

#6020=SHAPE_REPRESENTATION('boundary',(#379,#380,#381,#382,#383,#384,#385,#38
6),#34);
#6021=SHAPE_DEFINITION_REPRESENTATION(#6006,#6020);

#6026=PROPERTY_DEFINITION_REPRESENTATION(#6006,#14201);
#6027=NAME_ATTRIBUTE('axis of revolution',#6026);

#6032=PROPERTY_DEFINITION_REPRESENTATION(#6006,#15200);
#6033=NAME_ATTRIBUTE('profile curve',#6032);

#6034=PROPERTY_DEFINITION_REPRESENTATION(#6006,#15200);
#6035=NAME_ATTRIBUTE('profile curve',#6034);


/*  ---------------  dm_parameter_value_limits  --------------------*/

#14000=DM_PARAMETER_VALUE_LIMITS('',(#4292,#4293,#4294),#34);

#4292=DESCRIPTIVE_REPRESENTATION_ITEM('limits method','calculation stuff');

#4293=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       REPRESENTATION_ITEM('calculated limits'));

#4294=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       REPRESENTATION_ITEM('calculated limits'));

/*  ---------------  dm_vector_parameter  --------------------*/

#14081=DM_VECTOR_PARAMETER('',(#49),#34);
#14083=REPRESENTATION_RELATIONSHIP('','',#14081,#14000);
```



```
#14181=DM_VECTOR_PARAMETER('',(#49),#34);
#14183=REPRESENTATION_RELATIONSHIP('','',#14181,#14000);

#14201=DM_VECTOR_PARAMETER('',(#49,#14206),#34);
#14203=REPRESENTATION_RELATIONSHIP('','',#14201,#14000);

#14204=REPRESENTATION_RELATIONSHIP('specification','',#14081,#1220);
#14206=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#14085,#14086,#14087,#14088))
       REPRESENTATION_ITEM('calculated value'));
#14085=STANDARD_UNCERTAINTY('value       uncertainty','i       uncertainty',-
0.010000000000000);
#14086=STANDARD_UNCERTAINTY('value       uncertainty','j       uncertainty',-
0.010000000000000);
#14087=STANDARD_UNCERTAINTY('value       uncertainty','k       uncertainty',-
0.010000000000000);
#14088=PRECISION_QUALIFIER(4);


/*  ---------------  dm_point_parameter  --------------------*/

#15091=DM_POINT_PARAMETER('',(#44,#15002),#34);

#15191=DM_POINT_PARAMETER('',(#44,#15003),#34);
#15192=REPRESENTATION_RELATIONSHIP('','',#15191,#14000);

#15200=DM_POINT_PARAMETER('',(#44),#34);

#15002=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#15085,#15086,#15088))
       REPRESENTATION_ITEM('calculated value'));

#15003=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
       QUALIFIED_REPRESENTATION_ITEM((#15085,#15086,#15087))
       REPRESENTATION_ITEM('calculated value'));

#15085=STANDARD_UNCERTAINTY('value       uncertainty','x       uncertainty',-
0.010000000000000);
#15086=STANDARD_UNCERTAINTY('value       uncertainty','y       uncertainty',-
0.010000000000000);
#15087=STANDARD_UNCERTAINTY('value       uncertainty','z       uncertainty',-
0.010000000000000);
#15088=EXPANDED_UNCERTAINTY('value       uncertainty','z       uncertainty',-
0.010000000000000,1.0);

/*  ---------------  dm_dimension_parameter  --------------------*/

#16000=DM_DIMENSION_PARAMETER('',(#16002),#34);
#16002=(LENGTH_MEASURE_WITH_UNIT()
```

```
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
        QUALIFIED_REPRESENTATION_ITEM((#16085,#16086))
        REPRESENTATION_ITEM('calculated value'));


#16085=STANDARD_UNCERTAINTY('value uncertainty','',-0.010000000000000);
#16086=PRECISION_QUALIFIER(4);

/*  ---------------  dm_point  --------------------*/

#17490=DM_POINT('measurement points',(#17510,#17520),#34);
#17510=CARTESIAN_POINT('measured                    point',(1.040834E-17,-
0.186000000000000,0.630000000000000));
#17520=CARTESIAN_POINT('expected                    point',(1.040834E-17,-
0.186000000000000,0.630000000000000));


ENDSEC;
END-ISO-10303-21;
```

J.6 Manufacturing suite test case

This test case was generated to test 10303-240, 10303-238, ISO 10303-224 and this part of ISO 10303. The part was modified by adding specific data for this part of ISO 10303, and removing specific data from other parts of ISO 10303. This part of ISO 10303 has now been generated on a CAD system and output as an ISO 10303-224 file. This ISO 10303-224 file has been tested and validated to the schema of this part of ISO 10303. The data that is in common between all of the application protocols includes: Boundary-representation geometry, Features, Geometric and Dimensional Tolerances, Materials, and some product structure.

This part was successfully tested against the schema in this document.

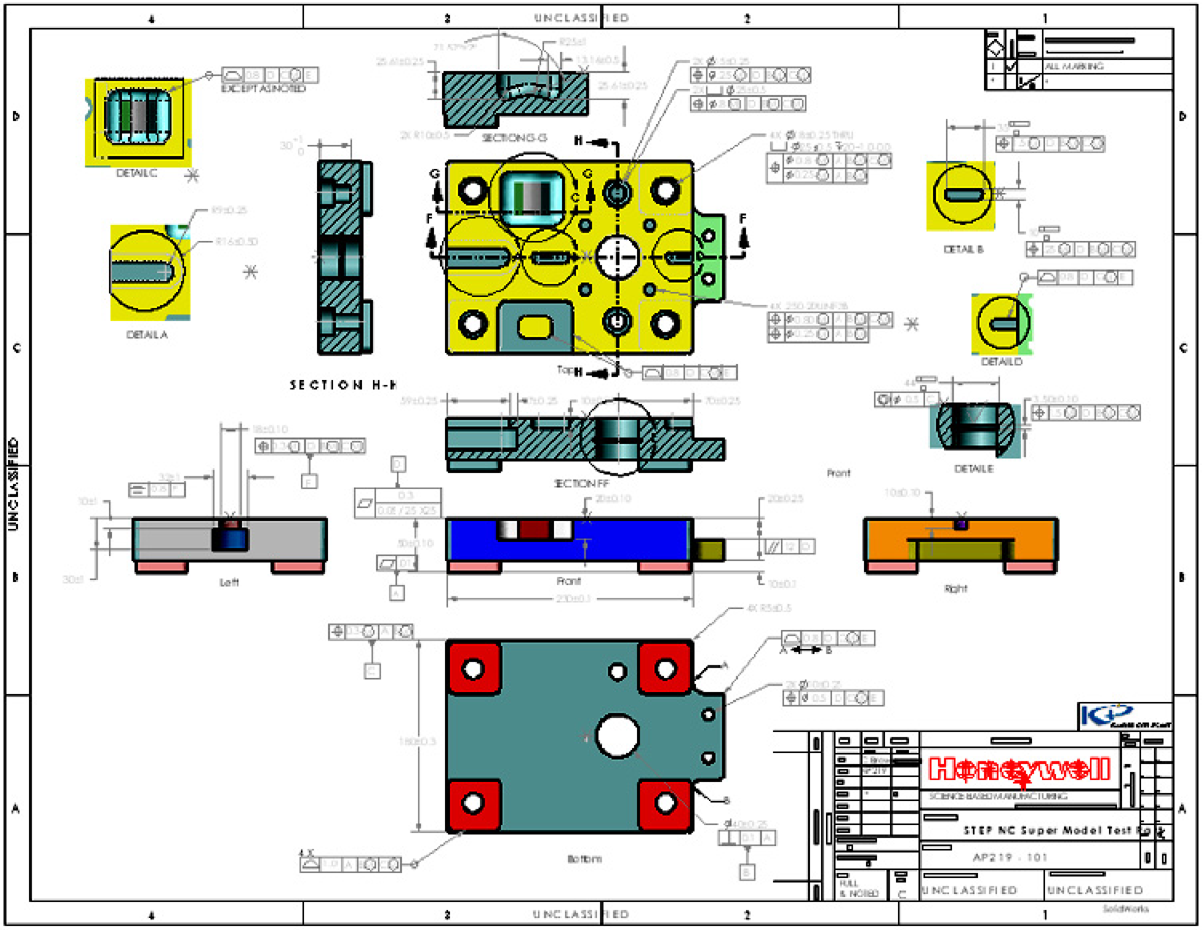


Figure J.1 — Manufacturing suite test case test part drawing

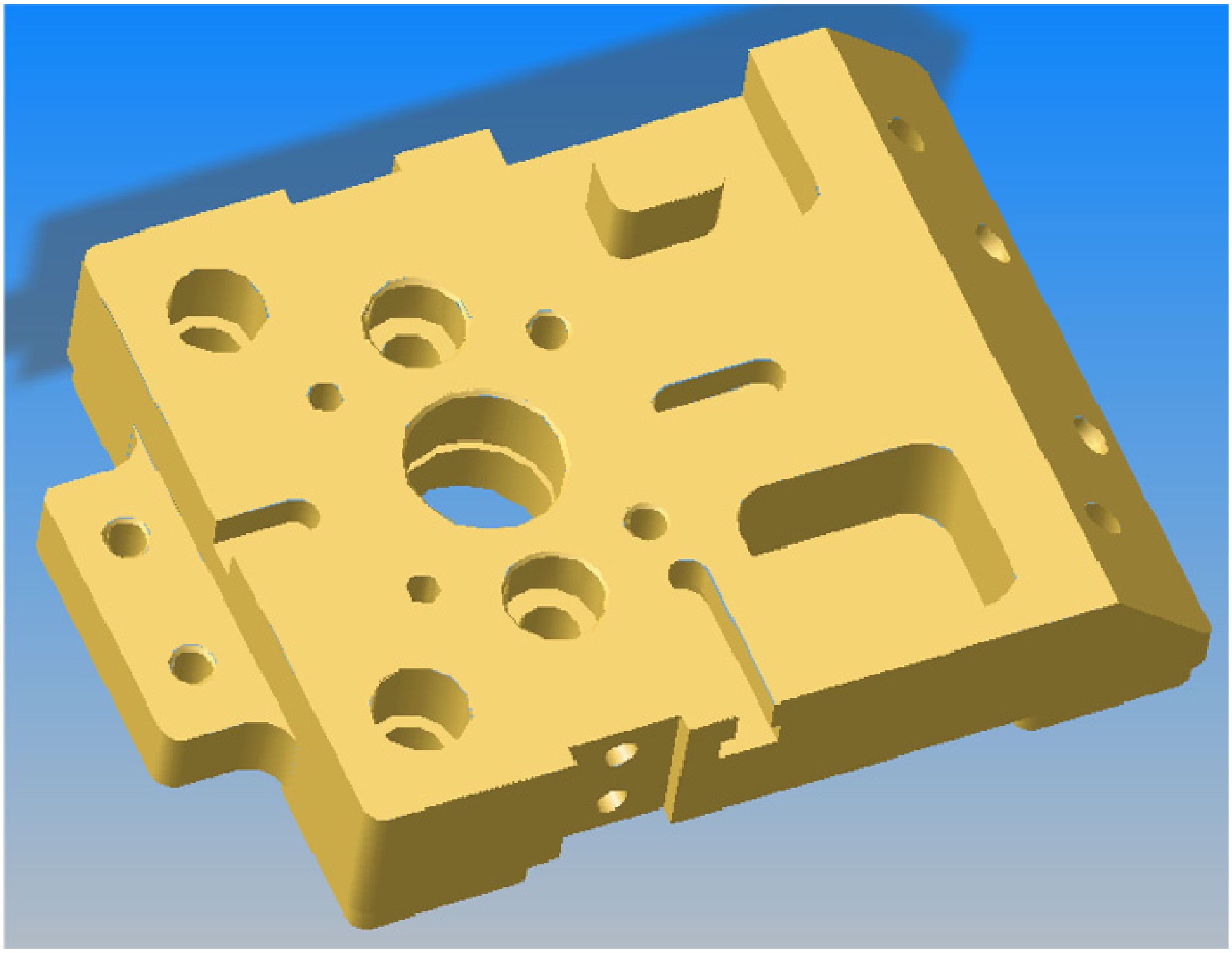


Figure J.2 — Manufacturing suite test case part

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('AP219 File'),'2;1');
FILE_NAME('92824_800172_g.219','1997-08-01T11:07:18-05:00',('RPTS
Operator'),('RPTS'),'RPTS   MP   6.0','PTC   Pro/ENGINEER   Version   18.0','RPTS
Operator');
FILE_SCHEMA(('DIMENSIONAL_INSPECTION_SCHEMA'));
ENDSEC;
DATA;
#5=DIMENSIONAL_EXPONENTS(1.0,0.0,0.0,0.0,0.0,0.0,0.0);
#6=DIMENSIONAL_EXPONENTS(0.0,0.0,0.0,0.0,0.0,0.0,0.0);
#10=(LENGTH_UNIT()NAMED_UNIT(#5)SI_UNIT(.MILLI.,.METRE.));
#11=RATIO_UNIT(#6);
#14=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#15=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.017453292519943),#14)
;
#19=(CONVERSION_BASED_UNIT('DEGREE',#15)NAMED_UNIT(#6)PLANE_ANGLE_UNIT());
#23=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
```

```
#24=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.000010000000000),#10,'clos
ure','maximum model space distance between geometric entities at asserted
connectivities');
#29=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((
#24))GLOBAL_UNIT_ASSIGNED_CONTEXT((#10,#19,#23))REPRESENTATION_CONTEXT('ID1',
'3D'));


#31=APPLICATION_CONTEXT('cast parts schema');
#32=APPLICATION_PROTOCOL_DEFINITION('IS','dimensional inspection
schema',2000,#31);
#33=PRODUCT_CONTEXT('dimensional inspection schema',#31,'Mechanical');
#34=PRODUCT('00000_ap223-101_01','STEP NC SUPER MODEL TEST PART',$,(#33));
#35=PRODUCT_DEFINITION_FORMATION('NA','NA',#34);
#36=PRODUCT_DEFINITION_CONTEXT('part definition',#31,'manufacturing
planning');
#37=PRODUCT_DEFINITION('00000_AP223-101','product design',#35,#36);
#38=PRODUCT_DEFINITION_SHAPE('product shape','shape for product',#37);
#39=CARTESIAN_POINT('',(225.0,-40.0,0.0));
#40=VERTEX_POINT('',#39);
#41=CARTESIAN_POINT('',(5.0,-40.0,0.0));
#42=VERTEX_POINT('',#41);
#43=DIRECTION('',(-1.0,0.0,0.0));
#44=VECTOR('',#43,220.0);
#45=LINE('',#39,#44);
#46=EDGE_CURVE('',#40,#42,#45,.T.);
#47=ORIENTED_EDGE('',*,*,#46,.F.);
#48=CARTESIAN_POINT('',(225.0,0.0,0.0));
#49=VERTEX_POINT('',#48);
#50=DIRECTION('',(0.0,-1.0,0.0));
#51=VECTOR('',#50,40.0);
#52=LINE('',#48,#51);
#53=EDGE_CURVE('',#49,#40,#52,.T.);
#54=ORIENTED_EDGE('',*,*,#53,.F.);
#55=CARTESIAN_POINT('',(5.0,0.0,0.0));
#56=VERTEX_POINT('',#55);
#57=DIRECTION('',(-1.0,0.0,0.0));
#58=VECTOR('',#57,220.0);
#59=LINE('',#48,#58);
#60=EDGE_CURVE('',#49,#56,#59,.T.);
#61=ORIENTED_EDGE('',*,*,#60,.T.);
#62=DIRECTION('',(0.0,-1.0,0.0));
#63=VECTOR('',#62,40.0);
#64=LINE('',#55,#63);
#65=EDGE_CURVE('',#56,#42,#64,.T.);
#66=ORIENTED_EDGE('',*,*,#65,.T.);
#67=EDGE_LOOP('',(#47,#54,#61,#66));
#68=FACE_BOUND('',#67,.F.);
#69=CARTESIAN_POINT('',(0.0,0.0,0.0));
#70=DIRECTION('',(0.0,0.0,1.0));
#71=DIRECTION('',(1.0,0.0,0.0));
#72=AXIS2_PLACEMENT_3D('',#69,#70,#71);
#73=PLANE('',#72);
#74=ADVANCED_FACE('',(#68),#73,.F.);
#75=SHAPE_ASPECT('featured shape: face 1',$,#38,.T.);
#76=PROPERTY_DEFINITION('',$,#75);
```

```
#77=FACE_SHAPE_REPRESENTATION('',(#74),#29);
#78=PROPERTY_DEFINITION_REPRESENTATION(#76,#77);
#79=CARTESIAN_POINT('',(47.999999999999979,-20.0,180.0));
#80=VERTEX_POINT('',#79);
#81=CARTESIAN_POINT('',(117.999999999999943,-20.0,180.0));
#82=VERTEX_POINT('',#81);
#83=DIRECTION('',(1.0,0.0,0.0));
#84=VECTOR('',#83,69.999999999999972);
#85=LINE('',#79,#84);
#86=EDGE_CURVE('',#80,#82,#85,.T.);
#87=ORIENTED_EDGE('',*,*,#86,.T.);
#88=CARTESIAN_POINT('',(117.999999999999943,0.0,180.0));
#89=VERTEX_POINT('',#88);
#90=DIRECTION('',(0.0,-1.0,0.0));
#91=VECTOR('',#90,20.0);
#92=LINE('',#88,#91);
#93=EDGE_CURVE('',#89,#82,#92,.T.);
#94=ORIENTED_EDGE('',*,*,#93,.F.);
#95=CARTESIAN_POINT('',(225.0,0.0,180.0));
#96=VERTEX_POINT('',#95);
#97=DIRECTION('',(-1.0,0.0,0.0));
#98=VECTOR('',#97,107.000000000000057);
#99=LINE('',#95,#98);
#100=EDGE_CURVE('',#96,#89,#99,.T.);
#101=ORIENTED_EDGE('',*,*,#100,.F.);
#102=CARTESIAN_POINT('',(225.0,-40.0,180.0));
#103=VERTEX_POINT('',#102);
#104=DIRECTION('',(0.0,1.0,0.0));
#105=VECTOR('',#104,40.0);
#106=LINE('',#102,#105);
#107=EDGE_CURVE('',#103,#96,#106,.T.);
#108=ORIENTED_EDGE('',*,*,#107,.F.);
#109=CARTESIAN_POINT('',(5.0,-40.0,180.0));
#110=VERTEX_POINT('',#109);
#111=DIRECTION('',(1.0,0.0,0.0));
#112=VECTOR('',#111,220.0);
#113=LINE('',#109,#112);
#114=EDGE_CURVE('',#110,#103,#113,.T.);
#115=ORIENTED_EDGE('',*,*,#114,.F.);
#116=CARTESIAN_POINT('',(5.0,0.0,180.0));
#117=VERTEX_POINT('',#116);
#118=DIRECTION('',(0.0,-1.0,0.0));
#119=VECTOR('',#118,40.0);
#120=LINE('',#116,#119);
#121=EDGE_CURVE('',#117,#110,#120,.T.);
#122=ORIENTED_EDGE('',*,*,#121,.F.);
#123=CARTESIAN_POINT('',(47.999999999999979,0.0,180.0));
#124=VERTEX_POINT('',#123);
#125=DIRECTION('',(-1.0,0.0,0.0));
#126=VECTOR('',#125,42.999999999999979);
#127=LINE('',#123,#126);
#128=EDGE_CURVE('',#124,#117,#127,.T.);
#129=ORIENTED_EDGE('',*,*,#128,.F.);
#130=DIRECTION('',(0.0,-1.0,0.0));
#131=VECTOR('',#130,20.0);
#132=LINE('',#123,#131);
```



```
#133=EDGE_CURVE('',#124,#80,#132,.T.);
#134=ORIENTED_EDGE('',*,*,#133,.T.);
#135=EDGE_LOOP('',(#87,#94,#101,#108,#115,#122,#129,#134));
#136=FACE_BOUND('',#135,.F.);
#137=CARTESIAN_POINT('',(0.0,0.0,180.0));
#138=DIRECTION('',(0.0,0.0,1.0));
#139=DIRECTION('',(1.0,0.0,0.0));
#140=AXIS2_PLACEMENT_3D('',#137,#138,#139);
#141=PLANE('',#140);
#142=ADVANCED_FACE('',(#136),#141,.T.);
#143=SHAPE_ASPECT('featured shape: face 2',$,#38,.T.);
#144=PROPERTY_DEFINITION('',$,#143);
#145=FACE_SHAPE_REPRESENTATION('',(#142),#29);
#146=PROPERTY_DEFINITION_REPRESENTATION(#144,#145);
#147=CARTESIAN_POINT('',(0.0,-30.0,106.0));
#148=VERTEX_POINT('',#147);
#149=CARTESIAN_POINT('',(0.0,-10.0,106.0));
#150=VERTEX_POINT('',#149);
#151=DIRECTION('',(0.0,1.0,0.0));
#152=VECTOR('',#151,20.0);
#153=LINE('',#147,#152);
#154=EDGE_CURVE('',#148,#150,#153,.T.);
#155=ORIENTED_EDGE('',*,*,#154,.T.);
#156=CARTESIAN_POINT('',(0.0,-10.0,99.0));
#157=VERTEX_POINT('',#156);
#158=DIRECTION('',(0.0,0.0,1.0));
#159=VECTOR('',#158,7.0);
#160=LINE('',#156,#159);
#161=EDGE_CURVE('',#157,#150,#160,.T.);
#162=ORIENTED_EDGE('',*,*,#161,.F.);
#163=CARTESIAN_POINT('',(0.0,0.0,99.0));
#164=VERTEX_POINT('',#163);
#165=DIRECTION('',(0.0,-1.0,0.0));
#166=VECTOR('',#165,10.0);
#167=LINE('',#163,#166);
#168=EDGE_CURVE('',#164,#157,#167,.T.);
#169=ORIENTED_EDGE('',*,*,#168,.F.);
#170=CARTESIAN_POINT('',(0.0,0.0,175.0));
#171=VERTEX_POINT('',#170);
#172=DIRECTION('',(0.0,0.0,1.0));
#173=VECTOR('',#172,76.0);
#174=LINE('',#163,#173);
#175=EDGE_CURVE('',#164,#171,#174,.T.);
#176=ORIENTED_EDGE('',*,*,#175,.T.);
#177=CARTESIAN_POINT('',(0.0,-40.0,175.0));
#178=VERTEX_POINT('',#177);
#179=DIRECTION('',(0.0,-1.0,0.0));
#180=VECTOR('',#179,40.0);
#181=LINE('',#170,#180);
#182=EDGE_CURVE('',#171,#178,#181,.T.);
#183=ORIENTED_EDGE('',*,*,#182,.T.);
#184=CARTESIAN_POINT('',(0.0,-40.0,5.0));
#185=VERTEX_POINT('',#184);
#186=DIRECTION('',(0.0,0.0,1.0));
#187=VECTOR('',#186,170.0);
#188=LINE('',#184,#187);
```



```
#189=EDGE_CURVE('',#185,#178,#188,.T.);
#190=ORIENTED_EDGE('',*,*,#189,.F.);
#191=CARTESIAN_POINT('',(0.0,0.0,5.0));
#192=VERTEX_POINT('',#191);
#193=DIRECTION('',(0.0,-1.0,0.0));
#194=VECTOR('',#193,40.0);
#195=LINE('',#191,#194);
#196=EDGE_CURVE('',#192,#185,#195,.T.);
#197=ORIENTED_EDGE('',*,*,#196,.F.);
#198=CARTESIAN_POINT('',(0.0,0.0,81.0));
#199=VERTEX_POINT('',#198);
#200=DIRECTION('',(0.0,0.0,1.0));
#201=VECTOR('',#200,76.0);
#202=LINE('',#191,#201);
#203=EDGE_CURVE('',#192,#199,#202,.T.);
#204=ORIENTED_EDGE('',*,*,#203,.T.);
#205=CARTESIAN_POINT('',(0.0,-10.0,81.0));
#206=VERTEX_POINT('',#205);
#207=DIRECTION('',(0.0,-1.0,0.0));
#208=VECTOR('',#207,10.0);
#209=LINE('',#198,#208);
#210=EDGE_CURVE('',#199,#206,#209,.T.);
#211=ORIENTED_EDGE('',*,*,#210,.T.);
#212=CARTESIAN_POINT('',(0.0,-10.0,74.0));
#213=VERTEX_POINT('',#212);
#214=DIRECTION('',(0.0,0.0,1.0));
#215=VECTOR('',#214,7.0);
#216=LINE('',#212,#215);
#217=EDGE_CURVE('',#213,#206,#216,.T.);
#218=ORIENTED_EDGE('',*,*,#217,.F.);
#219=CARTESIAN_POINT('',(0.0,-30.0,74.0));
#220=VERTEX_POINT('',#219);
#221=DIRECTION('',(0.0,1.0,0.0));
#222=VECTOR('',#221,20.0);
#223=LINE('',#219,#222);
#224=EDGE_CURVE('',#220,#213,#223,.T.);
#225=ORIENTED_EDGE('',*,*,#224,.F.);
#226=DIRECTION('',(0.0,0.0,1.0));
#227=VECTOR('',#226,32.0);
#228=LINE('',#219,#227);
#229=EDGE_CURVE('',#220,#148,#228,.T.);
#230=ORIENTED_EDGE('',*,*,#229,.T.);
#231=EDGE_LOOP('',(#155,#162,#169,#176,#183,#190,#197,#204,#211,#218,#225,#23
0));
#232=FACE_BOUND('',#231,.F.);
#233=CARTESIAN_POINT('',(0.0,0.0,0.0));
#234=DIRECTION('',(-1.0,0.0,0.0));
#235=DIRECTION('',(0.0,-1.0,0.0));
#236=AXIS2_PLACEMENT_3D('',#233,#234,#235);
#237=PLANE('',#236);
#238=ADVANCED_FACE('',(#232),#237,.T.);
#239=SHAPE_ASPECT('featured shape: face 3',$,#38,.T.);
#240=PROPERTY_DEFINITION('',$,#239);
#241=FACE_SHAPE_REPRESENTATION('',(#238),#29);
#242=PROPERTY_DEFINITION_REPRESENTATION(#240,#241);
#243=CARTESIAN_POINT('',(7.0,-50.0,50.000000000000007));
```



```
#244=VERTEX_POINT('',#243);
#245=CARTESIAN_POINT('',(45.0,-50.0,50.000000000000007));
#246=VERTEX_POINT('',#245);
#247=DIRECTION('',(1.0,0.0,0.0));
#248=VECTOR('',#247,38.0);
#249=LINE('',#243,#248);
#250=EDGE_CURVE('',#244,#246,#249,.T.);
#251=ORIENTED_EDGE('',*,*,#250,.T.);
#252=CARTESIAN_POINT('',(50.0,-50.0,45.000000000000007));
#253=VERTEX_POINT('',#252);
#254=CARTESIAN_POINT('',(45.0,-50.0,45.000000000000007));
#255=DIRECTION('',(0.0,-1.0,0.0));
#256=DIRECTION('',(1.0,0.0,0.0));
#257=AXIS2_PLACEMENT_3D('',#254,#255,#256);
#258=CIRCLE('',#257,5.0);
#259=EDGE_CURVE('',#253,#246,#258,.T.);
#260=ORIENTED_EDGE('',*,*,#259,.F.);
#261=CARTESIAN_POINT('',(50.0,-50.0,7.0));
#262=VERTEX_POINT('',#261);
#263=DIRECTION('',(0.0,0.0,1.0));
#264=VECTOR('',#263,38.000000000000007);
#265=LINE('',#261,#264);
#266=EDGE_CURVE('',#262,#253,#265,.T.);
#267=ORIENTED_EDGE('',*,*,#266,.F.);
#268=CARTESIAN_POINT('',(45.0,-50.0,2.0));
#269=VERTEX_POINT('',#268);
#270=CARTESIAN_POINT('',(45.0,-50.0,7.0));
#271=DIRECTION('',(0.0,-1.0,0.0));
#272=DIRECTION('',(0.0,0.0,-1.0));
#273=AXIS2_PLACEMENT_3D('',#270,#271,#272);
#274=CIRCLE('',#273,5.0);
#275=EDGE_CURVE('',#269,#262,#274,.T.);
#276=ORIENTED_EDGE('',*,*,#275,.F.);
#277=CARTESIAN_POINT('',(7.0,-50.0,2.0));
#278=VERTEX_POINT('',#277);
#279=DIRECTION('',(1.0,0.0,0.0));
#280=VECTOR('',#279,38.0);
#281=LINE('',#277,#280);
#282=EDGE_CURVE('',#278,#269,#281,.T.);
#283=ORIENTED_EDGE('',*,*,#282,.F.);
#284=CARTESIAN_POINT('',(2.0,-50.0,7.0));
#285=VERTEX_POINT('',#284);
#286=CARTESIAN_POINT('',(7.0,-50.0,7.0));
#287=DIRECTION('',(0.0,-1.0,0.0));
#288=DIRECTION('',(-1.0,0.0,0.0));
#289=AXIS2_PLACEMENT_3D('',#286,#287,#288);
#290=CIRCLE('',#289,5.0);
#291=EDGE_CURVE('',#285,#278,#290,.T.);
#292=ORIENTED_EDGE('',*,*,#291,.F.);
#293=CARTESIAN_POINT('',(2.0,-50.0,45.000000000000007));
#294=VERTEX_POINT('',#293);
#295=DIRECTION('',(0.0,0.0,1.0));
#296=VECTOR('',#295,38.000000000000007);
#297=LINE('',#284,#296);
#298=EDGE_CURVE('',#285,#294,#297,.T.);
#299=ORIENTED_EDGE('',*,*,#298,.T.);
```

```
#300=CARTESIAN_POINT('',(7.0,-50.0,45.000000000000007));
#301=DIRECTION('',(0.0,1.0,0.0));
#302=DIRECTION('',(-1.0,0.0,0.0));
#303=AXIS2_PLACEMENT_3D('',#300,#301,#302);
#304=CIRCLE('',#303,5.0);
#305=EDGE_CURVE('',#294,#244,#304,.T.);
#306=ORIENTED_EDGE('',*,*,#305,.T.);
#307=EDGE_LOOP('',(#251,#260,#267,#276,#283,#292,#299,#306));
#308=FACE_BOUND('',#307,.F.);
#309=CARTESIAN_POINT('',(34.0,-50.0,27.500000000000000));
#310=VERTEX_POINT('',#309);
#311=CARTESIAN_POINT('',(16.0,-50.0,27.500000000000000));
#312=VERTEX_POINT('',#311);
#313=CARTESIAN_POINT('',(25.0,-50.0,27.500000000000000));
#314=DIRECTION('',(0.0,1.0,0.0));
#315=DIRECTION('',(1.0,0.0,0.0));
#316=AXIS2_PLACEMENT_3D('',#313,#314,#315);
#317=CIRCLE('',#316,9.0);
#318=EDGE_CURVE('',#310,#312,#317,.T.);
#319=ORIENTED_EDGE('',*,*,#318,.F.);
#320=CARTESIAN_POINT('',(25.0,-50.0,27.500000000000000));
#321=DIRECTION('',(0.0,1.0,0.0));
#322=DIRECTION('',(-1.0,0.0,0.0));
#323=AXIS2_PLACEMENT_3D('',#320,#321,#322);
#324=CIRCLE('',#323,9.0);
#325=EDGE_CURVE('',#312,#310,#324,.T.);
#326=ORIENTED_EDGE('',*,*,#325,.F.);
#327=EDGE_LOOP('',(#319,#326));
#328=FACE_BOUND('',#327,.F.);
#329=CARTESIAN_POINT('',(0.0,-50.0,0.0));
#330=DIRECTION('',(0.0,-1.0,0.0));
#331=DIRECTION('',(1.0,0.0,0.0));
#332=AXIS2_PLACEMENT_3D('',#329,#330,#331);
#333=PLANE('',#332);
#334=ADVANCED_FACE('',(#308,#328),#333,.T.);
#335=SHAPE_ASPECT('featured shape: face 4',$,#38,.T.);
#336=PROPERTY_DEFINITION('',$,#335);
#337=FACE_SHAPE_REPRESENTATION('',(#334),#29);
#338=PROPERTY_DEFINITION_REPRESENTATION(#336,#337);
#339=CARTESIAN_POINT('',(180.000000000000028,-50.0,7.0));
#340=VERTEX_POINT('',#339);
#341=CARTESIAN_POINT('',(180.000000000000028,-50.0,45.0));
#342=VERTEX_POINT('',#341);
#343=DIRECTION('',(0.0,0.0,1.0));
#344=VECTOR('',#343,38.0);
#345=LINE('',#339,#344);
#346=EDGE_CURVE('',#340,#342,#345,.T.);
#347=ORIENTED_EDGE('',*,*,#346,.T.);
#348=CARTESIAN_POINT('',(185.000000000000028,-50.0,50.0));
#349=VERTEX_POINT('',#348);
#350=CARTESIAN_POINT('',(185.000000000000028,-50.0,45.0));
#351=DIRECTION('',(0.0,-1.0,0.0));
#352=DIRECTION('',(0.0,0.0,1.0));
#353=AXIS2_PLACEMENT_3D('',#350,#351,#352);
#354=CIRCLE('',#353,5.0);
#355=EDGE_CURVE('',#349,#342,#354,.T.);
```



```
#356=ORIENTED_EDGE('',*,*,#355,.F.);
#357=CARTESIAN_POINT('',(223.0,-50.0,50.0));
#358=VERTEX_POINT('',#357);
#359=DIRECTION('',(-1.0,0.0,0.0));
#360=VECTOR('',#359,37.999999999999972);
#361=LINE('',#357,#360);
#362=EDGE_CURVE('',#358,#349,#361,.T.);
#363=ORIENTED_EDGE('',*,*,#362,.F.);
#364=CARTESIAN_POINT('',(228.0,-50.0,45.0));
#365=VERTEX_POINT('',#364);
#366=CARTESIAN_POINT('',(223.0,-50.0,45.0));
#367=DIRECTION('',(0.0,-1.0,0.0));
#368=DIRECTION('',(1.0,0.0,0.0));
#369=AXIS2_PLACEMENT_3D('',#366,#367,#368);
#370=CIRCLE('',#369,5.0);
#371=EDGE_CURVE('',#365,#358,#370,.T.);
#372=ORIENTED_EDGE('',*,*,#371,.F.);
#373=CARTESIAN_POINT('',(228.0,-50.0,7.0));
#374=VERTEX_POINT('',#373);
#375=DIRECTION('',(0.0,0.0,1.0));
#376=VECTOR('',#375,38.0);
#377=LINE('',#373,#376);
#378=EDGE_CURVE('',#374,#365,#377,.T.);
#379=ORIENTED_EDGE('',*,*,#378,.F.);
#380=CARTESIAN_POINT('',(223.0,-50.0,2.0));
#381=VERTEX_POINT('',#380);
#382=CARTESIAN_POINT('',(223.0,-50.0,7.0));
#383=DIRECTION('',(0.0,1.0,0.0));
#384=DIRECTION('',(1.0,0.0,0.0));
#385=AXIS2_PLACEMENT_3D('',#382,#383,#384);
#386=CIRCLE('',#385,5.0);
#387=EDGE_CURVE('',#374,#381,#386,.T.);
#388=ORIENTED_EDGE('',*,*,#387,.T.);
#389=CARTESIAN_POINT('',(185.000000000000028,-50.0,2.0));
#390=VERTEX_POINT('',#389);
#391=DIRECTION('',(1.0,0.0,0.0));
#392=VECTOR('',#391,37.999999999999972);
#393=LINE('',#389,#392);
#394=EDGE_CURVE('',#390,#381,#393,.T.);
#395=ORIENTED_EDGE('',*,*,#394,.F.);
#396=CARTESIAN_POINT('',(185.000000000000028,-50.0,7.0));
#397=DIRECTION('',(0.0,1.0,0.0));
#398=DIRECTION('',(0.0,0.0,-1.0));
#399=AXIS2_PLACEMENT_3D('',#396,#397,#398);
#400=CIRCLE('',#399,5.0);
#401=EDGE_CURVE('',#390,#340,#400,.T.);
#402=ORIENTED_EDGE('',*,*,#401,.T.);
#403=EDGE_LOOP('',(#347,#356,#363,#372,#379,#388,#395,#402));
#404=FACE_BOUND('',#403,.F.);
#405=CARTESIAN_POINT('',(214.0,-50.0,27.500000000000000));
#406=VERTEX_POINT('',#405);
#407=CARTESIAN_POINT('',(196.0,-50.0,27.500000000000000));
#408=VERTEX_POINT('',#407);
#409=CARTESIAN_POINT('',(205.0,-50.0,27.500000000000000));
#410=DIRECTION('',(0.0,1.0,0.0));
#411=DIRECTION('',(1.0,0.0,0.0));
```

```
#412=AXIS2_PLACEMENT_3D('',#409,#410,#411);
#413=CIRCLE('',#412,9.0);
#414=EDGE_CURVE('',#406,#408,#413,.T.);
#415=ORIENTED_EDGE('',*,*,#414,.F.);
#416=CARTESIAN_POINT('',(205.0,-50.0,27.500000000000000));
#417=DIRECTION('',(0.0,1.0,0.0));
#418=DIRECTION('',(-1.0,0.0,0.0));
#419=AXIS2_PLACEMENT_3D('',#416,#417,#418);
#420=CIRCLE('',#419,9.0);
#421=EDGE_CURVE('',#408,#406,#420,.T.);
#422=ORIENTED_EDGE('',*,*,#421,.F.);
#423=EDGE_LOOP('',(#415,#422));
#424=FACE_BOUND('',#423,.F.);
#425=ADVANCED_FACE('',(#404,#424),#333,.T.);
#426=SHAPE_ASPECT('featured shape: face 5',$,#38,.T.);
#427=PROPERTY_DEFINITION('',$,#426);
#428=FACE_SHAPE_REPRESENTATION('',(#425),#29);
#429=PROPERTY_DEFINITION_REPRESENTATION(#427,#428);
#430=CARTESIAN_POINT('',(223.0,-50.0,130.000000000000028));
#431=VERTEX_POINT('',#430);
#432=CARTESIAN_POINT('',(184.999999999999972,-50.0,130.000000000000028));
#433=VERTEX_POINT('',#432);
#434=DIRECTION('',(-1.0,0.0,0.0));
#435=VECTOR('',#434,38.000000000000028);
#436=LINE('',#430,#435);
#437=EDGE_CURVE('',#431,#433,#436,.T.);
#438=ORIENTED_EDGE('',*,*,#437,.T.);
#439=CARTESIAN_POINT('',(179.999999999999972,-50.0,135.000000000000028));
#440=VERTEX_POINT('',#439);
#441=CARTESIAN_POINT('',(184.999999999999972,-50.0,135.000000000000028));
#442=DIRECTION('',(0.0,-1.0,0.0));
#443=DIRECTION('',(-1.0,0.0,0.0));
#444=AXIS2_PLACEMENT_3D('',#441,#442,#443);
#445=CIRCLE('',#444,5.0);
#446=EDGE_CURVE('',#440,#433,#445,.T.);
#447=ORIENTED_EDGE('',*,*,#446,.F.);
#448=CARTESIAN_POINT('',(179.999999999999972,-50.0,173.0));
#449=VERTEX_POINT('',#448);
#450=DIRECTION('',(0.0,0.0,-1.0));
#451=VECTOR('',#450,37.999999999999972);
#452=LINE('',#448,#451);
#453=EDGE_CURVE('',#449,#440,#452,.T.);
#454=ORIENTED_EDGE('',*,*,#453,.F.);
#455=CARTESIAN_POINT('',(184.999999999999972,-50.0,178.0));
#456=VERTEX_POINT('',#455);
#457=CARTESIAN_POINT('',(184.999999999999972,-50.0,173.0));
#458=DIRECTION('',(0.0,-1.0,0.0));
#459=DIRECTION('',(0.0,0.0,1.0));
#460=AXIS2_PLACEMENT_3D('',#457,#458,#459);
#461=CIRCLE('',#460,5.0);
#462=EDGE_CURVE('',#456,#449,#461,.T.);
#463=ORIENTED_EDGE('',*,*,#462,.F.);
#464=CARTESIAN_POINT('',(223.0,-50.0,178.0));
#465=VERTEX_POINT('',#464);
#466=DIRECTION('',(1.0,0.0,0.0));
#467=VECTOR('',#466,38.000000000000028);
```



```
#468=LINE('',#455,#467);
#469=EDGE_CURVE('',#456,#465,#468,.T.);
#470=ORIENTED_EDGE('',*,*,#469,.T.);
#471=CARTESIAN_POINT('',(228.0,-50.0,173.0));
#472=VERTEX_POINT('',#471);
#473=CARTESIAN_POINT('',(223.0,-50.0,173.0));
#474=DIRECTION('',(0.0,-1.0,0.0));
#475=DIRECTION('',(1.0,0.0,0.0));
#476=AXIS2_PLACEMENT_3D('',#473,#474,#475);
#477=CIRCLE('',#476,5.0);
#478=EDGE_CURVE('',#472,#465,#477,.T.);
#479=ORIENTED_EDGE('',*,*,#478,.F.);
#480=CARTESIAN_POINT('',(228.0,-50.0,135.000000000000028));
#481=VERTEX_POINT('',#480);
#482=DIRECTION('',(0.0,0.0,1.0));
#483=VECTOR('',#482,37.999999999999972);
#484=LINE('',#480,#483);
#485=EDGE_CURVE('',#481,#472,#484,.T.);
#486=ORIENTED_EDGE('',*,*,#485,.F.);
#487=CARTESIAN_POINT('',(223.0,-50.0,135.000000000000028));
#488=DIRECTION('',(0.0,1.0,0.0));
#489=DIRECTION('',(1.0,0.0,0.0));
#490=AXIS2_PLACEMENT_3D('',#487,#488,#489);
#491=CIRCLE('',#490,5.0);
#492=EDGE_CURVE('',#481,#431,#491,.T.);
#493=ORIENTED_EDGE('',*,*,#492,.T.);
#494=EDGE_LOOP('',(#438,#447,#454,#463,#470,#479,#486,#493));
#495=FACE_BOUND('',#494,.F.);
#496=CARTESIAN_POINT('',(214.0,-50.0,152.500000000000000));
#497=VERTEX_POINT('',#496);
#498=CARTESIAN_POINT('',(196.0,-50.0,152.500000000000000));
#499=VERTEX_POINT('',#498);
#500=CARTESIAN_POINT('',(205.0,-50.0,152.500000000000000));
#501=DIRECTION('',(0.0,1.0,0.0));
#502=DIRECTION('',(1.0,0.0,0.0));
#503=AXIS2_PLACEMENT_3D('',#500,#501,#502);
#504=CIRCLE('',#503,9.0);
#505=EDGE_CURVE('',#497,#499,#504,.T.);
#506=ORIENTED_EDGE('',*,*,#505,.F.);
#507=CARTESIAN_POINT('',(205.0,-50.0,152.500000000000000));
#508=DIRECTION('',(0.0,1.0,0.0));
#509=DIRECTION('',(-1.0,0.0,0.0));
#510=AXIS2_PLACEMENT_3D('',#507,#508,#509);
#511=CIRCLE('',#510,9.0);
#512=EDGE_CURVE('',#499,#497,#511,.T.);
#513=ORIENTED_EDGE('',*,*,#512,.F.);
#514=EDGE_LOOP('',(#506,#513));
#515=FACE_BOUND('',#514,.F.);
#516=ADVANCED_FACE('',(#495,#515),#333,.T.);
#517=SHAPE_ASPECT('featured shape: face 6',$,#38,.T.);
#518=PROPERTY_DEFINITION('',$,#517);
#519=FACE_SHAPE_REPRESENTATION('',(#516),#29);
#520=PROPERTY_DEFINITION_REPRESENTATION(#518,#519);
#521=CARTESIAN_POINT('',(49.999999999999972,-50.0,173.0));
#522=VERTEX_POINT('',#521);
#523=CARTESIAN_POINT('',(49.999999999999972,-50.0,134.999999999999972));
```

```
#524=VERTEX_POINT('',#523);
#525=DIRECTION('',(0.0,0.0,-1.0));
#526=VECTOR('',#525,38.000000000000028);
#527=LINE('',#521,#526);
#528=EDGE_CURVE('',#522,#524,#527,.T.);
#529=ORIENTED_EDGE('',*,*,#528,.T.);
#530=CARTESIAN_POINT('',(44.999999999999972,-50.0,129.999999999999972));
#531=VERTEX_POINT('',#530);
#532=CARTESIAN_POINT('',(44.999999999999972,-50.0,134.999999999999972));
#533=DIRECTION('',(0.0,-1.0,0.0));
#534=DIRECTION('',(0.0,0.0,-1.0));
#535=AXIS2_PLACEMENT_3D('',#532,#533,#534);
#536=CIRCLE('',#535,5.0);
#537=EDGE_CURVE('',#531,#524,#536,.T.);
#538=ORIENTED_EDGE('',*,*,#537,.F.);
#539=CARTESIAN_POINT('',(7.0,-50.0,129.999999999999972));
#540=VERTEX_POINT('',#539);
#541=DIRECTION('',(1.0,0.0,0.0));
#542=VECTOR('',#541,37.999999999999972);
#543=LINE('',#539,#542);
#544=EDGE_CURVE('',#540,#531,#543,.T.);
#545=ORIENTED_EDGE('',*,*,#544,.F.);
#546=CARTESIAN_POINT('',(2.0,-50.0,134.999999999999972));
#547=VERTEX_POINT('',#546);
#548=CARTESIAN_POINT('',(7.0,-50.0,134.999999999999972));
#549=DIRECTION('',(0.0,-1.0,0.0));
#550=DIRECTION('',(-1.0,0.0,0.0));
#551=AXIS2_PLACEMENT_3D('',#548,#549,#550);
#552=CIRCLE('',#551,5.0);
#553=EDGE_CURVE('',#547,#540,#552,.T.);
#554=ORIENTED_EDGE('',*,*,#553,.F.);
#555=CARTESIAN_POINT('',(2.0,-50.0,173.0));
#556=VERTEX_POINT('',#555);
#557=DIRECTION('',(0.0,0.0,1.0));
#558=VECTOR('',#557,38.000000000000028);
#559=LINE('',#546,#558);
#560=EDGE_CURVE('',#547,#556,#559,.T.);
#561=ORIENTED_EDGE('',*,*,#560,.T.);
#562=CARTESIAN_POINT('',(7.0,-50.0,178.0));
#563=VERTEX_POINT('',#562);
#564=CARTESIAN_POINT('',(7.0,-50.0,173.0));
#565=DIRECTION('',(0.0,1.0,0.0));
#566=DIRECTION('',(-1.0,0.0,0.0));
#567=AXIS2_PLACEMENT_3D('',#564,#565,#566);
#568=CIRCLE('',#567,5.0);
#569=EDGE_CURVE('',#556,#563,#568,.T.);
#570=ORIENTED_EDGE('',*,*,#569,.T.);
#571=CARTESIAN_POINT('',(44.999999999999972,-50.0,178.0));
#572=VERTEX_POINT('',#571);
#573=DIRECTION('',(1.0,0.0,0.0));
#574=VECTOR('',#573,37.999999999999972);
#575=LINE('',#562,#574);
#576=EDGE_CURVE('',#563,#572,#575,.T.);
#577=ORIENTED_EDGE('',*,*,#576,.T.);
#578=CARTESIAN_POINT('',(44.999999999999972,-50.0,173.0));
#579=DIRECTION('',(0.0,1.0,0.0));
```



```
#580=DIRECTION('',(0.0,0.0,1.0));
#581=AXIS2_PLACEMENT_3D('',#578,#579,#580);
#582=CIRCLE('',#581,5.0);
#583=EDGE_CURVE('',#572,#522,#582,.T.);
#584=ORIENTED_EDGE('',*,*,#583,.T.);
#585=EDGE_LOOP('',(#529,#538,#545,#554,#561,#570,#577,#584));
#586=FACE_BOUND('',#585,.F.);
#587=CARTESIAN_POINT('',(34.0,-50.0,152.500000000000000));
#588=VERTEX_POINT('',#587);
#589=CARTESIAN_POINT('',(16.0,-50.0,152.500000000000000));
#590=VERTEX_POINT('',#589);
#591=CARTESIAN_POINT('',(25.0,-50.0,152.500000000000000));
#592=DIRECTION('',(0.0,1.0,0.0));
#593=DIRECTION('',(1.0,0.0,0.0));
#594=AXIS2_PLACEMENT_3D('',#591,#592,#593);
#595=CIRCLE('',#594,9.0);
#596=EDGE_CURVE('',#588,#590,#595,.T.);
#597=ORIENTED_EDGE('',*,*,#596,.F.);
#598=CARTESIAN_POINT('',(25.0,-50.0,152.500000000000000));
#599=DIRECTION('',(0.0,1.0,0.0));
#600=DIRECTION('',(-1.0,0.0,0.0));
#601=AXIS2_PLACEMENT_3D('',#598,#599,#600);
#602=CIRCLE('',#601,9.0);
#603=EDGE_CURVE('',#590,#588,#602,.T.);
#604=ORIENTED_EDGE('',*,*,#603,.F.);
#605=EDGE_LOOP('',(#597,#604));
#606=FACE_BOUND('',#605,.F.);
#607=ADVANCED_FACE('',(#586,#606),#333,.T.);
#608=SHAPE_ASPECT('featured shape: face 7',$,#38,.T.);
#609=PROPERTY_DEFINITION('',$,#608);
#610=FACE_SHAPE_REPRESENTATION('',(#607),#29);
#611=PROPERTY_DEFINITION_REPRESENTATION(#609,#610);
#612=CARTESIAN_POINT('',(260.0,-40.0,55.0));
#613=VERTEX_POINT('',#612);
#614=CARTESIAN_POINT('',(260.0,-20.0,55.0));
#615=VERTEX_POINT('',#614);
#616=DIRECTION('',(0.0,1.0,0.0));
#617=VECTOR('',#616,20.0);
#618=LINE('',#612,#617);
#619=EDGE_CURVE('',#613,#615,#618,.T.);
#620=ORIENTED_EDGE('',*,*,#619,.F.);
#621=CARTESIAN_POINT('',(260.0,-40.0,125.0));
#622=VERTEX_POINT('',#621);
#623=DIRECTION('',(0.0,0.0,1.0));
#624=VECTOR('',#623,70.0);
#625=LINE('',#612,#624);
#626=EDGE_CURVE('',#613,#622,#625,.T.);
#627=ORIENTED_EDGE('',*,*,#626,.T.);
#628=CARTESIAN_POINT('',(260.0,-20.0,125.0));
#629=VERTEX_POINT('',#628);
#630=DIRECTION('',(0.0,1.0,0.0));
#631=VECTOR('',#630,20.0);
#632=LINE('',#621,#631);
#633=EDGE_CURVE('',#622,#629,#632,.T.);
#634=ORIENTED_EDGE('',*,*,#633,.T.);
#635=DIRECTION('',(0.0,0.0,1.0));
```

```
#636=VECTOR('',#635,70.0);
#637=LINE('',#614,#636);
#638=EDGE_CURVE('',#615,#629,#637,.T.);
#639=ORIENTED_EDGE('',*,*,#638,.F.);
#640=EDGE_LOOP('',(#620,#627,#634,#639));
#641=FACE_BOUND('',#640,.F.);
#642=CARTESIAN_POINT('',(260.0,-50.0,0.0));
#643=DIRECTION('',(1.0,0.0,0.0));
#644=DIRECTION('',(0.0,1.0,0.0));
#645=AXIS2_PLACEMENT_3D('',#642,#643,#644);
#646=PLANE('',#645);
#647=ADVANCED_FACE('',(#641),#646,.T.);
#648=SHAPE_ASPECT('featured shape: face 8',$,#38,.T.);
#649=PROPERTY_DEFINITION('',$,#648);
#650=FACE_SHAPE_REPRESENTATION('',(#647),#29);
#651=PROPERTY_DEFINITION_REPRESENTATION(#649,#650);
#652=CARTESIAN_POINT('',(117.999999999999943,0.0,135.0));
#653=VERTEX_POINT('',#652);
#654=DIRECTION('',(0.0,0.0,-1.0));
#655=VECTOR('',#654,45.0);
#656=LINE('',#88,#655);
#657=EDGE_CURVE('',#89,#653,#656,.T.);
#658=ORIENTED_EDGE('',*,*,#657,.T.);
#659=CARTESIAN_POINT('',(112.999999999999943,0.0,130.0));
#660=VERTEX_POINT('',#659);
#661=CARTESIAN_POINT('',(112.999999999999943,0.0,135.0));
#662=DIRECTION('',(0.0,1.0,0.0));
#663=DIRECTION('',(1.0,0.0,0.0));
#664=AXIS2_PLACEMENT_3D('',#661,#662,#663);
#665=CIRCLE('',#664,5.0);
#666=EDGE_CURVE('',#653,#660,#665,.T.);
#667=ORIENTED_EDGE('',*,*,#666,.T.);
#668=CARTESIAN_POINT('',(52.999999999999972,0.0,130.0));
#669=VERTEX_POINT('',#668);
#670=DIRECTION('',(-1.0,0.0,0.0));
#671=VECTOR('',#670,59.999999999999972);
#672=LINE('',#659,#671);
#673=EDGE_CURVE('',#660,#669,#672,.T.);
#674=ORIENTED_EDGE('',*,*,#673,.T.);
#675=CARTESIAN_POINT('',(47.999999999999972,0.0,135.0));
#676=VERTEX_POINT('',#675);
#677=CARTESIAN_POINT('',(52.999999999999979,0.0,135.0));
#678=DIRECTION('',(0.0,1.0,0.0));
#679=DIRECTION('',(-1.421085E-15,0.0,-1.0));
#680=AXIS2_PLACEMENT_3D('',#677,#678,#679);
#681=CIRCLE('',#680,5.0);
#682=EDGE_CURVE('',#669,#676,#681,.T.);
#683=ORIENTED_EDGE('',*,*,#682,.T.);
#684=DIRECTION('',(0.0,0.0,-1.0));
#685=VECTOR('',#684,45.0);
#686=LINE('',#123,#685);
#687=EDGE_CURVE('',#124,#676,#686,.T.);
#688=ORIENTED_EDGE('',*,*,#687,.F.);
#689=ORIENTED_EDGE('',*,*,#128,.T.);
#690=CARTESIAN_POINT('',(5.0,0.0,175.0));
#691=DIRECTION('',(0.0,-1.0,0.0));
```



```
#692=DIRECTION('',(0.0,0.0,1.0));
#693=AXIS2_PLACEMENT_3D('',#690,#691,#692);
#694=CIRCLE('',#693,5.0);
#695=EDGE_CURVE('',#117,#171,#694,.T.);
#696=ORIENTED_EDGE('',*,*,#695,.T.);
#697=ORIENTED_EDGE('',*,*,#175,.F.);
#698=CARTESIAN_POINT('',(49.999999694730889,0.0,99.0));
#699=VERTEX_POINT('',#698);
#700=DIRECTION('',(1.0,0.0,0.0));
#701=VECTOR('',#700,49.999999694730889);
#702=LINE('',#163,#701);
#703=EDGE_CURVE('',#164,#699,#702,.T.);
#704=ORIENTED_EDGE('',*,*,#703,.T.);
#705=CARTESIAN_POINT('',(50.0,0.0,81.0));
#706=VERTEX_POINT('',#705);
#707=CARTESIAN_POINT('',(50.0,0.0,90.0));
#708=DIRECTION('',(0.0,-1.0,0.0));
#709=DIRECTION('',(0.0,0.0,-1.0));
#710=AXIS2_PLACEMENT_3D('',#707,#708,#709);
#711=CIRCLE('',#710,9.0);
#712=EDGE_CURVE('',#706,#699,#711,.T.);
#713=ORIENTED_EDGE('',*,*,#712,.F.);
#714=DIRECTION('',(-1.0,0.0,0.0));
#715=VECTOR('',#714,50.0);
#716=LINE('',#705,#715);
#717=EDGE_CURVE('',#706,#199,#716,.T.);
#718=ORIENTED_EDGE('',*,*,#717,.T.);
#719=ORIENTED_EDGE('',*,*,#203,.F.);
#720=CARTESIAN_POINT('',(5.0,0.0,5.0));
#721=DIRECTION('',(0.0,1.0,0.0));
#722=DIRECTION('',(0.0,0.0,-1.0));
#723=AXIS2_PLACEMENT_3D('',#720,#721,#722);
#724=CIRCLE('',#723,5.0);
#725=EDGE_CURVE('',#56,#192,#724,.T.);
#726=ORIENTED_EDGE('',*,*,#725,.F.);
#727=ORIENTED_EDGE('',*,*,#60,.F.);
#728=CARTESIAN_POINT('',(230.0,0.0,5.0));
#729=VERTEX_POINT('',#728);
#730=CARTESIAN_POINT('',(225.0,0.0,5.0));
#731=DIRECTION('',(0.0,1.0,0.0));
#732=DIRECTION('',(1.0,0.0,0.0));
#733=AXIS2_PLACEMENT_3D('',#730,#731,#732);
#734=CIRCLE('',#733,5.0);
#735=EDGE_CURVE('',#729,#49,#734,.T.);
#736=ORIENTED_EDGE('',*,*,#735,.F.);
#737=CARTESIAN_POINT('',(230.0,0.0,85.0));
#738=VERTEX_POINT('',#737);
#739=DIRECTION('',(0.0,0.0,1.0));
#740=VECTOR('',#739,80.0);
#741=LINE('',#728,#740);
#742=EDGE_CURVE('',#729,#738,#741,.T.);
#743=ORIENTED_EDGE('',*,*,#742,.T.);
#744=CARTESIAN_POINT('',(210.0,0.0,85.0));
#745=VERTEX_POINT('',#744);
#746=DIRECTION('',(-1.0,0.0,0.0));
#747=VECTOR('',#746,20.0);
```

```
#748=LINE('',#737,#747);
#749=EDGE_CURVE('',#738,#745,#748,.T.);
#750=ORIENTED_EDGE('',*,*,#749,.T.);
#751=CARTESIAN_POINT('',(210.0,0.0,95.0));
#752=VERTEX_POINT('',#751);
#753=CARTESIAN_POINT('',(210.0,0.0,90.0));
#754=DIRECTION('',(0.0,-1.0,0.0));
#755=DIRECTION('',(0.0,0.0,1.0));
#756=AXIS2_PLACEMENT_3D('',#753,#754,#755);
#757=CIRCLE('',#756,5.0);
#758=EDGE_CURVE('',#752,#745,#757,.T.);
#759=ORIENTED_EDGE('',*,*,#758,.F.);
#760=CARTESIAN_POINT('',(230.0,0.0,95.0));
#761=VERTEX_POINT('',#760);
#762=DIRECTION('',(-1.0,0.0,0.0));
#763=VECTOR('',#762,20.0);
#764=LINE('',#760,#763);
#765=EDGE_CURVE('',#761,#752,#764,.T.);
#766=ORIENTED_EDGE('',*,*,#765,.F.);
#767=CARTESIAN_POINT('',(230.0,0.0,175.0));
#768=VERTEX_POINT('',#767);
#769=DIRECTION('',(0.0,0.0,1.0));
#770=VECTOR('',#769,80.0);
#771=LINE('',#760,#770);
#772=EDGE_CURVE('',#761,#768,#771,.T.);
#773=ORIENTED_EDGE('',*,*,#772,.T.);
#774=CARTESIAN_POINT('',(225.0,0.0,175.0));
#775=DIRECTION('',(0.0,-1.0,0.0));
#776=DIRECTION('',(1.0,0.0,0.0));
#777=AXIS2_PLACEMENT_3D('',#774,#775,#776);
#778=CIRCLE('',#777,5.0);
#779=EDGE_CURVE('',#768,#96,#778,.T.);
#780=ORIENTED_EDGE('',*,*,#779,.T.);
#781=ORIENTED_EDGE('',*,*,#100,.T.);
#782=EDGE_LOOP('',(#658,#667,#674,#683,#688,#689,#696,#697,#704,#713,#718,#71
9,#726,#727,#736,#743,#750,#759,#766,#773,#780,#781));
#783=FACE_BOUND('',#782,.F.);
#784=CARTESIAN_POINT('',(180.0,0.0,90.0));
#785=VERTEX_POINT('',#784);
#786=CARTESIAN_POINT('',(140.0,0.0,90.0));
#787=VERTEX_POINT('',#786);
#788=CARTESIAN_POINT('',(160.0,0.0,90.0));
#789=DIRECTION('',(0.0,-1.0,0.0));
#790=DIRECTION('',(1.0,0.0,0.0));
#791=AXIS2_PLACEMENT_3D('',#788,#789,#790);
#792=CIRCLE('',#791,20.0);
#793=EDGE_CURVE('',#785,#787,#792,.T.);
#794=ORIENTED_EDGE('',*,*,#793,.F.);
#795=CARTESIAN_POINT('',(160.0,0.0,90.0));
#796=DIRECTION('',(0.0,-1.0,0.0));
#797=DIRECTION('',(-1.0,0.0,0.0));
#798=AXIS2_PLACEMENT_3D('',#795,#796,#797);
#799=CIRCLE('',#798,20.0);
#800=EDGE_CURVE('',#787,#785,#799,.T.);
#801=ORIENTED_EDGE('',*,*,#800,.F.);
#802=EDGE_LOOP('',(#794,#801));
```



```
#803=FACE_BOUND('',#802,.F.);
#804=CARTESIAN_POINT('',(172.500000000000000,0.0,30.0));
#805=VERTEX_POINT('',#804);
#806=CARTESIAN_POINT('',(147.500000000000000,0.0,30.0));
#807=VERTEX_POINT('',#806);
#808=CARTESIAN_POINT('',(160.0,0.0,30.0));
#809=DIRECTION('',(0.0,1.0,0.0));
#810=DIRECTION('',(1.0,0.0,0.0));
#811=AXIS2_PLACEMENT_3D('',#808,#809,#810);
#812=CIRCLE('',#811,12.500000000000000);
#813=EDGE_CURVE('',#805,#807,#812,.T.);
#814=ORIENTED_EDGE('',*,*,#813,.T.);
#815=CARTESIAN_POINT('',(160.0,0.0,30.0));
#816=DIRECTION('',(0.0,1.0,0.0));
#817=DIRECTION('',(-1.0,0.0,0.0));
#818=AXIS2_PLACEMENT_3D('',#815,#816,#817);
#819=CIRCLE('',#818,12.500000000000000);
#820=EDGE_CURVE('',#807,#805,#819,.T.);
#821=ORIENTED_EDGE('',*,*,#820,.T.);
#822=EDGE_LOOP('',(#814,#821));
#823=FACE_BOUND('',#822,.F.);
#824=CARTESIAN_POINT('',(172.500000000000000,0.0,150.0));
#825=VERTEX_POINT('',#824);
#826=CARTESIAN_POINT('',(147.500000000000000,0.0,150.0));
#827=VERTEX_POINT('',#826);
#828=CARTESIAN_POINT('',(160.0,0.0,150.0));
#829=DIRECTION('',(0.0,1.0,0.0));
#830=DIRECTION('',(1.0,0.0,0.0));
#831=AXIS2_PLACEMENT_3D('',#828,#829,#830);
#832=CIRCLE('',#831,12.500000000000000);
#833=EDGE_CURVE('',#825,#827,#832,.T.);
#834=ORIENTED_EDGE('',*,*,#833,.T.);
#835=CARTESIAN_POINT('',(160.0,0.0,150.0));
#836=DIRECTION('',(0.0,1.0,0.0));
#837=DIRECTION('',(-1.0,0.0,0.0));
#838=AXIS2_PLACEMENT_3D('',#835,#836,#837);
#839=CIRCLE('',#838,12.500000000000000);
#840=EDGE_CURVE('',#827,#825,#839,.T.);
#841=ORIENTED_EDGE('',*,*,#840,.T.);
#842=EDGE_LOOP('',(#834,#841));
#843=FACE_BOUND('',#842,.F.);
#844=CARTESIAN_POINT('',(218.0,0.0,27.500000000000000));
#845=VERTEX_POINT('',#844);
#846=CARTESIAN_POINT('',(192.0,0.0,27.500000000000000));
#847=VERTEX_POINT('',#846);
#848=CARTESIAN_POINT('',(205.0,0.0,27.500000000000000));
#849=DIRECTION('',(0.0,1.0,0.0));
#850=DIRECTION('',(1.0,0.0,0.0));
#851=AXIS2_PLACEMENT_3D('',#848,#849,#850);
#852=CIRCLE('',#851,13.0);
#853=EDGE_CURVE('',#845,#847,#852,.T.);
#854=ORIENTED_EDGE('',*,*,#853,.T.);
#855=CARTESIAN_POINT('',(205.0,0.0,27.500000000000000));
#856=DIRECTION('',(0.0,1.0,0.0));
#857=DIRECTION('',(-1.0,0.0,0.0));
#858=AXIS2_PLACEMENT_3D('',#855,#856,#857);
```

```
#859=CIRCLE('',#858,13.0);
#860=EDGE_CURVE('',#847,#845,#859,.T.);
#861=ORIENTED_EDGE('',*,*,#860,.T.);
#862=EDGE_LOOP('',(#854,#861));
#863=FACE_BOUND('',#862,.F.);
#864=CARTESIAN_POINT('',(218.0,0.0,152.500000000000000));
#865=VERTEX_POINT('',#864);
#866=CARTESIAN_POINT('',(192.0,0.0,152.500000000000000));
#867=VERTEX_POINT('',#866);
#868=CARTESIAN_POINT('',(205.0,0.0,152.500000000000000));
#869=DIRECTION('',(0.0,1.0,0.0));
#870=DIRECTION('',(1.0,0.0,0.0));
#871=AXIS2_PLACEMENT_3D('',#868,#869,#870);
#872=CIRCLE('',#871,13.0);
#873=EDGE_CURVE('',#865,#867,#872,.T.);
#874=ORIENTED_EDGE('',*,*,#873,.T.);
#875=CARTESIAN_POINT('',(205.0,0.0,152.500000000000000));
#876=DIRECTION('',(0.0,1.0,0.0));
#877=DIRECTION('',(-1.0,0.0,0.0));
#878=AXIS2_PLACEMENT_3D('',#875,#876,#877);
#879=CIRCLE('',#878,13.0);
#880=EDGE_CURVE('',#867,#865,#879,.T.);
#881=ORIENTED_EDGE('',*,*,#880,.T.);
#882=EDGE_LOOP('',(#874,#881));
#883=FACE_BOUND('',#882,.F.);
#884=CARTESIAN_POINT('',(38.0,0.0,27.500000000000000));
#885=VERTEX_POINT('',#884);
#886=CARTESIAN_POINT('',(12.0,0.0,27.500000000000000));
#887=VERTEX_POINT('',#886);
#888=CARTESIAN_POINT('',(25.0,0.0,27.500000000000000));
#889=DIRECTION('',(0.0,1.0,0.0));
#890=DIRECTION('',(1.0,0.0,0.0));
#891=AXIS2_PLACEMENT_3D('',#888,#889,#890);
#892=CIRCLE('',#891,13.0);
#893=EDGE_CURVE('',#885,#887,#892,.T.);
#894=ORIENTED_EDGE('',*,*,#893,.T.);
#895=CARTESIAN_POINT('',(25.0,0.0,27.500000000000000));
#896=DIRECTION('',(0.0,1.0,0.0));
#897=DIRECTION('',(-1.0,0.0,0.0));
#898=AXIS2_PLACEMENT_3D('',#895,#896,#897);
#899=CIRCLE('',#898,13.0);
#900=EDGE_CURVE('',#887,#885,#899,.T.);
#901=ORIENTED_EDGE('',*,*,#900,.T.);
#902=EDGE_LOOP('',(#894,#901));
#903=FACE_BOUND('',#902,.F.);
#904=CARTESIAN_POINT('',(38.0,0.0,152.500000000000000));
#905=VERTEX_POINT('',#904);
#906=CARTESIAN_POINT('',(12.0,0.0,152.500000000000000));
#907=VERTEX_POINT('',#906);
#908=CARTESIAN_POINT('',(25.0,0.0,152.500000000000000));
#909=DIRECTION('',(0.0,1.0,0.0));
#910=DIRECTION('',(1.0,0.0,0.0));
#911=AXIS2_PLACEMENT_3D('',#908,#909,#910);
#912=CIRCLE('',#911,13.0);
#913=EDGE_CURVE('',#905,#907,#912,.T.);
#914=ORIENTED_EDGE('',*,*,#913,.T.);
```



```
#915=CARTESIAN_POINT('',(25.0,0.0,152.500000000000000));
#916=DIRECTION('',(0.0,1.0,0.0));
#917=DIRECTION('',(-1.0,0.0,0.0));
#918=AXIS2_PLACEMENT_3D('',#915,#916,#917);
#919=CIRCLE('',#918,13.0);
#920=EDGE_CURVE('',#907,#905,#919,.T.);
#921=ORIENTED_EDGE('',*,*,#920,.T.);
#922=EDGE_LOOP('',(#914,#921));
#923=FACE_BOUND('',#922,.F.);
#924=CARTESIAN_POINT('',(110.0,0.0,85.0));
#925=VERTEX_POINT('',#924);
#926=CARTESIAN_POINT('',(85.0,0.0,85.0));
#927=VERTEX_POINT('',#926);
#928=DIRECTION('',(-1.0,0.0,0.0));
#929=VECTOR('',#928,25.0);
#930=LINE('',#924,#929);
#931=EDGE_CURVE('',#925,#927,#930,.T.);
#932=ORIENTED_EDGE('',*,*,#931,.T.);
#933=CARTESIAN_POINT('',(85.0,0.0,95.0));
#934=VERTEX_POINT('',#933);
#935=CARTESIAN_POINT('',(85.0,0.0,90.0));
#936=DIRECTION('',(0.0,1.0,0.0));
#937=DIRECTION('',(0.0,0.0,-1.0));
#938=AXIS2_PLACEMENT_3D('',#935,#936,#937);
#939=CIRCLE('',#938,5.0);
#940=EDGE_CURVE('',#927,#934,#939,.T.);
#941=ORIENTED_EDGE('',*,*,#940,.T.);
#942=CARTESIAN_POINT('',(110.0,0.0,95.0));
#943=VERTEX_POINT('',#942);
#944=DIRECTION('',(1.0,0.0,0.0));
#945=VECTOR('',#944,25.0);
#946=LINE('',#933,#945);
#947=EDGE_CURVE('',#934,#943,#946,.T.);
#948=ORIENTED_EDGE('',*,*,#947,.T.);
#949=CARTESIAN_POINT('',(110.0,0.0,90.0));
#950=DIRECTION('',(0.0,1.0,0.0));
#951=DIRECTION('',(0.0,0.0,1.0));
#952=AXIS2_PLACEMENT_3D('',#949,#950,#951);
#953=CIRCLE('',#952,5.0);
#954=EDGE_CURVE('',#943,#925,#953,.T.);
#955=ORIENTED_EDGE('',*,*,#954,.T.);
#956=EDGE_LOOP('',(#932,#941,#948,#955));
#957=FACE_BOUND('',#956,.F.);
#958=CARTESIAN_POINT('',(192.568849999999998,0.0,120.0));
#959=VERTEX_POINT('',#958);
#960=CARTESIAN_POINT('',(187.431150000000002,0.0,120.0));
#961=VERTEX_POINT('',#960);
#962=CARTESIAN_POINT('',(190.0,0.0,120.0));
#963=DIRECTION('',(0.0,-1.0,0.0));
#964=DIRECTION('',(1.0,0.0,0.0));
#965=AXIS2_PLACEMENT_3D('',#962,#963,#964);
#966=CIRCLE('',#965,2.568850000000000);
#967=EDGE_CURVE('',#959,#961,#966,.T.);
#968=ORIENTED_EDGE('',*,*,#967,.F.);
#969=CARTESIAN_POINT('',(190.0,0.0,120.0));
#970=DIRECTION('',(0.0,-1.0,0.0));
```

```
#971=DIRECTION('',(-1.0,0.0,0.0));
#972=AXIS2_PLACEMENT_3D('',#969,#970,#971);
#973=CIRCLE('',#972,2.568850000000000);
#974=EDGE_CURVE('',#961,#959,#973,.T.);
#975=ORIENTED_EDGE('',*,*,#974,.F.);
#976=EDGE_LOOP('',(#968,#975));
#977=FACE_BOUND('',#976,.F.);
#978=CARTESIAN_POINT('',(192.568849999999998,0.0,60.0));
#979=VERTEX_POINT('',#978);
#980=CARTESIAN_POINT('',(187.431150000000002,0.0,60.0));
#981=VERTEX_POINT('',#980);
#982=CARTESIAN_POINT('',(190.0,0.0,60.0));
#983=DIRECTION('',(0.0,-1.0,0.0));
#984=DIRECTION('',(1.0,0.0,0.0));
#985=AXIS2_PLACEMENT_3D('',#982,#983,#984);
#986=CIRCLE('',#985,2.568850000000000);
#987=EDGE_CURVE('',#979,#981,#986,.T.);
#988=ORIENTED_EDGE('',*,*,#987,.F.);
#989=CARTESIAN_POINT('',(190.0,0.0,60.0));
#990=DIRECTION('',(0.0,-1.0,0.0));
#991=DIRECTION('',(-1.0,0.0,0.0));
#992=AXIS2_PLACEMENT_3D('',#989,#990,#991);
#993=CIRCLE('',#992,2.568850000000000);
#994=EDGE_CURVE('',#981,#979,#993,.T.);
#995=ORIENTED_EDGE('',*,*,#994,.F.);
#996=EDGE_LOOP('',(#988,#995));
#997=FACE_BOUND('',#996,.F.);
#998=CARTESIAN_POINT('',(132.568849999999998,0.0,120.0));
#999=VERTEX_POINT('',#998);
#1000=CARTESIAN_POINT('',(127.431150000000002,0.0,120.0));
#1001=VERTEX_POINT('',#1000);
#1002=CARTESIAN_POINT('',(130.0,0.0,120.0));
#1003=DIRECTION('',(0.0,-1.0,0.0));
#1004=DIRECTION('',(1.0,0.0,0.0));
#1005=AXIS2_PLACEMENT_3D('',#1002,#1003,#1004);
#1006=CIRCLE('',#1005,2.568850000000000);
#1007=EDGE_CURVE('',#999,#1001,#1006,.T.);
#1008=ORIENTED_EDGE('',*,*,#1007,.F.);
#1009=CARTESIAN_POINT('',(130.0,0.0,120.0));
#1010=DIRECTION('',(0.0,-1.0,0.0));
#1011=DIRECTION('',(-1.0,0.0,0.0));
#1012=AXIS2_PLACEMENT_3D('',#1009,#1010,#1011);
#1013=CIRCLE('',#1012,2.568850000000000);
#1014=EDGE_CURVE('',#1001,#999,#1013,.T.);
#1015=ORIENTED_EDGE('',*,*,#1014,.F.);
#1016=EDGE_LOOP('',(#1008,#1015));
#1017=FACE_BOUND('',#1016,.F.);
#1018=CARTESIAN_POINT('',(132.568849999999998,0.0,60.0));
#1019=VERTEX_POINT('',#1018);
#1020=CARTESIAN_POINT('',(127.431150000000002,0.0,60.0));
#1021=VERTEX_POINT('',#1020);
#1022=CARTESIAN_POINT('',(130.0,0.0,60.0));
#1023=DIRECTION('',(0.0,-1.0,0.0));
#1024=DIRECTION('',(1.0,0.0,0.0));
#1025=AXIS2_PLACEMENT_3D('',#1022,#1023,#1024);
#1026=CIRCLE('',#1025,2.568850000000000);
```



```
#1027=EDGE_CURVE('',#1019,#1021,#1026,.T.);
#1028=ORIENTED_EDGE('',*,*,#1027,.F.);
#1029=CARTESIAN_POINT('',(130.0,0.0,60.0));
#1030=DIRECTION('',(0.0,-1.0,0.0));
#1031=DIRECTION('',(-1.0,0.0,0.0));
#1032=AXIS2_PLACEMENT_3D('',#1029,#1030,#1031);
#1033=CIRCLE('',#1032,2.568850000000000);
#1034=EDGE_CURVE('',#1021,#1019,#1033,.T.);
#1035=ORIENTED_EDGE('',*,*,#1034,.F.);
#1036=EDGE_LOOP('',(#1028,#1035));
#1037=FACE_BOUND('',#1036,.F.);
#1038=CARTESIAN_POINT('',(100.0,0.0,10.0));
#1039=VERTEX_POINT('',#1038);
#1040=CARTESIAN_POINT('',(60.0,0.0,10.0));
#1041=VERTEX_POINT('',#1040);
#1042=DIRECTION('',(-1.0,0.0,0.0));
#1043=VECTOR('',#1042,40.0);
#1044=LINE('',#1038,#1043);
#1045=EDGE_CURVE('',#1039,#1041,#1044,.T.);
#1046=ORIENTED_EDGE('',*,*,#1045,.T.);
#1047=CARTESIAN_POINT('',(50.0,0.0,20.0));
#1048=VERTEX_POINT('',#1047);
#1049=CARTESIAN_POINT('',(59.999999999999993,1.421085E-14,20.0));
#1050=DIRECTION('',(5.532395E-16,-1.0,0.0));
#1051=DIRECTION('',(-1.0,-5.532395E-16,0.0));
#1052=AXIS2_PLACEMENT_3D('',#1049,#1050,#1051);
#1053=CIRCLE('',#1052,10.0);
#1054=EDGE_CURVE('',#1048,#1041,#1053,.T.);
#1055=ORIENTED_EDGE('',*,*,#1054,.F.);
#1056=CARTESIAN_POINT('',(50.0,0.0,50.0));
#1057=VERTEX_POINT('',#1056);
#1058=DIRECTION('',(0.0,0.0,-1.0));
#1059=VECTOR('',#1058,30.0);
#1060=LINE('',#1056,#1059);
#1061=EDGE_CURVE('',#1057,#1048,#1060,.T.);
#1062=ORIENTED_EDGE('',*,*,#1061,.F.);
#1063=CARTESIAN_POINT('',(60.0,0.0,60.0));
#1064=VERTEX_POINT('',#1063);
#1065=CARTESIAN_POINT('',(59.999999999999993,3.552714E-15,50.0));
#1066=DIRECTION('',(5.533360E-16,-1.0,1.261554E-33));
#1067=DIRECTION('',(7.105427E-16,3.944305E-31,1.0));
#1068=AXIS2_PLACEMENT_3D('',#1065,#1066,#1067);
#1069=CIRCLE('',#1068,10.0);
#1070=EDGE_CURVE('',#1064,#1057,#1069,.T.);
#1071=ORIENTED_EDGE('',*,*,#1070,.F.);
#1072=CARTESIAN_POINT('',(100.0,0.0,60.0));
#1073=VERTEX_POINT('',#1072);
#1074=DIRECTION('',(-1.0,0.0,0.0));
#1075=VECTOR('',#1074,40.0);
#1076=LINE('',#1072,#1075);
#1077=EDGE_CURVE('',#1073,#1064,#1076,.T.);
#1078=ORIENTED_EDGE('',*,*,#1077,.F.);
#1079=CARTESIAN_POINT('',(110.0,0.0,50.0));
#1080=VERTEX_POINT('',#1079);
#1081=CARTESIAN_POINT('',(99.999999999999986,-1.387779E-17,50.0));
#1082=DIRECTION('',(0.0,1.0,0.0));
```

```
#1083=DIRECTION('',(1.421085E-15,0.0,1.0));
#1084=AXIS2_PLACEMENT_3D('',#1081,#1082,#1083);
#1085=CIRCLE('',#1084,10.0);
#1086=EDGE_CURVE('',#1073,#1080,#1085,.T.);
#1087=ORIENTED_EDGE('',*,*,#1086,.T.);
#1088=CARTESIAN_POINT('',(110.0,0.0,20.0));
#1089=VERTEX_POINT('',#1088);
#1090=DIRECTION('',(0.0,0.0,-1.0));
#1091=VECTOR('',#1090,30.0);
#1092=LINE('',#1079,#1091);
#1093=EDGE_CURVE('',#1080,#1089,#1092,.T.);
#1094=ORIENTED_EDGE('',*,*,#1093,.T.);
#1095=CARTESIAN_POINT('',(99.99999999999986,4.163336E-17,20.0));
#1096=DIRECTION('',(0.0,1.0,0.0));
#1097=DIRECTION('',(1.0,0.0,0.0));
#1098=AXIS2_PLACEMENT_3D('',#1095,#1096,#1097);
#1099=CIRCLE('',#1098,10.0);
#1100=EDGE_CURVE('',#1089,#1039,#1099,.T.);
#1101=ORIENTED_EDGE('',*,*,#1100,.T.);
#1102=EDGE_LOOP('',(#1046,#1055,#1062,#1071,#1078,#1087,#1094,#1101));
#1103=FACE_BOUND('',#1102,.F.);
#1104=CARTESIAN_POINT('',(260.0,0.0,0.0));
#1105=DIRECTION('',(0.0,1.0,0.0));
#1106=DIRECTION('',(-1.0,0.0,0.0));
#1107=AXIS2_PLACEMENT_3D('',#1104,#1105,#1106);
#1108=PLANE('',#1107);
#1109=ADVANCED_FACE('',(#783,#803,#823,#843,#863,#883,#903,#923,#957,#977,#99
7,#1017,#1037,#1103),#1108,.T.);
#1110=SHAPE_ASPECT('featured shape: face 9',$,#38,.T.);
#1111=PROPERTY_DEFINITION('',$,#1110);
#1112=FACE_SHAPE_REPRESENTATION('',(#1109),#29);
#1113=PROPERTY_DEFINITION_REPRESENTATION(#1111,#1112);
#1114=CARTESIAN_POINT('',(69.999999999999972,0.0,145.0));
#1115=VERTEX_POINT('',#1114);
#1116=CARTESIAN_POINT('',(95.999999999999972,0.0,145.0));
#1117=VERTEX_POINT('',#1116);
#1118=DIRECTION('',(1.0,0.0,0.0));
#1119=VECTOR('',#1118,26.0);
#1120=LINE('',#1114,#1119);
#1121=EDGE_CURVE('',#1115,#1117,#1120,.T.);
#1122=ORIENTED_EDGE('',*,*,#1121,.T.);
#1123=CARTESIAN_POINT('',(97.999999999999972,0.0,147.0));
#1124=VERTEX_POINT('',#1123);
#1125=CARTESIAN_POINT('',(95.999999999999972,0.0,147.0));
#1126=DIRECTION('',(0.0,1.0,0.0));
#1127=DIRECTION('',(1.0,0.0,0.0));
#1128=AXIS2_PLACEMENT_3D('',#1125,#1126,#1127);
#1129=CIRCLE('',#1128,2.0);
#1130=EDGE_CURVE('',#1124,#1117,#1129,.T.);
#1131=ORIENTED_EDGE('',*,*,#1130,.F.);
#1132=CARTESIAN_POINT('',(97.999999999999972,0.0,163.0));
#1133=VERTEX_POINT('',#1132);
#1134=DIRECTION('',(0.0,0.0,1.0));
#1135=VECTOR('',#1134,16.0);
#1136=LINE('',#1123,#1135);
#1137=EDGE_CURVE('',#1124,#1133,#1136,.T.);
```



```
#1138=ORIENTED_EDGE('',*,*,#1137,.T.);
#1139=CARTESIAN_POINT('',(95.999999999999972,0.0,165.0));
#1140=VERTEX_POINT('',#1139);
#1141=CARTESIAN_POINT('',(95.999999999999972,0.0,163.0));
#1142=DIRECTION('',(0.0,1.0,0.0));
#1143=DIRECTION('',(0.0,0.0,1.0));
#1144=AXIS2_PLACEMENT_3D('',#1141,#1142,#1143);
#1145=CIRCLE('',#1144,2.0);
#1146=EDGE_CURVE('',#1140,#1133,#1145,.T.);
#1147=ORIENTED_EDGE('',*,*,#1146,.F.);
#1148=CARTESIAN_POINT('',(69.999999999999972,0.0,165.0));
#1149=VERTEX_POINT('',#1148);
#1150=DIRECTION('',(-1.0,0.0,0.0));
#1151=VECTOR('',#1150,26.0);
#1152=LINE('',#1139,#1151);
#1153=EDGE_CURVE('',#1140,#1149,#1152,.T.);
#1154=ORIENTED_EDGE('',*,*,#1153,.T.);
#1155=CARTESIAN_POINT('',(67.999999999999972,0.0,163.0));
#1156=VERTEX_POINT('',#1155);
#1157=CARTESIAN_POINT('',(69.999999999999972,0.0,163.0));
#1158=DIRECTION('',(0.0,1.0,0.0));
#1159=DIRECTION('',(-1.0,0.0,0.0));
#1160=AXIS2_PLACEMENT_3D('',#1157,#1158,#1159);
#1161=CIRCLE('',#1160,2.0);
#1162=EDGE_CURVE('',#1156,#1149,#1161,.T.);
#1163=ORIENTED_EDGE('',*,*,#1162,.F.);
#1164=CARTESIAN_POINT('',(67.999999999999972,0.0,147.0));
#1165=VERTEX_POINT('',#1164);
#1166=DIRECTION('',(0.0,0.0,-1.0));
#1167=VECTOR('',#1166,16.0);
#1168=LINE('',#1155,#1167);
#1169=EDGE_CURVE('',#1156,#1165,#1168,.T.);
#1170=ORIENTED_EDGE('',*,*,#1169,.T.);
#1171=CARTESIAN_POINT('',(69.999999999999972,0.0,147.0));
#1172=DIRECTION('',(0.0,1.0,0.0));
#1173=DIRECTION('',(0.0,0.0,-1.0));
#1174=AXIS2_PLACEMENT_3D('',#1171,#1172,#1173);
#1175=CIRCLE('',#1174,2.0);
#1176=EDGE_CURVE('',#1115,#1165,#1175,.T.);
#1177=ORIENTED_EDGE('',*,*,#1176,.F.);
#1178=EDGE_LOOP('',(#1122,#1131,#1138,#1147,#1154,#1163,#1170,#1177));
#1179=FACE_BOUND('',#1178,.F.);
#1180=ADVANCED_FACE('',(#1179),#1108,.T.);
#1181=SHAPE_ASPECT('featured shape: face 10',$,#38,.T.);
#1182=PROPERTY_DEFINITION('',$,#1181);
#1183=FACE_SHAPE_REPRESENTATION('',(#1180),#29);
#1184=PROPERTY_DEFINITION_REPRESENTATION(#1182,#1183);
#1185=CARTESIAN_POINT('',(230.0,-10.0,95.0));
#1186=VERTEX_POINT('',#1185);
#1187=DIRECTION('',(0.0,-1.0,0.0));
#1188=VECTOR('',#1187,10.0);
#1189=LINE('',#760,#1188);
#1190=EDGE_CURVE('',#761,#1186,#1189,.T.);
#1191=ORIENTED_EDGE('',*,*,#1190,.T.);
#1192=CARTESIAN_POINT('',(230.0,-10.0,85.0));
#1193=VERTEX_POINT('',#1192);
```

```
#1194=DIRECTION('',(0.0,0.0,-1.0));
#1195=VECTOR('',#1194,10.0);
#1196=LINE('',#1185,#1195);
#1197=EDGE_CURVE('',#1186,#1193,#1196,.T.);
#1198=ORIENTED_EDGE('',*,*,#1197,.T.);
#1199=DIRECTION('',(0.0,-1.0,0.0));
#1200=VECTOR('',#1199,10.0);
#1201=LINE('',#737,#1200);
#1202=EDGE_CURVE('',#738,#1193,#1201,.T.);
#1203=ORIENTED_EDGE('',*,*,#1202,.F.);
#1204=ORIENTED_EDGE('',*,*,#742,.F.);
#1205=CARTESIAN_POINT('',(230.0,-40.0,5.0));
#1206=VERTEX_POINT('',#1205);
#1207=DIRECTION('',(0.0,-1.0,0.0));
#1208=VECTOR('',#1207,40.0);
#1209=LINE('',#728,#1208);
#1210=EDGE_CURVE('',#729,#1206,#1209,.T.);
#1211=ORIENTED_EDGE('',*,*,#1210,.T.);
#1212=CARTESIAN_POINT('',(230.0,-40.0,40.0));
#1213=VERTEX_POINT('',#1212);
#1214=DIRECTION('',(0.0,0.0,1.0));
#1215=VECTOR('',#1214,35.0);
#1216=LINE('',#1205,#1215);
#1217=EDGE_CURVE('',#1206,#1213,#1216,.T.);
#1218=ORIENTED_EDGE('',*,*,#1217,.T.);
#1219=CARTESIAN_POINT('',(230.0,-20.0,40.0));
#1220=VERTEX_POINT('',#1219);
#1221=DIRECTION('',(0.0,1.0,0.0));
#1222=VECTOR('',#1221,20.0);
#1223=LINE('',#1212,#1222);
#1224=EDGE_CURVE('',#1213,#1220,#1223,.T.);
#1225=ORIENTED_EDGE('',*,*,#1224,.T.);
#1226=CARTESIAN_POINT('',(230.0,-20.0,140.000000000000028));
#1227=VERTEX_POINT('',#1226);
#1228=DIRECTION('',(0.0,0.0,1.0));
#1229=VECTOR('',#1228,100.000000000000028);
#1230=LINE('',#1219,#1229);
#1231=EDGE_CURVE('',#1220,#1227,#1230,.T.);
#1232=ORIENTED_EDGE('',*,*,#1231,.T.);
#1233=CARTESIAN_POINT('',(230.0,-40.0,140.0));
#1234=VERTEX_POINT('',#1233);
#1235=DIRECTION('',(0.0,-1.0,0.0));
#1236=VECTOR('',#1235,20.0);
#1237=LINE('',#1226,#1236);
#1238=EDGE_CURVE('',#1227,#1234,#1237,.T.);
#1239=ORIENTED_EDGE('',*,*,#1238,.T.);
#1240=CARTESIAN_POINT('',(230.0,-40.0,175.0));
#1241=VERTEX_POINT('',#1240);
#1242=DIRECTION('',(0.0,0.0,1.0));
#1243=VECTOR('',#1242,35.0);
#1244=LINE('',#1233,#1243);
#1245=EDGE_CURVE('',#1234,#1241,#1244,.T.);
#1246=ORIENTED_EDGE('',*,*,#1245,.T.);
#1247=DIRECTION('',(0.0,1.0,0.0));
#1248=VECTOR('',#1247,40.0);
#1249=LINE('',#1240,#1248);
```



```
#1250=EDGE_CURVE('',#1241,#768,#1249,.T.);
#1251=ORIENTED_EDGE('',*,*,#1250,.T.);
#1252=ORIENTED_EDGE('',*,*,#772,.F.);
#1253=EDGE_LOOP('',(#1191,#1198,#1203,#1204,#1211,#1218,#1225,#1232,#1239,#12
46,#1251,#1252));
#1254=FACE_BOUND('',#1253,.F.);
#1255=CARTESIAN_POINT('',(230.0,0.0,0.0));
#1256=DIRECTION('',(-1.0,0.0,0.0));
#1257=DIRECTION('',(0.0,-1.0,0.0));
#1258=AXIS2_PLACEMENT_3D('',#1255,#1256,#1257);
#1259=PLANE('',#1258);
#1260=ADVANCED_FACE('',(#1254),#1259,.F.);
#1261=SHAPE_ASPECT('featured shape: face 11',$,#38,.T.);
#1262=PROPERTY_DEFINITION('',$,#1261);
#1263=FACE_SHAPE_REPRESENTATION('',(#1260),#29);
#1264=PROPERTY_DEFINITION_REPRESENTATION(#1262,#1263);
#1265=CARTESIAN_POINT('',(240.0,-20.0,50.0));
#1266=VERTEX_POINT('',#1265);
#1267=CARTESIAN_POINT('',(240.0,-20.0,40.0));
#1268=DIRECTION('',(0.0,1.0,0.0));
#1269=DIRECTION('',(-1.0,0.0,0.0));
#1270=AXIS2_PLACEMENT_3D('',#1267,#1268,#1269);
#1271=CIRCLE('',#1270,10.0);
#1272=EDGE_CURVE('',#1220,#1266,#1271,.T.);
#1273=ORIENTED_EDGE('',*,*,#1272,.T.);
#1274=CARTESIAN_POINT('',(255.0,-20.0,50.0));
#1275=VERTEX_POINT('',#1274);
#1276=DIRECTION('',(-1.0,0.0,0.0));
#1277=VECTOR('',#1276,15.0);
#1278=LINE('',#1274,#1277);
#1279=EDGE_CURVE('',#1275,#1266,#1278,.T.);
#1280=ORIENTED_EDGE('',*,*,#1279,.F.);
#1281=CARTESIAN_POINT('',(255.0,-20.0,55.0));
#1282=DIRECTION('',(0.0,1.0,0.0));
#1283=DIRECTION('',(1.0,0.0,0.0));
#1284=AXIS2_PLACEMENT_3D('',#1281,#1282,#1283);
#1285=CIRCLE('',#1284,5.0);
#1286=EDGE_CURVE('',#615,#1275,#1285,.T.);
#1287=ORIENTED_EDGE('',*,*,#1286,.F.);
#1288=ORIENTED_EDGE('',*,*,#638,.T.);
#1289=CARTESIAN_POINT('',(255.0,-20.0,130.0));
#1290=VERTEX_POINT('',#1289);
#1291=CARTESIAN_POINT('',(255.0,-20.0,125.0));
#1292=DIRECTION('',(0.0,-1.0,0.0));
#1293=DIRECTION('',(1.0,0.0,0.0));
#1294=AXIS2_PLACEMENT_3D('',#1291,#1292,#1293);
#1295=CIRCLE('',#1294,5.0);
#1296=EDGE_CURVE('',#629,#1290,#1295,.T.);
#1297=ORIENTED_EDGE('',*,*,#1296,.T.);
#1298=CARTESIAN_POINT('',(240.0,-20.0,130.0));
#1299=VERTEX_POINT('',#1298);
#1300=DIRECTION('',(-1.0,0.0,0.0));
#1301=VECTOR('',#1300,15.0);
#1302=LINE('',#1289,#1301);
#1303=EDGE_CURVE('',#1290,#1299,#1302,.T.);
#1304=ORIENTED_EDGE('',*,*,#1303,.T.);
```

```
#1305=CARTESIAN_POINT('',(240.0,-20.0,140.0));
#1306=DIRECTION('',(0.0,-1.0,0.0));
#1307=DIRECTION('',(-1.0,0.0,2.842171E-15));
#1308=AXIS2_PLACEMENT_3D('',#1305,#1306,#1307);
#1309=CIRCLE('',#1308,10.0);
#1310=EDGE_CURVE('',#1227,#1299,#1309,.T.);
#1311=ORIENTED_EDGE('',*,*,#1310,.F.);
#1312=ORIENTED_EDGE('',*,*,#1231,.F.);
#1313=EDGE_LOOP('',(#1273,#1280,#1287,#1288,#1297,#1304,#1311,#1312));
#1314=FACE_BOUND('',#1313,.F.);
#1315=CARTESIAN_POINT('',(250.0,-20.0,110.0));
#1316=VERTEX_POINT('',#1315);
#1317=CARTESIAN_POINT('',(240.0,-20.0,110.0));
#1318=VERTEX_POINT('',#1317);
#1319=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#1320=DIRECTION('',(0.0,-1.0,0.0));
#1321=DIRECTION('',(1.0,0.0,0.0));
#1322=AXIS2_PLACEMENT_3D('',#1319,#1320,#1321);
#1323=CIRCLE('',#1322,5.0);
#1324=EDGE_CURVE('',#1316,#1318,#1323,.T.);
#1325=ORIENTED_EDGE('',*,*,#1324,.F.);
#1326=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#1327=DIRECTION('',(0.0,-1.0,0.0));
#1328=DIRECTION('',(-1.0,0.0,0.0));
#1329=AXIS2_PLACEMENT_3D('',#1326,#1327,#1328);
#1330=CIRCLE('',#1329,5.0);
#1331=EDGE_CURVE('',#1318,#1316,#1330,.T.);
#1332=ORIENTED_EDGE('',*,*,#1331,.F.);
#1333=EDGE_LOOP('',(#1325,#1332));
#1334=FACE_BOUND('',#1333,.F.);
#1335=CARTESIAN_POINT('',(250.0,-20.0,70.0));
#1336=VERTEX_POINT('',#1335);
#1337=CARTESIAN_POINT('',(240.0,-20.0,70.0));
#1338=VERTEX_POINT('',#1337);
#1339=CARTESIAN_POINT('',(245.0,-20.0,70.0));
#1340=DIRECTION('',(0.0,-1.0,0.0));
#1341=DIRECTION('',(1.0,0.0,0.0));
#1342=AXIS2_PLACEMENT_3D('',#1339,#1340,#1341);
#1343=CIRCLE('',#1342,5.0);
#1344=EDGE_CURVE('',#1336,#1338,#1343,.T.);
#1345=ORIENTED_EDGE('',*,*,#1344,.F.);
#1346=CARTESIAN_POINT('',(245.0,-20.0,70.0));
#1347=DIRECTION('',(0.0,-1.0,0.0));
#1348=DIRECTION('',(-1.0,0.0,0.0));
#1349=AXIS2_PLACEMENT_3D('',#1346,#1347,#1348);
#1350=CIRCLE('',#1349,5.0);
#1351=EDGE_CURVE('',#1338,#1336,#1350,.T.);
#1352=ORIENTED_EDGE('',*,*,#1351,.F.);
#1353=EDGE_LOOP('',(#1345,#1352));
#1354=FACE_BOUND('',#1353,.F.);
#1355=CARTESIAN_POINT('',(230.0,-20.0,0.0));
#1356=DIRECTION('',(0.0,-1.0,0.0));
#1357=DIRECTION('',(1.0,0.0,0.0));
#1358=AXIS2_PLACEMENT_3D('',#1355,#1356,#1357);
#1359=PLANE('',#1358);
#1360=ADVANCED_FACE('',(#1314,#1334,#1354),#1359,.F.);
```

```
#1361=SHAPE_ASPECT('featured shape: face 12',$,#38,.T.);
#1362=PROPERTY_DEFINITION('',$,#1361);
#1363=FACE_SHAPE_REPRESENTATION('',(#1360),#29);
#1364=PROPERTY_DEFINITION_REPRESENTATION(#1362,#1363);
#1365=ORIENTED_EDGE('',*,*,#46,.T.);
#1366=CARTESIAN_POINT('',(5.0,-40.0,5.0));
#1367=DIRECTION('',(0.0,1.0,0.0));
#1368=DIRECTION('',(0.0,0.0,-1.0));
#1369=AXIS2_PLACEMENT_3D('',#1366,#1367,#1368);
#1370=CIRCLE('',#1369,5.0);
#1371=EDGE_CURVE('',#42,#185,#1370,.T.);
#1372=ORIENTED_EDGE('',*,*,#1371,.T.);
#1373=ORIENTED_EDGE('',*,*,#189,.T.);
#1374=CARTESIAN_POINT('',(5.0,-40.0,175.0));
#1375=DIRECTION('',(0.0,-1.0,0.0));
#1376=DIRECTION('',(0.0,0.0,1.0));
#1377=AXIS2_PLACEMENT_3D('',#1374,#1375,#1376);
#1378=CIRCLE('',#1377,5.0);
#1379=EDGE_CURVE('',#110,#178,#1378,.T.);
#1380=ORIENTED_EDGE('',*,*,#1379,.F.);
#1381=ORIENTED_EDGE('',*,*,#114,.T.);
#1382=CARTESIAN_POINT('',(225.0,-40.0,175.0));
#1383=DIRECTION('',(0.0,-1.0,0.0));
#1384=DIRECTION('',(1.0,0.0,0.0));
#1385=AXIS2_PLACEMENT_3D('',#1382,#1383,#1384);
#1386=CIRCLE('',#1385,5.0);
#1387=EDGE_CURVE('',#1241,#103,#1386,.T.);
#1388=ORIENTED_EDGE('',*,*,#1387,.F.);
#1389=ORIENTED_EDGE('',*,*,#1245,.F.);
#1390=CARTESIAN_POINT('',(240.0,-40.0,130.0));
#1391=VERTEX_POINT('',#1390);
#1392=CARTESIAN_POINT('',(240.0,-40.0,140.0));
#1393=DIRECTION('',(0.0,-1.0,0.0));
#1394=DIRECTION('',(-1.0,0.0,5.684342E-15));
#1395=AXIS2_PLACEMENT_3D('',#1392,#1393,#1394);
#1396=CIRCLE('',#1395,10.0);
#1397=EDGE_CURVE('',#1234,#1391,#1396,.T.);
#1398=ORIENTED_EDGE('',*,*,#1397,.T.);
#1399=CARTESIAN_POINT('',(255.0,-40.0,130.0));
#1400=VERTEX_POINT('',#1399);
#1401=DIRECTION('',(-1.0,0.0,0.0));
#1402=VECTOR('',#1401,15.0);
#1403=LINE('',#1399,#1402);
#1404=EDGE_CURVE('',#1400,#1391,#1403,.T.);
#1405=ORIENTED_EDGE('',*,*,#1404,.F.);
#1406=CARTESIAN_POINT('',(255.0,-40.0,125.0));
#1407=DIRECTION('',(0.0,-1.0,0.0));
#1408=DIRECTION('',(1.0,0.0,0.0));
#1409=AXIS2_PLACEMENT_3D('',#1406,#1407,#1408);
#1410=CIRCLE('',#1409,5.0);
#1411=EDGE_CURVE('',#622,#1400,#1410,.T.);
#1412=ORIENTED_EDGE('',*,*,#1411,.F.);
#1413=ORIENTED_EDGE('',*,*,#626,.F.);
#1414=CARTESIAN_POINT('',(255.0,-40.0,50.0));
#1415=VERTEX_POINT('',#1414);
#1416=CARTESIAN_POINT('',(255.0,-40.0,55.0));
```

```
#1417=DIRECTION('',(0.0,1.0,0.0));
#1418=DIRECTION('',(1.0,0.0,0.0));
#1419=AXIS2_PLACEMENT_3D('',#1416,#1417,#1418);
#1420=CIRCLE('',#1419,5.0);
#1421=EDGE_CURVE('',#613,#1415,#1420,.T.);
#1422=ORIENTED_EDGE('',*,*,#1421,.T.);
#1423=CARTESIAN_POINT('',(240.0,-40.0,50.0));
#1424=VERTEX_POINT('',#1423);
#1425=DIRECTION('',(-1.0,0.0,0.0));
#1426=VECTOR('',#1425,15.0);
#1427=LINE('',#1414,#1426);
#1428=EDGE_CURVE('',#1415,#1424,#1427,.T.);
#1429=ORIENTED_EDGE('',*,*,#1428,.T.);
#1430=CARTESIAN_POINT('',(240.0,-40.0,40.0));
#1431=DIRECTION('',(0.0,1.0,0.0));
#1432=DIRECTION('',(-1.0,0.0,-5.684342E-15));
#1433=AXIS2_PLACEMENT_3D('',#1430,#1431,#1432);
#1434=CIRCLE('',#1433,10.0);
#1435=EDGE_CURVE('',#1213,#1424,#1434,.T.);
#1436=ORIENTED_EDGE('',*,*,#1435,.F.);
#1437=ORIENTED_EDGE('',*,*,#1217,.F.);
#1438=CARTESIAN_POINT('',(225.0,-40.0,5.0));
#1439=DIRECTION('',(0.0,1.0,0.0));
#1440=DIRECTION('',(1.0,0.0,0.0));
#1441=AXIS2_PLACEMENT_3D('',#1438,#1439,#1440);
#1442=CIRCLE('',#1441,5.0);
#1443=EDGE_CURVE('',#1206,#40,#1442,.T.);
#1444=ORIENTED_EDGE('',*,*,#1443,.T.);
#1445=EDGE_LOOP('',(#1365,#1372,#1373,#1380,#1381,#1388,#1389,#1398,#1405,#14
12,#1413,#1422,#1429,#1436,#1437,#1444));
#1446=FACE_BOUND('',#1445,.F.);
#1447=CARTESIAN_POINT('',(45.0,-40.0,2.0));
#1448=VERTEX_POINT('',#1447);
#1449=CARTESIAN_POINT('',(7.0,-40.0,2.0));
#1450=VERTEX_POINT('',#1449);
#1451=DIRECTION('',(-1.0,0.0,0.0));
#1452=VECTOR('',#1451,38.0);
#1453=LINE('',#1447,#1452);
#1454=EDGE_CURVE('',#1448,#1450,#1453,.T.);
#1455=ORIENTED_EDGE('',*,*,#1454,.F.);
#1456=CARTESIAN_POINT('',(50.0,-40.0,7.0));
#1457=VERTEX_POINT('',#1456);
#1458=CARTESIAN_POINT('',(45.0,-40.0,7.0));
#1459=DIRECTION('',(0.0,-1.0,0.0));
#1460=DIRECTION('',(0.0,0.0,-1.0));
#1461=AXIS2_PLACEMENT_3D('',#1458,#1459,#1460);
#1462=CIRCLE('',#1461,5.0);
#1463=EDGE_CURVE('',#1448,#1457,#1462,.T.);
#1464=ORIENTED_EDGE('',*,*,#1463,.T.);
#1465=CARTESIAN_POINT('',(50.0,-40.0,45.0));
#1466=VERTEX_POINT('',#1465);
#1467=DIRECTION('',(0.0,0.0,-1.0));
#1468=VECTOR('',#1467,38.0);
#1469=LINE('',#1465,#1468);
#1470=EDGE_CURVE('',#1466,#1457,#1469,.T.);
#1471=ORIENTED_EDGE('',*,*,#1470,.F.);
```



```
#1472=CARTESIAN_POINT('',(45.0,-40.0,50.0));
#1473=VERTEX_POINT('',#1472);
#1474=CARTESIAN_POINT('',(45.0,-40.0,45.000000000000007));
#1475=DIRECTION('',(0.0,1.0,0.0));
#1476=DIRECTION('',(0.0,0.0,1.0));
#1477=AXIS2_PLACEMENT_3D('',#1474,#1475,#1476);
#1478=CIRCLE('',#1477,5.0);
#1479=EDGE_CURVE('',#1473,#1466,#1478,.T.);
#1480=ORIENTED_EDGE('',*,*,#1479,.F.);
#1481=CARTESIAN_POINT('',(7.0,-40.0,50.0));
#1482=VERTEX_POINT('',#1481);
#1483=DIRECTION('',(1.0,0.0,0.0));
#1484=VECTOR('',#1483,38.0);
#1485=LINE('',#1481,#1484);
#1486=EDGE_CURVE('',#1482,#1473,#1485,.T.);
#1487=ORIENTED_EDGE('',*,*,#1486,.F.);
#1488=CARTESIAN_POINT('',(2.0,-40.0,45.0));
#1489=VERTEX_POINT('',#1488);
#1490=CARTESIAN_POINT('',(7.0,-40.0,45.000000000000007));
#1491=DIRECTION('',(0.0,1.0,0.0));
#1492=DIRECTION('',(-1.0,0.0,-1.421085E-15));
#1493=AXIS2_PLACEMENT_3D('',#1490,#1491,#1492);
#1494=CIRCLE('',#1493,5.0);
#1495=EDGE_CURVE('',#1489,#1482,#1494,.T.);
#1496=ORIENTED_EDGE('',*,*,#1495,.F.);
#1497=CARTESIAN_POINT('',(2.0,-40.0,7.0));
#1498=VERTEX_POINT('',#1497);
#1499=DIRECTION('',(0.0,0.0,1.0));
#1500=VECTOR('',#1499,38.0);
#1501=LINE('',#1497,#1500);
#1502=EDGE_CURVE('',#1498,#1489,#1501,.T.);
#1503=ORIENTED_EDGE('',*,*,#1502,.F.);
#1504=CARTESIAN_POINT('',(7.0,-40.0,7.0));
#1505=DIRECTION('',(0.0,-1.0,0.0));
#1506=DIRECTION('',(-1.0,0.0,0.0));
#1507=AXIS2_PLACEMENT_3D('',#1504,#1505,#1506);
#1508=CIRCLE('',#1507,5.0);
#1509=EDGE_CURVE('',#1498,#1450,#1508,.T.);
#1510=ORIENTED_EDGE('',*,*,#1509,.T.);
#1511=EDGE_LOOP('',(#1455,#1464,#1471,#1480,#1487,#1496,#1503,#1510));
#1512=FACE_BOUND('',#1511,.F.);
#1513=CARTESIAN_POINT('',(228.0,-40.0,45.0));
#1514=VERTEX_POINT('',#1513);
#1515=CARTESIAN_POINT('',(228.0,-40.0,7.0));
#1516=VERTEX_POINT('',#1515);
#1517=DIRECTION('',(0.0,0.0,-1.0));
#1518=VECTOR('',#1517,38.0);
#1519=LINE('',#1513,#1518);
#1520=EDGE_CURVE('',#1514,#1516,#1519,.T.);
#1521=ORIENTED_EDGE('',*,*,#1520,.F.);
#1522=CARTESIAN_POINT('',(223.0,-40.0,50.0));
#1523=VERTEX_POINT('',#1522);
#1524=CARTESIAN_POINT('',(223.0,-40.0,45.0));
#1525=DIRECTION('',(0.0,-1.0,0.0));
#1526=DIRECTION('',(1.0,0.0,0.0));
#1527=AXIS2_PLACEMENT_3D('',#1524,#1525,#1526);
```



```
#1528=CIRCLE('',#1527,5.0);
#1529=EDGE_CURVE('',#1514,#1523,#1528,.T.);
#1530=ORIENTED_EDGE('',*,*,#1529,.T.);
#1531=CARTESIAN_POINT('',(185.000000000000028,-40.0,50.0));
#1532=VERTEX_POINT('',#1531);
#1533=DIRECTION('',(1.0,0.0,0.0));
#1534=VECTOR('',#1533,37.999999999999972);
#1535=LINE('',#1531,#1534);
#1536=EDGE_CURVE('',#1532,#1523,#1535,.T.);
#1537=ORIENTED_EDGE('',*,*,#1536,.F.);
#1538=CARTESIAN_POINT('',(180.000000000000028,-40.0,45.0));
#1539=VERTEX_POINT('',#1538);
#1540=CARTESIAN_POINT('',(185.000000000000028,-40.0,45.0));
#1541=DIRECTION('',(0.0,1.0,0.0));
#1542=DIRECTION('',(-1.0,0.0,0.0));
#1543=AXIS2_PLACEMENT_3D('',#1540,#1541,#1542);
#1544=CIRCLE('',#1543,5.0);
#1545=EDGE_CURVE('',#1539,#1532,#1544,.T.);
#1546=ORIENTED_EDGE('',*,*,#1545,.F.);
#1547=CARTESIAN_POINT('',(180.000000000000028,-40.0,7.0));
#1548=VERTEX_POINT('',#1547);
#1549=DIRECTION('',(0.0,0.0,1.0));
#1550=VECTOR('',#1549,38.0);
#1551=LINE('',#1547,#1550);
#1552=EDGE_CURVE('',#1548,#1539,#1551,.T.);
#1553=ORIENTED_EDGE('',*,*,#1552,.F.);
#1554=CARTESIAN_POINT('',(185.000000000000028,-40.0,2.0));
#1555=VERTEX_POINT('',#1554);
#1556=CARTESIAN_POINT('',(185.000000000000028,-40.0,7.0));
#1557=DIRECTION('',(0.0,1.0,0.0));
#1558=DIRECTION('',(0.0,0.0,-1.0));
#1559=AXIS2_PLACEMENT_3D('',#1556,#1557,#1558);
#1560=CIRCLE('',#1559,5.0);
#1561=EDGE_CURVE('',#1555,#1548,#1560,.T.);
#1562=ORIENTED_EDGE('',*,*,#1561,.F.);
#1563=CARTESIAN_POINT('',(223.0,-40.0,2.0));
#1564=VERTEX_POINT('',#1563);
#1565=DIRECTION('',(-1.0,0.0,0.0));
#1566=VECTOR('',#1565,37.999999999999972);
#1567=LINE('',#1563,#1566);
#1568=EDGE_CURVE('',#1564,#1555,#1567,.T.);
#1569=ORIENTED_EDGE('',*,*,#1568,.F.);
#1570=CARTESIAN_POINT('',(223.0,-40.0,7.0));
#1571=DIRECTION('',(0.0,1.0,0.0));
#1572=DIRECTION('',(1.0,0.0,0.0));
#1573=AXIS2_PLACEMENT_3D('',#1570,#1571,#1572);
#1574=CIRCLE('',#1573,5.0);
#1575=EDGE_CURVE('',#1516,#1564,#1574,.T.);
#1576=ORIENTED_EDGE('',*,*,#1575,.F.);
#1577=EDGE_LOOP('',(#1521,#1530,#1537,#1546,#1553,#1562,#1569,#1576));
#1578=FACE_BOUND('',#1577,.F.);
#1579=CARTESIAN_POINT('',(184.999999999999972,-40.0,178.0));
#1580=VERTEX_POINT('',#1579);
#1581=CARTESIAN_POINT('',(223.0,-40.0,178.0));
#1582=VERTEX_POINT('',#1581);
#1583=DIRECTION('',(1.0,0.0,0.0));
```



```
#1584=VECTOR('',#1583,38.000000000000028);
#1585=LINE('',#1579,#1584);
#1586=EDGE_CURVE('',#1580,#1582,#1585,.T.);
#1587=ORIENTED_EDGE('',*,*,#1586,.F.);
#1588=CARTESIAN_POINT('',(179.999999999999972,-40.0,173.0));
#1589=VERTEX_POINT('',#1588);
#1590=CARTESIAN_POINT('',(184.999999999999972,-40.0,173.0));
#1591=DIRECTION('',(0.0,-1.0,0.0));
#1592=DIRECTION('',(0.0,0.0,1.0));
#1593=AXIS2_PLACEMENT_3D('',#1590,#1591,#1592);
#1594=CIRCLE('',#1593,5.0);
#1595=EDGE_CURVE('',#1580,#1589,#1594,.T.);
#1596=ORIENTED_EDGE('',*,*,#1595,.T.);
#1597=CARTESIAN_POINT('',(179.999999999999972,-40.0,135.000000000000028));
#1598=VERTEX_POINT('',#1597);
#1599=DIRECTION('',(0.0,0.0,1.0));
#1600=VECTOR('',#1599,37.999999999999972);
#1601=LINE('',#1597,#1600);
#1602=EDGE_CURVE('',#1598,#1589,#1601,.T.);
#1603=ORIENTED_EDGE('',*,*,#1602,.F.);
#1604=CARTESIAN_POINT('',(184.999999999999972,-40.0,130.000000000000028));
#1605=VERTEX_POINT('',#1604);
#1606=CARTESIAN_POINT('',(184.999999999999972,-40.0,135.000000000000028));
#1607=DIRECTION('',(0.0,1.0,0.0));
#1608=DIRECTION('',(0.0,0.0,-1.0));
#1609=AXIS2_PLACEMENT_3D('',#1606,#1607,#1608);
#1610=CIRCLE('',#1609,5.0);
#1611=EDGE_CURVE('',#1605,#1598,#1610,.T.);
#1612=ORIENTED_EDGE('',*,*,#1611,.F.);
#1613=CARTESIAN_POINT('',(223.0,-40.0,130.000000000000028));
#1614=VERTEX_POINT('',#1613);
#1615=DIRECTION('',(-1.0,0.0,0.0));
#1616=VECTOR('',#1615,38.000000000000028);
#1617=LINE('',#1613,#1616);
#1618=EDGE_CURVE('',#1614,#1605,#1617,.T.);
#1619=ORIENTED_EDGE('',*,*,#1618,.F.);
#1620=CARTESIAN_POINT('',(228.0,-40.0,135.000000000000028));
#1621=VERTEX_POINT('',#1620);
#1622=CARTESIAN_POINT('',(223.0,-40.0,135.000000000000028));
#1623=DIRECTION('',(0.0,1.0,0.0));
#1624=DIRECTION('',(1.0,0.0,0.0));
#1625=AXIS2_PLACEMENT_3D('',#1622,#1623,#1624);
#1626=CIRCLE('',#1625,5.0);
#1627=EDGE_CURVE('',#1621,#1614,#1626,.T.);
#1628=ORIENTED_EDGE('',*,*,#1627,.F.);
#1629=CARTESIAN_POINT('',(228.0,-40.0,173.0));
#1630=VERTEX_POINT('',#1629);
#1631=DIRECTION('',(0.0,0.0,-1.0));
#1632=VECTOR('',#1631,37.999999999999972);
#1633=LINE('',#1629,#1632);
#1634=EDGE_CURVE('',#1630,#1621,#1633,.T.);
#1635=ORIENTED_EDGE('',*,*,#1634,.F.);
#1636=CARTESIAN_POINT('',(223.0,-40.0,173.0));
#1637=DIRECTION('',(0.0,-1.0,0.0));
#1638=DIRECTION('',(1.0,0.0,0.0));
#1639=AXIS2_PLACEMENT_3D('',#1636,#1637,#1638);
```

```
#1640=CIRCLE('',#1639,5.0);
#1641=EDGE_CURVE('',#1630,#1582,#1640,.T.);
#1642=ORIENTED_EDGE('',*,*,#1641,.T.);
#1643=EDGE_LOOP('',(#1587,#1596,#1603,#1612,#1619,#1628,#1635,#1642));
#1644=FACE_BOUND('',#1643,.F.);
#1645=CARTESIAN_POINT('',(167.500000000000000,-40.0,150.0));
#1646=VERTEX_POINT('',#1645);
#1647=CARTESIAN_POINT('',(152.500000000000000,-40.0,150.0));
#1648=VERTEX_POINT('',#1647);
#1649=CARTESIAN_POINT('',(160.0,-40.0,150.0));
#1650=DIRECTION('',(0.0,1.0,0.0));
#1651=DIRECTION('',(1.0,0.0,0.0));
#1652=AXIS2_PLACEMENT_3D('',#1649,#1650,#1651);
#1653=CIRCLE('',#1652,7.500000000000000);
#1654=EDGE_CURVE('',#1646,#1648,#1653,.T.);
#1655=ORIENTED_EDGE('',*,*,#1654,.F.);
#1656=CARTESIAN_POINT('',(160.0,-40.0,150.0));
#1657=DIRECTION('',(0.0,1.0,0.0));
#1658=DIRECTION('',(-1.0,0.0,0.0));
#1659=AXIS2_PLACEMENT_3D('',#1656,#1657,#1658);
#1660=CIRCLE('',#1659,7.500000000000000);
#1661=EDGE_CURVE('',#1648,#1646,#1660,.T.);
#1662=ORIENTED_EDGE('',*,*,#1661,.F.);
#1663=EDGE_LOOP('',(#1655,#1662));
#1664=FACE_BOUND('',#1663,.F.);
#1665=CARTESIAN_POINT('',(180.0,-40.0,90.0));
#1666=VERTEX_POINT('',#1665);
#1667=CARTESIAN_POINT('',(140.0,-40.0,90.0));
#1668=VERTEX_POINT('',#1667);
#1669=CARTESIAN_POINT('',(160.0,-40.0,90.0));
#1670=DIRECTION('',(0.0,-1.0,0.0));
#1671=DIRECTION('',(1.0,0.0,0.0));
#1672=AXIS2_PLACEMENT_3D('',#1669,#1670,#1671);
#1673=CIRCLE('',#1672,20.0);
#1674=EDGE_CURVE('',#1666,#1668,#1673,.T.);
#1675=ORIENTED_EDGE('',*,*,#1674,.T.);
#1676=CARTESIAN_POINT('',(160.0,-40.0,90.0));
#1677=DIRECTION('',(0.0,-1.0,0.0));
#1678=DIRECTION('',(-1.0,0.0,0.0));
#1679=AXIS2_PLACEMENT_3D('',#1676,#1677,#1678);
#1680=CIRCLE('',#1679,20.0);
#1681=EDGE_CURVE('',#1668,#1666,#1680,.T.);
#1682=ORIENTED_EDGE('',*,*,#1681,.T.);
#1683=EDGE_LOOP('',(#1675,#1682));
#1684=FACE_BOUND('',#1683,.F.);
#1685=CARTESIAN_POINT('',(250.0,-40.0,110.0));
#1686=VERTEX_POINT('',#1685);
#1687=CARTESIAN_POINT('',(240.0,-40.0,110.0));
#1688=VERTEX_POINT('',#1687);
#1689=CARTESIAN_POINT('',(245.0,-40.0,110.0));
#1690=DIRECTION('',(0.0,-1.0,0.0));
#1691=DIRECTION('',(1.0,0.0,0.0));
#1692=AXIS2_PLACEMENT_3D('',#1689,#1690,#1691);
#1693=CIRCLE('',#1692,5.0);
#1694=EDGE_CURVE('',#1686,#1688,#1693,.T.);
#1695=ORIENTED_EDGE('',*,*,#1694,.T.);
```



```
#1696=CARTESIAN_POINT('',(245.0,-40.0,110.0));
#1697=DIRECTION('',(0.0,-1.0,0.0));
#1698=DIRECTION('',(-1.0,0.0,0.0));
#1699=AXIS2_PLACEMENT_3D('',#1696,#1697,#1698);
#1700=CIRCLE('',#1699,5.0);
#1701=EDGE_CURVE('',#1688,#1686,#1700,.T.);
#1702=ORIENTED_EDGE('',*,*,#1701,.T.);
#1703=EDGE_LOOP('',(#1695,#1702));
#1704=FACE_BOUND('',#1703,.F.);
#1705=CARTESIAN_POINT('',(250.0,-40.0,70.0));
#1706=VERTEX_POINT('',#1705);
#1707=CARTESIAN_POINT('',(240.0,-40.0,70.0));
#1708=VERTEX_POINT('',#1707);
#1709=CARTESIAN_POINT('',(245.0,-40.0,70.0));
#1710=DIRECTION('',(0.0,-1.0,0.0));
#1711=DIRECTION('',(1.0,0.0,0.0));
#1712=AXIS2_PLACEMENT_3D('',#1709,#1710,#1711);
#1713=CIRCLE('',#1712,5.0);
#1714=EDGE_CURVE('',#1706,#1708,#1713,.T.);
#1715=ORIENTED_EDGE('',*,*,#1714,.T.);
#1716=CARTESIAN_POINT('',(245.0,-40.0,70.0));
#1717=DIRECTION('',(0.0,-1.0,0.0));
#1718=DIRECTION('',(-1.0,0.0,0.0));
#1719=AXIS2_PLACEMENT_3D('',#1716,#1717,#1718);
#1720=CIRCLE('',#1719,5.0);
#1721=EDGE_CURVE('',#1708,#1706,#1720,.T.);
#1722=ORIENTED_EDGE('',*,*,#1721,.T.);
#1723=EDGE_LOOP('',(#1715,#1722));
#1724=FACE_BOUND('',#1723,.F.);
#1725=CARTESIAN_POINT('',(2.0,-40.0,134.999999999999972));
#1726=VERTEX_POINT('',#1725);
#1727=CARTESIAN_POINT('',(2.0,-40.0,173.0));
#1728=VERTEX_POINT('',#1727);
#1729=DIRECTION('',(0.0,0.0,1.0));
#1730=VECTOR('',#1729,38.000000000000028);
#1731=LINE('',#1725,#1730);
#1732=EDGE_CURVE('',#1726,#1728,#1731,.T.);
#1733=ORIENTED_EDGE('',*,*,#1732,.F.);
#1734=CARTESIAN_POINT('',(7.0,-40.0,129.999999999999972));
#1735=VERTEX_POINT('',#1734);
#1736=CARTESIAN_POINT('',(7.0,-40.0,134.999999999999972));
#1737=DIRECTION('',(0.0,-1.0,0.0));
#1738=DIRECTION('',(-1.0,0.0,0.0));
#1739=AXIS2_PLACEMENT_3D('',#1736,#1737,#1738);
#1740=CIRCLE('',#1739,5.0);
#1741=EDGE_CURVE('',#1726,#1735,#1740,.T.);
#1742=ORIENTED_EDGE('',*,*,#1741,.T.);
#1743=CARTESIAN_POINT('',(44.999999999999972,-40.0,129.999999999999972));
#1744=VERTEX_POINT('',#1743);
#1745=DIRECTION('',(-1.0,0.0,0.0));
#1746=VECTOR('',#1745,37.999999999999972);
#1747=LINE('',#1743,#1746);
#1748=EDGE_CURVE('',#1744,#1735,#1747,.T.);
#1749=ORIENTED_EDGE('',*,*,#1748,.F.);
#1750=CARTESIAN_POINT('',(49.999999999999972,-40.0,134.999999999999972));
#1751=VERTEX_POINT('',#1750);
```

```
#1752=CARTESIAN_POINT('',(44.999999999999972,-40.0,134.999999999999972));
#1753=DIRECTION('',(0.0,1.0,0.0));
#1754=DIRECTION('',(1.0,0.0,0.0));
#1755=AXIS2_PLACEMENT_3D('',#1752,#1753,#1754);
#1756=CIRCLE('',#1755,5.0);
#1757=EDGE_CURVE('',#1751,#1744,#1756,.T.);
#1758=ORIENTED_EDGE('',*,*,#1757,.F.);
#1759=CARTESIAN_POINT('',(49.999999999999972,-40.0,173.0));
#1760=VERTEX_POINT('',#1759);
#1761=DIRECTION('',(0.0,0.0,-1.0));
#1762=VECTOR('',#1761,38.000000000000028);
#1763=LINE('',#1759,#1762);
#1764=EDGE_CURVE('',#1760,#1751,#1763,.T.);
#1765=ORIENTED_EDGE('',*,*,#1764,.F.);
#1766=CARTESIAN_POINT('',(44.999999999999972,-40.0,178.0));
#1767=VERTEX_POINT('',#1766);
#1768=CARTESIAN_POINT('',(44.999999999999972,-40.0,173.0));
#1769=DIRECTION('',(0.0,1.0,0.0));
#1770=DIRECTION('',(0.0,0.0,1.0));
#1771=AXIS2_PLACEMENT_3D('',#1768,#1769,#1770);
#1772=CIRCLE('',#1771,5.0);
#1773=EDGE_CURVE('',#1767,#1760,#1772,.T.);
#1774=ORIENTED_EDGE('',*,*,#1773,.F.);
#1775=CARTESIAN_POINT('',(7.0,-40.0,178.0));
#1776=VERTEX_POINT('',#1775);
#1777=DIRECTION('',(1.0,0.0,0.0));
#1778=VECTOR('',#1777,37.999999999999972);
#1779=LINE('',#1775,#1778);
#1780=EDGE_CURVE('',#1776,#1767,#1779,.T.);
#1781=ORIENTED_EDGE('',*,*,#1780,.F.);
#1782=CARTESIAN_POINT('',(7.0,-40.0,173.0));
#1783=DIRECTION('',(0.0,1.0,0.0));
#1784=DIRECTION('',(-1.0,0.0,0.0));
#1785=AXIS2_PLACEMENT_3D('',#1782,#1783,#1784);
#1786=CIRCLE('',#1785,5.0);
#1787=EDGE_CURVE('',#1728,#1776,#1786,.T.);
#1788=ORIENTED_EDGE('',*,*,#1787,.F.);
#1789=EDGE_LOOP('',(#1733,#1742,#1749,#1758,#1765,#1774,#1781,#1788));
#1790=FACE_BOUND('',#1789,.F.);
#1791=CARTESIAN_POINT('',(230.0,-40.0,180.0));
#1792=DIRECTION('',(0.0,1.0,0.0));
#1793=DIRECTION('',(1.0,0.0,0.0));
#1794=AXIS2_PLACEMENT_3D('',#1791,#1792,#1793);
#1795=PLANE('',#1794);
#1796=ADVANCED_FACE('',(#1446,#1512,#1578,#1644,#1664,#1684,#1704,#1724,#1790
),#1795,.F.);
#1797=SHAPE_ASPECT('featured shape: face 13',$,#38,.T.);
#1798=PROPERTY_DEFINITION('',$,#1797);
#1799=FACE_SHAPE_REPRESENTATION('',(#1796),#29);
#1800=PROPERTY_DEFINITION_REPRESENTATION(#1798,#1799);
#1801=ORIENTED_EDGE('',*,*,#1397,.F.);
#1802=ORIENTED_EDGE('',*,*,#1238,.F.);
#1803=ORIENTED_EDGE('',*,*,#1310,.T.);
#1804=DIRECTION('',(0.0,-1.0,0.0));
#1805=VECTOR('',#1804,20.0);
#1806=LINE('',#1298,#1805);
```



```
#1807=EDGE_CURVE('',#1299,#1391,#1806,.T.);
#1808=ORIENTED_EDGE('',*,*,#1807,.T.);
#1809=EDGE_LOOP('',(#1801,#1802,#1803,#1808));
#1810=FACE_BOUND('',#1809,.F.);
#1811=CARTESIAN_POINT('',(240.0,-10.0,140.0));
#1812=DIRECTION('',(0.0,-1.0,0.0));
#1813=DIRECTION('',(0.0,0.0,-1.0));
#1814=AXIS2_PLACEMENT_3D('',#1811,#1812,#1813);
#1815=CYLINDRICAL_SURFACE('',#1814,10.0);
#1816=ADVANCED_FACE('',(#1810),#1815,.F.);
#1817=SHAPE_ASPECT('featured shape: face 14',$,#38,.T.);
#1818=PROPERTY_DEFINITION('',$,#1817);
#1819=FACE_SHAPE_REPRESENTATION('',(#1816),#29);
#1820=PROPERTY_DEFINITION_REPRESENTATION(#1818,#1819);
#1821=DIRECTION('',(0.0,1.0,0.0));
#1822=VECTOR('',#1821,20.0);
#1823=LINE('',#1399,#1822);
#1824=EDGE_CURVE('',#1400,#1290,#1823,.T.);
#1825=ORIENTED_EDGE('',*,*,#1824,.F.);
#1826=ORIENTED_EDGE('',*,*,#1404,.T.);
#1827=ORIENTED_EDGE('',*,*,#1807,.F.);
#1828=ORIENTED_EDGE('',*,*,#1303,.F.);
#1829=EDGE_LOOP('',(#1825,#1826,#1827,#1828));
#1830=FACE_BOUND('',#1829,.F.);
#1831=CARTESIAN_POINT('',(240.0,-10.0,130.0));
#1832=DIRECTION('',(0.0,0.0,-1.0));
#1833=DIRECTION('',(1.0,0.0,0.0));
#1834=AXIS2_PLACEMENT_3D('',#1831,#1832,#1833);
#1835=PLANE('',#1834);
#1836=ADVANCED_FACE('',(#1830),#1835,.F.);
#1837=SHAPE_ASPECT('featured shape: face 15',$,#38,.T.);
#1838=PROPERTY_DEFINITION('',$,#1837);
#1839=FACE_SHAPE_REPRESENTATION('',(#1836),#29);
#1840=PROPERTY_DEFINITION_REPRESENTATION(#1838,#1839);
#1841=ORIENTED_EDGE('',*,*,#1435,.T.);
#1842=DIRECTION('',(0.0,1.0,0.0));
#1843=VECTOR('',#1842,20.0);
#1844=LINE('',#1423,#1843);
#1845=EDGE_CURVE('',#1424,#1266,#1844,.T.);
#1846=ORIENTED_EDGE('',*,*,#1845,.T.);
#1847=ORIENTED_EDGE('',*,*,#1272,.F.);
#1848=ORIENTED_EDGE('',*,*,#1224,.F.);
#1849=EDGE_LOOP('',(#1841,#1846,#1847,#1848));
#1850=FACE_BOUND('',#1849,.F.);
#1851=CARTESIAN_POINT('',(240.0,-50.0,40.0));
#1852=DIRECTION('',(0.0,1.0,0.0));
#1853=DIRECTION('',(0.0,0.0,1.0));
#1854=AXIS2_PLACEMENT_3D('',#1851,#1852,#1853);
#1855=CYLINDRICAL_SURFACE('',#1854,10.0);
#1856=ADVANCED_FACE('',(#1850),#1855,.F.);
#1857=SHAPE_ASPECT('featured shape: face 16',$,#38,.T.);
#1858=PROPERTY_DEFINITION('',$,#1857);
#1859=FACE_SHAPE_REPRESENTATION('',(#1856),#29);
#1860=PROPERTY_DEFINITION_REPRESENTATION(#1858,#1859);
#1861=DIRECTION('',(0.0,1.0,0.0));
#1862=VECTOR('',#1861,20.0);
```

```
#1863=LINE('',#1414,#1862);
#1864=EDGE_CURVE('',#1415,#1275,#1863,.T.);
#1865=ORIENTED_EDGE('',*,*,#1864,.T.);
#1866=ORIENTED_EDGE('',*,*,#1279,.T.);
#1867=ORIENTED_EDGE('',*,*,#1845,.F.);
#1868=ORIENTED_EDGE('',*,*,#1428,.F.);
#1869=EDGE_LOOP('',(#1865,#1866,#1867,#1868));
#1870=FACE_BOUND('',#1869,.F.);
#1871=CARTESIAN_POINT('',(240.0,-50.0,50.0));
#1872=DIRECTION('',(0.0,0.0,1.0));
#1873=DIRECTION('',(1.0,0.0,0.0));
#1874=AXIS2_PLACEMENT_3D('',#1871,#1872,#1873);
#1875=PLANE('',#1874);
#1876=ADVANCED_FACE('',(#1870),#1875,.F.);
#1877=SHAPE_ASPECT('featured shape: face 17',$,#38,.T.);
#1878=PROPERTY_DEFINITION('',$,#1877);
#1879=FACE_SHAPE_REPRESENTATION('',(#1876),#29);
#1880=PROPERTY_DEFINITION_REPRESENTATION(#1878,#1879);
#1881=ORIENTED_EDGE('',*,*,#1324,.T.);
#1882=DIRECTION('',(0.0,-1.0,0.0));
#1883=VECTOR('',#1882,20.0);
#1884=LINE('',#1317,#1883);
#1885=EDGE_CURVE('',#1318,#1688,#1884,.T.);
#1886=ORIENTED_EDGE('',*,*,#1885,.T.);
#1887=ORIENTED_EDGE('',*,*,#1694,.F.);
#1888=DIRECTION('',(0.0,-1.0,0.0));
#1889=VECTOR('',#1888,20.0);
#1890=LINE('',#1315,#1889);
#1891=EDGE_CURVE('',#1316,#1686,#1890,.T.);
#1892=ORIENTED_EDGE('',*,*,#1891,.F.);
#1893=EDGE_LOOP('',(#1881,#1886,#1887,#1892));
#1894=FACE_BOUND('',#1893,.F.);
#1895=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#1896=DIRECTION('',(0.0,-1.0,0.0));
#1897=DIRECTION('',(1.0,0.0,0.0));
#1898=AXIS2_PLACEMENT_3D('',#1895,#1896,#1897);
#1899=CYLINDRICAL_SURFACE('',#1898,5.0);
#1900=ADVANCED_FACE('',(#1894),#1899,.F.);
#1901=SHAPE_ASPECT('featured shape: face 18',$,#38,.T.);
#1902=PROPERTY_DEFINITION('',$,#1901);
#1903=FACE_SHAPE_REPRESENTATION('',(#1900),#29);
#1904=PROPERTY_DEFINITION_REPRESENTATION(#1902,#1903);
#1905=ORIENTED_EDGE('',*,*,#1331,.T.);
#1906=ORIENTED_EDGE('',*,*,#1891,.T.);
#1907=ORIENTED_EDGE('',*,*,#1701,.F.);
#1908=ORIENTED_EDGE('',*,*,#1885,.F.);
#1909=EDGE_LOOP('',(#1905,#1906,#1907,#1908));
#1910=FACE_BOUND('',#1909,.F.);
#1911=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#1912=DIRECTION('',(0.0,-1.0,0.0));
#1913=DIRECTION('',(1.0,0.0,0.0));
#1914=AXIS2_PLACEMENT_3D('',#1911,#1912,#1913);
#1915=CYLINDRICAL_SURFACE('',#1914,5.0);
#1916=ADVANCED_FACE('',(#1910),#1915,.F.);
#1917=SHAPE_ASPECT('featured shape: face 19',$,#38,.T.);
#1918=PROPERTY_DEFINITION('',$,#1917);
```



```
#1919=FACE_SHAPE_REPRESENTATION('',(#1916),#29);
#1920=PROPERTY_DEFINITION_REPRESENTATION(#1918,#1919);
#1921=ORIENTED_EDGE('',*,*,#1344,.T.);
#1922=DIRECTION('',(0.0,-1.0,0.0));
#1923=VECTOR('',#1922,20.0);
#1924=LINE('',#1337,#1923);
#1925=EDGE_CURVE('',#1338,#1708,#1924,.T.);
#1926=ORIENTED_EDGE('',*,*,#1925,.T.);
#1927=ORIENTED_EDGE('',*,*,#1714,.F.);
#1928=DIRECTION('',(0.0,-1.0,0.0));
#1929=VECTOR('',#1928,20.0);
#1930=LINE('',#1335,#1929);
#1931=EDGE_CURVE('',#1336,#1706,#1930,.T.);
#1932=ORIENTED_EDGE('',*,*,#1931,.F.);
#1933=EDGE_LOOP('',(#1921,#1926,#1927,#1932));
#1934=FACE_BOUND('',#1933,.F.);
#1935=CARTESIAN_POINT('',(245.0,-20.0,70.0));
#1936=DIRECTION('',(0.0,-1.0,0.0));
#1937=DIRECTION('',(1.0,0.0,0.0));
#1938=AXIS2_PLACEMENT_3D('',#1935,#1936,#1937);
#1939=CYLINDRICAL_SURFACE('',#1938,5.0);
#1940=ADVANCED_FACE('',(#1934),#1939,.F.);
#1941=SHAPE_ASPECT('featured shape: face 20',$,#38,.T.);
#1942=PROPERTY_DEFINITION('',$,#1941);
#1943=FACE_SHAPE_REPRESENTATION('',(#1940),#29);
#1944=PROPERTY_DEFINITION_REPRESENTATION(#1942,#1943);
#1945=ORIENTED_EDGE('',*,*,#1351,.T.);
#1946=ORIENTED_EDGE('',*,*,#1931,.T.);
#1947=ORIENTED_EDGE('',*,*,#1721,.F.);
#1948=ORIENTED_EDGE('',*,*,#1925,.F.);
#1949=EDGE_LOOP('',(#1945,#1946,#1947,#1948));
#1950=FACE_BOUND('',#1949,.F.);
#1951=CARTESIAN_POINT('',(245.0,-20.0,70.0));
#1952=DIRECTION('',(0.0,-1.0,0.0));
#1953=DIRECTION('',(1.0,0.0,0.0));
#1954=AXIS2_PLACEMENT_3D('',#1951,#1952,#1953);
#1955=CYLINDRICAL_SURFACE('',#1954,5.0);
#1956=ADVANCED_FACE('',(#1950),#1955,.F.);
#1957=SHAPE_ASPECT('featured shape: face 21',$,#38,.T.);
#1958=PROPERTY_DEFINITION('',$,#1957);
#1959=FACE_SHAPE_REPRESENTATION('',(#1956),#29);
#1960=PROPERTY_DEFINITION_REPRESENTATION(#1958,#1959);
#1961=CARTESIAN_POINT('',(180.0,-18.250000000000000,90.000000000000014));
#1962=VERTEX_POINT('',#1961);
#1963=CARTESIAN_POINT('',(140.0,-18.250000000000000,90.0));
#1964=VERTEX_POINT('',#1963);
#1965=CARTESIAN_POINT('',(160.0,-18.250000000000000,90.0));
#1966=DIRECTION('',(0.0,-1.0,0.0));
#1967=DIRECTION('',(1.0,0.0,7.105427E-16));
#1968=AXIS2_PLACEMENT_3D('',#1965,#1966,#1967);
#1969=CIRCLE('',#1968,20.0);
#1970=EDGE_CURVE('',#1962,#1964,#1969,.T.);
#1971=ORIENTED_EDGE('',*,*,#1970,.F.);
#1972=DIRECTION('',(0.0,-1.0,7.786770E-16));
#1973=VECTOR('',#1972,18.250000000000000);
#1974=LINE('',#784,#1973);
```



```
#1975=EDGE_CURVE('',#785,#1962,#1974,.T.);
#1976=ORIENTED_EDGE('',*,*,#1975,.F.);
#1977=ORIENTED_EDGE('',*,*,#793,.T.);
#1978=DIRECTION('',(0.0,-1.0,0.0));
#1979=VECTOR('',#1978,18.250000000000000);
#1980=LINE('',#786,#1979);
#1981=EDGE_CURVE('',#787,#1964,#1980,.T.);
#1982=ORIENTED_EDGE('',*,*,#1981,.T.);
#1983=EDGE_LOOP('',(#1971,#1976,#1977,#1982));
#1984=FACE_BOUND('',#1983,.F.);
#1985=CARTESIAN_POINT('',(160.0,0.0,90.0));
#1986=DIRECTION('',(0.0,-1.0,0.0));
#1987=DIRECTION('',(1.0,0.0,0.0));
#1988=AXIS2_PLACEMENT_3D('',#1985,#1986,#1987);
#1989=CYLINDRICAL_SURFACE('',#1988,20.0);
#1990=ADVANCED_FACE('',(#1984),#1989,.F.);
#1991=SHAPE_ASPECT('featured shape: face 22',$,#38,.T.);
#1992=PROPERTY_DEFINITION('',$,#1991);
#1993=FACE_SHAPE_REPRESENTATION('',(#1990),#29);
#1994=PROPERTY_DEFINITION_REPRESENTATION(#1992,#1993);
#1995=ORIENTED_EDGE('',*,*,#1674,.F.);
#1996=CARTESIAN_POINT('',(180.0,-21.749999999999996,90.0));
#1997=VERTEX_POINT('',#1996);
#1998=DIRECTION('',(0.0,-1.0,0.0));
#1999=VECTOR('',#1998,18.250000000000011);
#2000=LINE('',#1996,#1999);
#2001=EDGE_CURVE('',#1997,#1666,#2000,.T.);
#2002=ORIENTED_EDGE('',*,*,#2001,.F.);
#2003=CARTESIAN_POINT('',(140.0,-21.750000000000000,90.0));
#2004=VERTEX_POINT('',#2003);
#2005=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2006=DIRECTION('',(0.0,-1.0,0.0));
#2007=DIRECTION('',(1.0,0.0,0.0));
#2008=AXIS2_PLACEMENT_3D('',#2005,#2006,#2007);
#2009=CIRCLE('',#2008,20.0);
#2010=EDGE_CURVE('',#1997,#2004,#2009,.T.);
#2011=ORIENTED_EDGE('',*,*,#2010,.T.);
#2012=DIRECTION('',(0.0,-1.0,0.0));
#2013=VECTOR('',#2012,18.250000000000007);
#2014=LINE('',#2003,#2013);
#2015=EDGE_CURVE('',#2004,#1668,#2014,.T.);
#2016=ORIENTED_EDGE('',*,*,#2015,.T.);
#2017=EDGE_LOOP('',(#1995,#2002,#2011,#2016));
#2018=FACE_BOUND('',#2017,.F.);
#2019=ADVANCED_FACE('',(#2018),#1989,.F.);
#2020=SHAPE_ASPECT('featured shape: face 23',$,#38,.T.);
#2021=PROPERTY_DEFINITION('',$,#2020);
#2022=FACE_SHAPE_REPRESENTATION('',(#2019),#29);
#2023=PROPERTY_DEFINITION_REPRESENTATION(#2021,#2022);
#2024=CARTESIAN_POINT('',(160.0,-18.250000000000000,90.0));
#2025=DIRECTION('',(0.0,-1.0,0.0));
#2026=DIRECTION('',(-1.0,0.0,0.0));
#2027=AXIS2_PLACEMENT_3D('',#2024,#2025,#2026);
#2028=CIRCLE('',#2027,20.0);
#2029=EDGE_CURVE('',#1964,#1962,#2028,.T.);
#2030=ORIENTED_EDGE('',*,*,#2029,.F.);
```

```
#2031=ORIENTED_EDGE('',*,*,#1981,.F.);
#2032=ORIENTED_EDGE('',*,*,#800,.T.);
#2033=ORIENTED_EDGE('',*,*,#1975,.T.);
#2034=EDGE_LOOP('',(#2030,#2031,#2032,#2033));
#2035=FACE_BOUND('',#2034,.F.);
#2036=CARTESIAN_POINT('',(160.0,0.0,90.0));
#2037=DIRECTION('',(0.0,-1.0,0.0));
#2038=DIRECTION('',(1.0,0.0,0.0));
#2039=AXIS2_PLACEMENT_3D('',#2036,#2037,#2038);
#2040=CYLINDRICAL_SURFACE('',#2039,20.0);
#2041=ADVANCED_FACE('',(#2035),#2040,.F.);
#2042=SHAPE_ASPECT('featured shape: face 24',$,#38,.T.);
#2043=PROPERTY_DEFINITION('',$,#2042);
#2044=FACE_SHAPE_REPRESENTATION('',(#2041),#29);
#2045=PROPERTY_DEFINITION_REPRESENTATION(#2043,#2044);
#2046=ORIENTED_EDGE('',*,*,#1681,.F.);
#2047=ORIENTED_EDGE('',*,*,#2015,.F.);
#2048=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2049=DIRECTION('',(0.0,-1.0,0.0));
#2050=DIRECTION('',(-1.0,0.0,0.0));
#2051=AXIS2_PLACEMENT_3D('',#2048,#2049,#2050);
#2052=CIRCLE('',#2051,20.0);
#2053=EDGE_CURVE('',#2004,#1997,#2052,.T.);
#2054=ORIENTED_EDGE('',*,*,#2053,.T.);
#2055=ORIENTED_EDGE('',*,*,#2001,.T.);
#2056=EDGE_LOOP('',(#2046,#2047,#2054,#2055));
#2057=FACE_BOUND('',#2056,.F.);
#2058=ADVANCED_FACE('',(#2057),#2040,.F.);
#2059=SHAPE_ASPECT('featured shape: face 25',$,#38,.T.);
#2060=PROPERTY_DEFINITION('',$,#2059);
#2061=FACE_SHAPE_REPRESENTATION('',(#2058),#29);
#2062=PROPERTY_DEFINITION_REPRESENTATION(#2060,#2061);
#2063=CARTESIAN_POINT('',(50.0,-10.0,99.0));
#2064=VERTEX_POINT('',#2063);
#2065=DIRECTION('',(1.0,0.0,0.0));
#2066=VECTOR('',#2065,50.0);
#2067=LINE('',#156,#2066);
#2068=EDGE_CURVE('',#157,#2064,#2067,.T.);
#2069=ORIENTED_EDGE('',*,*,#2068,.T.);
#2070=DIRECTION('',(0.0,-1.0,0.0));
#2071=VECTOR('',#2070,10.0);
#2072=LINE('',#698,#2071);
#2073=EDGE_CURVE('',#699,#2064,#2072,.T.);
#2074=ORIENTED_EDGE('',*,*,#2073,.F.);
#2075=ORIENTED_EDGE('',*,*,#703,.F.);
#2076=ORIENTED_EDGE('',*,*,#168,.T.);
#2077=EDGE_LOOP('',(#2069,#2074,#2075,#2076));
#2078=FACE_BOUND('',#2077,.F.);
#2079=CARTESIAN_POINT('',(0.0,0.0,99.0));
#2080=DIRECTION('',(0.0,0.0,1.0));
#2081=DIRECTION('',(1.0,0.0,0.0));
#2082=AXIS2_PLACEMENT_3D('',#2079,#2080,#2081);
#2083=PLANE('',#2082);
#2084=ADVANCED_FACE('',(#2078),#2083,.F.);
#2085=SHAPE_ASPECT('featured shape: face 26',$,#38,.T.);
#2086=PROPERTY_DEFINITION('',$,#2085);
```

```
#2087=FACE_SHAPE_REPRESENTATION('',(#2084),#29);
#2088=PROPERTY_DEFINITION_REPRESENTATION(#2086,#2087);
#2089=CARTESIAN_POINT('',(50.0,-10.0,81.0));
#2090=VERTEX_POINT('',#2089);
#2091=DIRECTION('',(-1.0,0.0,0.0));
#2092=VECTOR('',#2091,50.0);
#2093=LINE('',#2089,#2092);
#2094=EDGE_CURVE('',#2090,#206,#2093,.T.);
#2095=ORIENTED_EDGE('',*,*,#2094,.T.);
#2096=ORIENTED_EDGE('',*,*,#210,.F.);
#2097=ORIENTED_EDGE('',*,*,#717,.F.);
#2098=DIRECTION('',(0.0,-1.0,0.0));
#2099=VECTOR('',#2098,10.0);
#2100=LINE('',#705,#2099);
#2101=EDGE_CURVE('',#706,#2090,#2100,.T.);
#2102=ORIENTED_EDGE('',*,*,#2101,.T.);
#2103=EDGE_LOOP('',(#2095,#2096,#2097,#2102));
#2104=FACE_BOUND('',#2103,.F.);
#2105=CARTESIAN_POINT('',(50.0,0.0,81.0));
#2106=DIRECTION('',(0.0,0.0,-1.0));
#2107=DIRECTION('',(-1.0,0.0,0.0));
#2108=AXIS2_PLACEMENT_3D('',#2105,#2106,#2107);
#2109=PLANE('',#2108);
#2110=ADVANCED_FACE('',(#2104),#2109,.F.);
#2111=SHAPE_ASPECT('featured shape: face 27',$,#38,.T.);
#2112=PROPERTY_DEFINITION('',$,#2111);
#2113=FACE_SHAPE_REPRESENTATION('',(#2110),#29);
#2114=PROPERTY_DEFINITION_REPRESENTATION(#2112,#2113);
#2115=DIRECTION('',(0.0,1.0,0.0));
#2116=VECTOR('',#2115,10.0);
#2117=LINE('',#364,#2116);
#2118=EDGE_CURVE('',#365,#1514,#2117,.T.);
#2119=ORIENTED_EDGE('',*,*,#2118,.T.);
#2120=ORIENTED_EDGE('',*,*,#1520,.T.);
#2121=DIRECTION('',(0.0,1.0,0.0));
#2122=VECTOR('',#2121,10.0);
#2123=LINE('',#373,#2122);
#2124=EDGE_CURVE('',#374,#1516,#2123,.T.);
#2125=ORIENTED_EDGE('',*,*,#2124,.F.);
#2126=ORIENTED_EDGE('',*,*,#378,.T.);
#2127=EDGE_LOOP('',(#2119,#2120,#2125,#2126));
#2128=FACE_BOUND('',#2127,.F.);
#2129=CARTESIAN_POINT('',(228.0,-50.0,180.0));
#2130=DIRECTION('',(-1.0,0.0,0.0));
#2131=DIRECTION('',(0.0,1.0,0.0));
#2132=AXIS2_PLACEMENT_3D('',#2129,#2130,#2131);
#2133=PLANE('',#2132);
#2134=ADVANCED_FACE('',(#2128),#2133,.F.);
#2135=SHAPE_ASPECT('featured shape: face 28',$,#38,.T.);
#2136=PROPERTY_DEFINITION('',$,#2135);
#2137=FACE_SHAPE_REPRESENTATION('',(#2134),#29);
#2138=PROPERTY_DEFINITION_REPRESENTATION(#2136,#2137);
#2139=DIRECTION('',(0.0,1.0,0.0));
#2140=VECTOR('',#2139,10.0);
#2141=LINE('',#480,#2140);
#2142=EDGE_CURVE('',#481,#1621,#2141,.T.);
```



```
#2143=ORIENTED_EDGE('',*,*,#2142,.F.);
#2144=ORIENTED_EDGE('',*,*,#485,.T.);
#2145=DIRECTION('',(0.0,1.0,0.0));
#2146=VECTOR('',#2145,10.0);
#2147=LINE('',#471,#2146);
#2148=EDGE_CURVE('',#472,#1630,#2147,.T.);
#2149=ORIENTED_EDGE('',*,*,#2148,.T.);
#2150=ORIENTED_EDGE('',*,*,#1634,.T.);
#2151=EDGE_LOOP('',(#2143,#2144,#2149,#2150));
#2152=FACE_BOUND('',#2151,.F.);
#2153=ADVANCED_FACE('',(#2152),#2133,.F.);
#2154=SHAPE_ASPECT('featured shape: face 29',$,#38,.T.);
#2155=PROPERTY_DEFINITION('',$,#2154);
#2156=FACE_SHAPE_REPRESENTATION('',(#2153),#29);
#2157=PROPERTY_DEFINITION_REPRESENTATION(#2155,#2156);
#2158=DIRECTION('',(0.0,1.0,0.0));
#2159=VECTOR('',#2158,10.0);
#2160=LINE('',#293,#2159);
#2161=EDGE_CURVE('',#294,#1489,#2160,.T.);
#2162=ORIENTED_EDGE('',*,*,#2161,.F.);
#2163=ORIENTED_EDGE('',*,*,#298,.F.);
#2164=DIRECTION('',(0.0,1.0,0.0));
#2165=VECTOR('',#2164,10.0);
#2166=LINE('',#284,#2165);
#2167=EDGE_CURVE('',#285,#1498,#2166,.T.);
#2168=ORIENTED_EDGE('',*,*,#2167,.T.);
#2169=ORIENTED_EDGE('',*,*,#1502,.T.);
#2170=EDGE_LOOP('',(#2162,#2163,#2168,#2169));
#2171=FACE_BOUND('',#2170,.F.);
#2172=CARTESIAN_POINT('',(2.0,-50.0,0.0));
#2173=DIRECTION('',(1.0,0.0,0.0));
#2174=DIRECTION('',(0.0,1.0,0.0));
#2175=AXIS2_PLACEMENT_3D('',#2172,#2173,#2174);
#2176=PLANE('',#2175);
#2177=ADVANCED_FACE('',(#2171),#2176,.F.);
#2178=SHAPE_ASPECT('featured shape: face 30',$,#38,.T.);
#2179=PROPERTY_DEFINITION('',$,#2178);
#2180=FACE_SHAPE_REPRESENTATION('',(#2177),#29);
#2181=PROPERTY_DEFINITION_REPRESENTATION(#2179,#2180);
#2182=DIRECTION('',(0.0,1.0,0.0));
#2183=VECTOR('',#2182,10.0);
#2184=LINE('',#546,#2183);
#2185=EDGE_CURVE('',#547,#1726,#2184,.T.);
#2186=ORIENTED_EDGE('',*,*,#2185,.T.);
#2187=ORIENTED_EDGE('',*,*,#1732,.T.);
#2188=DIRECTION('',(0.0,1.0,0.0));
#2189=VECTOR('',#2188,10.0);
#2190=LINE('',#555,#2189);
#2191=EDGE_CURVE('',#556,#1728,#2190,.T.);
#2192=ORIENTED_EDGE('',*,*,#2191,.F.);
#2193=ORIENTED_EDGE('',*,*,#560,.F.);
#2194=EDGE_LOOP('',(#2186,#2187,#2192,#2193));
#2195=FACE_BOUND('',#2194,.F.);
#2196=ADVANCED_FACE('',(#2195),#2176,.F.);
#2197=SHAPE_ASPECT('featured shape: face 31',$,#38,.T.);
#2198=PROPERTY_DEFINITION('',$,#2197);
```

```
#2199=FACE_SHAPE_REPRESENTATION('',(#2196),#29);
#2200=PROPERTY_DEFINITION_REPRESENTATION(#2198,#2199);
#2201=DIRECTION('',(0.0,1.0,0.0));
#2202=VECTOR('',#2201,10.0);
#2203=LINE('',#455,#2202);
#2204=EDGE_CURVE('',#456,#1580,#2203,.T.);
#2205=ORIENTED_EDGE('',*,*,#2204,.T.);
#2206=ORIENTED_EDGE('',*,*,#1586,.T.);
#2207=DIRECTION('',(0.0,1.0,0.0));
#2208=VECTOR('',#2207,10.0);
#2209=LINE('',#464,#2208);
#2210=EDGE_CURVE('',#465,#1582,#2209,.T.);
#2211=ORIENTED_EDGE('',*,*,#2210,.F.);
#2212=ORIENTED_EDGE('',*,*,#469,.F.);
#2213=EDGE_LOOP('',(#2205,#2206,#2211,#2212));
#2214=FACE_BOUND('',#2213,.F.);
#2215=CARTESIAN_POINT('',(0.0,-50.0,178.0));
#2216=DIRECTION('',(0.0,0.0,-1.0));
#2217=DIRECTION('',(0.0,1.0,0.0));
#2218=AXIS2_PLACEMENT_3D('',#2215,#2216,#2217);
#2219=PLANE('',#2218);
#2220=ADVANCED_FACE('',(#2214),#2219,.F.);
#2221=SHAPE_ASPECT('featured shape: face 32',$,#38,.T.);
#2222=PROPERTY_DEFINITION('',$,#2221);
#2223=FACE_SHAPE_REPRESENTATION('',(#2220),#29);
#2224=PROPERTY_DEFINITION_REPRESENTATION(#2222,#2223);
#2225=DIRECTION('',(0.0,1.0,0.0));
#2226=VECTOR('',#2225,10.0);
#2227=LINE('',#571,#2226);
#2228=EDGE_CURVE('',#572,#1767,#2227,.T.);
#2229=ORIENTED_EDGE('',*,*,#2228,.F.);
#2230=ORIENTED_EDGE('',*,*,#576,.F.);
#2231=DIRECTION('',(0.0,1.0,0.0));
#2232=VECTOR('',#2231,10.0);
#2233=LINE('',#562,#2232);
#2234=EDGE_CURVE('',#563,#1776,#2233,.T.);
#2235=ORIENTED_EDGE('',*,*,#2234,.T.);
#2236=ORIENTED_EDGE('',*,*,#1780,.T.);
#2237=EDGE_LOOP('',(#2229,#2230,#2235,#2236));
#2238=FACE_BOUND('',#2237,.F.);
#2239=ADVANCED_FACE('',(#2238),#2219,.F.);
#2240=SHAPE_ASPECT('featured shape: face 33',$,#38,.T.);
#2241=PROPERTY_DEFINITION('',$,#2240);
#2242=FACE_SHAPE_REPRESENTATION('',(#2239),#29);
#2243=PROPERTY_DEFINITION_REPRESENTATION(#2241,#2242);
#2244=DIRECTION('',(0.0,1.0,0.0));
#2245=VECTOR('',#2244,10.0);
#2246=LINE('',#268,#2245);
#2247=EDGE_CURVE('',#269,#1448,#2246,.T.);
#2248=ORIENTED_EDGE('',*,*,#2247,.T.);
#2249=ORIENTED_EDGE('',*,*,#1454,.T.);
#2250=DIRECTION('',(0.0,1.0,0.0));
#2251=VECTOR('',#2250,10.0);
#2252=LINE('',#277,#2251);
#2253=EDGE_CURVE('',#278,#1450,#2252,.T.);
#2254=ORIENTED_EDGE('',*,*,#2253,.F.);
```

```
#2255=ORIENTED_EDGE('',*,*,#282,.T.);
#2256=EDGE_LOOP('',(#2248,#2249,#2254,#2255));
#2257=FACE_BOUND('',#2256,.F.);
#2258=CARTESIAN_POINT('',(230.0,-50.0,2.0));
#2259=DIRECTION('',(0.0,0.0,1.0));
#2260=DIRECTION('',(0.0,1.0,0.0));
#2261=AXIS2_PLACEMENT_3D('',#2258,#2259,#2260);
#2262=PLANE('',#2261);
#2263=ADVANCED_FACE('',(#2257),#2262,.F.);
#2264=SHAPE_ASPECT('featured shape: face 34',$,#38,.T.);
#2265=PROPERTY_DEFINITION('',$,#2264);
#2266=FACE_SHAPE_REPRESENTATION('',(#2263),#29);
#2267=PROPERTY_DEFINITION_REPRESENTATION(#2265,#2266);
#2268=DIRECTION('',(0.0,1.0,0.0));
#2269=VECTOR('',#2268,10.0);
#2270=LINE('',#389,#2269);
#2271=EDGE_CURVE('',#390,#1555,#2270,.T.);
#2272=ORIENTED_EDGE('',*,*,#2271,.F.);
#2273=ORIENTED_EDGE('',*,*,#394,.T.);
#2274=DIRECTION('',(0.0,1.0,0.0));
#2275=VECTOR('',#2274,10.0);
#2276=LINE('',#380,#2275);
#2277=EDGE_CURVE('',#381,#1564,#2276,.T.);
#2278=ORIENTED_EDGE('',*,*,#2277,.T.);
#2279=ORIENTED_EDGE('',*,*,#1568,.T.);
#2280=EDGE_LOOP('',(#2272,#2273,#2278,#2279));
#2281=FACE_BOUND('',#2280,.F.);
#2282=ADVANCED_FACE('',(#2281),#2262,.F.);
#2283=SHAPE_ASPECT('featured shape: face 35',$,#38,.T.);
#2284=PROPERTY_DEFINITION('',$,#2283);
#2285=FACE_SHAPE_REPRESENTATION('',(#2282),#29);
#2286=PROPERTY_DEFINITION_REPRESENTATION(#2284,#2285);
#2287=CARTESIAN_POINT('',(182.0,-18.250000000000000,90.0));
#2288=VERTEX_POINT('',#2287);
#2289=CARTESIAN_POINT('',(138.0,-18.250000000000000,90.0));
#2290=VERTEX_POINT('',#2289);
#2291=CARTESIAN_POINT('',(160.0,-18.250000000000000,90.0));
#2292=DIRECTION('',(0.0,1.0,0.0));
#2293=DIRECTION('',(1.0,0.0,0.0));
#2294=AXIS2_PLACEMENT_3D('',#2291,#2292,#2293);
#2295=CIRCLE('',#2294,22.0);
#2296=EDGE_CURVE('',#2288,#2290,#2295,.T.);
#2297=ORIENTED_EDGE('',*,*,#2296,.T.);
#2298=CARTESIAN_POINT('',(160.0,-18.250000000000000,90.0));
#2299=DIRECTION('',(0.0,-1.0,0.0));
#2300=DIRECTION('',(1.0,0.0,0.0));
#2301=AXIS2_PLACEMENT_3D('',#2298,#2299,#2300);
#2302=CIRCLE('',#2301,22.0);
#2303=EDGE_CURVE('',#2288,#2290,#2302,.T.);
#2304=ORIENTED_EDGE('',*,*,#2303,.F.);
#2305=EDGE_LOOP('',(#2297,#2304));
#2306=FACE_BOUND('',#2305,.F.);
#2307=ORIENTED_EDGE('',*,*,#1970,.T.);
#2308=ORIENTED_EDGE('',*,*,#2029,.T.);
#2309=EDGE_LOOP('',(#2307,#2308));
#2310=FACE_BOUND('',#2309,.F.);
```

```
#2311=CARTESIAN_POINT('',(160.0,-18.250000000000000,90.0));
#2312=DIRECTION('',(0.0,-1.0,0.0));
#2313=DIRECTION('',(1.0,0.0,0.0));
#2314=AXIS2_PLACEMENT_3D('',#2311,#2312,#2313);
#2315=PLANE('',#2314);
#2316=ADVANCED_FACE('',(#2306,#2310),#2315,.T.);
#2317=SHAPE_ASPECT('featured shape: face 36',$,#38,.T.);
#2318=PROPERTY_DEFINITION('',$,#2317);
#2319=FACE_SHAPE_REPRESENTATION('',(#2316),#29);
#2320=PROPERTY_DEFINITION_REPRESENTATION(#2318,#2319);
#2324=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MICRO.,.METRE.));
#2329=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(1.600000000000000),#2324)REPRESENTATION_ITEM('surface
finish'));
#2330=REPRESENTATION('surface finish value',(#2329),#29);
#2331=PROPERTY_DEFINITION('surface property','surface finish',#2317);
#2332=PROPERTY_DEFINITION('surface property','surface finish',#2317);
#2333=PROPERTY_DEFINITION_REPRESENTATION(#2332,#2330);
#2334=PROPERTY_DEFINITION_RELATIONSHIP('','',#2331,#2332);
#2335=CARTESIAN_POINT('',(182.0,-21.750000000000000,90.0));
#2336=VERTEX_POINT('',#2335);
#2337=DIRECTION('',(0.0,-1.0,0.0));
#2338=VECTOR('',#2337,3.500000000000000);
#2339=LINE('',#2287,#2338);
#2340=EDGE_CURVE('',#2288,#2336,#2339,.T.);
#2341=ORIENTED_EDGE('',*,*,#2340,.F.);
#2342=ORIENTED_EDGE('',*,*,#2303,.T.);
#2343=CARTESIAN_POINT('',(138.0,-21.750000000000000,90.0));
#2344=VERTEX_POINT('',#2343);
#2345=DIRECTION('',(0.0,-1.0,0.0));
#2346=VECTOR('',#2345,3.500000000000000);
#2347=LINE('',#2289,#2346);
#2348=EDGE_CURVE('',#2290,#2344,#2347,.T.);
#2349=ORIENTED_EDGE('',*,*,#2348,.T.);
#2350=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2351=DIRECTION('',(0.0,-1.0,0.0));
#2352=DIRECTION('',(1.0,0.0,0.0));
#2353=AXIS2_PLACEMENT_3D('',#2350,#2351,#2352);
#2354=CIRCLE('',#2353,22.0);
#2355=EDGE_CURVE('',#2336,#2344,#2354,.T.);
#2356=ORIENTED_EDGE('',*,*,#2355,.F.);
#2357=EDGE_LOOP('',(#2341,#2342,#2349,#2356));
#2358=FACE_BOUND('',#2357,.F.);
#2359=CARTESIAN_POINT('',(160.0,-18.074999999999999,90.0));
#2360=DIRECTION('',(0.0,-1.0,0.0));
#2361=DIRECTION('',(1.0,0.0,0.0));
#2362=AXIS2_PLACEMENT_3D('',#2359,#2360,#2361);
#2363=CYLINDRICAL_SURFACE('',#2362,22.0);
#2364=ADVANCED_FACE('',(#2358),#2363,.F.);
#2365=SHAPE_ASPECT('featured shape: face 37',$,#38,.T.);
#2366=PROPERTY_DEFINITION('',$,#2365);
#2367=FACE_SHAPE_REPRESENTATION('',(#2364),#29);
#2368=PROPERTY_DEFINITION_REPRESENTATION(#2366,#2367);
#2369=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2370=DIRECTION('',(0.0,1.0,0.0));
#2371=DIRECTION('',(1.0,0.0,0.0));
```



```
#2372=AXIS2_PLACEMENT_3D('',#2369,#2370,#2371);
#2373=CIRCLE('',#2372,22.0);
#2374=EDGE_CURVE('',#2336,#2344,#2373,.T.);
#2375=ORIENTED_EDGE('',*,*,#2374,.F.);
#2376=ORIENTED_EDGE('',*,*,#2355,.T.);
#2377=EDGE_LOOP('',(#2375,#2376));
#2378=FACE_BOUND('',#2377,.F.);
#2379=ORIENTED_EDGE('',*,*,#2010,.F.);
#2380=ORIENTED_EDGE('',*,*,#2053,.F.);
#2381=EDGE_LOOP('',(#2379,#2380));
#2382=FACE_BOUND('',#2381,.F.);
#2383=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2384=DIRECTION('',(0.0,-1.0,0.0));
#2385=DIRECTION('',(1.0,0.0,0.0));
#2386=AXIS2_PLACEMENT_3D('',#2383,#2384,#2385);
#2387=PLANE('',#2386);
#2388=ADVANCED_FACE('',(#2378,#2382),#2387,.F.);
#2389=SHAPE_ASPECT('featured shape: face 38',$,#38,.T.);
#2390=PROPERTY_DEFINITION('',$,#2389);
#2391=FACE_SHAPE_REPRESENTATION('',(#2388),#29);
#2392=PROPERTY_DEFINITION_REPRESENTATION(#2390,#2391);
#2393=ORIENTED_EDGE('',*,*,#2340,.T.);
#2394=ORIENTED_EDGE('',*,*,#2374,.T.);
#2395=ORIENTED_EDGE('',*,*,#2348,.F.);
#2396=ORIENTED_EDGE('',*,*,#2296,.F.);
#2397=EDGE_LOOP('',(#2393,#2394,#2395,#2396));
#2398=FACE_BOUND('',#2397,.F.);
#2399=CARTESIAN_POINT('',(160.0,-18.074999999999999,90.0));
#2400=DIRECTION('',(0.0,-1.0,0.0));
#2401=DIRECTION('',(1.0,0.0,0.0));
#2402=AXIS2_PLACEMENT_3D('',#2399,#2400,#2401);
#2403=CYLINDRICAL_SURFACE('',#2402,22.0);
#2404=ADVANCED_FACE('',(#2398),#2403,.F.);
#2405=SHAPE_ASPECT('featured shape: face 39',$,#38,.T.);
#2406=PROPERTY_DEFINITION('',$,#2405);
#2407=FACE_SHAPE_REPRESENTATION('',(#2404),#29);
#2408=PROPERTY_DEFINITION_REPRESENTATION(#2406,#2407);
#2409=CARTESIAN_POINT('',(167.500000000000000,-30.0,30.0));
#2410=VERTEX_POINT('',#2409);
#2411=CARTESIAN_POINT('',(152.500000000000000,-30.0,30.0));
#2412=VERTEX_POINT('',#2411);
#2413=CARTESIAN_POINT('',(160.0,-30.0,30.0));
#2414=DIRECTION('',(0.0,-1.0,0.0));
#2415=DIRECTION('',(1.0,0.0,0.0));
#2416=AXIS2_PLACEMENT_3D('',#2413,#2414,#2415);
#2417=CIRCLE('',#2416,7.500000000000000);
#2418=EDGE_CURVE('',#2410,#2412,#2417,.T.);
#2419=ORIENTED_EDGE('',*,*,#2418,.T.);
#2420=CARTESIAN_POINT('',(160.0,-30.0,30.0));
#2421=DIRECTION('',(0.0,1.0,0.0));
#2422=DIRECTION('',(1.0,0.0,0.0));
#2423=AXIS2_PLACEMENT_3D('',#2420,#2421,#2422);
#2424=CIRCLE('',#2423,7.500000000000000);
#2425=EDGE_CURVE('',#2410,#2412,#2424,.T.);
#2426=ORIENTED_EDGE('',*,*,#2425,.F.);
#2427=EDGE_LOOP('',(#2419,#2426));
```

```
#2428=FACE_BOUND('',#2427,.F.);
#2429=CARTESIAN_POINT('',(160.0,-30.0,30.0));
#2430=DIRECTION('',(0.0,1.0,0.0));
#2431=DIRECTION('',(1.0,0.0,0.0));
#2432=AXIS2_PLACEMENT_3D('',#2429,#2430,#2431);
#2433=PLANE('',#2432);
#2434=ADVANCED_FACE('',(#2428),#2433,.T.);
#2435=SHAPE_ASPECT('featured shape: face 40',$,#38,.T.);
#2436=PROPERTY_DEFINITION('',$,#2435);
#2437=FACE_SHAPE_REPRESENTATION('',(#2434),#29);
#2438=PROPERTY_DEFINITION_REPRESENTATION(#2436,#2437);
#2439=CARTESIAN_POINT('',(167.500000000000,-15.0,30.0));
#2440=VERTEX_POINT('',#2439);
#2441=DIRECTION('',(0.0,1.0,0.0));
#2442=VECTOR('',#2441,15.0);
#2443=LINE('',#2409,#2442);
#2444=EDGE_CURVE('',#2410,#2440,#2443,.T.);
#2445=ORIENTED_EDGE('',*,*,#2444,.F.);
#2446=ORIENTED_EDGE('',*,*,#2425,.T.);
#2447=CARTESIAN_POINT('',(152.500000000000,-15.0,30.0));
#2448=VERTEX_POINT('',#2447);
#2449=DIRECTION('',(0.0,1.0,0.0));
#2450=VECTOR('',#2449,15.0);
#2451=LINE('',#2411,#2450);
#2452=EDGE_CURVE('',#2412,#2448,#2451,.T.);
#2453=ORIENTED_EDGE('',*,*,#2452,.T.);
#2454=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2455=DIRECTION('',(0.0,1.0,0.0));
#2456=DIRECTION('',(1.0,0.0,0.0));
#2457=AXIS2_PLACEMENT_3D('',#2454,#2455,#2456);
#2458=CIRCLE('',#2457,7.500000000000000);
#2459=EDGE_CURVE('',#2440,#2448,#2458,.T.);
#2460=ORIENTED_EDGE('',*,*,#2459,.F.);
#2461=EDGE_LOOP('',(#2445,#2446,#2453,#2460));
#2462=FACE_BOUND('',#2461,.F.);
#2463=CARTESIAN_POINT('',(160.0,-31.500000000000000,30.0));
#2464=DIRECTION('',(0.0,1.0,0.0));
#2465=DIRECTION('',(1.0,0.0,0.0));
#2466=AXIS2_PLACEMENT_3D('',#2463,#2464,#2465);
#2467=CYLINDRICAL_SURFACE('',#2466,7.500000000000000);
#2468=ADVANCED_FACE('',(#2462),#2467,.F.);
#2469=SHAPE_ASPECT('featured shape: face 41',$,#38,.T.);
#2470=PROPERTY_DEFINITION('',$,#2469);
#2471=FACE_SHAPE_REPRESENTATION('',(#2468),#29);
#2472=PROPERTY_DEFINITION_REPRESENTATION(#2470,#2471);
#2473=CARTESIAN_POINT('',(172.500000000000,-15.0,30.0));
#2474=VERTEX_POINT('',#2473);
#2475=CARTESIAN_POINT('',(147.500000000000,-15.0,30.0));
#2476=VERTEX_POINT('',#2475);
#2477=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2478=DIRECTION('',(0.0,1.0,0.0));
#2479=DIRECTION('',(1.0,0.0,0.0));
#2480=AXIS2_PLACEMENT_3D('',#2477,#2478,#2479);
#2481=CIRCLE('',#2480,12.500000000000000);
#2482=EDGE_CURVE('',#2474,#2476,#2481,.T.);
#2483=ORIENTED_EDGE('',*,*,#2482,.F.);
```



```
#2484=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2485=DIRECTION('',(0.0,-1.0,0.0));
#2486=DIRECTION('',(1.0,0.0,0.0));
#2487=AXIS2_PLACEMENT_3D('',#2484,#2485,#2486);
#2488=CIRCLE('',#2487,12.500000000000000);
#2489=EDGE_CURVE('',#2474,#2476,#2488,.T.);
#2490=ORIENTED_EDGE('',*,*,#2489,.T.);
#2491=EDGE_LOOP('',(#2483,#2490));
#2492=FACE_BOUND('',#2491,.F.);
#2493=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2494=DIRECTION('',(0.0,-1.0,0.0));
#2495=DIRECTION('',(1.0,0.0,0.0));
#2496=AXIS2_PLACEMENT_3D('',#2493,#2494,#2495);
#2497=CIRCLE('',#2496,7.500000000000000);
#2498=EDGE_CURVE('',#2440,#2448,#2497,.T.);
#2499=ORIENTED_EDGE('',*,*,#2498,.F.);
#2500=ORIENTED_EDGE('',*,*,#2459,.T.);
#2501=EDGE_LOOP('',(#2499,#2500));
#2502=FACE_BOUND('',#2501,.F.);
#2503=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2504=DIRECTION('',(0.0,1.0,0.0));
#2505=DIRECTION('',(1.0,0.0,0.0));
#2506=AXIS2_PLACEMENT_3D('',#2503,#2504,#2505);
#2507=PLANE('',#2506);
#2508=ADVANCED_FACE('',(#2492,#2502),#2507,.T.);
#2509=SHAPE_ASPECT('featured shape: face 42',$,#38,.T.);
#2510=PROPERTY_DEFINITION('',$,#2509);
#2511=FACE_SHAPE_REPRESENTATION('',(#2508),#29);
#2512=PROPERTY_DEFINITION_REPRESENTATION(#2510,#2511);
#2513=ORIENTED_EDGE('',*,*,#813,.F.);
#2514=DIRECTION('',(0.0,1.0,0.0));
#2515=VECTOR('',#2514,15.0);
#2516=LINE('',#2473,#2515);
#2517=EDGE_CURVE('',#2474,#805,#2516,.T.);
#2518=ORIENTED_EDGE('',*,*,#2517,.F.);
#2519=ORIENTED_EDGE('',*,*,#2482,.T.);
#2520=DIRECTION('',(0.0,1.0,0.0));
#2521=VECTOR('',#2520,15.0);
#2522=LINE('',#2475,#2521);
#2523=EDGE_CURVE('',#2476,#807,#2522,.T.);
#2524=ORIENTED_EDGE('',*,*,#2523,.T.);
#2525=EDGE_LOOP('',(#2513,#2518,#2519,#2524));
#2526=FACE_BOUND('',#2525,.F.);
#2527=CARTESIAN_POINT('',(160.0,-31.500000000000000,30.0));
#2528=DIRECTION('',(0.0,1.0,0.0));
#2529=DIRECTION('',(1.0,0.0,0.0));
#2530=AXIS2_PLACEMENT_3D('',#2527,#2528,#2529);
#2531=CYLINDRICAL_SURFACE('',#2530,12.500000000000000);
#2532=ADVANCED_FACE('',(#2526),#2531,.F.);
#2533=SHAPE_ASPECT('featured shape: face 43',$,#38,.T.);
#2534=PROPERTY_DEFINITION('',$,#2533);
#2535=FACE_SHAPE_REPRESENTATION('',(#2532),#29);
#2536=PROPERTY_DEFINITION_REPRESENTATION(#2534,#2535);
#2537=ORIENTED_EDGE('',*,*,#2444,.T.);
#2538=ORIENTED_EDGE('',*,*,#2498,.T.);
#2539=ORIENTED_EDGE('',*,*,#2452,.F.);
```

```
#2540=ORIENTED_EDGE('',*,*,#2418,.F.);
#2541=EDGE_LOOP('',(#2537,#2538,#2539,#2540));
#2542=FACE_BOUND('',#2541,.F.);
#2543=CARTESIAN_POINT('',(160.0,-31.500000000000000,30.0));
#2544=DIRECTION('',(0.0,1.0,0.0));
#2545=DIRECTION('',(1.0,0.0,0.0));
#2546=AXIS2_PLACEMENT_3D('',#2543,#2544,#2545);
#2547=CYLINDRICAL_SURFACE('',#2546,7.500000000000000);
#2548=ADVANCED_FACE('',(#2542),#2547,.F.);
#2549=SHAPE_ASPECT('featured shape: face 44',$,#38,.T.);
#2550=PROPERTY_DEFINITION('',$,#2549);
#2551=FACE_SHAPE_REPRESENTATION('',(#2548),#29);
#2552=PROPERTY_DEFINITION_REPRESENTATION(#2550,#2551);
#2553=ORIENTED_EDGE('',*,*,#820,.F.);
#2554=ORIENTED_EDGE('',*,*,#2523,.F.);
#2555=ORIENTED_EDGE('',*,*,#2489,.F.);
#2556=ORIENTED_EDGE('',*,*,#2517,.T.);
#2557=EDGE_LOOP('',(#2553,#2554,#2555,#2556));
#2558=FACE_BOUND('',#2557,.F.);
#2559=CARTESIAN_POINT('',(160.0,-31.500000000000000,30.0));
#2560=DIRECTION('',(0.0,1.0,0.0));
#2561=DIRECTION('',(1.0,0.0,0.0));
#2562=AXIS2_PLACEMENT_3D('',#2559,#2560,#2561);
#2563=CYLINDRICAL_SURFACE('',#2562,12.500000000000000);
#2564=ADVANCED_FACE('',(#2558),#2563,.F.);
#2565=SHAPE_ASPECT('featured shape: face 45',$,#38,.T.);
#2566=PROPERTY_DEFINITION('',$,#2565);
#2567=FACE_SHAPE_REPRESENTATION('',(#2564),#29);
#2568=PROPERTY_DEFINITION_REPRESENTATION(#2566,#2567);
#2569=ORIENTED_EDGE('',*,*,#1654,.T.);
#2570=CARTESIAN_POINT('',(152.500000000000000,-15.0,150.0));
#2571=VERTEX_POINT('',#2570);
#2572=DIRECTION('',(0.0,1.0,0.0));
#2573=VECTOR('',#2572,25.0);
#2574=LINE('',#1647,#2573);
#2575=EDGE_CURVE('',#1648,#2571,#2574,.T.);
#2576=ORIENTED_EDGE('',*,*,#2575,.T.);
#2577=CARTESIAN_POINT('',(167.500000000000000,-15.0,150.0));
#2578=VERTEX_POINT('',#2577);
#2579=CARTESIAN_POINT('',(160.0,-15.0,150.0));
#2580=DIRECTION('',(0.0,1.0,0.0));
#2581=DIRECTION('',(1.0,0.0,0.0));
#2582=AXIS2_PLACEMENT_3D('',#2579,#2580,#2581);
#2583=CIRCLE('',#2582,7.500000000000000);
#2584=EDGE_CURVE('',#2578,#2571,#2583,.T.);
#2585=ORIENTED_EDGE('',*,*,#2584,.F.);
#2586=DIRECTION('',(0.0,1.0,0.0));
#2587=VECTOR('',#2586,25.0);
#2588=LINE('',#1645,#2587);
#2589=EDGE_CURVE('',#1646,#2578,#2588,.T.);
#2590=ORIENTED_EDGE('',*,*,#2589,.F.);
#2591=EDGE_LOOP('',(#2569,#2576,#2585,#2590));
#2592=FACE_BOUND('',#2591,.F.);
#2593=CARTESIAN_POINT('',(160.0,-42.0,150.0));
#2594=DIRECTION('',(0.0,1.0,0.0));
#2595=DIRECTION('',(1.0,0.0,0.0));
```



```
#2596=AXIS2_PLACEMENT_3D('',#2593,#2594,#2595);
#2597=CYLINDRICAL_SURFACE('',#2596,7.500000000000000);
#2598=ADVANCED_FACE('',(#2592),#2597,.F.);
#2599=SHAPE_ASPECT('featured shape: face 46',$,#38,.T.);
#2600=PROPERTY_DEFINITION('',$,#2599);
#2601=FACE_SHAPE_REPRESENTATION('',(#2598),#29);
#2602=PROPERTY_DEFINITION_REPRESENTATION(#2600,#2601);
#2603=CARTESIAN_POINT('',(172.500000000000000,-15.0,150.0));
#2604=VERTEX_POINT('',#2603);
#2605=CARTESIAN_POINT('',(147.500000000000000,-15.0,150.0));
#2606=VERTEX_POINT('',#2605);
#2607=CARTESIAN_POINT('',(160.0,-15.0,150.0));
#2608=DIRECTION('',(0.0,1.0,0.0));
#2609=DIRECTION('',(1.0,0.0,0.0));
#2610=AXIS2_PLACEMENT_3D('',#2607,#2608,#2609);
#2611=CIRCLE('',#2610,12.500000000000000);
#2612=EDGE_CURVE('',#2604,#2606,#2611,.T.);
#2613=ORIENTED_EDGE('',*,*,#2612,.F.);
#2614=CARTESIAN_POINT('',(160.0,-15.0,150.0));
#2615=DIRECTION('',(0.0,-1.0,0.0));
#2616=DIRECTION('',(1.0,0.0,0.0));
#2617=AXIS2_PLACEMENT_3D('',#2614,#2615,#2616);
#2618=CIRCLE('',#2617,12.500000000000000);
#2619=EDGE_CURVE('',#2604,#2606,#2618,.T.);
#2620=ORIENTED_EDGE('',*,*,#2619,.T.);
#2621=EDGE_LOOP('',(#2613,#2620));
#2622=FACE_BOUND('',#2621,.F.);
#2623=CARTESIAN_POINT('',(160.0,-15.0,150.0));
#2624=DIRECTION('',(0.0,-1.0,0.0));
#2625=DIRECTION('',(1.0,0.0,0.0));
#2626=AXIS2_PLACEMENT_3D('',#2623,#2624,#2625);
#2627=CIRCLE('',#2626,7.500000000000000);
#2628=EDGE_CURVE('',#2578,#2571,#2627,.T.);
#2629=ORIENTED_EDGE('',*,*,#2628,.F.);
#2630=ORIENTED_EDGE('',*,*,#2584,.T.);
#2631=EDGE_LOOP('',(#2629,#2630));
#2632=FACE_BOUND('',#2631,.F.);
#2633=CARTESIAN_POINT('',(160.0,-15.0,150.0));
#2634=DIRECTION('',(0.0,1.0,0.0));
#2635=DIRECTION('',(1.0,0.0,0.0));
#2636=AXIS2_PLACEMENT_3D('',#2633,#2634,#2635);
#2637=PLANE('',#2636);
#2638=ADVANCED_FACE('',(#2622,#2632),#2637,.T.);
#2639=SHAPE_ASPECT('featured shape: face 47',$,#38,.T.);
#2640=PROPERTY_DEFINITION('',$,#2639);
#2641=FACE_SHAPE_REPRESENTATION('',(#2638),#29);
#2642=PROPERTY_DEFINITION_REPRESENTATION(#2640,#2641);
#2643=ORIENTED_EDGE('',*,*,#833,.F.);
#2644=DIRECTION('',(0.0,1.0,0.0));
#2645=VECTOR('',#2644,15.0);
#2646=LINE('',#2603,#2645);
#2647=EDGE_CURVE('',#2604,#825,#2646,.T.);
#2648=ORIENTED_EDGE('',*,*,#2647,.F.);
#2649=ORIENTED_EDGE('',*,*,#2612,.T.);
#2650=DIRECTION('',(0.0,1.0,0.0));
#2651=VECTOR('',#2650,15.0);
```



```
#2652=LINE('',#2605,#2651);
#2653=EDGE_CURVE('',#2606,#827,#2652,.T.);
#2654=ORIENTED_EDGE('',*,*,#2653,.T.);
#2655=EDGE_LOOP('',(#2643,#2648,#2649,#2654));
#2656=FACE_BOUND('',#2655,.F.);
#2657=CARTESIAN_POINT('',(160.0,-42.0,150.0));
#2658=DIRECTION('',(0.0,1.0,0.0));
#2659=DIRECTION('',(1.0,0.0,0.0));
#2660=AXIS2_PLACEMENT_3D('',#2657,#2658,#2659);
#2661=CYLINDRICAL_SURFACE('',#2660,12.500000000000000);
#2662=ADVANCED_FACE('',(#2656),#2661,.F.);
#2663=SHAPE_ASPECT('featured shape: face 48',$,#38,.T.);
#2664=PROPERTY_DEFINITION('',$,#2663);
#2665=FACE_SHAPE_REPRESENTATION('',(#2662),#29);
#2666=PROPERTY_DEFINITION_REPRESENTATION(#2664,#2665);
#2667=ORIENTED_EDGE('',*,*,#1661,.T.);
#2668=ORIENTED_EDGE('',*,*,#2589,.T.);
#2669=ORIENTED_EDGE('',*,*,#2628,.T.);
#2670=ORIENTED_EDGE('',*,*,#2575,.F.);
#2671=EDGE_LOOP('',(#2667,#2668,#2669,#2670));
#2672=FACE_BOUND('',#2671,.F.);
#2673=CARTESIAN_POINT('',(160.0,-42.0,150.0));
#2674=DIRECTION('',(0.0,1.0,0.0));
#2675=DIRECTION('',(1.0,0.0,0.0));
#2676=AXIS2_PLACEMENT_3D('',#2673,#2674,#2675);
#2677=CYLINDRICAL_SURFACE('',#2676,7.500000000000000);
#2678=ADVANCED_FACE('',(#2672),#2677,.F.);
#2679=SHAPE_ASPECT('featured shape: face 49',$,#38,.T.);
#2680=PROPERTY_DEFINITION('',$,#2679);
#2681=FACE_SHAPE_REPRESENTATION('',(#2678),#29);
#2682=PROPERTY_DEFINITION_REPRESENTATION(#2680,#2681);
#2683=ORIENTED_EDGE('',*,*,#840,.F.);
#2684=ORIENTED_EDGE('',*,*,#2653,.F.);
#2685=ORIENTED_EDGE('',*,*,#2619,.F.);
#2686=ORIENTED_EDGE('',*,*,#2647,.T.);
#2687=EDGE_LOOP('',(#2683,#2684,#2685,#2686));
#2688=FACE_BOUND('',#2687,.F.);
#2689=CARTESIAN_POINT('',(160.0,-42.0,150.0));
#2690=DIRECTION('',(0.0,1.0,0.0));
#2691=DIRECTION('',(1.0,0.0,0.0));
#2692=AXIS2_PLACEMENT_3D('',#2689,#2690,#2691);
#2693=CYLINDRICAL_SURFACE('',#2692,12.500000000000000);
#2694=ADVANCED_FACE('',(#2688),#2693,.F.);
#2695=SHAPE_ASPECT('featured shape: face 50',$,#38,.T.);
#2696=PROPERTY_DEFINITION('',$,#2695);
#2697=FACE_SHAPE_REPRESENTATION('',(#2694),#29);
#2698=PROPERTY_DEFINITION_REPRESENTATION(#2696,#2697);
#2699=ORIENTED_EDGE('',*,*,#414,.T.);
#2700=CARTESIAN_POINT('',(196.0,-20.0,27.500000000000000));
#2701=VERTEX_POINT('',#2700);
#2702=DIRECTION('',(0.0,1.0,0.0));
#2703=VECTOR('',#2702,30.0);
#2704=LINE('',#407,#2703);
#2705=EDGE_CURVE('',#408,#2701,#2704,.T.);
#2706=ORIENTED_EDGE('',*,*,#2705,.T.);
#2707=CARTESIAN_POINT('',(214.0,-20.0,27.500000000000000));
```



```
#2708=VERTEX_POINT('',#2707);
#2709=CARTESIAN_POINT('',(205.0,-20.0,27.500000000000000));
#2710=DIRECTION('',(0.0,1.0,0.0));
#2711=DIRECTION('',(1.0,0.0,0.0));
#2712=AXIS2_PLACEMENT_3D('',#2709,#2710,#2711);
#2713=CIRCLE('',#2712,9.0);
#2714=EDGE_CURVE('',#2708,#2701,#2713,.T.);
#2715=ORIENTED_EDGE('',*,*,#2714,.F.);
#2716=DIRECTION('',(0.0,1.0,0.0));
#2717=VECTOR('',#2716,30.0);
#2718=LINE('',#405,#2717);
#2719=EDGE_CURVE('',#406,#2708,#2718,.T.);
#2720=ORIENTED_EDGE('',*,*,#2719,.F.);
#2721=EDGE_LOOP('',(#2699,#2706,#2715,#2720));
#2722=FACE_BOUND('',#2721,.F.);
#2723=CARTESIAN_POINT('',(205.0,-52.500000000000000,27.500000000000000));
#2724=DIRECTION('',(0.0,1.0,0.0));
#2725=DIRECTION('',(1.0,0.0,0.0));
#2726=AXIS2_PLACEMENT_3D('',#2723,#2724,#2725);
#2727=CYLINDRICAL_SURFACE('',#2726,9.0);
#2728=ADVANCED_FACE('',(#2722),#2727,.F.);
#2729=SHAPE_ASPECT('featured shape: face 51',$,#38,.T.);
#2730=PROPERTY_DEFINITION('',$,#2729);
#2731=FACE_SHAPE_REPRESENTATION('',(#2728),#29);
#2732=PROPERTY_DEFINITION_REPRESENTATION(#2730,#2731);
#2733=CARTESIAN_POINT('',(218.0,-20.0,27.500000000000000));
#2734=VERTEX_POINT('',#2733);
#2735=CARTESIAN_POINT('',(192.0,-20.0,27.500000000000000));
#2736=VERTEX_POINT('',#2735);
#2737=CARTESIAN_POINT('',(205.0,-20.0,27.500000000000000));
#2738=DIRECTION('',(0.0,1.0,0.0));
#2739=DIRECTION('',(1.0,0.0,0.0));
#2740=AXIS2_PLACEMENT_3D('',#2737,#2738,#2739);
#2741=CIRCLE('',#2740,13.0);
#2742=EDGE_CURVE('',#2734,#2736,#2741,.T.);
#2743=ORIENTED_EDGE('',*,*,#2742,.F.);
#2744=CARTESIAN_POINT('',(205.0,-20.0,27.500000000000000));
#2745=DIRECTION('',(0.0,-1.0,0.0));
#2746=DIRECTION('',(1.0,0.0,0.0));
#2747=AXIS2_PLACEMENT_3D('',#2744,#2745,#2746);
#2748=CIRCLE('',#2747,13.0);
#2749=EDGE_CURVE('',#2734,#2736,#2748,.T.);
#2750=ORIENTED_EDGE('',*,*,#2749,.T.);
#2751=EDGE_LOOP('',(#2743,#2750));
#2752=FACE_BOUND('',#2751,.F.);
#2753=CARTESIAN_POINT('',(205.0,-20.0,27.500000000000000));
#2754=DIRECTION('',(0.0,-1.0,0.0));
#2755=DIRECTION('',(1.0,0.0,0.0));
#2756=AXIS2_PLACEMENT_3D('',#2753,#2754,#2755);
#2757=CIRCLE('',#2756,9.0);
#2758=EDGE_CURVE('',#2708,#2701,#2757,.T.);
#2759=ORIENTED_EDGE('',*,*,#2758,.F.);
#2760=ORIENTED_EDGE('',*,*,#2714,.T.);
#2761=EDGE_LOOP('',(#2759,#2760));
#2762=FACE_BOUND('',#2761,.F.);
#2763=CARTESIAN_POINT('',(205.0,-20.0,27.500000000000000));
```

```
#2764=DIRECTION('',(0.0,1.0,0.0));
#2765=DIRECTION('',(1.0,0.0,0.0));
#2766=AXIS2_PLACEMENT_3D('',#2763,#2764,#2765);
#2767=PLANE('',#2766);
#2768=ADVANCED_FACE('',(#2752,#2762),#2767,.T.);
#2769=SHAPE_ASPECT('featured shape: face 52',$,#38,.T.);
#2770=PROPERTY_DEFINITION('',$,#2769);
#2771=FACE_SHAPE_REPRESENTATION('',(#2768),#29);
#2772=PROPERTY_DEFINITION_REPRESENTATION(#2770,#2771);
#2773=ORIENTED_EDGE('',*,*,#853,.F.);
#2774=DIRECTION('',(0.0,1.0,0.0));
#2775=VECTOR('',#2774,20.0);
#2776=LINE('',#2733,#2775);
#2777=EDGE_CURVE('',#2734,#845,#2776,.T.);
#2778=ORIENTED_EDGE('',*,*,#2777,.F.);
#2779=ORIENTED_EDGE('',*,*,#2742,.T.);
#2780=DIRECTION('',(0.0,1.0,0.0));
#2781=VECTOR('',#2780,20.0);
#2782=LINE('',#2735,#2781);
#2783=EDGE_CURVE('',#2736,#847,#2782,.T.);
#2784=ORIENTED_EDGE('',*,*,#2783,.T.);
#2785=EDGE_LOOP('',(#2773,#2778,#2779,#2784));
#2786=FACE_BOUND('',#2785,.F.);
#2787=CARTESIAN_POINT('',(205.0,-52.500000000000000,27.500000000000000));
#2788=DIRECTION('',(0.0,1.0,0.0));
#2789=DIRECTION('',(1.0,0.0,0.0));
#2790=AXIS2_PLACEMENT_3D('',#2787,#2788,#2789);
#2791=CYLINDRICAL_SURFACE('',#2790,13.0);
#2792=ADVANCED_FACE('',(#2786),#2791,.F.);
#2793=SHAPE_ASPECT('featured shape: face 53',$,#38,.T.);
#2794=PROPERTY_DEFINITION('',$,#2793);
#2795=FACE_SHAPE_REPRESENTATION('',(#2792),#29);
#2796=PROPERTY_DEFINITION_REPRESENTATION(#2794,#2795);
#2797=ORIENTED_EDGE('',*,*,#421,.T.);
#2798=ORIENTED_EDGE('',*,*,#2719,.T.);
#2799=ORIENTED_EDGE('',*,*,#2758,.T.);
#2800=ORIENTED_EDGE('',*,*,#2705,.F.);
#2801=EDGE_LOOP('',(#2797,#2798,#2799,#2800));
#2802=FACE_BOUND('',#2801,.F.);
#2803=CARTESIAN_POINT('',(205.0,-52.500000000000000,27.500000000000000));
#2804=DIRECTION('',(0.0,1.0,0.0));
#2805=DIRECTION('',(1.0,0.0,0.0));
#2806=AXIS2_PLACEMENT_3D('',#2803,#2804,#2805);
#2807=CYLINDRICAL_SURFACE('',#2806,9.0);
#2808=ADVANCED_FACE('',(#2802),#2807,.F.);
#2809=SHAPE_ASPECT('featured shape: face 54',$,#38,.T.);
#2810=PROPERTY_DEFINITION('',$,#2809);
#2811=FACE_SHAPE_REPRESENTATION('',(#2808),#29);
#2812=PROPERTY_DEFINITION_REPRESENTATION(#2810,#2811);
#2813=ORIENTED_EDGE('',*,*,#860,.F.);
#2814=ORIENTED_EDGE('',*,*,#2783,.F.);
#2815=ORIENTED_EDGE('',*,*,#2749,.F.);
#2816=ORIENTED_EDGE('',*,*,#2777,.T.);
#2817=EDGE_LOOP('',(#2813,#2814,#2815,#2816));
#2818=FACE_BOUND('',#2817,.F.);
#2819=CARTESIAN_POINT('',(205.0,-52.500000000000000,27.500000000000000));
```



```
#2820=DIRECTION('',(0.0,1.0,0.0));
#2821=DIRECTION('',(1.0,0.0,0.0));
#2822=AXIS2_PLACEMENT_3D('',#2819,#2820,#2821);
#2823=CYLINDRICAL_SURFACE('',#2822,13.0);
#2824=ADVANCED_FACE('',(#2818),#2823,.F.);
#2825=SHAPE_ASPECT('featured shape: face 55',$,#38,.T.);
#2826=PROPERTY_DEFINITION('',$,#2825);
#2827=FACE_SHAPE_REPRESENTATION('',(#2824),#29);
#2828=PROPERTY_DEFINITION_REPRESENTATION(#2826,#2827);
#2829=ORIENTED_EDGE('',*,*,#505,.T.);
#2830=CARTESIAN_POINT('',(196.0,-20.0,152.500000000000));
#2831=VERTEX_POINT('',#2830);
#2832=DIRECTION('',(0.0,1.0,0.0));
#2833=VECTOR('',#2832,30.0);
#2834=LINE('',#498,#2833);
#2835=EDGE_CURVE('',#499,#2831,#2834,.T.);
#2836=ORIENTED_EDGE('',*,*,#2835,.T.);
#2837=CARTESIAN_POINT('',(214.0,-20.0,152.500000000000));
#2838=VERTEX_POINT('',#2837);
#2839=CARTESIAN_POINT('',(205.0,-20.0,152.500000000000));
#2840=DIRECTION('',(0.0,1.0,0.0));
#2841=DIRECTION('',(1.0,0.0,0.0));
#2842=AXIS2_PLACEMENT_3D('',#2839,#2840,#2841);
#2843=CIRCLE('',#2842,9.0);
#2844=EDGE_CURVE('',#2838,#2831,#2843,.T.);
#2845=ORIENTED_EDGE('',*,*,#2844,.F.);
#2846=DIRECTION('',(0.0,1.0,0.0));
#2847=VECTOR('',#2846,30.0);
#2848=LINE('',#496,#2847);
#2849=EDGE_CURVE('',#497,#2838,#2848,.T.);
#2850=ORIENTED_EDGE('',*,*,#2849,.F.);
#2851=EDGE_LOOP('',(#2829,#2836,#2845,#2850));
#2852=FACE_BOUND('',#2851,.F.);
#2853=CARTESIAN_POINT('',(205.0,-52.500000000000,152.500000000000));
#2854=DIRECTION('',(0.0,1.0,0.0));
#2855=DIRECTION('',(1.0,0.0,0.0));
#2856=AXIS2_PLACEMENT_3D('',#2853,#2854,#2855);
#2857=CYLINDRICAL_SURFACE('',#2856,9.0);
#2858=ADVANCED_FACE('',(#2852),#2857,.F.);
#2859=SHAPE_ASPECT('featured shape: face 56',$,#38,.T.);
#2860=PROPERTY_DEFINITION('',$,#2859);
#2861=FACE_SHAPE_REPRESENTATION('',(#2858),#29);
#2862=PROPERTY_DEFINITION_REPRESENTATION(#2860,#2861);
#2863=CARTESIAN_POINT('',(218.0,-20.0,152.500000000000));
#2864=VERTEX_POINT('',#2863);
#2865=CARTESIAN_POINT('',(192.0,-20.0,152.500000000000));
#2866=VERTEX_POINT('',#2865);
#2867=CARTESIAN_POINT('',(205.0,-20.0,152.500000000000));
#2868=DIRECTION('',(0.0,1.0,0.0));
#2869=DIRECTION('',(1.0,0.0,0.0));
#2870=AXIS2_PLACEMENT_3D('',#2867,#2868,#2869);
#2871=CIRCLE('',#2870,13.0);
#2872=EDGE_CURVE('',#2864,#2866,#2871,.T.);
#2873=ORIENTED_EDGE('',*,*,#2872,.F.);
#2874=CARTESIAN_POINT('',(205.0,-20.0,152.500000000000));
#2875=DIRECTION('',(0.0,-1.0,0.0));
```

```
#2876=DIRECTION('',(1.0,0.0,0.0));
#2877=AXIS2_PLACEMENT_3D('',#2874,#2875,#2876);
#2878=CIRCLE('',#2877,13.0);
#2879=EDGE_CURVE('',#2864,#2866,#2878,.T.);
#2880=ORIENTED_EDGE('',*,*,#2879,.T.);
#2881=EDGE_LOOP('',(#2873,#2880));
#2882=FACE_BOUND('',#2881,.F.);
#2883=CARTESIAN_POINT('',(205.0,-20.0,152.500000000000000));
#2884=DIRECTION('',(0.0,-1.0,0.0));
#2885=DIRECTION('',(1.0,0.0,0.0));
#2886=AXIS2_PLACEMENT_3D('',#2883,#2884,#2885);
#2887=CIRCLE('',#2886,9.0);
#2888=EDGE_CURVE('',#2838,#2831,#2887,.T.);
#2889=ORIENTED_EDGE('',*,*,#2888,.F.);
#2890=ORIENTED_EDGE('',*,*,#2844,.T.);
#2891=EDGE_LOOP('',(#2889,#2890));
#2892=FACE_BOUND('',#2891,.F.);
#2893=CARTESIAN_POINT('',(205.0,-20.0,152.500000000000000));
#2894=DIRECTION('',(0.0,1.0,0.0));
#2895=DIRECTION('',(1.0,0.0,0.0));
#2896=AXIS2_PLACEMENT_3D('',#2893,#2894,#2895);
#2897=PLANE('',#2896);
#2898=ADVANCED_FACE('',(#2882,#2892),#2897,.T.);
#2899=SHAPE_ASPECT('featured shape: face 57',$,#38,.T.);
#2900=PROPERTY_DEFINITION('',$,#2899);
#2901=FACE_SHAPE_REPRESENTATION('',(#2898),#29);
#2902=PROPERTY_DEFINITION_REPRESENTATION(#2900,#2901);
#2903=ORIENTED_EDGE('',*,*,#873,.F.);
#2904=DIRECTION('',(0.0,1.0,0.0));
#2905=VECTOR('',#2904,20.0);
#2906=LINE('',#2863,#2905);
#2907=EDGE_CURVE('',#2864,#865,#2906,.T.);
#2908=ORIENTED_EDGE('',*,*,#2907,.F.);
#2909=ORIENTED_EDGE('',*,*,#2872,.T.);
#2910=DIRECTION('',(0.0,1.0,0.0));
#2911=VECTOR('',#2910,20.0);
#2912=LINE('',#2865,#2911);
#2913=EDGE_CURVE('',#2866,#867,#2912,.T.);
#2914=ORIENTED_EDGE('',*,*,#2913,.T.);
#2915=EDGE_LOOP('',(#2903,#2908,#2909,#2914));
#2916=FACE_BOUND('',#2915,.F.);
#2917=CARTESIAN_POINT('',(205.0,-52.500000000000000,152.500000000000000));
#2918=DIRECTION('',(0.0,1.0,0.0));
#2919=DIRECTION('',(1.0,0.0,0.0));
#2920=AXIS2_PLACEMENT_3D('',#2917,#2918,#2919);
#2921=CYLINDRICAL_SURFACE('',#2920,13.0);
#2922=ADVANCED_FACE('',(#2916),#2921,.F.);
#2923=SHAPE_ASPECT('featured shape: face 58',$,#38,.T.);
#2924=PROPERTY_DEFINITION('',$,#2923);
#2925=FACE_SHAPE_REPRESENTATION('',(#2922),#29);
#2926=PROPERTY_DEFINITION_REPRESENTATION(#2924,#2925);
#2927=ORIENTED_EDGE('',*,*,#512,.T.);
#2928=ORIENTED_EDGE('',*,*,#2849,.T.);
#2929=ORIENTED_EDGE('',*,*,#2888,.T.);
#2930=ORIENTED_EDGE('',*,*,#2835,.F.);
#2931=EDGE_LOOP('',(#2927,#2928,#2929,#2930));
```



```
#2932=FACE_BOUND('',#2931,.F.);
#2933=CARTESIAN_POINT('',(205.0,-52.500000000000000,152.500000000000000));
#2934=DIRECTION('',(0.0,1.0,0.0));
#2935=DIRECTION('',(1.0,0.0,0.0));
#2936=AXIS2_PLACEMENT_3D('',#2933,#2934,#2935);
#2937=CYLINDRICAL_SURFACE('',#2936,9.0);
#2938=ADVANCED_FACE('',(#2932),#2937,.F.);
#2939=SHAPE_ASPECT('featured shape: face 59',$,#38,.T.);
#2940=PROPERTY_DEFINITION('',$,#2939);
#2941=FACE_SHAPE_REPRESENTATION('',(#2938),#29);
#2942=PROPERTY_DEFINITION_REPRESENTATION(#2940,#2941);
#2943=ORIENTED_EDGE('',*,*,#880,.F.);
#2944=ORIENTED_EDGE('',*,*,#2913,.F.);
#2945=ORIENTED_EDGE('',*,*,#2879,.F.);
#2946=ORIENTED_EDGE('',*,*,#2907,.T.);
#2947=EDGE_LOOP('',(#2943,#2944,#2945,#2946));
#2948=FACE_BOUND('',#2947,.F.);
#2949=CARTESIAN_POINT('',(205.0,-52.500000000000000,152.500000000000000));
#2950=DIRECTION('',(0.0,1.0,0.0));
#2951=DIRECTION('',(1.0,0.0,0.0));
#2952=AXIS2_PLACEMENT_3D('',#2949,#2950,#2951);
#2953=CYLINDRICAL_SURFACE('',#2952,13.0);
#2954=ADVANCED_FACE('',(#2948),#2953,.F.);
#2955=SHAPE_ASPECT('featured shape: face 60',$,#38,.T.);
#2956=PROPERTY_DEFINITION('',$,#2955);
#2957=FACE_SHAPE_REPRESENTATION('',(#2954),#29);
#2958=PROPERTY_DEFINITION_REPRESENTATION(#2956,#2957);
#2959=ORIENTED_EDGE('',*,*,#318,.T.);
#2960=CARTESIAN_POINT('',(16.0,-20.0,27.500000000000000));
#2961=VERTEX_POINT('',#2960);
#2962=DIRECTION('',(0.0,1.0,0.0));
#2963=VECTOR('',#2962,30.0);
#2964=LINE('',#311,#2963);
#2965=EDGE_CURVE('',#312,#2961,#2964,.T.);
#2966=ORIENTED_EDGE('',*,*,#2965,.T.);
#2967=CARTESIAN_POINT('',(34.0,-20.0,27.500000000000000));
#2968=VERTEX_POINT('',#2967);
#2969=CARTESIAN_POINT('',(25.0,-20.0,27.500000000000000));
#2970=DIRECTION('',(0.0,1.0,0.0));
#2971=DIRECTION('',(1.0,0.0,0.0));
#2972=AXIS2_PLACEMENT_3D('',#2969,#2970,#2971);
#2973=CIRCLE('',#2972,9.0);
#2974=EDGE_CURVE('',#2968,#2961,#2973,.T.);
#2975=ORIENTED_EDGE('',*,*,#2974,.F.);
#2976=DIRECTION('',(0.0,1.0,0.0));
#2977=VECTOR('',#2976,30.0);
#2978=LINE('',#309,#2977);
#2979=EDGE_CURVE('',#310,#2968,#2978,.T.);
#2980=ORIENTED_EDGE('',*,*,#2979,.F.);
#2981=EDGE_LOOP('',(#2959,#2966,#2975,#2980));
#2982=FACE_BOUND('',#2981,.F.);
#2983=CARTESIAN_POINT('',(25.0,-52.500000000000000,27.500000000000000));
#2984=DIRECTION('',(0.0,1.0,0.0));
#2985=DIRECTION('',(1.0,0.0,0.0));
#2986=AXIS2_PLACEMENT_3D('',#2983,#2984,#2985);
#2987=CYLINDRICAL_SURFACE('',#2986,9.0);
```

```
#2988=ADVANCED_FACE('',(#2982),#2987,.F.);
#2989=SHAPE_ASPECT('featured shape: face 61',$,#38,.T.);
#2990=PROPERTY_DEFINITION('',$,#2989);
#2991=FACE_SHAPE_REPRESENTATION('',(#2988),#29);
#2992=PROPERTY_DEFINITION_REPRESENTATION(#2990,#2991);
#2993=CARTESIAN_POINT('',(38.0,-20.0,27.500000000000000));
#2994=VERTEX_POINT('',#2993);
#2995=CARTESIAN_POINT('',(12.0,-20.0,27.500000000000000));
#2996=VERTEX_POINT('',#2995);
#2997=CARTESIAN_POINT('',(25.0,-20.0,27.500000000000000));
#2998=DIRECTION('',(0.0,1.0,0.0));
#2999=DIRECTION('',(1.0,0.0,0.0));
#3000=AXIS2_PLACEMENT_3D('',#2997,#2998,#2999);
#3001=CIRCLE('',#3000,13.0);
#3002=EDGE_CURVE('',#2994,#2996,#3001,.T.);
#3003=ORIENTED_EDGE('',*,*,#3002,.F.);
#3004=CARTESIAN_POINT('',(25.0,-20.0,27.500000000000000));
#3005=DIRECTION('',(0.0,-1.0,0.0));
#3006=DIRECTION('',(1.0,0.0,0.0));
#3007=AXIS2_PLACEMENT_3D('',#3004,#3005,#3006);
#3008=CIRCLE('',#3007,13.0);
#3009=EDGE_CURVE('',#2994,#2996,#3008,.T.);
#3010=ORIENTED_EDGE('',*,*,#3009,.T.);
#3011=EDGE_LOOP('',(#3003,#3010));
#3012=FACE_BOUND('',#3011,.F.);
#3013=CARTESIAN_POINT('',(25.0,-20.0,27.500000000000000));
#3014=DIRECTION('',(0.0,-1.0,0.0));
#3015=DIRECTION('',(1.0,0.0,0.0));
#3016=AXIS2_PLACEMENT_3D('',#3013,#3014,#3015);
#3017=CIRCLE('',#3016,9.0);
#3018=EDGE_CURVE('',#2968,#2961,#3017,.T.);
#3019=ORIENTED_EDGE('',*,*,#3018,.F.);
#3020=ORIENTED_EDGE('',*,*,#2974,.T.);
#3021=EDGE_LOOP('',(#3019,#3020));
#3022=FACE_BOUND('',#3021,.F.);
#3023=CARTESIAN_POINT('',(25.0,-20.0,27.500000000000000));
#3024=DIRECTION('',(0.0,1.0,0.0));
#3025=DIRECTION('',(1.0,0.0,0.0));
#3026=AXIS2_PLACEMENT_3D('',#3023,#3024,#3025);
#3027=PLANE('',#3026);
#3028=ADVANCED_FACE('',(#3012,#3022),#3027,.T.);
#3029=SHAPE_ASPECT('featured shape: face 62',$,#38,.T.);
#3030=PROPERTY_DEFINITION('',$,#3029);
#3031=FACE_SHAPE_REPRESENTATION('',(#3028),#29);
#3032=PROPERTY_DEFINITION_REPRESENTATION(#3030,#3031);
#3033=ORIENTED_EDGE('',*,*,#893,.F.);
#3034=DIRECTION('',(0.0,1.0,0.0));
#3035=VECTOR('',#3034,20.0);
#3036=LINE('',#2993,#3035);
#3037=EDGE_CURVE('',#2994,#885,#3036,.T.);
#3038=ORIENTED_EDGE('',*,*,#3037,.F.);
#3039=ORIENTED_EDGE('',*,*,#3002,.T.);
#3040=DIRECTION('',(0.0,1.0,0.0));
#3041=VECTOR('',#3040,20.0);
#3042=LINE('',#2995,#3041);
#3043=EDGE_CURVE('',#2996,#887,#3042,.T.);
```



```
#3044=ORIENTED_EDGE('',*,*,#3043,.T.);
#3045=EDGE_LOOP('',(#3033,#3038,#3039,#3044));
#3046=FACE_BOUND('',#3045,.F.);
#3047=CARTESIAN_POINT('',(25.0,-52.500000000000000,27.500000000000000));
#3048=DIRECTION('',(0.0,1.0,0.0));
#3049=DIRECTION('',(1.0,0.0,0.0));
#3050=AXIS2_PLACEMENT_3D('',#3047,#3048,#3049);
#3051=CYLINDRICAL_SURFACE('',#3050,13.0);
#3052=ADVANCED_FACE('',(#3046),#3051,.F.);
#3053=SHAPE_ASPECT('featured shape: face 63',$,#38,.T.);
#3054=PROPERTY_DEFINITION('',$,#3053);
#3055=FACE_SHAPE_REPRESENTATION('',(#3052),#29);
#3056=PROPERTY_DEFINITION_REPRESENTATION(#3054,#3055);
#3057=ORIENTED_EDGE('',*,*,#325,.T.);
#3058=ORIENTED_EDGE('',*,*,#2979,.T.);
#3059=ORIENTED_EDGE('',*,*,#3018,.T.);
#3060=ORIENTED_EDGE('',*,*,#2965,.F.);
#3061=EDGE_LOOP('',(#3057,#3058,#3059,#3060));
#3062=FACE_BOUND('',#3061,.F.);
#3063=CARTESIAN_POINT('',(25.0,-52.500000000000000,27.500000000000000));
#3064=DIRECTION('',(0.0,1.0,0.0));
#3065=DIRECTION('',(1.0,0.0,0.0));
#3066=AXIS2_PLACEMENT_3D('',#3063,#3064,#3065);
#3067=CYLINDRICAL_SURFACE('',#3066,9.0);
#3068=ADVANCED_FACE('',(#3062),#3067,.F.);
#3069=SHAPE_ASPECT('featured shape: face 64',$,#38,.T.);
#3070=PROPERTY_DEFINITION('',$,#3069);
#3071=FACE_SHAPE_REPRESENTATION('',(#3068),#29);
#3072=PROPERTY_DEFINITION_REPRESENTATION(#3070,#3071);
#3073=ORIENTED_EDGE('',*,*,#900,.F.);
#3074=ORIENTED_EDGE('',*,*,#3043,.F.);
#3075=ORIENTED_EDGE('',*,*,#3009,.F.);
#3076=ORIENTED_EDGE('',*,*,#3037,.T.);
#3077=EDGE_LOOP('',(#3073,#3074,#3075,#3076));
#3078=FACE_BOUND('',#3077,.F.);
#3079=CARTESIAN_POINT('',(25.0,-52.500000000000000,27.500000000000000));
#3080=DIRECTION('',(0.0,1.0,0.0));
#3081=DIRECTION('',(1.0,0.0,0.0));
#3082=AXIS2_PLACEMENT_3D('',#3079,#3080,#3081);
#3083=CYLINDRICAL_SURFACE('',#3082,13.0);
#3084=ADVANCED_FACE('',(#3078),#3083,.F.);
#3085=SHAPE_ASPECT('featured shape: face 65',$,#38,.T.);
#3086=PROPERTY_DEFINITION('',$,#3085);
#3087=FACE_SHAPE_REPRESENTATION('',(#3084),#29);
#3088=PROPERTY_DEFINITION_REPRESENTATION(#3086,#3087);
#3089=ORIENTED_EDGE('',*,*,#596,.T.);
#3090=CARTESIAN_POINT('',(16.0,-20.0,152.500000000000000));
#3091=VERTEX_POINT('',#3090);
#3092=DIRECTION('',(0.0,1.0,0.0));
#3093=VECTOR('',#3092,30.0);
#3094=LINE('',#589,#3093);
#3095=EDGE_CURVE('',#590,#3091,#3094,.T.);
#3096=ORIENTED_EDGE('',*,*,#3095,.T.);
#3097=CARTESIAN_POINT('',(34.0,-20.0,152.500000000000000));
#3098=VERTEX_POINT('',#3097);
#3099=CARTESIAN_POINT('',(25.0,-20.0,152.500000000000000));
```

```
#3100=DIRECTION('',(0.0,1.0,0.0));
#3101=DIRECTION('',(1.0,0.0,0.0));
#3102=AXIS2_PLACEMENT_3D('',#3099,#3100,#3101);
#3103=CIRCLE('',#3102,9.0);
#3104=EDGE_CURVE('',#3098,#3091,#3103,.T.);
#3105=ORIENTED_EDGE('',*,*,#3104,.F.);
#3106=DIRECTION('',(0.0,1.0,0.0));
#3107=VECTOR('',#3106,30.0);
#3108=LINE('',#587,#3107);
#3109=EDGE_CURVE('',#588,#3098,#3108,.T.);
#3110=ORIENTED_EDGE('',*,*,#3109,.F.);
#3111=EDGE_LOOP('',(#3089,#3096,#3105,#3110));
#3112=FACE_BOUND('',#3111,.F.);
#3113=CARTESIAN_POINT('',(25.0,-52.500000000000000,152.500000000000000));
#3114=DIRECTION('',(0.0,1.0,0.0));
#3115=DIRECTION('',(1.0,0.0,0.0));
#3116=AXIS2_PLACEMENT_3D('',#3113,#3114,#3115);
#3117=CYLINDRICAL_SURFACE('',#3116,9.0);
#3118=ADVANCED_FACE('',(#3112),#3117,.F.);
#3119=SHAPE_ASPECT('featured shape: face 66',$,#38,.T.);
#3120=PROPERTY_DEFINITION('',$,#3119);
#3121=FACE_SHAPE_REPRESENTATION('',(#3118),#29);
#3122=PROPERTY_DEFINITION_REPRESENTATION(#3120,#3121);
#3123=CARTESIAN_POINT('',(38.0,-20.0,152.500000000000000));
#3124=VERTEX_POINT('',#3123);
#3125=CARTESIAN_POINT('',(12.0,-20.0,152.500000000000000));
#3126=VERTEX_POINT('',#3125);
#3127=CARTESIAN_POINT('',(25.0,-20.0,152.500000000000000));
#3128=DIRECTION('',(0.0,1.0,0.0));
#3129=DIRECTION('',(1.0,0.0,0.0));
#3130=AXIS2_PLACEMENT_3D('',#3127,#3128,#3129);
#3131=CIRCLE('',#3130,13.0);
#3132=EDGE_CURVE('',#3124,#3126,#3131,.T.);
#3133=ORIENTED_EDGE('',*,*,#3132,.F.);
#3134=CARTESIAN_POINT('',(25.0,-20.0,152.500000000000000));
#3135=DIRECTION('',(0.0,-1.0,0.0));
#3136=DIRECTION('',(1.0,0.0,0.0));
#3137=AXIS2_PLACEMENT_3D('',#3134,#3135,#3136);
#3138=CIRCLE('',#3137,13.0);
#3139=EDGE_CURVE('',#3124,#3126,#3138,.T.);
#3140=ORIENTED_EDGE('',*,*,#3139,.T.);
#3141=EDGE_LOOP('',(#3133,#3140));
#3142=FACE_BOUND('',#3141,.F.);
#3143=CARTESIAN_POINT('',(25.0,-20.0,152.500000000000000));
#3144=DIRECTION('',(0.0,-1.0,0.0));
#3145=DIRECTION('',(1.0,0.0,0.0));
#3146=AXIS2_PLACEMENT_3D('',#3143,#3144,#3145);
#3147=CIRCLE('',#3146,9.0);
#3148=EDGE_CURVE('',#3098,#3091,#3147,.T.);
#3149=ORIENTED_EDGE('',*,*,#3148,.F.);
#3150=ORIENTED_EDGE('',*,*,#3104,.T.);
#3151=EDGE_LOOP('',(#3149,#3150));
#3152=FACE_BOUND('',#3151,.F.);
#3153=CARTESIAN_POINT('',(25.0,-20.0,152.500000000000000));
#3154=DIRECTION('',(0.0,1.0,0.0));
#3155=DIRECTION('',(1.0,0.0,0.0));
```



```
#3156=AXIS2_PLACEMENT_3D('',#3153,#3154,#3155);
#3157=PLANE('',#3156);
#3158=ADVANCED_FACE('',(#3142,#3152),#3157,.T.);
#3159=SHAPE_ASPECT('featured shape: face 67',$,#38,.T.);
#3160=PROPERTY_DEFINITION('',$,#3159);
#3161=FACE_SHAPE_REPRESENTATION('',(#3158),#29);
#3162=PROPERTY_DEFINITION_REPRESENTATION(#3160,#3161);
#3163=ORIENTED_EDGE('',*,*,#913,.F.);
#3164=DIRECTION('',(0.0,1.0,0.0));
#3165=VECTOR('',#3164,20.0);
#3166=LINE('',#3123,#3165);
#3167=EDGE_CURVE('',#3124,#905,#3166,.T.);
#3168=ORIENTED_EDGE('',*,*,#3167,.F.);
#3169=ORIENTED_EDGE('',*,*,#3132,.T.);
#3170=DIRECTION('',(0.0,1.0,0.0));
#3171=VECTOR('',#3170,20.0);
#3172=LINE('',#3125,#3171);
#3173=EDGE_CURVE('',#3126,#907,#3172,.T.);
#3174=ORIENTED_EDGE('',*,*,#3173,.T.);
#3175=EDGE_LOOP('',(#3163,#3168,#3169,#3174));
#3176=FACE_BOUND('',#3175,.F.);
#3177=CARTESIAN_POINT('',(25.0,-52.500000000000000,152.500000000000000));
#3178=DIRECTION('',(0.0,1.0,0.0));
#3179=DIRECTION('',(1.0,0.0,0.0));
#3180=AXIS2_PLACEMENT_3D('',#3177,#3178,#3179);
#3181=CYLINDRICAL_SURFACE('',#3180,13.0);
#3182=ADVANCED_FACE('',(#3176),#3181,.F.);
#3183=SHAPE_ASPECT('featured shape: face 68',$,#38,.T.);
#3184=PROPERTY_DEFINITION('',$,#3183);
#3185=FACE_SHAPE_REPRESENTATION('',(#3182),#29);
#3186=PROPERTY_DEFINITION_REPRESENTATION(#3184,#3185);
#3187=ORIENTED_EDGE('',*,*,#603,.T.);
#3188=ORIENTED_EDGE('',*,*,#3109,.T.);
#3189=ORIENTED_EDGE('',*,*,#3148,.T.);
#3190=ORIENTED_EDGE('',*,*,#3095,.F.);
#3191=EDGE_LOOP('',(#3187,#3188,#3189,#3190));
#3192=FACE_BOUND('',#3191,.F.);
#3193=CARTESIAN_POINT('',(25.0,-52.500000000000000,152.500000000000000));
#3194=DIRECTION('',(0.0,1.0,0.0));
#3195=DIRECTION('',(1.0,0.0,0.0));
#3196=AXIS2_PLACEMENT_3D('',#3193,#3194,#3195);
#3197=CYLINDRICAL_SURFACE('',#3196,9.0);
#3198=ADVANCED_FACE('',(#3192),#3197,.F.);
#3199=SHAPE_ASPECT('featured shape: face 69',$,#38,.T.);
#3200=PROPERTY_DEFINITION('',$,#3199);
#3201=FACE_SHAPE_REPRESENTATION('',(#3198),#29);
#3202=PROPERTY_DEFINITION_REPRESENTATION(#3200,#3201);
#3203=ORIENTED_EDGE('',*,*,#920,.F.);
#3204=ORIENTED_EDGE('',*,*,#3173,.F.);
#3205=ORIENTED_EDGE('',*,*,#3139,.F.);
#3206=ORIENTED_EDGE('',*,*,#3167,.T.);
#3207=EDGE_LOOP('',(#3203,#3204,#3205,#3206));
#3208=FACE_BOUND('',#3207,.F.);
#3209=CARTESIAN_POINT('',(25.0,-52.500000000000000,152.500000000000000));
#3210=DIRECTION('',(0.0,1.0,0.0));
#3211=DIRECTION('',(1.0,0.0,0.0));
```

```
#3212=AXIS2_PLACEMENT_3D('',#3209,#3210,#3211);
#3213=CYLINDRICAL_SURFACE('',#3212,13.0);
#3214=ADVANCED_FACE('',(#3208),#3213,.F.);
#3215=SHAPE_ASPECT('featured shape: face 70',$,#38,.T.);
#3216=PROPERTY_DEFINITION('',$,#3215);
#3217=FACE_SHAPE_REPRESENTATION('',(#3214),#29);
#3218=PROPERTY_DEFINITION_REPRESENTATION(#3216,#3217);
#3219=ORIENTED_EDGE('',*,*,#765,.T.);
#3220=CARTESIAN_POINT('',(210.0,-10.0,95.0));
#3221=VERTEX_POINT('',#3220);
#3222=DIRECTION('',(0.0,-1.0,0.0));
#3223=VECTOR('',#3222,10.0);
#3224=LINE('',#751,#3223);
#3225=EDGE_CURVE('',#752,#3221,#3224,.T.);
#3226=ORIENTED_EDGE('',*,*,#3225,.T.);
#3227=DIRECTION('',(1.0,0.0,0.0));
#3228=VECTOR('',#3227,20.0);
#3229=LINE('',#3220,#3228);
#3230=EDGE_CURVE('',#3221,#1186,#3229,.T.);
#3231=ORIENTED_EDGE('',*,*,#3230,.T.);
#3232=ORIENTED_EDGE('',*,*,#1190,.F.);
#3233=EDGE_LOOP('',(#3219,#3226,#3231,#3232));
#3234=FACE_BOUND('',#3233,.F.);
#3235=CARTESIAN_POINT('',(205.0,0.0,95.0));
#3236=DIRECTION('',(0.0,0.0,1.0));
#3237=DIRECTION('',(0.0,-1.0,0.0));
#3238=AXIS2_PLACEMENT_3D('',#3235,#3236,#3237);
#3239=PLANE('',#3238);
#3240=ADVANCED_FACE('',(#3234),#3239,.F.);
#3241=SHAPE_ASPECT('featured shape: face 71',$,#38,.T.);
#3242=PROPERTY_DEFINITION('',$,#3241);
#3243=FACE_SHAPE_REPRESENTATION('',(#3240),#29);
#3244=PROPERTY_DEFINITION_REPRESENTATION(#3242,#3243);
#3245=ORIENTED_EDGE('',*,*,#1197,.F.);
#3246=ORIENTED_EDGE('',*,*,#3230,.F.);
#3247=CARTESIAN_POINT('',(210.0,-10.0,85.0));
#3248=VERTEX_POINT('',#3247);
#3249=CARTESIAN_POINT('',(210.0,-10.0,90.0));
#3250=DIRECTION('',(0.0,1.0,0.0));
#3251=DIRECTION('',(0.0,0.0,-1.0));
#3252=AXIS2_PLACEMENT_3D('',#3249,#3250,#3251);
#3253=CIRCLE('',#3252,5.0);
#3254=EDGE_CURVE('',#3248,#3221,#3253,.T.);
#3255=ORIENTED_EDGE('',*,*,#3254,.F.);
#3256=DIRECTION('',(1.0,0.0,0.0));
#3257=VECTOR('',#3256,20.0);
#3258=LINE('',#3247,#3257);
#3259=EDGE_CURVE('',#3248,#1193,#3258,.T.);
#3260=ORIENTED_EDGE('',*,*,#3259,.T.);
#3261=EDGE_LOOP('',(#3245,#3246,#3255,#3260));
#3262=FACE_BOUND('',#3261,.F.);
#3263=CARTESIAN_POINT('',(205.0,-10.0,95.0));
#3264=DIRECTION('',(0.0,-1.0,0.0));
#3265=DIRECTION('',(0.0,0.0,-1.0));
#3266=AXIS2_PLACEMENT_3D('',#3263,#3264,#3265);
#3267=PLANE('',#3266);
```



```
#3268=ADVANCED_FACE('',(#3262),#3267,.F.);
#3269=SHAPE_ASPECT('featured shape: face 72',$,#38,.T.);
#3270=PROPERTY_DEFINITION('',$,#3269);
#3271=FACE_SHAPE_REPRESENTATION('',(#3268),#29);
#3272=PROPERTY_DEFINITION_REPRESENTATION(#3270,#3271);
#3273=ORIENTED_EDGE('',*,*,#749,.F.);
#3274=ORIENTED_EDGE('',*,*,#1202,.T.);
#3275=ORIENTED_EDGE('',*,*,#3259,.F.);
#3276=DIRECTION('',(0.0,1.0,0.0));
#3277=VECTOR('',#3276,10.0);
#3278=LINE('',#3247,#3277);
#3279=EDGE_CURVE('',#3248,#745,#3278,.T.);
#3280=ORIENTED_EDGE('',*,*,#3279,.T.);
#3281=EDGE_LOOP('',(#3273,#3274,#3275,#3280));
#3282=FACE_BOUND('',#3281,.F.);
#3283=CARTESIAN_POINT('',(205.0,-10.0,85.0));
#3284=DIRECTION('',(0.0,0.0,-1.0));
#3285=DIRECTION('',(0.0,1.0,0.0));
#3286=AXIS2_PLACEMENT_3D('',#3283,#3284,#3285);
#3287=PLANE('',#3286);
#3288=ADVANCED_FACE('',(#3282),#3287,.F.);
#3289=SHAPE_ASPECT('featured shape: face 73',$,#38,.T.);
#3290=PROPERTY_DEFINITION('',$,#3289);
#3291=FACE_SHAPE_REPRESENTATION('',(#3288),#29);
#3292=PROPERTY_DEFINITION_REPRESENTATION(#3290,#3291);
#3293=ORIENTED_EDGE('',*,*,#3254,.T.);
#3294=ORIENTED_EDGE('',*,*,#3225,.F.);
#3295=ORIENTED_EDGE('',*,*,#758,.T.);
#3296=ORIENTED_EDGE('',*,*,#3279,.F.);
#3297=EDGE_LOOP('',(#3293,#3294,#3295,#3296));
#3298=FACE_BOUND('',#3297,.F.);
#3299=CARTESIAN_POINT('',(210.0,-10.0,90.0));
#3300=DIRECTION('',(0.0,-1.0,0.0));
#3301=DIRECTION('',(-1.0,0.0,0.0));
#3302=AXIS2_PLACEMENT_3D('',#3299,#3300,#3301);
#3303=CYLINDRICAL_SURFACE('',#3302,5.0);
#3304=ADVANCED_FACE('',(#3298),#3303,.F.);
#3305=SHAPE_ASPECT('featured shape: face 74',$,#38,.T.);
#3306=PROPERTY_DEFINITION('',$,#3305);
#3307=FACE_SHAPE_REPRESENTATION('',(#3304),#29);
#3308=PROPERTY_DEFINITION_REPRESENTATION(#3306,#3307);
#3309=CARTESIAN_POINT('',(110.0,-10.0,85.0));
#3310=VERTEX_POINT('',#3309);
#3311=CARTESIAN_POINT('',(85.0,-10.0,85.0));
#3312=VERTEX_POINT('',#3311);
#3313=DIRECTION('',(-1.0,0.0,0.0));
#3314=VECTOR('',#3313,25.0);
#3315=LINE('',#3309,#3314);
#3316=EDGE_CURVE('',#3310,#3312,#3315,.T.);
#3317=ORIENTED_EDGE('',*,*,#3316,.F.);
#3318=CARTESIAN_POINT('',(110.0,-10.0,95.0));
#3319=VERTEX_POINT('',#3318);
#3320=CARTESIAN_POINT('',(110.0,-10.0,90.0));
#3321=DIRECTION('',(0.0,1.0,0.0));
#3322=DIRECTION('',(0.0,0.0,1.0));
#3323=AXIS2_PLACEMENT_3D('',#3320,#3321,#3322);
```



```
#3324=CIRCLE('',#3323,5.0);
#3325=EDGE_CURVE('',#3319,#3310,#3324,.T.);
#3326=ORIENTED_EDGE('',*,*,#3325,.F.);
#3327=CARTESIAN_POINT('',(85.0,-10.0,95.0));
#3328=VERTEX_POINT('',#3327);
#3329=DIRECTION('',(1.0,0.0,0.0));
#3330=VECTOR('',#3329,25.0);
#3331=LINE('',#3327,#3330);
#3332=EDGE_CURVE('',#3328,#3319,#3331,.T.);
#3333=ORIENTED_EDGE('',*,*,#3332,.F.);
#3334=CARTESIAN_POINT('',(85.0,-10.0,90.0));
#3335=DIRECTION('',(0.0,1.0,0.0));
#3336=DIRECTION('',(0.0,0.0,-1.0));
#3337=AXIS2_PLACEMENT_3D('',#3334,#3335,#3336);
#3338=CIRCLE('',#3337,5.0);
#3339=EDGE_CURVE('',#3312,#3328,#3338,.T.);
#3340=ORIENTED_EDGE('',*,*,#3339,.F.);
#3341=EDGE_LOOP('',(#3317,#3326,#3333,#3340));
#3342=FACE_BOUND('',#3341,.F.);
#3343=CARTESIAN_POINT('',(260.0,-10.0,0.0));
#3344=DIRECTION('',(0.0,1.0,0.0));
#3345=DIRECTION('',(0.0,0.0,1.0));
#3346=AXIS2_PLACEMENT_3D('',#3343,#3344,#3345);
#3347=PLANE('',#3346);
#3348=ADVANCED_FACE('',(#3342),#3347,.T.);
#3349=SHAPE_ASPECT('featured shape: face 75',$,#38,.T.);
#3350=PROPERTY_DEFINITION('',$,#3349);
#3351=FACE_SHAPE_REPRESENTATION('',(#3348),#29);
#3352=PROPERTY_DEFINITION_REPRESENTATION(#3350,#3351);
#3353=ORIENTED_EDGE('',*,*,#931,.F.);
#3354=DIRECTION('',(0.0,1.0,0.0));
#3355=VECTOR('',#3354,10.0);
#3356=LINE('',#3309,#3355);
#3357=EDGE_CURVE('',#3310,#925,#3356,.T.);
#3358=ORIENTED_EDGE('',*,*,#3357,.F.);
#3359=ORIENTED_EDGE('',*,*,#3316,.T.);
#3360=DIRECTION('',(0.0,1.0,0.0));
#3361=VECTOR('',#3360,10.0);
#3362=LINE('',#3311,#3361);
#3363=EDGE_CURVE('',#3312,#927,#3362,.T.);
#3364=ORIENTED_EDGE('',*,*,#3363,.T.);
#3365=EDGE_LOOP('',(#3353,#3358,#3359,#3364));
#3366=FACE_BOUND('',#3365,.F.);
#3367=CARTESIAN_POINT('',(110.0,-10.0,85.0));
#3368=DIRECTION('',(0.0,0.0,-1.0));
#3369=DIRECTION('',(-1.0,0.0,0.0));
#3370=AXIS2_PLACEMENT_3D('',#3367,#3368,#3369);
#3371=PLANE('',#3370);
#3372=ADVANCED_FACE('',(#3366),#3371,.F.);
#3373=SHAPE_ASPECT('featured shape: face 76',$,#38,.T.);
#3374=PROPERTY_DEFINITION('',$,#3373);
#3375=FACE_SHAPE_REPRESENTATION('',(#3372),#29);
#3376=PROPERTY_DEFINITION_REPRESENTATION(#3374,#3375);
#3377=ORIENTED_EDGE('',*,*,#940,.F.);
#3378=ORIENTED_EDGE('',*,*,#3363,.F.);
#3379=ORIENTED_EDGE('',*,*,#3339,.T.);
```



```
#3380=DIRECTION('',(0.0,1.0,0.0));
#3381=VECTOR('',#3380,10.0);
#3382=LINE('',#3327,#3381);
#3383=EDGE_CURVE('',#3328,#934,#3382,.T.);
#3384=ORIENTED_EDGE('',*,*,#3383,.T.);
#3385=EDGE_LOOP('',(#3377,#3378,#3379,#3384));
#3386=FACE_BOUND('',#3385,.F.);
#3387=CARTESIAN_POINT('',(85.0,-10.0,90.0));
#3388=DIRECTION('',(0.0,1.0,0.0));
#3389=DIRECTION('',(0.0,0.0,1.0));
#3390=AXIS2_PLACEMENT_3D('',#3387,#3388,#3389);
#3391=CYLINDRICAL_SURFACE('',#3390,5.0);
#3392=ADVANCED_FACE('',(#3386),#3391,.F.);
#3393=SHAPE_ASPECT('featured shape: face 77',$,#38,.T.);
#3394=PROPERTY_DEFINITION('',$,#3393);
#3395=FACE_SHAPE_REPRESENTATION('',(#3392),#29);
#3396=PROPERTY_DEFINITION_REPRESENTATION(#3394,#3395);
#3397=ORIENTED_EDGE('',*,*,#947,.F.);
#3398=ORIENTED_EDGE('',*,*,#3383,.F.);
#3399=ORIENTED_EDGE('',*,*,#3332,.T.);
#3400=DIRECTION('',(0.0,1.0,0.0));
#3401=VECTOR('',#3400,10.0);
#3402=LINE('',#3318,#3401);
#3403=EDGE_CURVE('',#3319,#943,#3402,.T.);
#3404=ORIENTED_EDGE('',*,*,#3403,.T.);
#3405=EDGE_LOOP('',(#3397,#3398,#3399,#3404));
#3406=FACE_BOUND('',#3405,.F.);
#3407=CARTESIAN_POINT('',(85.0,-10.0,95.0));
#3408=DIRECTION('',(0.0,0.0,1.0));
#3409=DIRECTION('',(1.0,0.0,0.0));
#3410=AXIS2_PLACEMENT_3D('',#3407,#3408,#3409);
#3411=PLANE('',#3410);
#3412=ADVANCED_FACE('',(#3406),#3411,.F.);
#3413=SHAPE_ASPECT('featured shape: face 78',$,#38,.T.);
#3414=PROPERTY_DEFINITION('',$,#3413);
#3415=FACE_SHAPE_REPRESENTATION('',(#3412),#29);
#3416=PROPERTY_DEFINITION_REPRESENTATION(#3414,#3415);
#3417=ORIENTED_EDGE('',*,*,#954,.F.);
#3418=ORIENTED_EDGE('',*,*,#3403,.F.);
#3419=ORIENTED_EDGE('',*,*,#3325,.T.);
#3420=ORIENTED_EDGE('',*,*,#3357,.T.);
#3421=EDGE_LOOP('',(#3417,#3418,#3419,#3420));
#3422=FACE_BOUND('',#3421,.F.);
#3423=CARTESIAN_POINT('',(110.0,-10.0,90.0));
#3424=DIRECTION('',(0.0,1.0,0.0));
#3425=DIRECTION('',(0.0,0.0,1.0));
#3426=AXIS2_PLACEMENT_3D('',#3423,#3424,#3425);
#3427=CYLINDRICAL_SURFACE('',#3426,5.0);
#3428=ADVANCED_FACE('',(#3422),#3427,.F.);
#3429=SHAPE_ASPECT('featured shape: face 79',$,#38,.T.);
#3430=PROPERTY_DEFINITION('',$,#3429);
#3431=FACE_SHAPE_REPRESENTATION('',(#3428),#29);
#3432=PROPERTY_DEFINITION_REPRESENTATION(#3430,#3431);
#3433=CARTESIAN_POINT('',(192.568849999999998,-19.050000000000001,120.0));
#3434=VERTEX_POINT('',#3433);
#3435=CARTESIAN_POINT('',(187.431150000000002,-19.050000000000001,120.0));
```

```
#3436=VERTEX_POINT('',#3435);
#3437=CARTESIAN_POINT('',(190.0,-19.049999999999969,120.0));
#3438=DIRECTION('',(0.0,-1.0,0.0));
#3439=DIRECTION('',(1.0,0.0,0.0));
#3440=AXIS2_PLACEMENT_3D('',#3437,#3438,#3439);
#3441=CIRCLE('',#3440,2.568850000000000);
#3442=EDGE_CURVE('',#3434,#3436,#3441,.T.);
#3443=ORIENTED_EDGE('',*,*,#3442,.T.);
#3444=CARTESIAN_POINT('',(190.0,-19.049999999999969,120.0));
#3445=DIRECTION('',(0.0,-1.0,0.0));
#3446=DIRECTION('',(-1.0,0.0,0.0));
#3447=AXIS2_PLACEMENT_3D('',#3444,#3445,#3446);
#3448=CIRCLE('',#3447,2.568850000000000);
#3449=EDGE_CURVE('',#3436,#3434,#3448,.T.);
#3450=ORIENTED_EDGE('',*,*,#3449,.T.);
#3451=EDGE_LOOP('',(#3443,#3450));
#3452=FACE_BOUND('',#3451,.F.);
#3453=CARTESIAN_POINT('',(190.0,-19.050000000000001,120.0));
#3454=DIRECTION('',(0.0,-1.0,0.0));
#3455=DIRECTION('',(1.0,0.0,0.0));
#3456=AXIS2_PLACEMENT_3D('',#3453,#3454,#3455);
#3457=PLANE('',#3456);
#3458=ADVANCED_FACE('',(#3452),#3457,.F.);
#3459=SHAPE_ASPECT('featured shape: face 80',$,#38,.T.);
#3460=PROPERTY_DEFINITION('',$,#3459);
#3461=FACE_SHAPE_REPRESENTATION('',(#3458),#29);
#3462=PROPERTY_DEFINITION_REPRESENTATION(#3460,#3461);
#3463=ORIENTED_EDGE('',*,*,#967,.T.);
#3464=DIRECTION('',(0.0,-1.0,0.0));
#3465=VECTOR('',#3464,19.050000000000001);
#3466=LINE('',#960,#3465);
#3467=EDGE_CURVE('',#961,#3436,#3466,.T.);
#3468=ORIENTED_EDGE('',*,*,#3467,.T.);
#3469=ORIENTED_EDGE('',*,*,#3442,.F.);
#3470=DIRECTION('',(0.0,-1.0,0.0));
#3471=VECTOR('',#3470,19.050000000000001);
#3472=LINE('',#958,#3471);
#3473=EDGE_CURVE('',#959,#3434,#3472,.T.);
#3474=ORIENTED_EDGE('',*,*,#3473,.F.);
#3475=EDGE_LOOP('',(#3463,#3468,#3469,#3474));
#3476=FACE_BOUND('',#3475,.F.);
#3477=CARTESIAN_POINT('',(190.0,0.0,120.0));
#3478=DIRECTION('',(0.0,-1.0,0.0));
#3479=DIRECTION('',(1.0,0.0,0.0));
#3480=AXIS2_PLACEMENT_3D('',#3477,#3478,#3479);
#3481=CYLINDRICAL_SURFACE('',#3480,2.568850000000000);
#3482=ADVANCED_FACE('',(#3476),#3481,.F.);
#3483=SHAPE_ASPECT('featured shape: face 81',$,#38,.T.);
#3484=PROPERTY_DEFINITION('',$,#3483);
#3485=FACE_SHAPE_REPRESENTATION('',(#3482),#29);
#3486=PROPERTY_DEFINITION_REPRESENTATION(#3484,#3485);
#3487=ORIENTED_EDGE('',*,*,#974,.T.);
#3488=ORIENTED_EDGE('',*,*,#3473,.T.);
#3489=ORIENTED_EDGE('',*,*,#3449,.F.);
#3490=ORIENTED_EDGE('',*,*,#3467,.F.);
#3491=EDGE_LOOP('',(#3487,#3488,#3489,#3490));
```



```
#3492=FACE_BOUND('',#3491,.F.);
#3493=CARTESIAN_POINT('',(190.0,0.0,120.0));
#3494=DIRECTION('',(0.0,-1.0,0.0));
#3495=DIRECTION('',(1.0,0.0,0.0));
#3496=AXIS2_PLACEMENT_3D('',#3493,#3494,#3495);
#3497=CYLINDRICAL_SURFACE('',#3496,2.568850000000000);
#3498=ADVANCED_FACE('',(#3492),#3497,.F.);
#3499=SHAPE_ASPECT('featured shape: face 82',$,#38,.T.);
#3500=PROPERTY_DEFINITION('',$,#3499);
#3501=FACE_SHAPE_REPRESENTATION('',(#3498),#29);
#3502=PROPERTY_DEFINITION_REPRESENTATION(#3500,#3501);
#3503=CARTESIAN_POINT('',(192.568849999999998,-19.050000000000001,60.0));
#3504=VERTEX_POINT('',#3503);
#3505=CARTESIAN_POINT('',(187.431150000000002,-19.050000000000001,60.0));
#3506=VERTEX_POINT('',#3505);
#3507=CARTESIAN_POINT('',(190.0,-19.049999999999969,60.0));
#3508=DIRECTION('',(0.0,-1.0,0.0));
#3509=DIRECTION('',(1.0,0.0,0.0));
#3510=AXIS2_PLACEMENT_3D('',#3507,#3508,#3509);
#3511=CIRCLE('',#3510,2.568850000000000);
#3512=EDGE_CURVE('',#3504,#3506,#3511,.T.);
#3513=ORIENTED_EDGE('',*,*,#3512,.T.);
#3514=CARTESIAN_POINT('',(190.0,-19.049999999999969,60.0));
#3515=DIRECTION('',(0.0,-1.0,0.0));
#3516=DIRECTION('',(-1.0,0.0,0.0));
#3517=AXIS2_PLACEMENT_3D('',#3514,#3515,#3516);
#3518=CIRCLE('',#3517,2.568850000000000);
#3519=EDGE_CURVE('',#3506,#3504,#3518,.T.);
#3520=ORIENTED_EDGE('',*,*,#3519,.T.);
#3521=EDGE_LOOP('',(#3513,#3520));
#3522=FACE_BOUND('',#3521,.F.);
#3523=CARTESIAN_POINT('',(190.0,-19.050000000000001,60.0));
#3524=DIRECTION('',(0.0,-1.0,0.0));
#3525=DIRECTION('',(1.0,0.0,0.0));
#3526=AXIS2_PLACEMENT_3D('',#3523,#3524,#3525);
#3527=PLANE('',#3526);
#3528=ADVANCED_FACE('',(#3522),#3527,.F.);
#3529=SHAPE_ASPECT('featured shape: face 83',$,#38,.T.);
#3530=PROPERTY_DEFINITION('',$,#3529);
#3531=FACE_SHAPE_REPRESENTATION('',(#3528),#29);
#3532=PROPERTY_DEFINITION_REPRESENTATION(#3530,#3531);
#3533=ORIENTED_EDGE('',*,*,#987,.T.);
#3534=DIRECTION('',(0.0,-1.0,0.0));
#3535=VECTOR('',#3534,19.050000000000001);
#3536=LINE('',#980,#3535);
#3537=EDGE_CURVE('',#981,#3506,#3536,.T.);
#3538=ORIENTED_EDGE('',*,*,#3537,.T.);
#3539=ORIENTED_EDGE('',*,*,#3512,.F.);
#3540=DIRECTION('',(0.0,-1.0,0.0));
#3541=VECTOR('',#3540,19.050000000000001);
#3542=LINE('',#978,#3541);
#3543=EDGE_CURVE('',#979,#3504,#3542,.T.);
#3544=ORIENTED_EDGE('',*,*,#3543,.F.);
#3545=EDGE_LOOP('',(#3533,#3538,#3539,#3544));
#3546=FACE_BOUND('',#3545,.F.);
#3547=CARTESIAN_POINT('',(190.0,0.0,60.0));
```

```
#3548=DIRECTION('',(0.0,-1.0,0.0));
#3549=DIRECTION('',(1.0,0.0,0.0));
#3550=AXIS2_PLACEMENT_3D('',#3547,#3548,#3549);
#3551=CYLINDRICAL_SURFACE('',#3550,2.568850000000000);
#3552=ADVANCED_FACE('',(#3546),#3551,.F.);
#3553=SHAPE_ASPECT('featured shape: face 84',$,#38,.T.);
#3554=PROPERTY_DEFINITION('',$,#3553);
#3555=FACE_SHAPE_REPRESENTATION('',(#3552),#29);
#3556=PROPERTY_DEFINITION_REPRESENTATION(#3554,#3555);
#3557=ORIENTED_EDGE('',*,*,#994,.T.);
#3558=ORIENTED_EDGE('',*,*,#3543,.T.);
#3559=ORIENTED_EDGE('',*,*,#3519,.F.);
#3560=ORIENTED_EDGE('',*,*,#3537,.F.);
#3561=EDGE_LOOP('',(#3557,#3558,#3559,#3560));
#3562=FACE_BOUND('',#3561,.F.);
#3563=CARTESIAN_POINT('',(190.0,0.0,60.0));
#3564=DIRECTION('',(0.0,-1.0,0.0));
#3565=DIRECTION('',(1.0,0.0,0.0));
#3566=AXIS2_PLACEMENT_3D('',#3563,#3564,#3565);
#3567=CYLINDRICAL_SURFACE('',#3566,2.568850000000000);
#3568=ADVANCED_FACE('',(#3562),#3567,.F.);
#3569=SHAPE_ASPECT('featured shape: face 85',$,#38,.T.);
#3570=PROPERTY_DEFINITION('',$,#3569);
#3571=FACE_SHAPE_REPRESENTATION('',(#3568),#29);
#3572=PROPERTY_DEFINITION_REPRESENTATION(#3570,#3571);
#3573=CARTESIAN_POINT('',(132.568849999999998,-19.050000000000001,120.0));
#3574=VERTEX_POINT('',#3573);
#3575=CARTESIAN_POINT('',(127.431150000000002,-19.050000000000001,120.0));
#3576=VERTEX_POINT('',#3575);
#3577=CARTESIAN_POINT('',(130.0,-19.049999999999969,120.0));
#3578=DIRECTION('',(0.0,-1.0,0.0));
#3579=DIRECTION('',(1.0,0.0,0.0));
#3580=AXIS2_PLACEMENT_3D('',#3577,#3578,#3579);
#3581=CIRCLE('',#3580,2.568850000000000);
#3582=EDGE_CURVE('',#3574,#3576,#3581,.T.);
#3583=ORIENTED_EDGE('',*,*,#3582,.T.);
#3584=CARTESIAN_POINT('',(130.0,-19.049999999999969,120.0));
#3585=DIRECTION('',(0.0,-1.0,0.0));
#3586=DIRECTION('',(-1.0,0.0,0.0));
#3587=AXIS2_PLACEMENT_3D('',#3584,#3585,#3586);
#3588=CIRCLE('',#3587,2.568850000000000);
#3589=EDGE_CURVE('',#3576,#3574,#3588,.T.);
#3590=ORIENTED_EDGE('',*,*,#3589,.T.);
#3591=EDGE_LOOP('',(#3583,#3590));
#3592=FACE_BOUND('',#3591,.F.);
#3593=CARTESIAN_POINT('',(130.0,-19.050000000000001,120.0));
#3594=DIRECTION('',(0.0,-1.0,0.0));
#3595=DIRECTION('',(1.0,0.0,0.0));
#3596=AXIS2_PLACEMENT_3D('',#3593,#3594,#3595);
#3597=PLANE('',#3596);
#3598=ADVANCED_FACE('',(#3592),#3597,.F.);
#3599=SHAPE_ASPECT('featured shape: face 86',$,#38,.T.);
#3600=PROPERTY_DEFINITION('',$,#3599);
#3601=FACE_SHAPE_REPRESENTATION('',(#3598),#29);
#3602=PROPERTY_DEFINITION_REPRESENTATION(#3600,#3601);
#3603=ORIENTED_EDGE('',*,*,#1007,.T.);
```



```
#3604=DIRECTION('',(0.0,-1.0,0.0));
#3605=VECTOR('',#3604,19.050000000000001);
#3606=LINE('',#1000,#3605);
#3607=EDGE_CURVE('',#1001,#3576,#3606,.T.);
#3608=ORIENTED_EDGE('',*,*,#3607,.T.);
#3609=ORIENTED_EDGE('',*,*,#3582,.F.);
#3610=DIRECTION('',(0.0,-1.0,0.0));
#3611=VECTOR('',#3610,19.050000000000001);
#3612=LINE('',#998,#3611);
#3613=EDGE_CURVE('',#999,#3574,#3612,.T.);
#3614=ORIENTED_EDGE('',*,*,#3613,.F.);
#3615=EDGE_LOOP('',(#3603,#3608,#3609,#3614));
#3616=FACE_BOUND('',#3615,.F.);
#3617=CARTESIAN_POINT('',(130.0,0.0,120.0));
#3618=DIRECTION('',(0.0,-1.0,0.0));
#3619=DIRECTION('',(1.0,0.0,0.0));
#3620=AXIS2_PLACEMENT_3D('',#3617,#3618,#3619);
#3621=CYLINDRICAL_SURFACE('',#3620,2.568850000000000);
#3622=ADVANCED_FACE('',(#3616),#3621,.F.);
#3623=SHAPE_ASPECT('featured shape: face 87',$,#38,.T.);
#3624=PROPERTY_DEFINITION('',$,#3623);
#3625=FACE_SHAPE_REPRESENTATION('',(#3622),#29);
#3626=PROPERTY_DEFINITION_REPRESENTATION(#3624,#3625);
#3627=ORIENTED_EDGE('',*,*,#1014,.T.);
#3628=ORIENTED_EDGE('',*,*,#3613,.T.);
#3629=ORIENTED_EDGE('',*,*,#3589,.F.);
#3630=ORIENTED_EDGE('',*,*,#3607,.F.);
#3631=EDGE_LOOP('',(#3627,#3628,#3629,#3630));
#3632=FACE_BOUND('',#3631,.F.);
#3633=CARTESIAN_POINT('',(130.0,0.0,120.0));
#3634=DIRECTION('',(0.0,-1.0,0.0));
#3635=DIRECTION('',(1.0,0.0,0.0));
#3636=AXIS2_PLACEMENT_3D('',#3633,#3634,#3635);
#3637=CYLINDRICAL_SURFACE('',#3636,2.568850000000000);
#3638=ADVANCED_FACE('',(#3632),#3637,.F.);
#3639=SHAPE_ASPECT('featured shape: face 88',$,#38,.T.);
#3640=PROPERTY_DEFINITION('',$,#3639);
#3641=FACE_SHAPE_REPRESENTATION('',(#3638),#29);
#3642=PROPERTY_DEFINITION_REPRESENTATION(#3640,#3641);
#3643=CARTESIAN_POINT('',(132.568849999999998,-19.050000000000001,60.0));
#3644=VERTEX_POINT('',#3643);
#3645=CARTESIAN_POINT('',(127.431150000000002,-19.050000000000001,60.0));
#3646=VERTEX_POINT('',#3645);
#3647=CARTESIAN_POINT('',(130.0,-19.049999999999969,60.0));
#3648=DIRECTION('',(0.0,-1.0,0.0));
#3649=DIRECTION('',(1.0,0.0,0.0));
#3650=AXIS2_PLACEMENT_3D('',#3647,#3648,#3649);
#3651=CIRCLE('',#3650,2.568850000000000);
#3652=EDGE_CURVE('',#3644,#3646,#3651,.T.);
#3653=ORIENTED_EDGE('',*,*,#3652,.T.);
#3654=CARTESIAN_POINT('',(130.0,-19.049999999999969,60.0));
#3655=DIRECTION('',(0.0,-1.0,0.0));
#3656=DIRECTION('',(-1.0,0.0,0.0));
#3657=AXIS2_PLACEMENT_3D('',#3654,#3655,#3656);
#3658=CIRCLE('',#3657,2.568850000000000);
#3659=EDGE_CURVE('',#3646,#3644,#3658,.T.);
```

```
#3660=ORIENTED_EDGE('',*,*,#3659,.T.);
#3661=EDGE_LOOP('',(#3653,#3660));
#3662=FACE_BOUND('',#3661,.F.);
#3663=CARTESIAN_POINT('',(130.0,-19.050000000000001,60.0));
#3664=DIRECTION('',(0.0,-1.0,0.0));
#3665=DIRECTION('',(1.0,0.0,0.0));
#3666=AXIS2_PLACEMENT_3D('',#3663,#3664,#3665);
#3667=PLANE('',#3666);
#3668=ADVANCED_FACE('',(#3662),#3667,.F.);
#3669=SHAPE_ASPECT('featured shape: face 89',$,#38,.T.);
#3670=PROPERTY_DEFINITION('',$,#3669);
#3671=FACE_SHAPE_REPRESENTATION('',(#3668),#29);
#3672=PROPERTY_DEFINITION_REPRESENTATION(#3670,#3671);
#3673=ORIENTED_EDGE('',*,*,#1027,.T.);
#3674=DIRECTION('',(0.0,-1.0,0.0));
#3675=VECTOR('',#3674,19.050000000000001);
#3676=LINE('',#1020,#3675);
#3677=EDGE_CURVE('',#1021,#3646,#3676,.T.);
#3678=ORIENTED_EDGE('',*,*,#3677,.T.);
#3679=ORIENTED_EDGE('',*,*,#3652,.F.);
#3680=DIRECTION('',(0.0,-1.0,0.0));
#3681=VECTOR('',#3680,19.050000000000001);
#3682=LINE('',#1018,#3681);
#3683=EDGE_CURVE('',#1019,#3644,#3682,.T.);
#3684=ORIENTED_EDGE('',*,*,#3683,.F.);
#3685=EDGE_LOOP('',(#3673,#3678,#3679,#3684));
#3686=FACE_BOUND('',#3685,.F.);
#3687=CARTESIAN_POINT('',(130.0,0.0,60.0));
#3688=DIRECTION('',(0.0,-1.0,0.0));
#3689=DIRECTION('',(1.0,0.0,0.0));
#3690=AXIS2_PLACEMENT_3D('',#3687,#3688,#3689);
#3691=CYLINDRICAL_SURFACE('',#3690,2.568850000000000);
#3692=ADVANCED_FACE('',(#3686),#3691,.F.);
#3693=SHAPE_ASPECT('featured shape: face 90',$,#38,.T.);
#3694=PROPERTY_DEFINITION('',$,#3693);
#3695=FACE_SHAPE_REPRESENTATION('',(#3692),#29);
#3696=PROPERTY_DEFINITION_REPRESENTATION(#3694,#3695);
#3697=ORIENTED_EDGE('',*,*,#1034,.T.);
#3698=ORIENTED_EDGE('',*,*,#3683,.T.);
#3699=ORIENTED_EDGE('',*,*,#3659,.F.);
#3700=ORIENTED_EDGE('',*,*,#3677,.F.);
#3701=EDGE_LOOP('',(#3697,#3698,#3699,#3700));
#3702=FACE_BOUND('',#3701,.F.);
#3703=CARTESIAN_POINT('',(130.0,0.0,60.0));
#3704=DIRECTION('',(0.0,-1.0,0.0));
#3705=DIRECTION('',(1.0,0.0,0.0));
#3706=AXIS2_PLACEMENT_3D('',#3703,#3704,#3705);
#3707=CYLINDRICAL_SURFACE('',#3706,2.568850000000000);
#3708=ADVANCED_FACE('',(#3702),#3707,.F.);
#3709=SHAPE_ASPECT('featured shape: face 91',$,#38,.T.);
#3710=PROPERTY_DEFINITION('',$,#3709);
#3711=FACE_SHAPE_REPRESENTATION('',(#3708),#29);
#3712=PROPERTY_DEFINITION_REPRESENTATION(#3710,#3711);
#3713=ORIENTED_EDGE('',*,*,#1210,.F.);
#3714=ORIENTED_EDGE('',*,*,#735,.T.);
#3715=ORIENTED_EDGE('',*,*,#53,.T.);
```



```
#3716=ORIENTED_EDGE('',*,*,#1443,.F.);
#3717=EDGE_LOOP('',(#3713,#3714,#3715,#3716));
#3718=FACE_BOUND('',#3717,.F.);
#3719=CARTESIAN_POINT('',(225.0,0.0,5.0));
#3720=DIRECTION('',(0.0,1.0,0.0));
#3721=DIRECTION('',(1.0,0.0,0.0));
#3722=AXIS2_PLACEMENT_3D('',#3719,#3720,#3721);
#3723=CYLINDRICAL_SURFACE('',#3722,5.0);
#3724=ADVANCED_FACE('',(#3718),#3723,.T.);
#3725=SHAPE_ASPECT('featured shape: face 92',$,#38,.T.);
#3726=PROPERTY_DEFINITION('',$,#3725);
#3727=FACE_SHAPE_REPRESENTATION('',(#3724),#29);
#3728=PROPERTY_DEFINITION_REPRESENTATION(#3726,#3727);
#3729=ORIENTED_EDGE('',*,*,#1250,.F.);
#3730=ORIENTED_EDGE('',*,*,#1387,.T.);
#3731=ORIENTED_EDGE('',*,*,#107,.T.);
#3732=ORIENTED_EDGE('',*,*,#779,.F.);
#3733=EDGE_LOOP('',(#3729,#3730,#3731,#3732));
#3734=FACE_BOUND('',#3733,.F.);
#3735=CARTESIAN_POINT('',(225.0,-40.0,175.0));
#3736=DIRECTION('',(0.0,-1.0,0.0));
#3737=DIRECTION('',(1.0,0.0,0.0));
#3738=AXIS2_PLACEMENT_3D('',#3735,#3736,#3737);
#3739=CYLINDRICAL_SURFACE('',#3738,5.0);
#3740=ADVANCED_FACE('',(#3734),#3739,.T.);
#3741=SHAPE_ASPECT('featured shape: face 93',$,#38,.T.);
#3742=PROPERTY_DEFINITION('',$,#3741);
#3743=FACE_SHAPE_REPRESENTATION('',(#3740),#29);
#3744=PROPERTY_DEFINITION_REPRESENTATION(#3742,#3743);
#3745=ORIENTED_EDGE('',*,*,#121,.T.);
#3746=ORIENTED_EDGE('',*,*,#1379,.T.);
#3747=ORIENTED_EDGE('',*,*,#182,.F.);
#3748=ORIENTED_EDGE('',*,*,#695,.F.);
#3749=EDGE_LOOP('',(#3745,#3746,#3747,#3748));
#3750=FACE_BOUND('',#3749,.F.);
#3751=CARTESIAN_POINT('',(5.0,0.0,175.0));
#3752=DIRECTION('',(0.0,-1.0,0.0));
#3753=DIRECTION('',(0.0,0.0,1.0));
#3754=AXIS2_PLACEMENT_3D('',#3751,#3752,#3753);
#3755=CYLINDRICAL_SURFACE('',#3754,5.0);
#3756=ADVANCED_FACE('',(#3750),#3755,.T.);
#3757=SHAPE_ASPECT('featured shape: face 94',$,#38,.T.);
#3758=PROPERTY_DEFINITION('',$,#3757);
#3759=FACE_SHAPE_REPRESENTATION('',(#3756),#29);
#3760=PROPERTY_DEFINITION_REPRESENTATION(#3758,#3759);
#3761=ORIENTED_EDGE('',*,*,#65,.F.);
#3762=ORIENTED_EDGE('',*,*,#725,.T.);
#3763=ORIENTED_EDGE('',*,*,#196,.T.);
#3764=ORIENTED_EDGE('',*,*,#1371,.F.);
#3765=EDGE_LOOP('',(#3761,#3762,#3763,#3764));
#3766=FACE_BOUND('',#3765,.F.);
#3767=CARTESIAN_POINT('',(5.0,0.0,5.0));
#3768=DIRECTION('',(0.0,1.0,0.0));
#3769=DIRECTION('',(0.0,0.0,-1.0));
#3770=AXIS2_PLACEMENT_3D('',#3767,#3768,#3769);
#3771=CYLINDRICAL_SURFACE('',#3770,5.0);
```

```
#3772=ADVANCED_FACE('',(#3766),#3771,.T.);
#3773=SHAPE_ASPECT('featured shape: face 95',$,#38,.T.);
#3774=PROPERTY_DEFINITION('',$,#3773);
#3775=FACE_SHAPE_REPRESENTATION('',(#3772),#29);
#3776=PROPERTY_DEFINITION_REPRESENTATION(#3774,#3775);
#3777=ORIENTED_EDGE('',*,*,#619,.T.);
#3778=ORIENTED_EDGE('',*,*,#1286,.T.);
#3779=ORIENTED_EDGE('',*,*,#1864,.F.);
#3780=ORIENTED_EDGE('',*,*,#1421,.F.);
#3781=EDGE_LOOP('',(#3777,#3778,#3779,#3780));
#3782=FACE_BOUND('',#3781,.F.);
#3783=CARTESIAN_POINT('',(255.0,-40.0,55.0));
#3784=DIRECTION('',(0.0,1.0,0.0));
#3785=DIRECTION('',(1.0,0.0,0.0));
#3786=AXIS2_PLACEMENT_3D('',#3783,#3784,#3785);
#3787=CYLINDRICAL_SURFACE('',#3786,5.0);
#3788=ADVANCED_FACE('',(#3782),#3787,.T.);
#3789=SHAPE_ASPECT('featured shape: face 96',$,#38,.T.);
#3790=PROPERTY_DEFINITION('',$,#3789);
#3791=FACE_SHAPE_REPRESENTATION('',(#3788),#29);
#3792=PROPERTY_DEFINITION_REPRESENTATION(#3790,#3791);
#3793=ORIENTED_EDGE('',*,*,#633,.F.);
#3794=ORIENTED_EDGE('',*,*,#1411,.T.);
#3795=ORIENTED_EDGE('',*,*,#1824,.T.);
#3796=ORIENTED_EDGE('',*,*,#1296,.F.);
#3797=EDGE_LOOP('',(#3793,#3794,#3795,#3796));
#3798=FACE_BOUND('',#3797,.F.);
#3799=CARTESIAN_POINT('',(255.0,-40.0,125.0));
#3800=DIRECTION('',(0.0,-1.0,0.0));
#3801=DIRECTION('',(1.0,0.0,0.0));
#3802=AXIS2_PLACEMENT_3D('',#3799,#3800,#3801);
#3803=CYLINDRICAL_SURFACE('',#3802,5.0);
#3804=ADVANCED_FACE('',(#3798),#3803,.T.);
#3805=SHAPE_ASPECT('featured shape: face 97',$,#38,.T.);
#3806=PROPERTY_DEFINITION('',$,#3805);
#3807=FACE_SHAPE_REPRESENTATION('',(#3804),#29);
#3808=PROPERTY_DEFINITION_REPRESENTATION(#3806,#3807);
#3809=ORIENTED_EDGE('',*,*,#2148,.F.);
#3810=ORIENTED_EDGE('',*,*,#478,.T.);
#3811=ORIENTED_EDGE('',*,*,#2210,.T.);
#3812=ORIENTED_EDGE('',*,*,#1641,.F.);
#3813=EDGE_LOOP('',(#3809,#3810,#3811,#3812));
#3814=FACE_BOUND('',#3813,.F.);
#3815=CARTESIAN_POINT('',(223.0,-50.0,173.0));
#3816=DIRECTION('',(0.0,-1.0,0.0));
#3817=DIRECTION('',(1.0,0.0,0.0));
#3818=AXIS2_PLACEMENT_3D('',#3815,#3816,#3817);
#3819=CYLINDRICAL_SURFACE('',#3818,5.0);
#3820=ADVANCED_FACE('',(#3814),#3819,.T.);
#3821=SHAPE_ASPECT('featured shape: face 98',$,#38,.T.);
#3822=PROPERTY_DEFINITION('',$,#3821);
#3823=FACE_SHAPE_REPRESENTATION('',(#3820),#29);
#3824=PROPERTY_DEFINITION_REPRESENTATION(#3822,#3823);
#3825=ORIENTED_EDGE('',*,*,#2124,.T.);
#3826=ORIENTED_EDGE('',*,*,#1575,.T.);
#3827=ORIENTED_EDGE('',*,*,#2277,.F.);
```

```
#3828=ORIENTED_EDGE('',*,*,#387,.F.);
#3829=EDGE_LOOP('',(#3825,#3826,#3827,#3828));
#3830=FACE_BOUND('',#3829,.F.);
#3831=CARTESIAN_POINT('',(223.0,-50.0,7.0));
#3832=DIRECTION('',(0.0,1.0,0.0));
#3833=DIRECTION('',(1.0,0.0,0.0));
#3834=AXIS2_PLACEMENT_3D('',#3831,#3832,#3833);
#3835=CYLINDRICAL_SURFACE('',#3834,5.0);
#3836=ADVANCED_FACE('',(#3830),#3835,.T.);
#3837=SHAPE_ASPECT('featured shape: face 99',$,#38,.T.);
#3838=PROPERTY_DEFINITION('',$,#3837);
#3839=FACE_SHAPE_REPRESENTATION('',(#3836),#29);
#3840=PROPERTY_DEFINITION_REPRESENTATION(#3838,#3839);
#3841=ORIENTED_EDGE('',*,*,#2191,.T.);
#3842=ORIENTED_EDGE('',*,*,#1787,.T.);
#3843=ORIENTED_EDGE('',*,*,#2234,.F.);
#3844=ORIENTED_EDGE('',*,*,#569,.F.);
#3845=EDGE_LOOP('',(#3841,#3842,#3843,#3844));
#3846=FACE_BOUND('',#3845,.F.);
#3847=CARTESIAN_POINT('',(7.0,-50.0,173.0));
#3848=DIRECTION('',(0.0,1.0,0.0));
#3849=DIRECTION('',(-1.0,0.0,0.0));
#3850=AXIS2_PLACEMENT_3D('',#3847,#3848,#3849);
#3851=CYLINDRICAL_SURFACE('',#3850,5.0);
#3852=ADVANCED_FACE('',(#3846),#3851,.T.);
#3853=SHAPE_ASPECT('featured shape: face 100',$,#38,.T.);
#3854=PROPERTY_DEFINITION('',$,#3853);
#3855=FACE_SHAPE_REPRESENTATION('',(#3852),#29);
#3856=PROPERTY_DEFINITION_REPRESENTATION(#3854,#3855);
#3857=ORIENTED_EDGE('',*,*,#2167,.F.);
#3858=ORIENTED_EDGE('',*,*,#291,.T.);
#3859=ORIENTED_EDGE('',*,*,#2253,.T.);
#3860=ORIENTED_EDGE('',*,*,#1509,.F.);
#3861=EDGE_LOOP('',(#3857,#3858,#3859,#3860));
#3862=FACE_BOUND('',#3861,.F.);
#3863=CARTESIAN_POINT('',(7.0,-50.0,7.0));
#3864=DIRECTION('',(0.0,-1.0,0.0));
#3865=DIRECTION('',(-1.0,0.0,0.0));
#3866=AXIS2_PLACEMENT_3D('',#3863,#3864,#3865);
#3867=CYLINDRICAL_SURFACE('',#3866,5.0);
#3868=ADVANCED_FACE('',(#3862),#3867,.T.);
#3869=SHAPE_ASPECT('featured shape: face 101',$,#38,.T.);
#3870=PROPERTY_DEFINITION('',$,#3869);
#3871=FACE_SHAPE_REPRESENTATION('',(#3868),#29);
#3872=PROPERTY_DEFINITION_REPRESENTATION(#3870,#3871);
#3873=ORIENTED_EDGE('',*,*,#1045,.F.);
#3874=CARTESIAN_POINT('',(99.999999999999986,-15.609999999999999,10.0));
#3875=VERTEX_POINT('',#3874);
#3876=DIRECTION('',(0.0,1.0,0.0));
#3877=VECTOR('',#3876,15.610000000000003);
#3878=LINE('',#3874,#3877);
#3879=EDGE_CURVE('',#3875,#1039,#3878,.T.);
#3880=ORIENTED_EDGE('',*,*,#3879,.F.);
#3881=CARTESIAN_POINT('',(91.065103839083633,-12.632558739516522,10.0));
#3882=VERTEX_POINT('',#3881);
#3883=DIRECTION('',(-0.948710608132913,0.316145823973808,0.0));
```

```
#3884=VECTOR('',#3883,9.417936391056951);
#3885=LINE('',#3874,#3884);
#3886=EDGE_CURVE('',#3875,#3882,#3885,.T.);
#3887=ORIENTED_EDGE('',*,*,#3886,.T.);
#3888=CARTESIAN_POINT('',(68.934896160916992,-12.632558739516311,10.0));
#3889=VERTEX_POINT('',#3888);
#3890=CARTESIAN_POINT('',(80.0,-45.837430024168363,10.0));
#3891=DIRECTION('',(0.0,0.0,1.0));
#3892=DIRECTION('',(0.316145823973818,0.948710608132910,0.0));
#3893=AXIS2_PLACEMENT_3D('',#3890,#3891,#3892);
#3894=CIRCLE('',#3893,35.000000000004434);
#3895=EDGE_CURVE('',#3882,#3889,#3894,.T.);
#3896=ORIENTED_EDGE('',*,*,#3895,.T.);
#3897=CARTESIAN_POINT('',(60.0,-15.610000000000003,10.0));
#3898=VERTEX_POINT('',#3897);
#3899=DIRECTION('',(-0.948710608132913,-0.316145823973808,0.0));
#3900=VECTOR('',#3899,9.417936391056960);
#3901=LINE('',#3888,#3900);
#3902=EDGE_CURVE('',#3889,#3898,#3901,.T.);
#3903=ORIENTED_EDGE('',*,*,#3902,.T.);
#3904=DIRECTION('',(4.551843E-16,-1.0,0.0));
#3905=VECTOR('',#3904,15.610000000000001);
#3906=LINE('',#1040,#3905);
#3907=EDGE_CURVE('',#1041,#3898,#3906,.T.);
#3908=ORIENTED_EDGE('',*,*,#3907,.F.);
#3909=EDGE_LOOP('',(#3873,#3880,#3887,#3896,#3903,#3908));
#3910=FACE_BOUND('',#3909,.F.);
#3911=CARTESIAN_POINT('',(50.0,0.0,10.0));
#3912=DIRECTION('',(0.0,0.0,1.0));
#3913=DIRECTION('',(1.0,0.0,0.0));
#3914=AXIS2_PLACEMENT_3D('',#3911,#3912,#3913);
#3915=PLANE('',#3914);
#3916=ADVANCED_FACE('',(#3910),#3915,.T.);
#3917=SHAPE_ASPECT('featured shape: face 102',$,#38,.T.);
#3918=PROPERTY_DEFINITION('',$,#3917);
#3919=FACE_SHAPE_REPRESENTATION('',(#3916),#29);
#3920=PROPERTY_DEFINITION_REPRESENTATION(#3918,#3919);
#3921=ORIENTED_EDGE('',*,*,#1093,.F.);
#3922=CARTESIAN_POINT('',(109.999999999999986,-15.610000000000001,50.0));
#3923=VERTEX_POINT('',#3922);
#3924=DIRECTION('',(0.0,1.0,0.0));
#3925=VECTOR('',#3924,15.610000000000001);
#3926=LINE('',#3922,#3925);
#3927=EDGE_CURVE('',#3923,#1080,#3926,.T.);
#3928=ORIENTED_EDGE('',*,*,#3927,.F.);
#3929=CARTESIAN_POINT('',(109.999999999999986,-15.609999999999999,20.0));
#3930=VERTEX_POINT('',#3929);
#3931=DIRECTION('',(0.0,0.0,-1.0));
#3932=VECTOR('',#3931,30.0);
#3933=LINE('',#3922,#3932);
#3934=EDGE_CURVE('',#3923,#3930,#3933,.T.);
#3935=ORIENTED_EDGE('',*,*,#3934,.T.);
#3936=DIRECTION('',(0.0,1.0,0.0));
#3937=VECTOR('',#3936,15.609999999999999);
#3938=LINE('',#3929,#3937);
#3939=EDGE_CURVE('',#3930,#1089,#3938,.T.);
```



```
#3940=ORIENTED_EDGE('',*,*,#3939,.T.);
#3941=EDGE_LOOP('',(#3921,#3928,#3935,#3940));
#3942=FACE_BOUND('',#3941,.F.);
#3943=CARTESIAN_POINT('',(109.999999999999986,-15.609999999999999,90.0));
#3944=DIRECTION('',(1.0,0.0,0.0));
#3945=DIRECTION('',(0.0,1.0,0.0));
#3946=AXIS2_PLACEMENT_3D('',#3943,#3944,#3945);
#3947=PLANE('',#3946);
#3948=ADVANCED_FACE('',(#3942),#3947,.F.);
#3949=SHAPE_ASPECT('featured shape: face 103',$,#38,.T.);
#3950=PROPERTY_DEFINITION('',$,#3949);
#3951=FACE_SHAPE_REPRESENTATION('',(#3948),#29);
#3952=PROPERTY_DEFINITION_REPRESENTATION(#3950,#3951);
#3953=CARTESIAN_POINT('',(96.838541760261919,-25.097106081329134,20.0));
#3954=VERTEX_POINT('',#3953);
#3955=CARTESIAN_POINT('',(99.999999999999986,-15.609999999999999,20.0));
#3956=DIRECTION('',(0.0,0.0,-1.0));
#3957=DIRECTION('',(1.0,-1.776357E-16,0.0));
#3958=AXIS2_PLACEMENT_3D('',#3955,#3956,#3957);
#3959=CIRCLE('',#3958,10.0);
#3960=EDGE_CURVE('',#3930,#3954,#3959,.T.);
#3961=ORIENTED_EDGE('',*,*,#3960,.F.);
#3962=ORIENTED_EDGE('',*,*,#3934,.F.);
#3963=CARTESIAN_POINT('',(96.838541760261919,-25.097106081329134,50.0));
#3964=VERTEX_POINT('',#3963);
#3965=CARTESIAN_POINT('',(99.999999999999986,-15.609999999999999,50.0));
#3966=DIRECTION('',(0.0,0.0,1.000000000000000));
#3967=DIRECTION('',(-0.316145823973807,-0.948710608132914,0.0));
#3968=AXIS2_PLACEMENT_3D('',#3965,#3966,#3967);
#3969=CIRCLE('',#3968,10.0);
#3970=EDGE_CURVE('',#3964,#3923,#3969,.T.);
#3971=ORIENTED_EDGE('',*,*,#3970,.F.);
#3972=DIRECTION('',(0.0,0.0,-1.0));
#3973=VECTOR('',#3972,30.0);
#3974=LINE('',#3963,#3973);
#3975=EDGE_CURVE('',#3964,#3954,#3974,.T.);
#3976=ORIENTED_EDGE('',*,*,#3975,.T.);
#3977=EDGE_LOOP('',(#3961,#3962,#3971,#3976));
#3978=FACE_BOUND('',#3977,.F.);
#3979=CARTESIAN_POINT('',(99.999999999999986,-15.609999999999999,90.0));
#3980=DIRECTION('',(0.0,0.0,1.0));
#3981=DIRECTION('',(-1.0,0.0,0.0));
#3982=AXIS2_PLACEMENT_3D('',#3979,#3980,#3981);
#3983=CYLINDRICAL_SURFACE('',#3982,10.0);
#3984=ADVANCED_FACE('',(#3978),#3983,.F.);
#3985=SHAPE_ASPECT('featured shape: face 104',$,#38,.T.);
#3986=PROPERTY_DEFINITION('',$,#3985);
#3987=FACE_SHAPE_REPRESENTATION('',(#3984),#29);
#3988=PROPERTY_DEFINITION_REPRESENTATION(#3986,#3987);
#3989=CARTESIAN_POINT('',(87.903645599345182,-22.119664820845532,20.0));
#3990=VERTEX_POINT('',#3989);
#3991=DIRECTION('',(-0.948710608132913,0.316145823973807,0.0));
#3992=VECTOR('',#3991,9.417936391056967);
#3993=LINE('',#3953,#3992);
#3994=EDGE_CURVE('',#3954,#3990,#3993,.T.);
#3995=ORIENTED_EDGE('',*,*,#3994,.F.);
```

```
#3996=ORIENTED_EDGE('',*,*,#3975,.F.);
#3997=CARTESIAN_POINT('',(87.903645599345182,-22.119664820845532,50.0));
#3998=VERTEX_POINT('',#3997);
#3999=DIRECTION('',(-0.948710608132914,0.316145823973807,0.0));
#4000=VECTOR('',#3999,9.417936391056950);
#4001=LINE('',#3963,#4000);
#4002=EDGE_CURVE('',#3964,#3998,#4001,.T.);
#4003=ORIENTED_EDGE('',*,*,#4002,.T.);
#4004=DIRECTION('',(-4.736952E-16,2.368476E-16,-1.0));
#4005=VECTOR('',#4004,30.0);
#4006=LINE('',#3997,#4005);
#4007=EDGE_CURVE('',#3998,#3990,#4006,.T.);
#4008=ORIENTED_EDGE('',*,*,#4007,.T.);
#4009=EDGE_LOOP('',(#3995,#3996,#4003,#4008));
#4010=FACE_BOUND('',#4009,.F.);
#4011=CARTESIAN_POINT('',(87.903645599345182,-22.119664820845532,90.0));
#4012=DIRECTION('',(-0.316145823973807,-0.948710608132913,0.0));
#4013=DIRECTION('',(0.948710608132913,-0.316145823973807,0.0));
#4014=AXIS2_PLACEMENT_3D('',#4011,#4012,#4013);
#4015=PLANE('',#4014);
#4016=ADVANCED_FACE('',(#4010),#4015,.F.);
#4017=SHAPE_ASPECT('featured shape: face 105',$,#38,.T.);
#4018=PROPERTY_DEFINITION('',$,#4017);
#4019=FACE_SHAPE_REPRESENTATION('',(#4016),#29);
#4020=PROPERTY_DEFINITION_REPRESENTATION(#4018,#4019);
#4021=CARTESIAN_POINT('',(72.096354400654803,-22.119664820845529,20.0));
#4022=VERTEX_POINT('',#4021);
#4023=CARTESIAN_POINT('',(80.0,-45.837430024168363,20.000000000000021));
#4024=DIRECTION('',(0.0,0.0,1.000000000000000));
#4025=DIRECTION('',(0.316145823973807,0.948710608132914,0.0));
#4026=AXIS2_PLACEMENT_3D('',#4023,#4024,#4025);
#4027=CIRCLE('',#4026,25.0);
#4028=EDGE_CURVE('',#3990,#4022,#4027,.T.);
#4029=ORIENTED_EDGE('',*,*,#4028,.F.);
#4030=ORIENTED_EDGE('',*,*,#4007,.F.);
#4031=CARTESIAN_POINT('',(72.096354400654803,-22.119664820845529,50.0));
#4032=VERTEX_POINT('',#4031);
#4033=CARTESIAN_POINT('',(80.0,-45.837430024168363,49.999999999999979));
#4034=DIRECTION('',(0.0,0.0,1.000000000000000));
#4035=DIRECTION('',(0.316145823973808,0.948710608132913,0.0));
#4036=AXIS2_PLACEMENT_3D('',#4033,#4034,#4035);
#4037=CIRCLE('',#4036,24.999999999999993);
#4038=EDGE_CURVE('',#3998,#4032,#4037,.T.);
#4039=ORIENTED_EDGE('',*,*,#4038,.T.);
#4040=DIRECTION('',(0.0,0.0,-1.0));
#4041=VECTOR('',#4040,30.0);
#4042=LINE('',#4031,#4041);
#4043=EDGE_CURVE('',#4032,#4022,#4042,.T.);
#4044=ORIENTED_EDGE('',*,*,#4043,.T.);
#4045=EDGE_LOOP('',(#4029,#4030,#4039,#4044));
#4046=FACE_BOUND('',#4045,.F.);
#4047=CARTESIAN_POINT('',(80.0,-45.837430024168363,90.0));
#4048=DIRECTION('',(0.0,0.0,1.0));
#4049=DIRECTION('',(-1.0,0.0,0.0));
#4050=AXIS2_PLACEMENT_3D('',#4047,#4048,#4049);
#4051=CYLINDRICAL_SURFACE('',#4050,25.0);
```



```
#4052=ADVANCED_FACE('',(#4046),#4051,.T.);
#4053=SHAPE_ASPECT('featured shape: face 106',$,#38,.T.);
#4054=PROPERTY_DEFINITION('',$,#4053);
#4055=FACE_SHAPE_REPRESENTATION('',(#4052),#29);
#4056=PROPERTY_DEFINITION_REPRESENTATION(#4054,#4055);
#4057=CARTESIAN_POINT('',(63.161458239738074,-25.097106081329137,20.0));
#4058=VERTEX_POINT('',#4057);
#4059=DIRECTION('',(-0.948710608132914,-0.316145823973807,0.0));
#4060=VECTOR('',#4059,9.417936391056950);
#4061=LINE('',#4021,#4060);
#4062=EDGE_CURVE('',#4022,#4058,#4061,.T.);
#4063=ORIENTED_EDGE('',*,*,#4062,.F.);
#4064=ORIENTED_EDGE('',*,*,#4043,.F.);
#4065=CARTESIAN_POINT('',(63.161458239738074,-25.097106081329137,50.0));
#4066=VERTEX_POINT('',#4065);
#4067=DIRECTION('',(-0.948710608132913,-0.316145823973807,0.0));
#4068=VECTOR('',#4067,9.417936391056951);
#4069=LINE('',#4031,#4068);
#4070=EDGE_CURVE('',#4032,#4066,#4069,.T.);
#4071=ORIENTED_EDGE('',*,*,#4070,.T.);
#4072=DIRECTION('',(0.0,0.0,-1.0));
#4073=VECTOR('',#4072,30.0);
#4074=LINE('',#4065,#4073);
#4075=EDGE_CURVE('',#4066,#4058,#4074,.T.);
#4076=ORIENTED_EDGE('',*,*,#4075,.T.);
#4077=EDGE_LOOP('',(#4063,#4064,#4071,#4076));
#4078=FACE_BOUND('',#4077,.F.);
#4079=CARTESIAN_POINT('',(63.161458239738060,-25.097106081329141,90.0));
#4080=DIRECTION('',(0.316145823973807,-0.948710608132914,0.0));
#4081=DIRECTION('',(0.948710608132914,0.316145823973807,0.0));
#4082=AXIS2_PLACEMENT_3D('',#4079,#4080,#4081);
#4083=PLANE('',#4082);
#4084=ADVANCED_FACE('',(#4078),#4083,.F.);
#4085=SHAPE_ASPECT('featured shape: face 107',$,#38,.T.);
#4086=PROPERTY_DEFINITION('',$,#4085);
#4087=FACE_SHAPE_REPRESENTATION('',(#4084),#29);
#4088=PROPERTY_DEFINITION_REPRESENTATION(#4086,#4087);
#4089=CARTESIAN_POINT('',(50.0,-15.609999999999998,20.0));
#4090=VERTEX_POINT('',#4089);
#4091=CARTESIAN_POINT('',(60.0,-15.609999999999992,20.0));
#4092=DIRECTION('',(0.0,0.0,-1.0));
#4093=DIRECTION('',(0.316145823973808,-0.948710608132913,0.0));
#4094=AXIS2_PLACEMENT_3D('',#4091,#4092,#4093);
#4095=CIRCLE('',#4094,10.0);
#4096=EDGE_CURVE('',#4058,#4090,#4095,.T.);
#4097=ORIENTED_EDGE('',*,*,#4096,.F.);
#4098=ORIENTED_EDGE('',*,*,#4075,.F.);
#4099=CARTESIAN_POINT('',(50.0,-15.609999999999998,50.0));
#4100=VERTEX_POINT('',#4099);
#4101=CARTESIAN_POINT('',(60.0,-15.609999999999992,50.0));
#4102=DIRECTION('',(0.0,0.0,1.0));
#4103=DIRECTION('',(-1.0,-5.329071E-16,0.0));
#4104=AXIS2_PLACEMENT_3D('',#4101,#4102,#4103);
#4105=CIRCLE('',#4104,10.0);
#4106=EDGE_CURVE('',#4100,#4066,#4105,.T.);
#4107=ORIENTED_EDGE('',*,*,#4106,.F.);
```

```
#4108=DIRECTION('',(0.0,0.0,-1.0));
#4109=VECTOR('',#4108,30.0);
#4110=LINE('',#4099,#4109);
#4111=EDGE_CURVE('',#4100,#4090,#4110,.T.);
#4112=ORIENTED_EDGE('',*,*,#4111,.T.);
#4113=EDGE_LOOP('',(#4097,#4098,#4107,#4112));
#4114=FACE_BOUND('',#4113,.F.);
#4115=CARTESIAN_POINT('',(60.0,-15.609999999999992,90.0));
#4116=DIRECTION('',(0.0,0.0,1.0));
#4117=DIRECTION('',(-1.0,0.0,0.0));
#4118=AXIS2_PLACEMENT_3D('',#4115,#4116,#4117);
#4119=CYLINDRICAL_SURFACE('',#4118,10.0);
#4120=ADVANCED_FACE('',(#4114),#4119,.F.);
#4121=SHAPE_ASPECT('featured shape: face 108',$,#38,.T.);
#4122=PROPERTY_DEFINITION('',$,#4121);
#4123=FACE_SHAPE_REPRESENTATION('',(#4120),#29);
#4124=PROPERTY_DEFINITION_REPRESENTATION(#4122,#4123);
#4125=ORIENTED_EDGE('',*,*,#1061,.T.);
#4126=DIRECTION('',(4.551843E-16,-1.0,0.0));
#4127=VECTOR('',#4126,15.609999999999999);
#4128=LINE('',#1047,#4127);
#4129=EDGE_CURVE('',#1048,#4090,#4128,.T.);
#4130=ORIENTED_EDGE('',*,*,#4129,.T.);
#4131=ORIENTED_EDGE('',*,*,#4111,.F.);
#4132=DIRECTION('',(4.551843E-16,-1.0,0.0));
#4133=VECTOR('',#4132,15.609999999999998);
#4134=LINE('',#1056,#4133);
#4135=EDGE_CURVE('',#1057,#4100,#4134,.T.);
#4136=ORIENTED_EDGE('',*,*,#4135,.F.);
#4137=EDGE_LOOP('',(#4125,#4130,#4131,#4136));
#4138=FACE_BOUND('',#4137,.F.);
#4139=CARTESIAN_POINT('',(49.999999999999993,0.0,90.0));
#4140=DIRECTION('',(-1.0,-4.551843E-16,0.0));
#4141=DIRECTION('',(4.551843E-16,-1.0,0.0));
#4142=AXIS2_PLACEMENT_3D('',#4139,#4140,#4141);
#4143=PLANE('',#4142);
#4144=ADVANCED_FACE('',(#4138),#4143,.F.);
#4145=SHAPE_ASPECT('featured shape: face 109',$,#38,.T.);
#4146=PROPERTY_DEFINITION('',$,#4145);
#4147=FACE_SHAPE_REPRESENTATION('',(#4144),#29);
#4148=PROPERTY_DEFINITION_REPRESENTATION(#4146,#4147);
#4149=ORIENTED_EDGE('',*,*,#1077,.T.);
#4150=CARTESIAN_POINT('',(60.0,-15.610000000000003,60.0));
#4151=VERTEX_POINT('',#4150);
#4152=DIRECTION('',(4.551843E-16,-1.0,0.0));
#4153=VECTOR('',#4152,15.609999999999999);
#4154=LINE('',#1063,#4153);
#4155=EDGE_CURVE('',#1064,#4151,#4154,.T.);
#4156=ORIENTED_EDGE('',*,*,#4155,.T.);
#4157=CARTESIAN_POINT('',(68.934896160916736,-12.632558739516394,60.0));
#4158=VERTEX_POINT('',#4157);
#4159=DIRECTION('',(-0.948710608132913,-0.316145823973807,0.0));
#4160=VECTOR('',#4159,9.417936391056944);
#4161=LINE('',#4157,#4160);
#4162=EDGE_CURVE('',#4158,#4151,#4161,.T.);
#4163=ORIENTED_EDGE('',*,*,#4162,.F.);
```



```
#4164=CARTESIAN_POINT('',(91.065103839083264,-12.632558739516396,60.0));
#4165=VERTEX_POINT('',#4164);
#4166=CARTESIAN_POINT('',(80.0,-45.837430024168363,60.0));
#4167=DIRECTION('',(0.0,0.0,1.000000000000000));
#4168=DIRECTION('',(0.316145823973807,0.948710608132913,0.0));
#4169=AXIS2_PLACEMENT_3D('',#4166,#4167,#4168);
#4170=CIRCLE('',#4169,35.000000000005265);
#4171=EDGE_CURVE('',#4165,#4158,#4170,.T.);
#4172=ORIENTED_EDGE('',*,*,#4171,.F.);
#4173=CARTESIAN_POINT('',(99.999999999999986,-15.610000000000001,60.0));
#4174=VERTEX_POINT('',#4173);
#4175=DIRECTION('',(-0.948710608132913,0.316145823973808,0.0));
#4176=VECTOR('',#4175,9.417936391056955);
#4177=LINE('',#4173,#4176);
#4178=EDGE_CURVE('',#4174,#4165,#4177,.T.);
#4179=ORIENTED_EDGE('',*,*,#4178,.F.);
#4180=DIRECTION('',(0.0,1.0,0.0));
#4181=VECTOR('',#4180,15.610000000000003);
#4182=LINE('',#4173,#4181);
#4183=EDGE_CURVE('',#4174,#1073,#4182,.T.);
#4184=ORIENTED_EDGE('',*,*,#4183,.T.);
#4185=EDGE_LOOP('',(#4149,#4156,#4163,#4172,#4179,#4184));
#4186=FACE_BOUND('',#4185,.F.);
#4187=CARTESIAN_POINT('',(50.0,0.0,60.0));
#4188=DIRECTION('',(0.0,0.0,1.0));
#4189=DIRECTION('',(1.0,0.0,0.0));
#4190=AXIS2_PLACEMENT_3D('',#4187,#4188,#4189);
#4191=PLANE('',#4190);
#4192=ADVANCED_FACE('',(#4186),#4191,.F.);
#4193=SHAPE_ASPECT('featured shape: face 110',$,#38,.T.);
#4194=PROPERTY_DEFINITION('',$,#4193);
#4195=FACE_SHAPE_REPRESENTATION('',(#4192),#29);
#4196=PROPERTY_DEFINITION_REPRESENTATION(#4194,#4195);
#4197=ORIENTED_EDGE('',*,*,#1054,.T.);
#4198=ORIENTED_EDGE('',*,*,#3907,.T.);
#4199=CARTESIAN_POINT('',(60.0,-15.609999999999992,20.0));
#4200=DIRECTION('',(-5.532395E-16,1.0,0.0));
#4201=DIRECTION('',(0.0,0.0,-1.0));
#4202=AXIS2_PLACEMENT_3D('',#4199,#4200,#4201);
#4203=CIRCLE('',#4202,10.0);
#4204=EDGE_CURVE('',#3898,#4090,#4203,.T.);
#4205=ORIENTED_EDGE('',*,*,#4204,.T.);
#4206=ORIENTED_EDGE('',*,*,#4129,.F.);
#4207=EDGE_LOOP('',(#4197,#4198,#4205,#4206));
#4208=FACE_BOUND('',#4207,.F.);
#4209=CARTESIAN_POINT('',(60.000000000000007,-25.643252392135473,20.0));
#4210=DIRECTION('',(-5.532395E-16,1.0,0.0));
#4211=DIRECTION('',(0.0,0.0,-1.0));
#4212=AXIS2_PLACEMENT_3D('',#4209,#4210,#4211);
#4213=CYLINDRICAL_SURFACE('',#4212,10.0);
#4214=ADVANCED_FACE('',(#4208),#4213,.F.);
#4215=SHAPE_ASPECT('featured shape: face 111',$,#38,.T.);
#4216=PROPERTY_DEFINITION('',$,#4215);
#4217=FACE_SHAPE_REPRESENTATION('',(#4214),#29);
#4218=PROPERTY_DEFINITION_REPRESENTATION(#4216,#4217);
#4219=ORIENTED_EDGE('',*,*,#3902,.F.);
```

```
#4220=CARTESIAN_POINT('',(68.934896160916736,-12.632558739516401,20.0));
#4221=DIRECTION('',(-0.948710608132914,-0.316145823973807,-1.684763E-16));
#4222=DIRECTION('',(0.0,5.329071E-16,-1.0));
#4223=AXIS2_PLACEMENT_3D('',#4220,#4221,#4222);
#4224=CIRCLE('',#4223,10.0);
#4225=EDGE_CURVE('',#3889,#4022,#4224,.T.);
#4226=ORIENTED_EDGE('',*,*,#4225,.T.);
#4227=ORIENTED_EDGE('',*,*,#4062,.T.);
#4228=CARTESIAN_POINT('',(60.000000000000043,-15.609999999999989,20.0));
#4229=DIRECTION('',(-0.948710608132914,-0.316145823973807,1.123176E-16));
#4230=DIRECTION('',(0.0,-3.552714E-16,-1.0));
#4231=AXIS2_PLACEMENT_3D('',#4228,#4229,#4230);
#4232=CIRCLE('',#4231,10.0);
#4233=EDGE_CURVE('',#3898,#4058,#4232,.T.);
#4234=ORIENTED_EDGE('',*,*,#4233,.F.);
#4235=EDGE_LOOP('',(#4219,#4226,#4227,#4234));
#4236=FACE_BOUND('',#4235,.F.);
#4237=CARTESIAN_POINT('',(106.973637467630653,0.043371212405010,20.0));
#4238=DIRECTION('',(-0.948710608132914,-0.316145823973807,0.0));
#4239=DIRECTION('',(0.0,0.0,-1.0));
#4240=AXIS2_PLACEMENT_3D('',#4237,#4238,#4239);
#4241=CYLINDRICAL_SURFACE('',#4240,10.0);
#4242=ADVANCED_FACE('',(#4236),#4241,.F.);
#4243=SHAPE_ASPECT('featured shape: face 112',$,#38,.T.);
#4244=PROPERTY_DEFINITION('',$,#4243);
#4245=FACE_SHAPE_REPRESENTATION('',(#4242),#29);
#4246=PROPERTY_DEFINITION_REPRESENTATION(#4244,#4245);
#4247=ORIENTED_EDGE('',*,*,#3895,.F.);
#4248=CARTESIAN_POINT('',(91.065103839083250,-12.632558739516391,20.0));
#4249=DIRECTION('',(-0.948710608132914,0.316145823973807,0.0));
#4250=DIRECTION('',(0.0,0.0,-1.0));
#4251=AXIS2_PLACEMENT_3D('',#4248,#4249,#4250);
#4252=CIRCLE('',#4251,10.0);
#4253=EDGE_CURVE('',#3882,#3990,#4252,.T.);
#4254=ORIENTED_EDGE('',*,*,#4253,.T.);
#4255=ORIENTED_EDGE('',*,*,#4028,.T.);
#4256=ORIENTED_EDGE('',*,*,#4225,.F.);
#4257=EDGE_LOOP('',(#4247,#4254,#4255,#4256));
#4258=FACE_BOUND('',#4257,.F.);
#4259=CARTESIAN_POINT('',(80.0,-45.837430024168363,20.0));
#4260=DIRECTION('',(0.0,0.0,1.0));
#4261=DIRECTION('',(0.316336760367410,0.948646959643181,0.0));
#4262=AXIS2_PLACEMENT_3D('',#4259,#4260,#4261);
#4263=TOROIDAL_SURFACE('',#4262,35.0,10.0);
#4264=ADVANCED_FACE('',(#4258),#4263,.F.);
#4265=SHAPE_ASPECT('featured shape: face 113',$,#38,.T.);
#4266=PROPERTY_DEFINITION('',$,#4265);
#4267=FACE_SHAPE_REPRESENTATION('',(#4264),#29);
#4268=PROPERTY_DEFINITION_REPRESENTATION(#4266,#4267);
#4269=ORIENTED_EDGE('',*,*,#3994,.T.);
#4270=ORIENTED_EDGE('',*,*,#4253,.F.);
#4271=ORIENTED_EDGE('',*,*,#3886,.F.);
#4272=CARTESIAN_POINT('',(99.999999999999986,-15.610000000000003,20.0));
#4273=DIRECTION('',(-0.948710608132914,0.316145823973807,1.684763E-16));
#4274=DIRECTION('',(0.0,5.329071E-16,-1.0));
#4275=AXIS2_PLACEMENT_3D('',#4272,#4273,#4274);
```



```
#4276=CIRCLE('',#4275,10.0);
#4277=EDGE_CURVE('',#3875,#3954,#4276,.T.);
#4278=ORIENTED_EDGE('',*,*,#4277,.T.);
#4279=EDGE_LOOP('',(#4269,#4270,#4271,#4278));
#4280=FACE_BOUND('',#4279,.F.);
#4281=CARTESIAN_POINT('',(110.060467428838379,-18.962523664841502,20.0));
#4282=DIRECTION('',(-0.948710608132914,0.316145823973807,0.0));
#4283=DIRECTION('',(0.0,0.0,-1.0));
#4284=AXIS2_PLACEMENT_3D('',#4281,#4282,#4283);
#4285=CYLINDRICAL_SURFACE('',#4284,10.0);
#4286=ADVANCED_FACE('',(#4280),#4285,.F.);
#4287=SHAPE_ASPECT('featured shape: face 114',$,#38,.T.);
#4288=PROPERTY_DEFINITION('',$,#4287);
#4289=FACE_SHAPE_REPRESENTATION('',(#4286),#29);
#4290=PROPERTY_DEFINITION_REPRESENTATION(#4288,#4289);
#4291=ORIENTED_EDGE('',*,*,#1100,.F.);
#4292=ORIENTED_EDGE('',*,*,#3939,.F.);
#4293=CARTESIAN_POINT('',(99.999999999999986,-15.609999999999994,20.0));
#4294=DIRECTION('',(0.0,-1.0,0.0));
#4295=DIRECTION('',(0.0,0.0,-1.0));
#4296=AXIS2_PLACEMENT_3D('',#4293,#4294,#4295);
#4297=CIRCLE('',#4296,10.0);
#4298=EDGE_CURVE('',#3875,#3930,#4297,.T.);
#4299=ORIENTED_EDGE('',*,*,#4298,.F.);
#4300=ORIENTED_EDGE('',*,*,#3879,.T.);
#4301=EDGE_LOOP('',(#4291,#4292,#4299,#4300));
#4302=FACE_BOUND('',#4301,.F.);
#4303=CARTESIAN_POINT('',(99.999999999999986,0.043371212405010,20.0));
#4304=DIRECTION('',(0.0,-1.0,0.0));
#4305=DIRECTION('',(0.0,0.0,-1.0));
#4306=AXIS2_PLACEMENT_3D('',#4303,#4304,#4305);
#4307=CYLINDRICAL_SURFACE('',#4306,10.0);
#4308=ADVANCED_FACE('',(#4302),#4307,.F.);
#4309=SHAPE_ASPECT('featured shape: face 115',$,#38,.T.);
#4310=PROPERTY_DEFINITION('',$,#4309);
#4311=FACE_SHAPE_REPRESENTATION('',(#4308),#29);
#4312=PROPERTY_DEFINITION_REPRESENTATION(#4310,#4311);
#4313=ORIENTED_EDGE('',*,*,#3960,.T.);
#4314=ORIENTED_EDGE('',*,*,#4277,.F.);
#4315=ORIENTED_EDGE('',*,*,#4298,.T.);
#4316=EDGE_LOOP('',(#4313,#4314,#4315));
#4317=FACE_BOUND('',#4316,.F.);
#4318=CARTESIAN_POINT('',(99.999999999999986,-15.610000000000003,20.0));
#4319=DIRECTION('',(0.223548855975687,0.670839704394397,-0.707106781186547));
#4320=DIRECTION('',(-0.223548855975687,-0.670839704394397,-
0.707106781186548));
#4321=AXIS2_PLACEMENT_3D('',#4318,#4319,#4320);
#4322=SPHERICAL_SURFACE('',#4321,10.0);
#4323=ADVANCED_FACE('',(#4317),#4322,.F.);
#4324=SHAPE_ASPECT('featured shape: face 116',$,#38,.T.);
#4325=PROPERTY_DEFINITION('',$,#4324);
#4326=FACE_SHAPE_REPRESENTATION('',(#4323),#29);
#4327=PROPERTY_DEFINITION_REPRESENTATION(#4325,#4326);
#4328=ORIENTED_EDGE('',*,*,#4096,.T.);
#4329=ORIENTED_EDGE('',*,*,#4204,.F.);
#4330=ORIENTED_EDGE('',*,*,#4233,.T.);
```



```
#4331=EDGE_LOOP('',(#4328,#4329,#4330));
#4332=FACE_BOUND('',#4331,.F.);
#4333=CARTESIAN_POINT('',(60.0,-15.610000000000001,20.0));
#4334=DIRECTION('',(0.707106781186547,3.768222E-16,-0.707106781186548));
#4335=DIRECTION('',(-0.707106781186547,-3.768222E-16,-0.707106781186547));
#4336=AXIS2_PLACEMENT_3D('',#4333,#4334,#4335);
#4337=SPHERICAL_SURFACE('',#4336,10.0);
#4338=ADVANCED_FACE('',(#4332),#4337,.F.);
#4339=SHAPE_ASPECT('featured shape: face 117',$,#38,.T.);
#4340=PROPERTY_DEFINITION('',$,#4339);
#4341=FACE_SHAPE_REPRESENTATION('',(#4338),#29);
#4342=PROPERTY_DEFINITION_REPRESENTATION(#4340,#4341);
#4343=ORIENTED_EDGE('',*,*,#1070,.T.);
#4344=ORIENTED_EDGE('',*,*,#4135,.T.);
#4345=CARTESIAN_POINT('',(60.0,-15.610000000000007,50.0));
#4346=DIRECTION('',(-5.329071E-16,1.0,0.0));
#4347=DIRECTION('',(-1.0,-5.329071E-16,0.0));
#4348=AXIS2_PLACEMENT_3D('',#4345,#4346,#4347);
#4349=CIRCLE('',#4348,10.0);
#4350=EDGE_CURVE('',#4100,#4151,#4349,.T.);
#4351=ORIENTED_EDGE('',*,*,#4350,.T.);
#4352=ORIENTED_EDGE('',*,*,#4155,.F.);
#4353=EDGE_LOOP('',(#4343,#4344,#4351,#4352));
#4354=FACE_BOUND('',#4353,.F.);
#4355=CARTESIAN_POINT('',(60.000000000000007,-25.649428979163300,50.0));
#4356=DIRECTION('',(-5.533360E-16,1.0,0.0));
#4357=DIRECTION('',(-1.0,-5.533360E-16,0.0));
#4358=AXIS2_PLACEMENT_3D('',#4355,#4356,#4357);
#4359=CYLINDRICAL_SURFACE('',#4358,10.0);
#4360=ADVANCED_FACE('',(#4354),#4359,.F.);
#4361=SHAPE_ASPECT('featured shape: face 118',$,#38,.T.);
#4362=PROPERTY_DEFINITION('',$,#4361);
#4363=FACE_SHAPE_REPRESENTATION('',(#4360),#29);
#4364=PROPERTY_DEFINITION_REPRESENTATION(#4362,#4363);
#4365=ORIENTED_EDGE('',*,*,#4070,.F.);
#4366=CARTESIAN_POINT('',(68.934896160916765,-12.632558739516384,50.0));
#4367=DIRECTION('',(-0.948710608132914,-0.316145823973807,0.0));
#4368=DIRECTION('',(0.316145823973807,-0.948710608132914,0.0));
#4369=AXIS2_PLACEMENT_3D('',#4366,#4367,#4368);
#4370=CIRCLE('',#4369,10.0);
#4371=EDGE_CURVE('',#4032,#4158,#4370,.T.);
#4372=ORIENTED_EDGE('',*,*,#4371,.T.);
#4373=ORIENTED_EDGE('',*,*,#4162,.T.);
#4374=CARTESIAN_POINT('',(59.999999999999957,-15.610000000000014,50.0));
#4375=DIRECTION('',(-0.948710608132913,-0.316145823973807,0.0));
#4376=DIRECTION('',(0.316145823973807,-0.948710608132913,0.0));
#4377=AXIS2_PLACEMENT_3D('',#4374,#4375,#4376);
#4378=CIRCLE('',#4377,10.0);
#4379=EDGE_CURVE('',#4066,#4151,#4378,.T.);
#4380=ORIENTED_EDGE('',*,*,#4379,.F.);
#4381=EDGE_LOOP('',(#4365,#4372,#4373,#4380));
#4382=FACE_BOUND('',#4381,.F.);
#4383=CARTESIAN_POINT('',(106.941654523051668,0.032713300202545,50.0));
#4384=DIRECTION('',(-0.948710608132914,-0.316145823973807,0.0));
#4385=DIRECTION('',(0.316145823973807,-0.948710608132914,0.0));
#4386=AXIS2_PLACEMENT_3D('',#4383,#4384,#4385);
```

```
#4387=CYLINDRICAL_SURFACE('',#4386,10.0);
#4388=ADVANCED_FACE('',(#4382),#4387,.F.);
#4389=SHAPE_ASPECT('featured shape: face 119',$,#38,.T.);
#4390=PROPERTY_DEFINITION('',$,#4389);
#4391=FACE_SHAPE_REPRESENTATION('',(#4388),#29);
#4392=PROPERTY_DEFINITION_REPRESENTATION(#4390,#4391);
#4393=ORIENTED_EDGE('',*,*,#4038,.F.);
#4394=CARTESIAN_POINT('',(91.065103839083264,-12.632558739516391,50.0));
#4395=DIRECTION('',(-0.948710608132914,0.316145823973807,0.0));
#4396=DIRECTION('',(-0.316145823973807,-0.948710608132914,0.0));
#4397=AXIS2_PLACEMENT_3D('',#4394,#4395,#4396);
#4398=CIRCLE('',#4397,10.0);
#4399=EDGE_CURVE('',#3998,#4165,#4398,.T.);
#4400=ORIENTED_EDGE('',*,*,#4399,.T.);
#4401=ORIENTED_EDGE('',*,*,#4171,.T.);
#4402=ORIENTED_EDGE('',*,*,#4371,.F.);
#4403=EDGE_LOOP('',(#4393,#4400,#4401,#4402));
#4404=FACE_BOUND('',#4403,.F.);
#4405=CARTESIAN_POINT('',(80.0,-45.837430024168363,50.0));
#4406=DIRECTION('',(0.0,0.0,1.0));
#4407=DIRECTION('',(0.316345968619440,0.948643889000624,0.0));
#4408=AXIS2_PLACEMENT_3D('',#4405,#4406,#4407);
#4409=TOROIDAL_SURFACE('',#4408,34.999999999999993,10.0);
#4410=ADVANCED_FACE('',(#4404),#4409,.F.);
#4411=SHAPE_ASPECT('featured shape: face 120',$,#38,.T.);
#4412=PROPERTY_DEFINITION('',$,#4411);
#4413=FACE_SHAPE_REPRESENTATION('',(#4410),#29);
#4414=PROPERTY_DEFINITION_REPRESENTATION(#4412,#4413);
#4415=ORIENTED_EDGE('',*,*,#4178,.T.);
#4416=ORIENTED_EDGE('',*,*,#4399,.F.);
#4417=ORIENTED_EDGE('',*,*,#4002,.F.);
#4418=CARTESIAN_POINT('',(100.0,-15.610000000000007,50.0));
#4419=DIRECTION('',(-0.948710608132914,0.316145823973807,0.0));
#4420=DIRECTION('',(-0.316145823973807,-0.948710608132914,0.0));
#4421=AXIS2_PLACEMENT_3D('',#4418,#4419,#4420);
#4422=CIRCLE('',#4421,10.0);
#4423=EDGE_CURVE('',#3964,#4174,#4422,.T.);
#4424=ORIENTED_EDGE('',*,*,#4423,.T.);
#4425=EDGE_LOOP('',(#4415,#4416,#4417,#4424));
#4426=FACE_BOUND('',#4425,.F.);
#4427=CARTESIAN_POINT('',(110.080242558619432,-18.969113476998370,50.0));
#4428=DIRECTION('',(-0.948710608132914,0.316145823973807,0.0));
#4429=DIRECTION('',(-0.316145823973807,-0.948710608132914,0.0));
#4430=AXIS2_PLACEMENT_3D('',#4427,#4428,#4429);
#4431=CYLINDRICAL_SURFACE('',#4430,10.0);
#4432=ADVANCED_FACE('',(#4426),#4431,.F.);
#4433=SHAPE_ASPECT('featured shape: face 121',$,#38,.T.);
#4434=PROPERTY_DEFINITION('',$,#4433);
#4435=FACE_SHAPE_REPRESENTATION('',(#4432),#29);
#4436=PROPERTY_DEFINITION_REPRESENTATION(#4434,#4435);
#4437=ORIENTED_EDGE('',*,*,#1086,.F.);
#4438=ORIENTED_EDGE('',*,*,#4183,.F.);
#4439=CARTESIAN_POINT('',(99.999999999999986,-15.610000000000010,50.0));
#4440=DIRECTION('',(0.0,-1.0,0.0));
#4441=DIRECTION('',(1.0,0.0,0.0));
#4442=AXIS2_PLACEMENT_3D('',#4439,#4440,#4441);
```

```
#4443=CIRCLE('',#4442,10.0);
#4444=EDGE_CURVE('',#3923,#4174,#4443,.T.);
#4445=ORIENTED_EDGE('',*,*,#4444,.F.);
#4446=ORIENTED_EDGE('',*,*,#3927,.T.);
#4447=EDGE_LOOP('',(#4437,#4438,#4445,#4446));
#4448=FACE_BOUND('',#4447,.F.);
#4449=CARTESIAN_POINT('',(99.999999999999986,0.032713300202545,50.0));
#4450=DIRECTION('',(0.0,-1.0,0.0));
#4451=DIRECTION('',(1.0,0.0,0.0));
#4452=AXIS2_PLACEMENT_3D('',#4449,#4450,#4451);
#4453=CYLINDRICAL_SURFACE('',#4452,10.0);
#4454=ADVANCED_FACE('',(#4448),#4453,.F.);
#4455=SHAPE_ASPECT('featured shape: face 122',$,#38,.T.);
#4456=PROPERTY_DEFINITION('',$,#4455);
#4457=FACE_SHAPE_REPRESENTATION('',(#4454),#29);
#4458=PROPERTY_DEFINITION_REPRESENTATION(#4456,#4457);
#4459=ORIENTED_EDGE('',*,*,#3970,.T.);
#4460=ORIENTED_EDGE('',*,*,#4444,.T.);
#4461=ORIENTED_EDGE('',*,*,#4423,.F.);
#4462=EDGE_LOOP('',(#4459,#4460,#4461));
#4463=FACE_BOUND('',#4462,.F.);
#4464=CARTESIAN_POINT('',(99.999999999999986,-15.610000000000003,50.0));
#4465=DIRECTION('',(-0.223548855975687,-0.670839704394397,-
0.707106781186547));
#4466=DIRECTION('',(-0.223548855975687,-
0.670839704394397,0.707106781186548));
#4467=AXIS2_PLACEMENT_3D('',#4464,#4465,#4466);
#4468=SPHERICAL_SURFACE('',#4467,10.0);
#4469=ADVANCED_FACE('',(#4463),#4468,.F.);
#4470=SHAPE_ASPECT('featured shape: face 123',$,#38,.T.);
#4471=PROPERTY_DEFINITION('',$,#4470);
#4472=FACE_SHAPE_REPRESENTATION('',(#4469),#29);
#4473=PROPERTY_DEFINITION_REPRESENTATION(#4471,#4472);
#4474=ORIENTED_EDGE('',*,*,#4106,.T.);
#4475=ORIENTED_EDGE('',*,*,#4379,.T.);
#4476=ORIENTED_EDGE('',*,*,#4350,.F.);
#4477=EDGE_LOOP('',(#4474,#4475,#4476));
#4478=FACE_BOUND('',#4477,.F.);
#4479=CARTESIAN_POINT('',(60.0,-15.609999999999999,50.0));
#4480=DIRECTION('',(-0.707106781186548,-3.768222E-16,-0.707106781186547));
#4481=DIRECTION('',(-0.707106781186547,-3.768222E-16,0.707106781186547));
#4482=AXIS2_PLACEMENT_3D('',#4479,#4480,#4481);
#4483=SPHERICAL_SURFACE('',#4482,10.0);
#4484=ADVANCED_FACE('',(#4478),#4483,.F.);
#4485=SHAPE_ASPECT('featured shape: face 124',$,#38,.T.);
#4486=PROPERTY_DEFINITION('',$,#4485);
#4487=FACE_SHAPE_REPRESENTATION('',(#4484),#29);
#4488=PROPERTY_DEFINITION_REPRESENTATION(#4486,#4487);
#4489=CARTESIAN_POINT('',(50.0,-10.0,90.0));
#4490=DIRECTION('',(0.0,-1.0,0.0));
#4491=DIRECTION('',(0.0,0.0,-1.0));
#4492=AXIS2_PLACEMENT_3D('',#4489,#4490,#4491);
#4493=CIRCLE('',#4492,9.0);
#4494=EDGE_CURVE('',#2090,#2064,#4493,.T.);
#4495=ORIENTED_EDGE('',*,*,#4494,.F.);
#4496=ORIENTED_EDGE('',*,*,#2101,.F.);
```



```
#4497=ORIENTED_EDGE('',*,*,#712,.T.);
#4498=ORIENTED_EDGE('',*,*,#2073,.T.);
#4499=EDGE_LOOP('',(#4495,#4496,#4497,#4498));
#4500=FACE_BOUND('',#4499,.F.);
#4501=CARTESIAN_POINT('',(50.0,0.0,90.0));
#4502=DIRECTION('',(0.0,-1.0,0.0));
#4503=DIRECTION('',(1.0,0.0,0.0));
#4504=AXIS2_PLACEMENT_3D('',#4501,#4502,#4503);
#4505=CYLINDRICAL_SURFACE('',#4504,9.0);
#4506=ADVANCED_FACE('',(#4500),#4505,.F.);
#4507=SHAPE_ASPECT('featured shape: face 125',$,#38,.T.);
#4508=PROPERTY_DEFINITION('',$,#4507);
#4509=FACE_SHAPE_REPRESENTATION('',(#4506),#29);
#4510=PROPERTY_DEFINITION_REPRESENTATION(#4508,#4509);
#4511=CARTESIAN_POINT('',(47.999999999999979,-20.0,135.0));
#4512=VERTEX_POINT('',#4511);
#4513=DIRECTION('',(0.0,0.0,1.0));
#4514=VECTOR('',#4513,45.0);
#4515=LINE('',#4511,#4514);
#4516=EDGE_CURVE('',#4512,#80,#4515,.T.);
#4517=ORIENTED_EDGE('',*,*,#4516,.F.);
#4518=CARTESIAN_POINT('',(52.999999999999979,-20.0,130.0));
#4519=VERTEX_POINT('',#4518);
#4520=CARTESIAN_POINT('',(52.999999999999979,-20.0,135.0));
#4521=DIRECTION('',(0.0,1.0,0.0));
#4522=DIRECTION('',(0.0,0.0,-1.0));
#4523=AXIS2_PLACEMENT_3D('',#4520,#4521,#4522);
#4524=CIRCLE('',#4523,5.0);
#4525=EDGE_CURVE('',#4519,#4512,#4524,.T.);
#4526=ORIENTED_EDGE('',*,*,#4525,.F.);
#4527=CARTESIAN_POINT('',(112.999999999999943,-20.0,130.0));
#4528=VERTEX_POINT('',#4527);
#4529=DIRECTION('',(-1.0,0.0,0.0));
#4530=VECTOR('',#4529,59.999999999999964);
#4531=LINE('',#4527,#4530);
#4532=EDGE_CURVE('',#4528,#4519,#4531,.T.);
#4533=ORIENTED_EDGE('',*,*,#4532,.F.);
#4534=CARTESIAN_POINT('',(117.999999999999943,-20.0,135.0));
#4535=VERTEX_POINT('',#4534);
#4536=CARTESIAN_POINT('',(112.999999999999943,-20.0,135.0));
#4537=DIRECTION('',(0.0,1.0,0.0));
#4538=DIRECTION('',(1.0,0.0,0.0));
#4539=AXIS2_PLACEMENT_3D('',#4536,#4537,#4538);
#4540=CIRCLE('',#4539,5.0);
#4541=EDGE_CURVE('',#4535,#4528,#4540,.T.);
#4542=ORIENTED_EDGE('',*,*,#4541,.F.);
#4543=DIRECTION('',(0.0,0.0,-1.0));
#4544=VECTOR('',#4543,45.0);
#4545=LINE('',#81,#4544);
#4546=EDGE_CURVE('',#82,#4535,#4545,.T.);
#4547=ORIENTED_EDGE('',*,*,#4546,.F.);
#4548=ORIENTED_EDGE('',*,*,#86,.F.);
#4549=EDGE_LOOP('',(#4517,#4526,#4533,#4542,#4547,#4548));
#4550=FACE_BOUND('',#4549,.F.);
#4551=CARTESIAN_POINT('',(69.999999999999972,-20.0,145.0));
#4552=VERTEX_POINT('',#4551);
```



```
#4553=CARTESIAN_POINT('',(95.999999999999972,-20.0,145.0));
#4554=VERTEX_POINT('',#4553);
#4555=DIRECTION('',(1.0,0.0,0.0));
#4556=VECTOR('',#4555,26.0);
#4557=LINE('',#4551,#4556);
#4558=EDGE_CURVE('',#4552,#4554,#4557,.T.);
#4559=ORIENTED_EDGE('',*,*,#4558,.F.);
#4560=CARTESIAN_POINT('',(67.999999999999972,-20.0,147.0));
#4561=VERTEX_POINT('',#4560);
#4562=CARTESIAN_POINT('',(69.999999999999972,-20.0,147.0));
#4563=DIRECTION('',(0.0,-1.0,0.0));
#4564=DIRECTION('',(-1.0,0.0,0.0));
#4565=AXIS2_PLACEMENT_3D('',#4562,#4563,#4564);
#4566=CIRCLE('',#4565,2.0);
#4567=EDGE_CURVE('',#4561,#4552,#4566,.T.);
#4568=ORIENTED_EDGE('',*,*,#4567,.F.);
#4569=CARTESIAN_POINT('',(67.999999999999972,-20.0,163.0));
#4570=VERTEX_POINT('',#4569);
#4571=DIRECTION('',(0.0,0.0,-1.0));
#4572=VECTOR('',#4571,16.0);
#4573=LINE('',#4569,#4572);
#4574=EDGE_CURVE('',#4570,#4561,#4573,.T.);
#4575=ORIENTED_EDGE('',*,*,#4574,.F.);
#4576=CARTESIAN_POINT('',(69.999999999999972,-20.0,165.0));
#4577=VERTEX_POINT('',#4576);
#4578=CARTESIAN_POINT('',(69.999999999999972,-20.0,163.0));
#4579=DIRECTION('',(0.0,-1.0,0.0));
#4580=DIRECTION('',(0.0,0.0,1.0));
#4581=AXIS2_PLACEMENT_3D('',#4578,#4579,#4580);
#4582=CIRCLE('',#4581,2.0);
#4583=EDGE_CURVE('',#4577,#4570,#4582,.T.);
#4584=ORIENTED_EDGE('',*,*,#4583,.F.);
#4585=CARTESIAN_POINT('',(95.999999999999972,-20.0,165.0));
#4586=VERTEX_POINT('',#4585);
#4587=DIRECTION('',(-1.0,0.0,0.0));
#4588=VECTOR('',#4587,26.0);
#4589=LINE('',#4585,#4588);
#4590=EDGE_CURVE('',#4586,#4577,#4589,.T.);
#4591=ORIENTED_EDGE('',*,*,#4590,.F.);
#4592=CARTESIAN_POINT('',(97.999999999999972,-20.0,163.0));
#4593=VERTEX_POINT('',#4592);
#4594=CARTESIAN_POINT('',(95.999999999999972,-20.0,163.0));
#4595=DIRECTION('',(0.0,-1.0,0.0));
#4596=DIRECTION('',(1.0,0.0,0.0));
#4597=AXIS2_PLACEMENT_3D('',#4594,#4595,#4596);
#4598=CIRCLE('',#4597,2.0);
#4599=EDGE_CURVE('',#4593,#4586,#4598,.T.);
#4600=ORIENTED_EDGE('',*,*,#4599,.F.);
#4601=CARTESIAN_POINT('',(97.999999999999972,-20.0,147.0));
#4602=VERTEX_POINT('',#4601);
#4603=DIRECTION('',(0.0,0.0,1.0));
#4604=VECTOR('',#4603,16.0);
#4605=LINE('',#4601,#4604);
#4606=EDGE_CURVE('',#4602,#4593,#4605,.T.);
#4607=ORIENTED_EDGE('',*,*,#4606,.F.);
#4608=CARTESIAN_POINT('',(95.999999999999972,-20.0,147.0));
```



```
#4609=DIRECTION('',(0.0,-1.0,0.0));
#4610=DIRECTION('',(0.0,0.0,-1.0));
#4611=AXIS2_PLACEMENT_3D('',#4608,#4609,#4610);
#4612=CIRCLE('',#4611,2.0);
#4613=EDGE_CURVE('',#4554,#4602,#4612,.T.);
#4614=ORIENTED_EDGE('',*,*,#4613,.F.);
#4615=EDGE_LOOP('',(#4559,#4568,#4575,#4584,#4591,#4600,#4607,#4614));
#4616=FACE_BOUND('',#4615,.F.);
#4617=CARTESIAN_POINT('',(0.0,-20.0,180.0));
#4618=DIRECTION('',(0.0,1.0,0.0));
#4619=DIRECTION('',(1.0,0.0,0.0));
#4620=AXIS2_PLACEMENT_3D('',#4617,#4618,#4619);
#4621=PLANE('',#4620);
#4622=ADVANCED_FACE('',(#4550,#4616),#4621,.T.);
#4623=SHAPE_ASPECT('featured shape: face 126',$,#38,.T.);
#4624=PROPERTY_DEFINITION('',$,#4623);
#4625=FACE_SHAPE_REPRESENTATION('',(#4622),#29);
#4626=PROPERTY_DEFINITION_REPRESENTATION(#4624,#4625);
#4627=ORIENTED_EDGE('',*,*,#93,.T.);
#4628=ORIENTED_EDGE('',*,*,#4546,.T.);
#4629=DIRECTION('',(0.0,-1.0,0.0));
#4630=VECTOR('',#4629,20.0);
#4631=LINE('',#652,#4630);
#4632=EDGE_CURVE('',#653,#4535,#4631,.T.);
#4633=ORIENTED_EDGE('',*,*,#4632,.F.);
#4634=ORIENTED_EDGE('',*,*,#657,.F.);
#4635=EDGE_LOOP('',(#4627,#4628,#4633,#4634));
#4636=FACE_BOUND('',#4635,.F.);
#4637=CARTESIAN_POINT('',(117.999999999999943,0.0,180.0));
#4638=DIRECTION('',(1.0,0.0,0.0));
#4639=DIRECTION('',(0.0,0.0,-1.0));
#4640=AXIS2_PLACEMENT_3D('',#4637,#4638,#4639);
#4641=PLANE('',#4640);
#4642=ADVANCED_FACE('',(#4636),#4641,.F.);
#4643=SHAPE_ASPECT('featured shape: face 127',$,#38,.T.);
#4644=PROPERTY_DEFINITION('',$,#4643);
#4645=FACE_SHAPE_REPRESENTATION('',(#4642),#29);
#4646=PROPERTY_DEFINITION_REPRESENTATION(#4644,#4645);
#4647=ORIENTED_EDGE('',*,*,#666,.F.);
#4648=ORIENTED_EDGE('',*,*,#4632,.T.);
#4649=ORIENTED_EDGE('',*,*,#4541,.T.);
#4650=DIRECTION('',(0.0,-1.0,0.0));
#4651=VECTOR('',#4650,20.0);
#4652=LINE('',#659,#4651);
#4653=EDGE_CURVE('',#660,#4528,#4652,.T.);
#4654=ORIENTED_EDGE('',*,*,#4653,.F.);
#4655=EDGE_LOOP('',(#4647,#4648,#4649,#4654));
#4656=FACE_BOUND('',#4655,.F.);
#4657=CARTESIAN_POINT('',(112.999999999999943,0.0,135.0));
#4658=DIRECTION('',(0.0,1.0,0.0));
#4659=DIRECTION('',(1.0,0.0,0.0));
#4660=AXIS2_PLACEMENT_3D('',#4657,#4658,#4659);
#4661=CYLINDRICAL_SURFACE('',#4660,5.0);
#4662=ADVANCED_FACE('',(#4656),#4661,.F.);
#4663=SHAPE_ASPECT('featured shape: face 128',$,#38,.T.);
#4664=PROPERTY_DEFINITION('',$,#4663);
```



```
#4665=FACE_SHAPE_REPRESENTATION('',(#4662),#29);
#4666=PROPERTY_DEFINITION_REPRESENTATION(#4664,#4665);
#4667=ORIENTED_EDGE('',*,*,#673,.F.);
#4668=ORIENTED_EDGE('',*,*,#4653,.T.);
#4669=ORIENTED_EDGE('',*,*,#4532,.T.);
#4670=DIRECTION('',(0.0,-1.0,0.0));
#4671=VECTOR('',#4670,20.0);
#4672=LINE('',#668,#4671);
#4673=EDGE_CURVE('',#669,#4519,#4672,.T.);
#4674=ORIENTED_EDGE('',*,*,#4673,.F.);
#4675=EDGE_LOOP('',(#4667,#4668,#4669,#4674));
#4676=FACE_BOUND('',#4675,.F.);
#4677=CARTESIAN_POINT('',(112.999999999999943,0.0,130.0));
#4678=DIRECTION('',(0.0,0.0,-1.0));
#4679=DIRECTION('',(-1.0,0.0,0.0));
#4680=AXIS2_PLACEMENT_3D('',#4677,#4678,#4679);
#4681=PLANE('',#4680);
#4682=ADVANCED_FACE('',(#4676),#4681,.F.);
#4683=SHAPE_ASPECT('featured shape: face 129',$,#38,.T.);
#4684=PROPERTY_DEFINITION('',$,#4683);
#4685=FACE_SHAPE_REPRESENTATION('',(#4682),#29);
#4686=PROPERTY_DEFINITION_REPRESENTATION(#4684,#4685);
#4687=ORIENTED_EDGE('',*,*,#682,.F.);
#4688=ORIENTED_EDGE('',*,*,#4673,.T.);
#4689=ORIENTED_EDGE('',*,*,#4525,.T.);
#4690=DIRECTION('',(0.0,-1.0,0.0));
#4691=VECTOR('',#4690,20.0);
#4692=LINE('',#675,#4691);
#4693=EDGE_CURVE('',#676,#4512,#4692,.T.);
#4694=ORIENTED_EDGE('',*,*,#4693,.F.);
#4695=EDGE_LOOP('',(#4687,#4688,#4689,#4694));
#4696=FACE_BOUND('',#4695,.F.);
#4697=CARTESIAN_POINT('',(52.999999999999979,0.0,135.0));
#4698=DIRECTION('',(0.0,1.0,0.0));
#4699=DIRECTION('',(1.0,0.0,0.0));
#4700=AXIS2_PLACEMENT_3D('',#4697,#4698,#4699);
#4701=CYLINDRICAL_SURFACE('',#4700,5.0);
#4702=ADVANCED_FACE('',(#4696),#4701,.F.);
#4703=SHAPE_ASPECT('featured shape: face 130',$,#38,.T.);
#4704=PROPERTY_DEFINITION('',$,#4703);
#4705=FACE_SHAPE_REPRESENTATION('',(#4702),#29);
#4706=PROPERTY_DEFINITION_REPRESENTATION(#4704,#4705);
#4707=ORIENTED_EDGE('',*,*,#133,.F.);
#4708=ORIENTED_EDGE('',*,*,#687,.T.);
#4709=ORIENTED_EDGE('',*,*,#4693,.T.);
#4710=ORIENTED_EDGE('',*,*,#4516,.T.);
#4711=EDGE_LOOP('',(#4707,#4708,#4709,#4710));
#4712=FACE_BOUND('',#4711,.F.);
#4713=CARTESIAN_POINT('',(47.999999999999979,0.0,135.0));
#4714=DIRECTION('',(-1.0,0.0,0.0));
#4715=DIRECTION('',(0.0,0.0,1.0));
#4716=AXIS2_PLACEMENT_3D('',#4713,#4714,#4715);
#4717=PLANE('',#4716);
#4718=ADVANCED_FACE('',(#4712),#4717,.F.);
#4719=SHAPE_ASPECT('featured shape: face 131',$,#38,.T.);
#4720=PROPERTY_DEFINITION('',$,#4719);
```



```
#4721=FACE_SHAPE_REPRESENTATION('',(#4718),#29);
#4722=PROPERTY_DEFINITION_REPRESENTATION(#4720,#4721);
#4723=ORIENTED_EDGE('',*,*,#1121,.F.);
#4724=DIRECTION('',(0.0,-1.0,0.0));
#4725=VECTOR('',#4724,20.0);
#4726=LINE('',#1114,#4725);
#4727=EDGE_CURVE('',#1115,#4552,#4726,.T.);
#4728=ORIENTED_EDGE('',*,*,#4727,.T.);
#4729=ORIENTED_EDGE('',*,*,#4558,.T.);
#4730=DIRECTION('',(0.0,-1.0,0.0));
#4731=VECTOR('',#4730,20.0);
#4732=LINE('',#1116,#4731);
#4733=EDGE_CURVE('',#1117,#4554,#4732,.T.);
#4734=ORIENTED_EDGE('',*,*,#4733,.F.);
#4735=EDGE_LOOP('',(#4723,#4728,#4729,#4734));
#4736=FACE_BOUND('',#4735,.F.);
#4737=CARTESIAN_POINT('',(69.999999999999972,0.0,145.0));
#4738=DIRECTION('',(0.0,0.0,1.0));
#4739=DIRECTION('',(1.0,0.0,0.0));
#4740=AXIS2_PLACEMENT_3D('',#4737,#4738,#4739);
#4741=PLANE('',#4740);
#4742=ADVANCED_FACE('',(#4736),#4741,.F.);
#4743=SHAPE_ASPECT('featured shape: face 132',$,#38,.T.);
#4744=PROPERTY_DEFINITION('',$,#4743);
#4745=FACE_SHAPE_REPRESENTATION('',(#4742),#29);
#4746=PROPERTY_DEFINITION_REPRESENTATION(#4744,#4745);
#4747=ORIENTED_EDGE('',*,*,#1130,.T.);
#4748=ORIENTED_EDGE('',*,*,#4733,.T.);
#4749=ORIENTED_EDGE('',*,*,#4613,.T.);
#4750=DIRECTION('',(0.0,-1.0,0.0));
#4751=VECTOR('',#4750,20.0);
#4752=LINE('',#1123,#4751);
#4753=EDGE_CURVE('',#1124,#4602,#4752,.T.);
#4754=ORIENTED_EDGE('',*,*,#4753,.F.);
#4755=EDGE_LOOP('',(#4747,#4748,#4749,#4754));
#4756=FACE_BOUND('',#4755,.F.);
#4757=CARTESIAN_POINT('',(95.999999999999972,0.0,147.0));
#4758=DIRECTION('',(0.0,1.0,0.0));
#4759=DIRECTION('',(1.0,0.0,0.0));
#4760=AXIS2_PLACEMENT_3D('',#4757,#4758,#4759);
#4761=CYLINDRICAL_SURFACE('',#4760,2.0);
#4762=ADVANCED_FACE('',(#4756),#4761,.T.);
#4763=SHAPE_ASPECT('featured shape: face 133',$,#38,.T.);
#4764=PROPERTY_DEFINITION('',$,#4763);
#4765=FACE_SHAPE_REPRESENTATION('',(#4762),#29);
#4766=PROPERTY_DEFINITION_REPRESENTATION(#4764,#4765);
#4767=ORIENTED_EDGE('',*,*,#1137,.F.);
#4768=ORIENTED_EDGE('',*,*,#4753,.T.);
#4769=ORIENTED_EDGE('',*,*,#4606,.T.);
#4770=DIRECTION('',(0.0,-1.0,0.0));
#4771=VECTOR('',#4770,20.0);
#4772=LINE('',#1132,#4771);
#4773=EDGE_CURVE('',#1133,#4593,#4772,.T.);
#4774=ORIENTED_EDGE('',*,*,#4773,.F.);
#4775=EDGE_LOOP('',(#4767,#4768,#4769,#4774));
#4776=FACE_BOUND('',#4775,.F.);
```

```
#4777=CARTESIAN_POINT('',(97.999999999999972,0.0,147.0));
#4778=DIRECTION('',(-1.0,0.0,0.0));
#4779=DIRECTION('',(0.0,0.0,1.0));
#4780=AXIS2_PLACEMENT_3D('',#4777,#4778,#4779);
#4781=PLANE('',#4780);
#4782=ADVANCED_FACE('',(#4776),#4781,.F.);
#4783=SHAPE_ASPECT('featured shape: face 134',$,#38,.T.);
#4784=PROPERTY_DEFINITION('',$,#4783);
#4785=FACE_SHAPE_REPRESENTATION('',(#4782),#29);
#4786=PROPERTY_DEFINITION_REPRESENTATION(#4784,#4785);
#4787=ORIENTED_EDGE('',*,*,#1146,.T.);
#4788=ORIENTED_EDGE('',*,*,#4773,.T.);
#4789=ORIENTED_EDGE('',*,*,#4599,.T.);
#4790=DIRECTION('',(0.0,-1.0,0.0));
#4791=VECTOR('',#4790,20.0);
#4792=LINE('',#1139,#4791);
#4793=EDGE_CURVE('',#1140,#4586,#4792,.T.);
#4794=ORIENTED_EDGE('',*,*,#4793,.F.);
#4795=EDGE_LOOP('',(#4787,#4788,#4789,#4794));
#4796=FACE_BOUND('',#4795,.F.);
#4797=CARTESIAN_POINT('',(95.999999999999972,0.0,163.0));
#4798=DIRECTION('',(0.0,1.0,0.0));
#4799=DIRECTION('',(1.0,0.0,0.0));
#4800=AXIS2_PLACEMENT_3D('',#4797,#4798,#4799);
#4801=CYLINDRICAL_SURFACE('',#4800,2.0);
#4802=ADVANCED_FACE('',(#4796),#4801,.T.);
#4803=SHAPE_ASPECT('featured shape: face 135',$,#38,.T.);
#4804=PROPERTY_DEFINITION('',$,#4803);
#4805=FACE_SHAPE_REPRESENTATION('',(#4802),#29);
#4806=PROPERTY_DEFINITION_REPRESENTATION(#4804,#4805);
#4807=ORIENTED_EDGE('',*,*,#1153,.F.);
#4808=ORIENTED_EDGE('',*,*,#4793,.T.);
#4809=ORIENTED_EDGE('',*,*,#4590,.T.);
#4810=DIRECTION('',(0.0,-1.0,0.0));
#4811=VECTOR('',#4810,20.0);
#4812=LINE('',#1148,#4811);
#4813=EDGE_CURVE('',#1149,#4577,#4812,.T.);
#4814=ORIENTED_EDGE('',*,*,#4813,.F.);
#4815=EDGE_LOOP('',(#4807,#4808,#4809,#4814));
#4816=FACE_BOUND('',#4815,.F.);
#4817=CARTESIAN_POINT('',(95.999999999999972,0.0,165.0));
#4818=DIRECTION('',(0.0,0.0,-1.0));
#4819=DIRECTION('',(-1.0,0.0,0.0));
#4820=AXIS2_PLACEMENT_3D('',#4817,#4818,#4819);
#4821=PLANE('',#4820);
#4822=ADVANCED_FACE('',(#4816),#4821,.F.);
#4823=SHAPE_ASPECT('featured shape: face 136',$,#38,.T.);
#4824=PROPERTY_DEFINITION('',$,#4823);
#4825=FACE_SHAPE_REPRESENTATION('',(#4822),#29);
#4826=PROPERTY_DEFINITION_REPRESENTATION(#4824,#4825);
#4827=ORIENTED_EDGE('',*,*,#1162,.T.);
#4828=ORIENTED_EDGE('',*,*,#4813,.T.);
#4829=ORIENTED_EDGE('',*,*,#4583,.T.);
#4830=DIRECTION('',(0.0,-1.0,0.0));
#4831=VECTOR('',#4830,20.0);
#4832=LINE('',#1155,#4831);
```

```
#4833=EDGE_CURVE('',#1156,#4570,#4832,.T.);
#4834=ORIENTED_EDGE('',*,*,#4833,.F.);
#4835=EDGE_LOOP('',(#4827,#4828,#4829,#4834));
#4836=FACE_BOUND('',#4835,.F.);
#4837=CARTESIAN_POINT('',(69.999999999999972,0.0,163.0));
#4838=DIRECTION('',(0.0,1.0,0.0));
#4839=DIRECTION('',(1.0,0.0,0.0));
#4840=AXIS2_PLACEMENT_3D('',#4837,#4838,#4839);
#4841=CYLINDRICAL_SURFACE('',#4840,2.0);
#4842=ADVANCED_FACE('',(#4836),#4841,.T.);
#4843=SHAPE_ASPECT('featured shape: face 137',$,#38,.T.);
#4844=PROPERTY_DEFINITION('',$,#4843);
#4845=FACE_SHAPE_REPRESENTATION('',(#4842),#29);
#4846=PROPERTY_DEFINITION_REPRESENTATION(#4844,#4845);
#4847=ORIENTED_EDGE('',*,*,#1169,.F.);
#4848=ORIENTED_EDGE('',*,*,#4833,.T.);
#4849=ORIENTED_EDGE('',*,*,#4574,.T.);
#4850=DIRECTION('',(0.0,-1.0,0.0));
#4851=VECTOR('',#4850,20.0);
#4852=LINE('',#1164,#4851);
#4853=EDGE_CURVE('',#1165,#4561,#4852,.T.);
#4854=ORIENTED_EDGE('',*,*,#4853,.F.);
#4855=EDGE_LOOP('',(#4847,#4848,#4849,#4854));
#4856=FACE_BOUND('',#4855,.F.);
#4857=CARTESIAN_POINT('',(67.999999999999972,0.0,163.0));
#4858=DIRECTION('',(1.0,0.0,0.0));
#4859=DIRECTION('',(0.0,0.0,-1.0));
#4860=AXIS2_PLACEMENT_3D('',#4857,#4858,#4859);
#4861=PLANE('',#4860);
#4862=ADVANCED_FACE('',(#4856),#4861,.F.);
#4863=SHAPE_ASPECT('featured shape: face 138',$,#38,.T.);
#4864=PROPERTY_DEFINITION('',$,#4863);
#4865=FACE_SHAPE_REPRESENTATION('',(#4862),#29);
#4866=PROPERTY_DEFINITION_REPRESENTATION(#4864,#4865);
#4867=ORIENTED_EDGE('',*,*,#1176,.T.);
#4868=ORIENTED_EDGE('',*,*,#4853,.T.);
#4869=ORIENTED_EDGE('',*,*,#4567,.T.);
#4870=ORIENTED_EDGE('',*,*,#4727,.F.);
#4871=EDGE_LOOP('',(#4867,#4868,#4869,#4870));
#4872=FACE_BOUND('',#4871,.F.);
#4873=CARTESIAN_POINT('',(69.999999999999972,0.0,147.0));
#4874=DIRECTION('',(0.0,1.0,0.0));
#4875=DIRECTION('',(1.0,0.0,0.0));
#4876=AXIS2_PLACEMENT_3D('',#4873,#4874,#4875);
#4877=CYLINDRICAL_SURFACE('',#4876,2.0);
#4878=ADVANCED_FACE('',(#4872),#4877,.T.);
#4879=SHAPE_ASPECT('featured shape: face 139',$,#38,.T.);
#4880=PROPERTY_DEFINITION('',$,#4879);
#4881=FACE_SHAPE_REPRESENTATION('',(#4878),#29);
#4882=PROPERTY_DEFINITION_REPRESENTATION(#4880,#4881);
#4883=ORIENTED_EDGE('',*,*,#250,.F.);
#4884=DIRECTION('',(0.0,1.0,0.0));
#4885=VECTOR('',#4884,10.0);
#4886=LINE('',#243,#4885);
#4887=EDGE_CURVE('',#244,#1482,#4886,.T.);
#4888=ORIENTED_EDGE('',*,*,#4887,.T.);
```

```
#4889=ORIENTED_EDGE('',*,*,#1486,.T.);
#4890=DIRECTION('',(0.0,1.0,0.0));
#4891=VECTOR('',#4890,10.0);
#4892=LINE('',#245,#4891);
#4893=EDGE_CURVE('',#246,#1473,#4892,.T.);
#4894=ORIENTED_EDGE('',*,*,#4893,.F.);
#4895=EDGE_LOOP('',(#4883,#4888,#4889,#4894));
#4896=FACE_BOUND('',#4895,.F.);
#4897=CARTESIAN_POINT('',(2.0,-50.0,50.000000000000007));
#4898=DIRECTION('',(0.0,0.0,-1.0));
#4899=DIRECTION('',(1.0,0.0,0.0));
#4900=AXIS2_PLACEMENT_3D('',#4897,#4898,#4899);
#4901=PLANE('',#4900);
#4902=ADVANCED_FACE('',(#4896),#4901,.F.);
#4903=SHAPE_ASPECT('featured shape: face 140',$,#38,.T.);
#4904=PROPERTY_DEFINITION('',$,#4903);
#4905=FACE_SHAPE_REPRESENTATION('',(#4902),#29);
#4906=PROPERTY_DEFINITION_REPRESENTATION(#4904,#4905);
#4907=ORIENTED_EDGE('',*,*,#259,.T.);
#4908=ORIENTED_EDGE('',*,*,#4893,.T.);
#4909=ORIENTED_EDGE('',*,*,#1479,.T.);
#4910=DIRECTION('',(0.0,1.0,0.0));
#4911=VECTOR('',#4910,10.0);
#4912=LINE('',#252,#4911);
#4913=EDGE_CURVE('',#253,#1466,#4912,.T.);
#4914=ORIENTED_EDGE('',*,*,#4913,.F.);
#4915=EDGE_LOOP('',(#4907,#4908,#4909,#4914));
#4916=FACE_BOUND('',#4915,.F.);
#4917=CARTESIAN_POINT('',(45.0,-50.0,45.000000000000007));
#4918=DIRECTION('',(0.0,-1.0,0.0));
#4919=DIRECTION('',(1.0,0.0,0.0));
#4920=AXIS2_PLACEMENT_3D('',#4917,#4918,#4919);
#4921=CYLINDRICAL_SURFACE('',#4920,5.0);
#4922=ADVANCED_FACE('',(#4916),#4921,.T.);
#4923=SHAPE_ASPECT('featured shape: face 141',$,#38,.T.);
#4924=PROPERTY_DEFINITION('',$,#4923);
#4925=FACE_SHAPE_REPRESENTATION('',(#4922),#29);
#4926=PROPERTY_DEFINITION_REPRESENTATION(#4924,#4925);
#4927=ORIENTED_EDGE('',*,*,#266,.T.);
#4928=ORIENTED_EDGE('',*,*,#4913,.T.);
#4929=ORIENTED_EDGE('',*,*,#1470,.T.);
#4930=DIRECTION('',(0.0,1.0,0.0));
#4931=VECTOR('',#4930,10.0);
#4932=LINE('',#261,#4931);
#4933=EDGE_CURVE('',#262,#1457,#4932,.T.);
#4934=ORIENTED_EDGE('',*,*,#4933,.F.);
#4935=EDGE_LOOP('',(#4927,#4928,#4929,#4934));
#4936=FACE_BOUND('',#4935,.F.);
#4937=CARTESIAN_POINT('',(50.0,-50.0,45.000000000000007));
#4938=DIRECTION('',(-1.0,0.0,0.0));
#4939=DIRECTION('',(0.0,0.0,-1.0));
#4940=AXIS2_PLACEMENT_3D('',#4937,#4938,#4939);
#4941=PLANE('',#4940);
#4942=ADVANCED_FACE('',(#4936),#4941,.F.);
#4943=SHAPE_ASPECT('featured shape: face 142',$,#38,.T.);
#4944=PROPERTY_DEFINITION('',$,#4943);
```



```
#4945=FACE_SHAPE_REPRESENTATION('',(#4942),#29);
#4946=PROPERTY_DEFINITION_REPRESENTATION(#4944,#4945);
#4947=ORIENTED_EDGE('',*,*,#346,.F.);
#4948=DIRECTION('',(0.0,1.0,0.0));
#4949=VECTOR('',#4948,10.0);
#4950=LINE('',#339,#4949);
#4951=EDGE_CURVE('',#340,#1548,#4950,.T.);
#4952=ORIENTED_EDGE('',*,*,#4951,.T.);
#4953=ORIENTED_EDGE('',*,*,#1552,.T.);
#4954=DIRECTION('',(0.0,1.0,0.0));
#4955=VECTOR('',#4954,10.0);
#4956=LINE('',#341,#4955);
#4957=EDGE_CURVE('',#342,#1539,#4956,.T.);
#4958=ORIENTED_EDGE('',*,*,#4957,.F.);
#4959=EDGE_LOOP('',(#4947,#4952,#4953,#4958));
#4960=FACE_BOUND('',#4959,.F.);
#4961=CARTESIAN_POINT('',(180.000000000000028,-50.0,2.0));
#4962=DIRECTION('',(1.0,0.0,0.0));
#4963=DIRECTION('',(0.0,0.0,1.0));
#4964=AXIS2_PLACEMENT_3D('',#4961,#4962,#4963);
#4965=PLANE('',#4964);
#4966=ADVANCED_FACE('',(#4960),#4965,.F.);
#4967=SHAPE_ASPECT('featured shape: face 143',$,#38,.T.);
#4968=PROPERTY_DEFINITION('',$,#4967);
#4969=FACE_SHAPE_REPRESENTATION('',(#4966),#29);
#4970=PROPERTY_DEFINITION_REPRESENTATION(#4968,#4969);
#4971=ORIENTED_EDGE('',*,*,#355,.T.);
#4972=ORIENTED_EDGE('',*,*,#4957,.T.);
#4973=ORIENTED_EDGE('',*,*,#1545,.T.);
#4974=DIRECTION('',(0.0,1.0,0.0));
#4975=VECTOR('',#4974,10.0);
#4976=LINE('',#348,#4975);
#4977=EDGE_CURVE('',#349,#1532,#4976,.T.);
#4978=ORIENTED_EDGE('',*,*,#4977,.F.);
#4979=EDGE_LOOP('',(#4971,#4972,#4973,#4978));
#4980=FACE_BOUND('',#4979,.F.);
#4981=CARTESIAN_POINT('',(185.000000000000028,-50.0,45.0));
#4982=DIRECTION('',(0.0,-1.0,0.0));
#4983=DIRECTION('',(1.0,0.0,0.0));
#4984=AXIS2_PLACEMENT_3D('',#4981,#4982,#4983);
#4985=CYLINDRICAL_SURFACE('',#4984,5.0);
#4986=ADVANCED_FACE('',(#4980),#4985,.T.);
#4987=SHAPE_ASPECT('featured shape: face 144',$,#38,.T.);
#4988=PROPERTY_DEFINITION('',$,#4987);
#4989=FACE_SHAPE_REPRESENTATION('',(#4986),#29);
#4990=PROPERTY_DEFINITION_REPRESENTATION(#4988,#4989);
#4991=ORIENTED_EDGE('',*,*,#362,.T.);
#4992=ORIENTED_EDGE('',*,*,#4977,.T.);
#4993=ORIENTED_EDGE('',*,*,#1536,.T.);
#4994=DIRECTION('',(0.0,1.0,0.0));
#4995=VECTOR('',#4994,10.0);
#4996=LINE('',#357,#4995);
#4997=EDGE_CURVE('',#358,#1523,#4996,.T.);
#4998=ORIENTED_EDGE('',*,*,#4997,.F.);
#4999=EDGE_LOOP('',(#4991,#4992,#4993,#4998));
#5000=FACE_BOUND('',#4999,.F.);
```

```
#5001=CARTESIAN_POINT('',(185.000000000000028,-50.0,50.0));
#5002=DIRECTION('',(0.0,0.0,-1.0));
#5003=DIRECTION('',(1.0,0.0,0.0));
#5004=AXIS2_PLACEMENT_3D('',#5001,#5002,#5003);
#5005=PLANE('',#5004);
#5006=ADVANCED_FACE('',(#5000),#5005,.F.);
#5007=SHAPE_ASPECT('featured shape: face 145',$,#38,.T.);
#5008=PROPERTY_DEFINITION('',$,#5007);
#5009=FACE_SHAPE_REPRESENTATION('',(#5006),#29);
#5010=PROPERTY_DEFINITION_REPRESENTATION(#5008,#5009);
#5011=ORIENTED_EDGE('',*,*,#437,.F.);
#5012=DIRECTION('',(0.0,1.0,0.0));
#5013=VECTOR('',#5012,10.0);
#5014=LINE('',#430,#5013);
#5015=EDGE_CURVE('',#431,#1614,#5014,.T.);
#5016=ORIENTED_EDGE('',*,*,#5015,.T.);
#5017=ORIENTED_EDGE('',*,*,#1618,.T.);
#5018=DIRECTION('',(0.0,1.0,0.0));
#5019=VECTOR('',#5018,10.0);
#5020=LINE('',#432,#5019);
#5021=EDGE_CURVE('',#433,#1605,#5020,.T.);
#5022=ORIENTED_EDGE('',*,*,#5021,.F.);
#5023=EDGE_LOOP('',(#5011,#5016,#5017,#5022));
#5024=FACE_BOUND('',#5023,.F.);
#5025=CARTESIAN_POINT('',(228.0,-50.0,130.000000000000028));
#5026=DIRECTION('',(0.0,0.0,1.0));
#5027=DIRECTION('',(-1.0,0.0,0.0));
#5028=AXIS2_PLACEMENT_3D('',#5025,#5026,#5027);
#5029=PLANE('',#5028);
#5030=ADVANCED_FACE('',(#5024),#5029,.F.);
#5031=SHAPE_ASPECT('featured shape: face 146',$,#38,.T.);
#5032=PROPERTY_DEFINITION('',$,#5031);
#5033=FACE_SHAPE_REPRESENTATION('',(#5030),#29);
#5034=PROPERTY_DEFINITION_REPRESENTATION(#5032,#5033);
#5035=ORIENTED_EDGE('',*,*,#446,.T.);
#5036=ORIENTED_EDGE('',*,*,#5021,.T.);
#5037=ORIENTED_EDGE('',*,*,#1611,.T.);
#5038=DIRECTION('',(0.0,1.0,0.0));
#5039=VECTOR('',#5038,10.0);
#5040=LINE('',#439,#5039);
#5041=EDGE_CURVE('',#440,#1598,#5040,.T.);
#5042=ORIENTED_EDGE('',*,*,#5041,.F.);
#5043=EDGE_LOOP('',(#5035,#5036,#5037,#5042));
#5044=FACE_BOUND('',#5043,.F.);
#5045=CARTESIAN_POINT('',(184.999999999999972,-50.0,135.000000000000028));
#5046=DIRECTION('',(0.0,-1.0,0.0));
#5047=DIRECTION('',(1.0,0.0,0.0));
#5048=AXIS2_PLACEMENT_3D('',#5045,#5046,#5047);
#5049=CYLINDRICAL_SURFACE('',#5048,5.0);
#5050=ADVANCED_FACE('',(#5044),#5049,.T.);
#5051=SHAPE_ASPECT('featured shape: face 147',$,#38,.T.);
#5052=PROPERTY_DEFINITION('',$,#5051);
#5053=FACE_SHAPE_REPRESENTATION('',(#5050),#29);
#5054=PROPERTY_DEFINITION_REPRESENTATION(#5052,#5053);
#5055=ORIENTED_EDGE('',*,*,#453,.T.);
#5056=ORIENTED_EDGE('',*,*,#5041,.T.);
```

```
#5057=ORIENTED_EDGE('',*,*,#1602,.T.);
#5058=DIRECTION('',(0.0,1.0,0.0));
#5059=VECTOR('',#5058,10.0);
#5060=LINE('',#448,#5059);
#5061=EDGE_CURVE('',#449,#1589,#5060,.T.);
#5062=ORIENTED_EDGE('',*,*,#5061,.F.);
#5063=EDGE_LOOP('',(#5055,#5056,#5057,#5062));
#5064=FACE_BOUND('',#5063,.F.);
#5065=CARTESIAN_POINT('',(179.999999999999972,-50.0,135.000000000000028));
#5066=DIRECTION('',(1.0,0.0,0.0));
#5067=DIRECTION('',(0.0,0.0,1.0));
#5068=AXIS2_PLACEMENT_3D('',#5065,#5066,#5067);
#5069=PLANE('',#5068);
#5070=ADVANCED_FACE('',(#5064),#5069,.F.);
#5071=SHAPE_ASPECT('featured shape: face 148',$,#38,.T.);
#5072=PROPERTY_DEFINITION('',$,#5071);
#5073=FACE_SHAPE_REPRESENTATION('',(#5070),#29);
#5074=PROPERTY_DEFINITION_REPRESENTATION(#5072,#5073);
#5075=ORIENTED_EDGE('',*,*,#528,.F.);
#5076=DIRECTION('',(0.0,1.0,0.0));
#5077=VECTOR('',#5076,10.0);
#5078=LINE('',#521,#5077);
#5079=EDGE_CURVE('',#522,#1760,#5078,.T.);
#5080=ORIENTED_EDGE('',*,*,#5079,.T.);
#5081=ORIENTED_EDGE('',*,*,#1764,.T.);
#5082=DIRECTION('',(0.0,1.0,0.0));
#5083=VECTOR('',#5082,10.0);
#5084=LINE('',#523,#5083);
#5085=EDGE_CURVE('',#524,#1751,#5084,.T.);
#5086=ORIENTED_EDGE('',*,*,#5085,.F.);
#5087=EDGE_LOOP('',(#5075,#5080,#5081,#5086));
#5088=FACE_BOUND('',#5087,.F.);
#5089=CARTESIAN_POINT('',(49.999999999999972,-50.0,178.0));
#5090=DIRECTION('',(-1.0,0.0,0.0));
#5091=DIRECTION('',(0.0,0.0,-1.0));
#5092=AXIS2_PLACEMENT_3D('',#5089,#5090,#5091);
#5093=PLANE('',#5092);
#5094=ADVANCED_FACE('',(#5088),#5093,.F.);
#5095=SHAPE_ASPECT('featured shape: face 149',$,#38,.T.);
#5096=PROPERTY_DEFINITION('',$,#5095);
#5097=FACE_SHAPE_REPRESENTATION('',(#5094),#29);
#5098=PROPERTY_DEFINITION_REPRESENTATION(#5096,#5097);
#5099=ORIENTED_EDGE('',*,*,#537,.T.);
#5100=ORIENTED_EDGE('',*,*,#5085,.T.);
#5101=ORIENTED_EDGE('',*,*,#1757,.T.);
#5102=DIRECTION('',(0.0,1.0,0.0));
#5103=VECTOR('',#5102,10.0);
#5104=LINE('',#530,#5103);
#5105=EDGE_CURVE('',#531,#1744,#5104,.T.);
#5106=ORIENTED_EDGE('',*,*,#5105,.F.);
#5107=EDGE_LOOP('',(#5099,#5100,#5101,#5106));
#5108=FACE_BOUND('',#5107,.F.);
#5109=CARTESIAN_POINT('',(44.999999999999972,-50.0,134.999999999999972));
#5110=DIRECTION('',(0.0,-1.0,0.0));
#5111=DIRECTION('',(1.0,0.0,0.0));
#5112=AXIS2_PLACEMENT_3D('',#5109,#5110,#5111);
```

```
#5113=CYLINDRICAL_SURFACE('',#5112,5.0);
#5114=ADVANCED_FACE('',(#5108),#5113,.T.);
#5115=SHAPE_ASPECT('featured shape: face 150',$,#38,.T.);
#5116=PROPERTY_DEFINITION('',$,#5115);
#5117=FACE_SHAPE_REPRESENTATION('',(#5114),#29);
#5118=PROPERTY_DEFINITION_REPRESENTATION(#5116,#5117);
#5119=ORIENTED_EDGE('',*,*,#544,.T.);
#5120=ORIENTED_EDGE('',*,*,#5105,.T.);
#5121=ORIENTED_EDGE('',*,*,#1748,.T.);
#5122=DIRECTION('',(0.0,1.0,0.0));
#5123=VECTOR('',#5122,10.0);
#5124=LINE('',#539,#5123);
#5125=EDGE_CURVE('',#540,#1735,#5124,.T.);
#5126=ORIENTED_EDGE('',*,*,#5125,.F.);
#5127=EDGE_LOOP('',(#5119,#5120,#5121,#5126));
#5128=FACE_BOUND('',#5127,.F.);
#5129=CARTESIAN_POINT('',(44.999999999999972,-50.0,129.999999999999972));
#5130=DIRECTION('',(0.0,0.0,1.0));
#5131=DIRECTION('',(-1.0,0.0,0.0));
#5132=AXIS2_PLACEMENT_3D('',#5129,#5130,#5131);
#5133=PLANE('',#5132);
#5134=ADVANCED_FACE('',(#5128),#5133,.F.);
#5135=SHAPE_ASPECT('featured shape: face 151',$,#38,.T.);
#5136=PROPERTY_DEFINITION('',$,#5135);
#5137=FACE_SHAPE_REPRESENTATION('',(#5134),#29);
#5138=PROPERTY_DEFINITION_REPRESENTATION(#5136,#5137);
#5139=ORIENTED_EDGE('',*,*,#2271,.T.);
#5140=ORIENTED_EDGE('',*,*,#1561,.T.);
#5141=ORIENTED_EDGE('',*,*,#4951,.F.);
#5142=ORIENTED_EDGE('',*,*,#401,.F.);
#5143=EDGE_LOOP('',(#5139,#5140,#5141,#5142));
#5144=FACE_BOUND('',#5143,.F.);
#5145=CARTESIAN_POINT('',(185.000000000000028,-50.0,7.0));
#5146=DIRECTION('',(0.0,1.0,0.0));
#5147=DIRECTION('',(0.0,0.0,-1.0));
#5148=AXIS2_PLACEMENT_3D('',#5145,#5146,#5147);
#5149=CYLINDRICAL_SURFACE('',#5148,5.0);
#5150=ADVANCED_FACE('',(#5144),#5149,.T.);
#5151=SHAPE_ASPECT('featured shape: face 152',$,#38,.T.);
#5152=PROPERTY_DEFINITION('',$,#5151);
#5153=FACE_SHAPE_REPRESENTATION('',(#5150),#29);
#5154=PROPERTY_DEFINITION_REPRESENTATION(#5152,#5153);
#5155=ORIENTED_EDGE('',*,*,#2204,.F.);
#5156=ORIENTED_EDGE('',*,*,#462,.T.);
#5157=ORIENTED_EDGE('',*,*,#5061,.T.);
#5158=ORIENTED_EDGE('',*,*,#1595,.F.);
#5159=EDGE_LOOP('',(#5155,#5156,#5157,#5158));
#5160=FACE_BOUND('',#5159,.F.);
#5161=CARTESIAN_POINT('',(184.999999999999972,-50.0,173.0));
#5162=DIRECTION('',(0.0,-1.0,0.0));
#5163=DIRECTION('',(0.0,0.0,1.0));
#5164=AXIS2_PLACEMENT_3D('',#5161,#5162,#5163);
#5165=CYLINDRICAL_SURFACE('',#5164,5.0);
#5166=ADVANCED_FACE('',(#5160),#5165,.T.);
#5167=SHAPE_ASPECT('featured shape: face 153',$,#38,.T.);
#5168=PROPERTY_DEFINITION('',$,#5167);
```

```
#5169=FACE_SHAPE_REPRESENTATION('',(#5166),#29);
#5170=PROPERTY_DEFINITION_REPRESENTATION(#5168,#5169);
#5171=ORIENTED_EDGE('',*,*,#2161,.T.);
#5172=ORIENTED_EDGE('',*,*,#1495,.T.);
#5173=ORIENTED_EDGE('',*,*,#4887,.F.);
#5174=ORIENTED_EDGE('',*,*,#305,.F.);
#5175=EDGE_LOOP('',(#5171,#5172,#5173,#5174));
#5176=FACE_BOUND('',#5175,.F.);
#5177=CARTESIAN_POINT('',(7.0,-50.0,45.000000000000007));
#5178=DIRECTION('',(0.0,1.0,0.0));
#5179=DIRECTION('',(-1.0,0.0,0.0));
#5180=AXIS2_PLACEMENT_3D('',#5177,#5178,#5179);
#5181=CYLINDRICAL_SURFACE('',#5180,5.0);
#5182=ADVANCED_FACE('',(#5176),#5181,.T.);
#5183=SHAPE_ASPECT('featured shape: face 154',$,#38,.T.);
#5184=PROPERTY_DEFINITION('',$,#5183);
#5185=FACE_SHAPE_REPRESENTATION('',(#5182),#29);
#5186=PROPERTY_DEFINITION_REPRESENTATION(#5184,#5185);
#5187=ORIENTED_EDGE('',*,*,#2118,.F.);
#5188=ORIENTED_EDGE('',*,*,#371,.T.);
#5189=ORIENTED_EDGE('',*,*,#4997,.T.);
#5190=ORIENTED_EDGE('',*,*,#1529,.F.);
#5191=EDGE_LOOP('',(#5187,#5188,#5189,#5190));
#5192=FACE_BOUND('',#5191,.F.);
#5193=CARTESIAN_POINT('',(223.0,-50.0,45.0));
#5194=DIRECTION('',(0.0,-1.0,0.0));
#5195=DIRECTION('',(1.0,0.0,0.0));
#5196=AXIS2_PLACEMENT_3D('',#5193,#5194,#5195);
#5197=CYLINDRICAL_SURFACE('',#5196,5.0);
#5198=ADVANCED_FACE('',(#5192),#5197,.T.);
#5199=SHAPE_ASPECT('featured shape: face 155',$,#38,.T.);
#5200=PROPERTY_DEFINITION('',$,#5199);
#5201=FACE_SHAPE_REPRESENTATION('',(#5198),#29);
#5202=PROPERTY_DEFINITION_REPRESENTATION(#5200,#5201);
#5203=ORIENTED_EDGE('',*,*,#2142,.T.);
#5204=ORIENTED_EDGE('',*,*,#1627,.T.);
#5205=ORIENTED_EDGE('',*,*,#5015,.F.);
#5206=ORIENTED_EDGE('',*,*,#492,.F.);
#5207=EDGE_LOOP('',(#5203,#5204,#5205,#5206));
#5208=FACE_BOUND('',#5207,.F.);
#5209=CARTESIAN_POINT('',(223.0,-50.0,135.000000000000028));
#5210=DIRECTION('',(0.0,1.0,0.0));
#5211=DIRECTION('',(1.0,0.0,0.0));
#5212=AXIS2_PLACEMENT_3D('',#5209,#5210,#5211);
#5213=CYLINDRICAL_SURFACE('',#5212,5.0);
#5214=ADVANCED_FACE('',(#5208),#5213,.T.);
#5215=SHAPE_ASPECT('featured shape: face 156',$,#38,.T.);
#5216=PROPERTY_DEFINITION('',$,#5215);
#5217=FACE_SHAPE_REPRESENTATION('',(#5214),#29);
#5218=PROPERTY_DEFINITION_REPRESENTATION(#5216,#5217);
#5219=ORIENTED_EDGE('',*,*,#2247,.F.);
#5220=ORIENTED_EDGE('',*,*,#275,.T.);
#5221=ORIENTED_EDGE('',*,*,#4933,.T.);
#5222=ORIENTED_EDGE('',*,*,#1463,.F.);
#5223=EDGE_LOOP('',(#5219,#5220,#5221,#5222));
#5224=FACE_BOUND('',#5223,.F.);
```

```
#5225=CARTESIAN_POINT('',(45.0,-50.0,7.0));
#5226=DIRECTION('',(0.0,-1.0,0.0));
#5227=DIRECTION('',(0.0,0.0,-1.0));
#5228=AXIS2_PLACEMENT_3D('',#5225,#5226,#5227);
#5229=CYLINDRICAL_SURFACE('',#5228,5.0);
#5230=ADVANCED_FACE('',(#5224),#5229,.T.);
#5231=SHAPE_ASPECT('featured shape: face 157',$,#38,.T.);
#5232=PROPERTY_DEFINITION('',$,#5231);
#5233=FACE_SHAPE_REPRESENTATION('',(#5230),#29);
#5234=PROPERTY_DEFINITION_REPRESENTATION(#5232,#5233);
#5235=ORIENTED_EDGE('',*,*,#2228,.T.);
#5236=ORIENTED_EDGE('',*,*,#1773,.T.);
#5237=ORIENTED_EDGE('',*,*,#5079,.F.);
#5238=ORIENTED_EDGE('',*,*,#583,.F.);
#5239=EDGE_LOOP('',(#5235,#5236,#5237,#5238));
#5240=FACE_BOUND('',#5239,.F.);
#5241=CARTESIAN_POINT('',(44.999999999999972,-50.0,173.0));
#5242=DIRECTION('',(0.0,1.0,0.0));
#5243=DIRECTION('',(0.0,0.0,1.0));
#5244=AXIS2_PLACEMENT_3D('',#5241,#5242,#5243);
#5245=CYLINDRICAL_SURFACE('',#5244,5.0);
#5246=ADVANCED_FACE('',(#5240),#5245,.T.);
#5247=SHAPE_ASPECT('featured shape: face 158',$,#38,.T.);
#5248=PROPERTY_DEFINITION('',$,#5247);
#5249=FACE_SHAPE_REPRESENTATION('',(#5246),#29);
#5250=PROPERTY_DEFINITION_REPRESENTATION(#5248,#5249);
#5251=ORIENTED_EDGE('',*,*,#2185,.F.);
#5252=ORIENTED_EDGE('',*,*,#553,.T.);
#5253=ORIENTED_EDGE('',*,*,#5125,.T.);
#5254=ORIENTED_EDGE('',*,*,#1741,.F.);
#5255=EDGE_LOOP('',(#5251,#5252,#5253,#5254));
#5256=FACE_BOUND('',#5255,.F.);
#5257=CARTESIAN_POINT('',(7.0,-50.0,134.999999999999972));
#5258=DIRECTION('',(0.0,-1.0,0.0));
#5259=DIRECTION('',(-1.0,0.0,0.0));
#5260=AXIS2_PLACEMENT_3D('',#5257,#5258,#5259);
#5261=CYLINDRICAL_SURFACE('',#5260,5.0);
#5262=ADVANCED_FACE('',(#5256),#5261,.T.);
#5263=SHAPE_ASPECT('featured shape: face 159',$,#38,.T.);
#5264=PROPERTY_DEFINITION('',$,#5263);
#5265=FACE_SHAPE_REPRESENTATION('',(#5262),#29);
#5266=PROPERTY_DEFINITION_REPRESENTATION(#5264,#5265);
#5267=ORIENTED_EDGE('',*,*,#217,.T.);
#5268=ORIENTED_EDGE('',*,*,#2094,.F.);
#5269=ORIENTED_EDGE('',*,*,#4494,.T.);
#5270=ORIENTED_EDGE('',*,*,#2068,.F.);
#5271=ORIENTED_EDGE('',*,*,#161,.T.);
#5272=CARTESIAN_POINT('',(50.099999999999994,-10.0,106.0));
#5273=VERTEX_POINT('',#5272);
#5274=DIRECTION('',(1.0,0.0,0.0));
#5275=VECTOR('',#5274,50.099999999999994);
#5276=LINE('',#149,#5275);
#5277=EDGE_CURVE('',#150,#5273,#5276,.T.);
#5278=ORIENTED_EDGE('',*,*,#5277,.T.);
#5279=CARTESIAN_POINT('',(66.0,-10.0,90.099999999999994));
#5280=VERTEX_POINT('',#5279);
```



```
#5281=CARTESIAN_POINT('',(50.099999999999994,-10.0,90.099999999999994));
#5282=DIRECTION('',(0.0,1.0,0.0));
#5283=DIRECTION('',(0.0,0.0,1.0));
#5284=AXIS2_PLACEMENT_3D('',#5281,#5282,#5283);
#5285=CIRCLE('',#5284,15.900000000000000);
#5286=EDGE_CURVE('',#5273,#5280,#5285,.T.);
#5287=ORIENTED_EDGE('',*,*,#5286,.T.);
#5288=CARTESIAN_POINT('',(66.0,-10.0,89.900000000000006));
#5289=VERTEX_POINT('',#5288);
#5290=DIRECTION('',(0.0,0.0,-1.0));
#5291=VECTOR('',#5290,0.199999999999989);
#5292=LINE('',#5279,#5291);
#5293=EDGE_CURVE('',#5280,#5289,#5292,.T.);
#5294=ORIENTED_EDGE('',*,*,#5293,.T.);
#5295=CARTESIAN_POINT('',(50.099999999999994,-10.0,74.0));
#5296=VERTEX_POINT('',#5295);
#5297=CARTESIAN_POINT('',(50.099999999999994,-10.0,89.900000000000006));
#5298=DIRECTION('',(0.0,1.0,0.0));
#5299=DIRECTION('',(1.0,0.0,0.0));
#5300=AXIS2_PLACEMENT_3D('',#5297,#5298,#5299);
#5301=CIRCLE('',#5300,15.900000000000000);
#5302=EDGE_CURVE('',#5289,#5296,#5301,.T.);
#5303=ORIENTED_EDGE('',*,*,#5302,.T.);
#5304=DIRECTION('',(-1.0,0.0,0.0));
#5305=VECTOR('',#5304,50.099999999999994);
#5306=LINE('',#5295,#5305);
#5307=EDGE_CURVE('',#5296,#213,#5306,.T.);
#5308=ORIENTED_EDGE('',*,*,#5307,.T.);
#5309=EDGE_LOOP('',(#5267,#5268,#5269,#5270,#5271,#5278,#5287,#5294,#5303,#53
08));
#5310=FACE_BOUND('',#5309,.F.);
#5311=CARTESIAN_POINT('',(260.0,-10.0,0.0));
#5312=DIRECTION('',(0.0,1.0,0.0));
#5313=DIRECTION('',(0.0,0.0,1.0));
#5314=AXIS2_PLACEMENT_3D('',#5311,#5312,#5313);
#5315=PLANE('',#5314);
#5316=ADVANCED_FACE('',(#5310),#5315,.F.);
#5317=SHAPE_ASPECT('featured shape: face 160',$,#38,.T.);
#5318=PROPERTY_DEFINITION('',$,#5317);
#5319=FACE_SHAPE_REPRESENTATION('',(#5316),#29);
#5320=PROPERTY_DEFINITION_REPRESENTATION(#5318,#5319);
#5321=ORIENTED_EDGE('',*,*,#229,.F.);
#5322=CARTESIAN_POINT('',(50.099999999999994,-30.0,74.0));
#5323=VERTEX_POINT('',#5322);
#5324=DIRECTION('',(-1.0,0.0,0.0));
#5325=VECTOR('',#5324,50.099999999999994);
#5326=LINE('',#5322,#5325);
#5327=EDGE_CURVE('',#5323,#220,#5326,.T.);
#5328=ORIENTED_EDGE('',*,*,#5327,.F.);
#5329=CARTESIAN_POINT('',(66.0,-30.0,89.900000000000006));
#5330=VERTEX_POINT('',#5329);
#5331=CARTESIAN_POINT('',(50.099999999999994,-30.0,89.900000000000006));
#5332=DIRECTION('',(0.0,1.0,0.0));
#5333=DIRECTION('',(1.0,0.0,0.0));
#5334=AXIS2_PLACEMENT_3D('',#5331,#5332,#5333);
#5335=CIRCLE('',#5334,15.900000000000000);
```

```
#5336=EDGE_CURVE('',#5330,#5323,#5335,.T.);
#5337=ORIENTED_EDGE('',*,*,#5336,.F.);
#5338=CARTESIAN_POINT('',(66.0,-30.0,90.099999999999994));
#5339=VERTEX_POINT('',#5338);
#5340=DIRECTION('',(0.0,0.0,-1.0));
#5341=VECTOR('',#5340,0.199999999999989);
#5342=LINE('',#5338,#5341);
#5343=EDGE_CURVE('',#5339,#5330,#5342,.T.);
#5344=ORIENTED_EDGE('',*,*,#5343,.F.);
#5345=CARTESIAN_POINT('',(50.099999999999994,-30.0,106.0));
#5346=VERTEX_POINT('',#5345);
#5347=CARTESIAN_POINT('',(50.099999999999994,-30.0,90.099999999999994));
#5348=DIRECTION('',(0.0,1.0,0.0));
#5349=DIRECTION('',(0.0,0.0,1.0));
#5350=AXIS2_PLACEMENT_3D('',#5347,#5348,#5349);
#5351=CIRCLE('',#5350,15.900000000000000);
#5352=EDGE_CURVE('',#5346,#5339,#5351,.T.);
#5353=ORIENTED_EDGE('',*,*,#5352,.F.);
#5354=DIRECTION('',(1.0,0.0,0.0));
#5355=VECTOR('',#5354,50.099999999999994);
#5356=LINE('',#147,#5355);
#5357=EDGE_CURVE('',#148,#5346,#5356,.T.);
#5358=ORIENTED_EDGE('',*,*,#5357,.F.);
#5359=EDGE_LOOP('',(#5321,#5328,#5337,#5344,#5353,#5358));
#5360=FACE_BOUND('',#5359,.F.);
#5361=CARTESIAN_POINT('',(260.0,-30.0,0.0));
#5362=DIRECTION('',(0.0,1.0,0.0));
#5363=DIRECTION('',(0.0,0.0,1.0));
#5364=AXIS2_PLACEMENT_3D('',#5361,#5362,#5363);
#5365=PLANE('',#5364);
#5366=ADVANCED_FACE('',(#5360),#5365,.T.);
#5367=SHAPE_ASPECT('featured shape: face 161',$,#38,.T.);
#5368=PROPERTY_DEFINITION('',$,#5367);
#5369=FACE_SHAPE_REPRESENTATION('',(#5366),#29);
#5370=PROPERTY_DEFINITION_REPRESENTATION(#5368,#5369);
#5371=ORIENTED_EDGE('',*,*,#154,.F.);
#5372=ORIENTED_EDGE('',*,*,#5357,.T.);
#5373=DIRECTION('',(0.0,-1.0,0.0));
#5374=VECTOR('',#5373,20.0);
#5375=LINE('',#5272,#5374);
#5376=EDGE_CURVE('',#5273,#5346,#5375,.T.);
#5377=ORIENTED_EDGE('',*,*,#5376,.F.);
#5378=ORIENTED_EDGE('',*,*,#5277,.F.);
#5379=EDGE_LOOP('',(#5371,#5372,#5377,#5378));
#5380=FACE_BOUND('',#5379,.F.);
#5381=CARTESIAN_POINT('',(0.0,-10.0,106.0));
#5382=DIRECTION('',(0.0,0.0,1.0));
#5383=DIRECTION('',(1.0,0.0,0.0));
#5384=AXIS2_PLACEMENT_3D('',#5381,#5382,#5383);
#5385=PLANE('',#5384);
#5386=ADVANCED_FACE('',(#5380),#5385,.F.);
#5387=SHAPE_ASPECT('featured shape: face 162',$,#38,.T.);
#5388=PROPERTY_DEFINITION('',$,#5387);
#5389=FACE_SHAPE_REPRESENTATION('',(#5386),#29);
#5390=PROPERTY_DEFINITION_REPRESENTATION(#5388,#5389);
#5391=ORIENTED_EDGE('',*,*,#5286,.F.);
```

```
#5392=ORIENTED_EDGE('',*,*,#5376,.T.);
#5393=ORIENTED_EDGE('',*,*,#5352,.T.);
#5394=DIRECTION('',(0.0,-1.0,0.0));
#5395=VECTOR('',#5394,20.0);
#5396=LINE('',#5279,#5395);
#5397=EDGE_CURVE('',#5280,#5339,#5396,.T.);
#5398=ORIENTED_EDGE('',*,*,#5397,.F.);
#5399=EDGE_LOOP('',(#5391,#5392,#5393,#5398));
#5400=FACE_BOUND('',#5399,.F.);
#5401=CARTESIAN_POINT('',(50.099999999999994,-10.0,90.099999999999994));
#5402=DIRECTION('',(0.0,1.0,0.0));
#5403=DIRECTION('',(0.0,0.0,1.0));
#5404=AXIS2_PLACEMENT_3D('',#5401,#5402,#5403);
#5405=CYLINDRICAL_SURFACE('',#5404,15.900000000000000);
#5406=ADVANCED_FACE('',(#5400),#5405,.F.);
#5407=SHAPE_ASPECT('featured shape: face 163',$,#38,.T.);
#5408=PROPERTY_DEFINITION('',$,#5407);
#5409=FACE_SHAPE_REPRESENTATION('',(#5406),#29);
#5410=PROPERTY_DEFINITION_REPRESENTATION(#5408,#5409);
#5411=ORIENTED_EDGE('',*,*,#5293,.F.);
#5412=ORIENTED_EDGE('',*,*,#5397,.T.);
#5413=ORIENTED_EDGE('',*,*,#5343,.T.);
#5414=DIRECTION('',(0.0,-1.0,0.0));
#5415=VECTOR('',#5414,20.0);
#5416=LINE('',#5288,#5415);
#5417=EDGE_CURVE('',#5289,#5330,#5416,.T.);
#5418=ORIENTED_EDGE('',*,*,#5417,.F.);
#5419=EDGE_LOOP('',(#5411,#5412,#5413,#5418));
#5420=FACE_BOUND('',#5419,.F.);
#5421=CARTESIAN_POINT('',(66.0,-10.0,90.099999999999994));
#5422=DIRECTION('',(1.0,0.0,0.0));
#5423=DIRECTION('',(0.0,0.0,-1.0));
#5424=AXIS2_PLACEMENT_3D('',#5421,#5422,#5423);
#5425=PLANE('',#5424);
#5426=ADVANCED_FACE('',(#5420),#5425,.F.);
#5427=SHAPE_ASPECT('featured shape: face 164',$,#38,.T.);
#5428=PROPERTY_DEFINITION('',$,#5427);
#5429=FACE_SHAPE_REPRESENTATION('',(#5426),#29);
#5430=PROPERTY_DEFINITION_REPRESENTATION(#5428,#5429);
#5431=ORIENTED_EDGE('',*,*,#5302,.F.);
#5432=ORIENTED_EDGE('',*,*,#5417,.T.);
#5433=ORIENTED_EDGE('',*,*,#5336,.T.);
#5434=DIRECTION('',(0.0,-1.0,0.0));
#5435=VECTOR('',#5434,20.0);
#5436=LINE('',#5295,#5435);
#5437=EDGE_CURVE('',#5296,#5323,#5436,.T.);
#5438=ORIENTED_EDGE('',*,*,#5437,.F.);
#5439=EDGE_LOOP('',(#5431,#5432,#5433,#5438));
#5440=FACE_BOUND('',#5439,.F.);
#5441=CARTESIAN_POINT('',(50.099999999999994,-10.0,89.900000000000006));
#5442=DIRECTION('',(0.0,1.0,0.0));
#5443=DIRECTION('',(0.0,0.0,1.0));
#5444=AXIS2_PLACEMENT_3D('',#5441,#5442,#5443);
#5445=CYLINDRICAL_SURFACE('',#5444,15.900000000000000);
#5446=ADVANCED_FACE('',(#5440),#5445,.F.);
#5447=SHAPE_ASPECT('featured shape: face 165',$,#38,.T.);
```

```
#5448=PROPERTY_DEFINITION('',$,#5447);
#5449=FACE_SHAPE_REPRESENTATION('',(#5446),#29);
#5450=PROPERTY_DEFINITION_REPRESENTATION(#5448,#5449);
#5451=ORIENTED_EDGE('',*,*,#224,.T.);
#5452=ORIENTED_EDGE('',*,*,#5307,.F.);
#5453=ORIENTED_EDGE('',*,*,#5437,.T.);
#5454=ORIENTED_EDGE('',*,*,#5327,.T.);
#5455=EDGE_LOOP('',(#5451,#5452,#5453,#5454));
#5456=FACE_BOUND('',#5455,.F.);
#5457=CARTESIAN_POINT('',(50.099999999999994,-10.0,74.0));
#5458=DIRECTION('',(0.0,0.0,-1.0));
#5459=DIRECTION('',(-1.0,0.0,0.0));
#5460=AXIS2_PLACEMENT_3D('',#5457,#5458,#5459);
#5461=PLANE('',#5460);
#5462=ADVANCED_FACE('',(#5456),#5461,.F.);
#5463=SHAPE_ASPECT('featured shape: face 166',$,#38,.T.);
#5464=PROPERTY_DEFINITION('',$,#5463);
#5465=FACE_SHAPE_REPRESENTATION('',(#5462),#29);
#5466=PROPERTY_DEFINITION_REPRESENTATION(#5464,#5465);
#5467=CLOSED_SHELL('',(#74,#142,#238,#334,#425,#516,#607,#647,#1109,#1180,#12
60,#1360,#1796,#1816,#1836,#1856,#1876,#1900,#1916,#1940,#1956,#1990,#2019,#2
041,#2058,#2084,#2110,#2134,#2153,#2177,#2196,#2220,#2239,#2263,#2282,#2316,#
2364,#2388,#2404,#2434,#2468,#2508,#2532,#2548,#2564,#2598,#2638,#2662,#2678,
#2694,#2728,#2768,#2792,#2808,#2824,#2858,#2898,#2922,#2938,#2954,#2988,#3028
,#3052,#3068,#3084,#3118,#3158,#3182,#3198,#3214,#3240,#3268,#3288,#3304,#334
8,#3372,#3392,#3412,#3428,#3458,#3482,#3498,#3528,#3552,#3568,#3598,#3622,#36
38,#3668,#3692,#3708,#3724,#3740,#3756,#3772,#3788,#3804,#3820,#3836,#3852,#3
868,#3916,#3948,#3984,#4016,#4052,#4084,#4120,#4144,#4192,#4214,#4242,#4264,#
4286,#4308,#4323,#4338,#4360,#4388,#4410,#4432,#4454,#4469,#4484,#4506,#4622,
#4642,#4662,#4682,#4702,#4718,#4742,#4762,#4782,#4802,#4822,#4842,#4862,#4878
,#4902,#4922,#4942,#4966,#4986,#5006,#5030,#5050,#5070,#5094,#5114,#5134,#515
0,#5166,#5182,#5198,#5214,#5230,#5246,#5262,#5316,#5366,#5386,#5406,#5426,#54
46,#5462));
#5468=MANIFOLD_SOLID_BREP('',#5467);
#5469=ADVANCED_BREP_SHAPE_REPRESENTATION('',(#5468),#29);
#5470=SHAPE_DEFINITION_REPRESENTATION(#38,#5469);
#5471=DATUM_FEATURE('',$,#38,.T.);
#5472=PROPERTY_DEFINITION('',$,#5471);
#5473=REPRESENTATION('',(#5316),#29);
#5474=PROPERTY_DEFINITION_REPRESENTATION(#5472,#5473);
#5475=DATUM('DTM1',$,#38,.T.,'DTM1');
#5476=SHAPE_ASPECT_RELATIONSHIP('',$,#5471,#5475);
#5477=DATUM_REFERENCE(1,#5475);
#5478=DIMENSIONAL_LOCATION('',$,#1110,#5317);
#5479=PRECISION_QUALIFIER(4);
#5485=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5479))REPRESENTAT
ION_ITEM(''));
#5486=SHAPE_DIMENSION_REPRESENTATION('',(#5485),#29);
#5487=PRECISION_QUALIFIER(4);
#5488=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#5489=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#5490=MEASURE_QUALIFICATION('','',#5488,(#5487));
#5491=MEASURE_QUALIFICATION('','',#5489,(#5487));
#5492=TOLERANCE_VALUE(#5489,#5488);
#5493=PLUS_MINUS_TOLERANCE(#5492,#5478);
```



```
#5494=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5478,#5486);
#5495=CARTESIAN_POINT('',(0.0,0.0,90.0));
#5496=DIRECTION('',(0.0,0.0,-1.0));
#5497=DIRECTION('',(-1.0,0.0,0.0));
#5498=AXIS2_PLACEMENT_3D('DTM2',#5495,#5496,#5497);
#5499=DATUM_FEATURE('',$,#38,.T.);
#5500=PROPERTY_DEFINITION('',$,#5499);
#5501=REPRESENTATION('',(#5498),#29);
#5502=PROPERTY_DEFINITION_REPRESENTATION(#5500,#5501);
#5503=DATUM('DTM2',$,#38,.F.,'DTM2');
#5504=SHAPE_ASPECT_RELATIONSHIP('',$,#5499,#5503);
#5505=DATUM_REFERENCE(1,#5503);
#5506=DIMENSIONAL_LOCATION('',$,#75,#5508);
#5507=PRECISION_QUALIFIER(4);
#5508=SHAPE_ASPECT('',$,#38,.T.);
#5509=PROPERTY_DEFINITION('',$,#5508);
#5510=PROPERTY_DEFINITION_REPRESENTATION(#5509,#5501);
#5513=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(90.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5507))REPRESENTAT
ION_ITEM(''));
#5514=SHAPE_DIMENSION_REPRESENTATION('',(#5513),#29);
#5515=PRECISION_QUALIFIER(4);
#5516=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.150000000000000),#10);
#5517=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.150000000000000),#10);
#5518=MEASURE_QUALIFICATION('','',#5516,(#5515));
#5519=MEASURE_QUALIFICATION('','',#5517,(#5515));
#5520=TOLERANCE_VALUE(#5517,#5516);
#5521=PLUS_MINUS_TOLERANCE(#5520,#5506);
#5522=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5506,#5514);
#5523=DATUM_FEATURE('',$,#38,.T.);
#5524=PROPERTY_DEFINITION('',$,#5523);
#5525=REPRESENTATION('',(#334),#29);
#5526=PROPERTY_DEFINITION_REPRESENTATION(#5524,#5525);
#5527=DATUM('A',$,#38,.T.,'A');
#5528=SHAPE_ASPECT_RELATIONSHIP('',$,#5523,#5527);
#5529=DATUM_REFERENCE(1,#5527);
#5530=DATUM_FEATURE('',$,#38,.T.);
#5531=PROPERTY_DEFINITION('',$,#5530);
#5532=REPRESENTATION('',(#238),#29);
#5533=PROPERTY_DEFINITION_REPRESENTATION(#5531,#5532);
#5534=DATUM('C',$,#38,.T.,'C');
#5535=SHAPE_ASPECT_RELATIONSHIP('',$,#5530,#5534);
#5536=DATUM_REFERENCE(1,#5534);
#5537=DATUM_FEATURE('',$,#38,.T.);
#5538=PROPERTY_DEFINITION('',$,#5537);
#5539=REPRESENTATION('',(#1109),#29);
#5540=PROPERTY_DEFINITION_REPRESENTATION(#5538,#5539);
#5541=DATUM('D',$,#38,.T.,'D');
#5542=SHAPE_ASPECT_RELATIONSHIP('',$,#5537,#5541);
#5543=DATUM_REFERENCE(1,#5541);
#5544=DATUM_FEATURE('',$,#38,.T.);
#5545=PROPERTY_DEFINITION('',$,#5544);
#5546=REPRESENTATION('',(#2110),#29);
#5547=PROPERTY_DEFINITION_REPRESENTATION(#5545,#5546);
#5548=DATUM('F',$,#38,.T.,'F');
#5549=SHAPE_ASPECT_RELATIONSHIP('',$,#5544,#5548);
```

```
#5550=DATUM_REFERENCE(1,#5548);
#5551=DATUM_FEATURE('',$,#38,.T.);
#5552=PROPERTY_DEFINITION('',$,#5551);
#5553=REPRESENTATION('',(#1260),#29);
#5554=PROPERTY_DEFINITION_REPRESENTATION(#5552,#5553);
#5555=DATUM('E',$,#38,.T.,'E');
#5556=SHAPE_ASPECT_RELATIONSHIP('',$,#5551,#5555);
#5557=DATUM_REFERENCE(1,#5555);


#5595=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('',$,#38,.T.)STEP());
#5596=PROPERTY_DEFINITION('',$,#5595);
#5597=PRODUCT_DEFINITION_SHAPE('',$,#5595);
#5598=CARTESIAN_POINT('',(230.0,-20.0,0.0));
#5599=DIRECTION('',(0.0,0.0,1.0));
#5600=DIRECTION('',(0.707106781186547,0.707106781186547,0.0));
#5601=AXIS2_PLACEMENT_3D('orientation',#5598,#5599,#5600);
#5602=REPRESENTATION('',(#1260,#1360),#29);
#5603=PROPERTY_DEFINITION_REPRESENTATION(#5596,#5602);
#5604=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5601),#29);
#5605=PROPERTY_DEFINITION_REPRESENTATION(#5596,#5604);
#5606=FEATURE_COMPONENT_DEFINITION('',$);
#5607=PRODUCT_DEFINITION_SHAPE('',$,#5606);
#5608=PATH_FEATURE_COMPONENT('linear path','linear',#5607,.F.);
#5609=STANDARD_UNCERTAINTY('upper limit','plus',0.300000000000000);
#5610=STANDARD_UNCERTAINTY('lower limit','minus',-0.300000000000000);
#5611=PRECISION_QUALIFIER(4);
#5617=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(180.0),#10)

QUALIFIED_REPRESENTATION_ITEM((#5609,#5610,#5611))REPRESENTATION_ITEM('distan
ce'));
#5618=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5601,#5617),#29);
#5619=PROPERTY_DEFINITION('',$,#5608);
#5620=PROPERTY_DEFINITION_REPRESENTATION(#5619,#5618);
#5621=DIRECTION_SHAPE_REPRESENTATION('',(#5599),#29);
#5622=PROPERTY_DEFINITION_REPRESENTATION(#5619,#5621);
#5623=FEATURE_COMPONENT_DEFINITION('',$);
#5624=PRODUCT_DEFINITION_SHAPE('',$,#5623);
#5625=VEE_PROFILE('vee profile','v-shape',#5624,.F.);
#5630=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)
      PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#5635=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-
45.0),#19)
      PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#5636=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5601,#5630,#5635),#29);
#5637=PROPERTY_DEFINITION('',$,#5625);
#5638=PROPERTY_DEFINITION_REPRESENTATION(#5637,#5636);
#5639=SHAPE_ASPECT('','removal boundary occurrence',#5597,.T.);
#5640=SHAPE_DEFINING_RELATIONSHIP('vee boundary','profile
usage',#5625,#5639);
#5641=SHAPE_ASPECT('','course of travel occurrence',#5597,.T.);
#5642=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#5608,#5641);
```



```
#5643=DIMENSIONAL_LOCATION('',$,#1110,#1361);
#5644=PRECISION_QUALIFIER(4);
#5650=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5644))REPRESENTAT
ION_ITEM(''));
#5651=SHAPE_DIMENSION_REPRESENTATION('',(#5650),#29);
#5652=PRECISION_QUALIFIER(4);
#5653=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#5654=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.250000000000000),#10);
#5655=MEASURE_QUALIFICATION('','',#5653,(#5652));
#5656=MEASURE_QUALIFICATION('','',#5654,(#5652));
#5657=TOLERANCE_VALUE(#5654,#5653);
#5658=PLUS_MINUS_TOLERANCE(#5657,#5643);
#5659=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5643,#5651);
#5660=DIMENSIONAL_LOCATION('',$,#239,#1261);
#5661=PRECISION_QUALIFIER(4);
#5667=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(230.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5661))REPRESENTA
TION_ITEM(''));
#5668=SHAPE_DIMENSION_REPRESENTATION('',(#5667),#29);
#5669=PRECISION_QUALIFIER(4);
#5670=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#5671=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#5672=MEASURE_QUALIFICATION('','',#5670,(#5669));
#5673=MEASURE_QUALIFICATION('','',#5671,(#5669));
#5674=TOLERANCE_VALUE(#5671,#5670);
#5675=PLUS_MINUS_TOLERANCE(#5674,#5660);
#5676=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5660,#5668);

#5682=(CHARACTERIZED_OBJECT('V_LIN',$)FEATURE_DEFINITION()INSTANCED_FEATURE()
REMOVAL_VOLUME()SHAPE_ASPECT('V_LIN',$,#38,.T.));
#5683=PROPERTY_DEFINITION('',$,#5682);
#5684=PRODUCT_DEFINITION_SHAPE('',$,#5682);
#5685=CARTESIAN_POINT('',(0.0,0.0,0.0));
#5686=DIRECTION('',(0.0,0.0,1.0));
#5687=DIRECTION('',(1.0,0.0,0.0));
#5688=AXIS2_PLACEMENT_3D('orientation',#5685,#5686,#5687);
#5689=REPRESENTATION('',(#1260,#1796),#29);
#5690=PROPERTY_DEFINITION_REPRESENTATION(#5683,#5689);
#5691=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5688),#29);
#5692=PROPERTY_DEFINITION_REPRESENTATION(#5683,#5691);
#5693=SHAPE_ASPECT('','volume shape',#38,.T.);
#5694=PROPERTY_DEFINITION('',$,#5693);
#5695=PROPERTY_DEFINITION_REPRESENTATION(#5694,#5689);
#5696=SHAPE_ASPECT('','shape volume occurrence',#5684,.T.);
#5697=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#5693,#5696);
#5703=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('',$,#38,.T.)STEP());
#5704=PROPERTY_DEFINITION('',$,#5703);
#5705=PRODUCT_DEFINITION_SHAPE('',$,#5703);
#5706=CARTESIAN_POINT('',(230.0,-10.0,130.0));
#5707=DIRECTION('',(0.0,-1.0,0.0));
#5708=DIRECTION('',(0.707106781186547,0.0,0.707106781186547));
#5709=AXIS2_PLACEMENT_3D('orientation',#5706,#5707,#5708);
#5710=REPRESENTATION('',(#1260,#1816,#1836),#29);
#5711=PROPERTY_DEFINITION_REPRESENTATION(#5704,#5710);
```

```
#5712=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5709),#29);
#5713=PROPERTY_DEFINITION_REPRESENTATION(#5704,#5712);
#5714=FEATURE_COMPONENT_DEFINITION('',$);
#5715=PRODUCT_DEFINITION_SHAPE('',$,#5714);
#5716=PATH_FEATURE_COMPONENT('linear path','linear',#5715,.F.);
#5717=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#5718=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#5719=PRECISION_QUALIFIER(4);
#5725=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5717,#5718,#5719)
)REPRESENTATION_ITEM('distance'));
#5726=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5709,#5725),#29);
#5727=PROPERTY_DEFINITION('',$,#5716);
#5728=PROPERTY_DEFINITION_REPRESENTATION(#5727,#5726);
#5729=DIRECTION_SHAPE_REPRESENTATION('',(#5707),#29);
#5730=PROPERTY_DEFINITION_REPRESENTATION(#5727,#5729);
#5731=FEATURE_COMPONENT_DEFINITION('',$);
#5732=PRODUCT_DEFINITION_SHAPE('',$,#5731);
#5733=VEE_PROFILE('vee profile','v-shape',#5732,.F.);
#5738=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#5743=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-
45.0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#5744=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#5745=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#5746=PRECISION_QUALIFIER(4);
#5752=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5744,#5745,#5746)
)REPRESENTATION_ITEM('profile radius'));
#5753=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5709,#5738,#5743,#5752),#29);
#5754=PROPERTY_DEFINITION('',$,#5733);
#5755=PROPERTY_DEFINITION_REPRESENTATION(#5754,#5753);
#5756=SHAPE_ASPECT('','removal boundary occurrence',#5705,.T.);
#5757=SHAPE_DEFINING_RELATIONSHIP('vee boundary','profile
usage',#5733,#5756);
#5758=SHAPE_ASPECT('','course of travel occurrence',#5705,.T.);
#5759=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#5716,#5758);
#5760=DIMENSIONAL_LOCATION('',$,#143,#1837);
#5761=PRECISION_QUALIFIER(4);
#5767=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5761))REPRESENTAT
ION_ITEM(''));
#5768=SHAPE_DIMENSION_REPRESENTATION('',(#5767),#29);
#5769=PRECISION_QUALIFIER(4);
#5770=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.300000000000000),#10);
#5771=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.300000000000000),#10);
#5772=MEASURE_QUALIFICATION('','',#5770,(#5769));
#5773=MEASURE_QUALIFICATION('','',#5771,(#5769));
#5774=TOLERANCE_VALUE(#5771,#5770);
#5775=PLUS_MINUS_TOLERANCE(#5774,#5760);
#5776=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5760,#5768);
#5777=CARTESIAN_POINT('',(230.0,-10.0,180.0));
#5778=DIRECTION('',(-1.0,0.0,0.0));
#5779=DIRECTION('',(0.0,0.0,-1.0));
#5780=AXIS2_PLACEMENT_3D('',#5777,#5778,#5779);
```



```
#5781=SHAPE_ASPECT('',$,#38,.F.);
#5782=PROPERTY_DEFINITION('',$,#5781);
#5783=REPRESENTATION('',(#5780),#29);
#5784=PROPERTY_DEFINITION_REPRESENTATION(#5782,#5783);
#5785=DIMENSIONAL_LOCATION('',$,#239,#5781);
#5786=PRECISION_QUALIFIER(4);
#5792=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(230.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5786))REPRESENTA
TION_ITEM(''));
#5793=SHAPE_DIMENSION_REPRESENTATION('',(#5792),#29);
#5794=PRECISION_QUALIFIER(4);
#5795=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#5796=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#5797=MEASURE_QUALIFICATION('','',#5795,(#5794));
#5798=MEASURE_QUALIFICATION('','',#5796,(#5794));
#5799=TOLERANCE_VALUE(#5796,#5795);
#5800=PLUS_MINUS_TOLERANCE(#5799,#5785);
#5801=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5785,#5793);
#5807=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('',$,#38,.T.)STEP());
#5808=PROPERTY_DEFINITION('',$,#5807);
#5809=PRODUCT_DEFINITION_SHAPE('',$,#5807);
#5810=CARTESIAN_POINT('',(230.0,-50.0,50.0));
#5811=DIRECTION('',(0.0,1.0,0.0));
#5812=DIRECTION('',(0.707106781186547,0.0,-0.707106781186547));
#5813=AXIS2_PLACEMENT_3D('orientation',#5810,#5811,#5812);
#5814=REPRESENTATION('',(#1260,#1856,#1876),#29);
#5815=PROPERTY_DEFINITION_REPRESENTATION(#5808,#5814);
#5816=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5813),#29);
#5817=PROPERTY_DEFINITION_REPRESENTATION(#5808,#5816);
#5818=FEATURE_COMPONENT_DEFINITION('',$);
#5819=PRODUCT_DEFINITION_SHAPE('',$,#5818);
#5820=PATH_FEATURE_COMPONENT('linear path','linear',#5819,.F.);
#5821=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#5822=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#5823=PRECISION_QUALIFIER(4);
#5829=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5821,#5822,#5823)
)REPRESENTATION_ITEM('distance'));
#5830=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5813,#5829),#29);
#5831=PROPERTY_DEFINITION('',$,#5820);
#5832=PROPERTY_DEFINITION_REPRESENTATION(#5831,#5830);
#5833=DIRECTION_SHAPE_REPRESENTATION('',(#5811),#29);
#5834=PROPERTY_DEFINITION_REPRESENTATION(#5831,#5833);
#5835=FEATURE_COMPONENT_DEFINITION('',$);
#5836=PRODUCT_DEFINITION_SHAPE('',$,#5835);
#5837=VEE_PROFILE('vee profile','v-shape',#5836,.F.);
#5842=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#5847=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-
45.0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#5848=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#5849=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#5850=PRECISION_QUALIFIER(4);
```

```
#5856=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5848,#5849,#5850)
)REPRESENTATION_ITEM('profile radius'));
#5857=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5813,#5842,#5847,#5856),#29);
#5858=PROPERTY_DEFINITION('',$,#5837);
#5859=PROPERTY_DEFINITION_REPRESENTATION(#5858,#5857);
#5860=SHAPE_ASPECT('','removal boundary occurrence',#5809,.T.);
#5861=SHAPE_DEFINING_RELATIONSHIP('vee boundary','profile
usage',#5837,#5860);
#5862=SHAPE_ASPECT('','course of travel occurrence',#5809,.T.);
#5863=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#5820,#5862);
#5864=DIMENSIONAL_LOCATION('',$,#75,#1877);
#5865=PRECISION_QUALIFIER(4);
#5871=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5865))REPRESENTAT
ION_ITEM(''));
#5872=SHAPE_DIMENSION_REPRESENTATION('',(#5871),#29);
#5873=PRECISION_QUALIFIER(4);
#5874=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.300000000000000),#10);
#5875=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.300000000000000),#10);
#5876=MEASURE_QUALIFICATION('','',#5874,(#5873));
#5877=MEASURE_QUALIFICATION('','',#5875,(#5873));
#5878=TOLERANCE_VALUE(#5875,#5874);
#5879=PLUS_MINUS_TOLERANCE(#5878,#5864);
#5880=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5864,#5872);
#5881=CARTESIAN_POINT('',(230.0,-50.0,0.0));
#5882=DIRECTION('',(-1.0,0.0,0.0));
#5883=DIRECTION('',(0.0,0.0,1.0));
#5884=AXIS2_PLACEMENT_3D('',#5881,#5882,#5883);
#5885=SHAPE_ASPECT('',$,#38,.F.);
#5886=PROPERTY_DEFINITION('',$,#5885);
#5887=REPRESENTATION('',(#5884),#29);
#5888=PROPERTY_DEFINITION_REPRESENTATION(#5886,#5887);
#5889=DIMENSIONAL_LOCATION('',$,#239,#5885);
#5890=PRECISION_QUALIFIER(4);
#5896=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(230.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5890))REPRESENTA
TION_ITEM(''));
#5897=SHAPE_DIMENSION_REPRESENTATION('',(#5896),#29);
#5898=PRECISION_QUALIFIER(4);
#5899=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#5900=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#5901=MEASURE_QUALIFICATION('','',#5899,(#5898));
#5902=MEASURE_QUALIFICATION('','',#5900,(#5898));
#5903=TOLERANCE_VALUE(#5900,#5899);
#5904=PLUS_MINUS_TOLERANCE(#5903,#5889);
#5905=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5889,#5897);
#5911=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#5912=PROPERTY_DEFINITION('',$,#5911);
#5913=PRODUCT_DEFINITION_SHAPE('',$,#5911);
#5914=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#5915=DIRECTION('',(0.0,-1.0,0.0));
#5916=DIRECTION('',(1.0,0.0,0.0));
#5917=AXIS2_PLACEMENT_3D('orientation',#5914,#5915,#5916);
```



```
#5918=REPRESENTATION('',(#1900,#1916),#29);
#5919=PROPERTY_DEFINITION_REPRESENTATION(#5912,#5918);
#5920=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5917),#29);
#5921=PROPERTY_DEFINITION_REPRESENTATION(#5912,#5920);
#5922=FEATURE_COMPONENT_DEFINITION('',$);
#5923=PRODUCT_DEFINITION_SHAPE('',$,#5922);
#5924=CIRCULAR_CLOSED_PROFILE('circular profile',$,#5923,.F.);
#5925=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#5926=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#5927=PRECISION_QUALIFIER(4);
#5933=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5925,#5926,#5927)
)REPRESENTATION_ITEM('diameter'));
#5934=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5917,#5933),#29);
#5935=PROPERTY_DEFINITION('',$,#5924);
#5936=PROPERTY_DEFINITION_REPRESENTATION(#5935,#5934);
#5937=FEATURE_COMPONENT_DEFINITION('',$);
#5938=PRODUCT_DEFINITION_SHAPE('',$,#5937);
#5939=PATH_FEATURE_COMPONENT('linear path','linear',#5938,.F.);
#5944=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)REPRESENTATION_ITEM('distance'));
#5945=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5917,#5944),#29);
#5946=PROPERTY_DEFINITION('',$,#5939);
#5947=PROPERTY_DEFINITION_REPRESENTATION(#5946,#5945);
#5948=DIRECTION_SHAPE_REPRESENTATION('',(#5915),#29);
#5949=PROPERTY_DEFINITION_REPRESENTATION(#5946,#5948);
#5950=SHAPE_ASPECT('','diameter occurrence',#5913,.T.);
#5951=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#5924,#5950);
#5952=SHAPE_ASPECT('','hole depth occurrence',#5913,.T.);
#5953=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#5939,#5952);
#5954=FEATURE_COMPONENT_DEFINITION('',$);
#5955=PRODUCT_DEFINITION_SHAPE('',$,#5954);
#5956=HOLE_BOTTOM('bottom condition','through',#5955,.F.);
#5957=SHAPE_ASPECT('','bottom condition occurrence',#5913,.T.);
#5958=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#5956,#5957);
#5959=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#5960=DIRECTION('',(0.0,-1.0,0.0));
#5961=AXIS1_PLACEMENT('',#5959,#5960);
#5962=SHAPE_ASPECT('',$,#38,.F.);
#5963=PROPERTY_DEFINITION('',$,#5962);
#5964=REPRESENTATION('',(#5961),#29);
#5965=PROPERTY_DEFINITION_REPRESENTATION(#5963,#5964);
#5966=DIMENSIONAL_LOCATION('',$,#648,#5962);
#5967=PRECISION_QUALIFIER(4);
#5973=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5967))REPRESENTAT
ION_ITEM(''));
#5974=SHAPE_DIMENSION_REPRESENTATION('',(#5973),#29);
#5975=PRECISION_QUALIFIER(4);
#5976=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#5977=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#5978=MEASURE_QUALIFICATION('','',#5976,(#5975));
#5979=MEASURE_QUALIFICATION('','',#5977,(#5975));
#5980=TOLERANCE_VALUE(#5977,#5976);
```

```
#5981=PLUS_MINUS_TOLERANCE(#5980,#5966);
#5982=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5966,#5974);
#5983=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#5984=DIRECTION('',(0.0,-1.0,0.0));
#5985=AXIS1_PLACEMENT('',#5983,#5984);
#5986=SHAPE_ASPECT('',$,#38,.F.);
#5987=PROPERTY_DEFINITION('',$,#5986);
#5988=REPRESENTATION('',(#5985),#29);
#5989=PROPERTY_DEFINITION_REPRESENTATION(#5987,#5988);
#5990=DIMENSIONAL_LOCATION('',$,#1837,#5986);
#5991=PRECISION_QUALIFIER(4);
#5997=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5991))REPRESENTAT
ION_ITEM(''));
#5998=SHAPE_DIMENSION_REPRESENTATION('',(#5997),#29);
#5999=PRECISION_QUALIFIER(4);
#6000=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#6001=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#6002=MEASURE_QUALIFICATION('','',#6000,(#5999));
#6003=MEASURE_QUALIFICATION('','',#6001,(#5999));
#6004=TOLERANCE_VALUE(#6001,#6000);
#6005=PLUS_MINUS_TOLERANCE(#6004,#5990);
#6006=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5990,#5998);
#6012=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#6013=PROPERTY_DEFINITION('',$,#6012);
#6014=PRODUCT_DEFINITION_SHAPE('',$,#6012);
#6015=CARTESIAN_POINT('',(245.0,-20.0,70.0));
#6016=DIRECTION('',(0.0,-1.0,0.0));
#6017=DIRECTION('',(1.0,0.0,0.0));
#6018=AXIS2_PLACEMENT_3D('orientation',#6015,#6016,#6017);
#6019=REPRESENTATION('',(#1940,#1956),#29);
#6020=PROPERTY_DEFINITION_REPRESENTATION(#6013,#6019);
#6021=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6018),#29);
#6022=PROPERTY_DEFINITION_REPRESENTATION(#6013,#6021);
#6023=FEATURE_COMPONENT_DEFINITION('',$);
#6024=PRODUCT_DEFINITION_SHAPE('',$,#6023);
#6025=CIRCULAR_CLOSED_PROFILE('circular profile',$,#6024,.F.);
#6026=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#6027=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#6028=PRECISION_QUALIFIER(4);
#6034=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6026,#6027,#6028)
)REPRESENTATION_ITEM('diameter'));
#6035=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6018,#6034),#29);
#6036=PROPERTY_DEFINITION('',$,#6025);
#6037=PROPERTY_DEFINITION_REPRESENTATION(#6036,#6035);
#6038=FEATURE_COMPONENT_DEFINITION('',$);
#6039=PRODUCT_DEFINITION_SHAPE('',$,#6038);
#6040=PATH_FEATURE_COMPONENT('linear path','linear',#6039,.F.);
#6045=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)REPRESENTATION_ITEM('distance'));
#6046=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6018,#6045),#29);
#6047=PROPERTY_DEFINITION('',$,#6040);
#6048=PROPERTY_DEFINITION_REPRESENTATION(#6047,#6046);
#6049=DIRECTION_SHAPE_REPRESENTATION('',(#6016),#29);
```



```
#6050=PROPERTY_DEFINITION_REPRESENTATION(#6047,#6049);
#6051=SHAPE_ASPECT('','diameter occurrence',#6014,.T.);
#6052=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#6025,#6051);
#6053=SHAPE_ASPECT('','hole depth occurrence',#6014,.T.);
#6054=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#6040,#6053);
#6055=FEATURE_COMPONENT_DEFINITION('',$);
#6056=PRODUCT_DEFINITION_SHAPE('',$,#6055);
#6057=HOLE_BOTTOM('bottom condition','through',#6056,.F.);
#6058=SHAPE_ASPECT('','bottom condition occurrence',#6014,.T.);
#6059=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#6057,#6058);
#6060=CARTESIAN_POINT('',(245.0,-20.0,70.0));
#6061=DIRECTION('',(0.0,-1.0,0.0));
#6062=AXIS1_PLACEMENT('',#6060,#6061);
#6063=SHAPE_ASPECT('',$,#38,.F.);
#6064=PROPERTY_DEFINITION('',$,#6063);
#6065=REPRESENTATION('',(#6062),#29);
#6066=PROPERTY_DEFINITION_REPRESENTATION(#6064,#6065);
#6067=DIMENSIONAL_LOCATION('',$,#648,#6063);
#6068=PRECISION_QUALIFIER(4);
#6074=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6068))REPRESENTAT
ION_ITEM(''));
#6075=SHAPE_DIMENSION_REPRESENTATION('',(#6074),#29);
#6076=PRECISION_QUALIFIER(4);
#6077=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#6078=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#6079=MEASURE_QUALIFICATION('','',#6077,(#6076));
#6080=MEASURE_QUALIFICATION('','',#6078,(#6076));
#6081=TOLERANCE_VALUE(#6078,#6077);
#6082=PLUS_MINUS_TOLERANCE(#6081,#6067);
#6083=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6067,#6075);
#6084=CARTESIAN_POINT('',(245.0,-20.0,70.0));
#6085=DIRECTION('',(0.0,-1.0,0.0));
#6086=AXIS1_PLACEMENT('',#6084,#6085);
#6087=SHAPE_ASPECT('',$,#38,.F.);
#6088=PROPERTY_DEFINITION('',$,#6087);
#6089=REPRESENTATION('',(#6086),#29);
#6090=PROPERTY_DEFINITION_REPRESENTATION(#6088,#6089);
#6091=DIMENSIONAL_LOCATION('',$,#1877,#6087);
#6092=PRECISION_QUALIFIER(4);
#6098=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6092))REPRESENTAT
ION_ITEM(''));
#6099=SHAPE_DIMENSION_REPRESENTATION('',(#6098),#29);
#6100=PRECISION_QUALIFIER(4);
#6101=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#6102=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#6103=MEASURE_QUALIFICATION('','',#6101,(#6100));
#6104=MEASURE_QUALIFICATION('','',#6102,(#6100));
#6105=TOLERANCE_VALUE(#6102,#6101);
#6106=PLUS_MINUS_TOLERANCE(#6105,#6091);
#6107=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6091,#6099);
#6113=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
```

```
#6114=PROPERTY_DEFINITION('',$,#6113);
#6115=PRODUCT_DEFINITION_SHAPE('',$,#6113);
#6116=CARTESIAN_POINT('',(160.0,0.0,90.0));
#6117=DIRECTION('',(0.0,-1.0,0.0));
#6118=DIRECTION('',(1.0,0.0,0.0));
#6119=AXIS2_PLACEMENT_3D('orientation',#6116,#6117,#6118);
#6120=REPRESENTATION('',(#1990,#2019,#2041,#2058),#29);
#6121=PROPERTY_DEFINITION_REPRESENTATION(#6114,#6120);
#6122=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6119),#29);
#6123=PROPERTY_DEFINITION_REPRESENTATION(#6114,#6122);
#6124=FEATURE_COMPONENT_DEFINITION('',$);
#6125=PRODUCT_DEFINITION_SHAPE('',$,#6124);
#6126=CIRCULAR_CLOSED_PROFILE('circular profile',$,#6125,.F.);
#6127=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#6128=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#6129=PRECISION_QUALIFIER(4);
#6135=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(40.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6127,#6128,#6129)
)REPRESENTATION_ITEM('diameter'));
#6136=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6119,#6135),#29);
#6137=PROPERTY_DEFINITION('',$,#6126);
#6138=PROPERTY_DEFINITION_REPRESENTATION(#6137,#6136);
#6139=FEATURE_COMPONENT_DEFINITION('',$);
#6140=PRODUCT_DEFINITION_SHAPE('',$,#6139);
#6141=PATH_FEATURE_COMPONENT('linear path','linear',#6140,.F.);
#6146=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.000000000000007),#10)REPRESENTATION_ITEM('distance'));
#6147=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6119,#6146),#29);
#6148=PROPERTY_DEFINITION('',$,#6141);
#6149=PROPERTY_DEFINITION_REPRESENTATION(#6148,#6147);
#6150=DIRECTION_SHAPE_REPRESENTATION('',(#6117),#29);
#6151=PROPERTY_DEFINITION_REPRESENTATION(#6148,#6150);
#6152=SHAPE_ASPECT('','diameter occurrence',#6115,.T.);
#6153=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#6126,#6152);
#6154=SHAPE_ASPECT('','hole depth occurrence',#6115,.T.);
#6155=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#6141,#6154);
#6156=FEATURE_COMPONENT_DEFINITION('',$);
#6157=PRODUCT_DEFINITION_SHAPE('',$,#6156);
#6158=HOLE_BOTTOM('bottom condition','through',#6157,.F.);
#6159=SHAPE_ASPECT('','bottom condition occurrence',#6115,.T.);
#6160=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#6158,#6159);
#6161=CARTESIAN_POINT('',(160.0,0.0,90.0));
#6162=DIRECTION('',(0.0,-1.0,0.0));
#6163=AXIS1_PLACEMENT('',#6161,#6162);
#6164=SHAPE_ASPECT('',$,#38,.F.);
#6165=PROPERTY_DEFINITION('',$,#6164);
#6166=REPRESENTATION('',(#6163),#29);
#6167=PROPERTY_DEFINITION_REPRESENTATION(#6165,#6166);
#6168=DIMENSIONAL_LOCATION('',$,#1261,#6164);
#6169=PRECISION_QUALIFIER(4);
#6175=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(70.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6169))REPRESENTAT
ION_ITEM(''));
#6176=SHAPE_DIMENSION_REPRESENTATION('',(#6175),#29);
```



```
#6177=PRECISION_QUALIFIER(4);
#6178=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#6179=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.250000000000000),#10);
#6180=MEASURE_QUALIFICATION('','',#6178,(#6177));
#6181=MEASURE_QUALIFICATION('','',#6179,(#6177));
#6182=TOLERANCE_VALUE(#6179,#6178);
#6183=PLUS_MINUS_TOLERANCE(#6182,#6168);
#6184=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6168,#6176);
#6185=CARTESIAN_POINT('',(160.0,0.0,90.0));
#6186=DIRECTION('',(0.0,-1.0,0.0));
#6187=AXIS1_PLACEMENT('',#6185,#6186);
#6188=SHAPE_ASPECT('',$,#38,.F.);
#6189=PROPERTY_DEFINITION('',$,#6188);
#6190=REPRESENTATION('',(#6187),#29);
#6191=PROPERTY_DEFINITION_REPRESENTATION(#6189,#6190);
#6192=DIMENSIONAL_LOCATION('',$,#143,#6188);
#6193=PRECISION_QUALIFIER(4);
#6199=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(90.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6193))REPRESENTAT
ION_ITEM(''));
#6200=SHAPE_DIMENSION_REPRESENTATION('',(#6199),#29);
#6201=PRECISION_QUALIFIER(4);
#6202=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.150000000000000),#10);
#6203=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.150000000000000),#10);
#6204=MEASURE_QUALIFICATION('','',#6202,(#6201));
#6205=MEASURE_QUALIFICATION('','',#6203,(#6201));
#6206=TOLERANCE_VALUE(#6203,#6202);
#6207=PLUS_MINUS_TOLERANCE(#6206,#6192);
#6208=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6192,#6200);
#6214=(CHARACTERIZED_OBJECT('','open
rectangular')FEATURE_DEFINITION()INSTANCED_FEATURE()POCKET()SHAPE_ASPECT('','
open rectangular',#38,.T.));
#6215=PROPERTY_DEFINITION('',$,#6214);
#6216=PRODUCT_DEFINITION_SHAPE('',$,#6214);
#6217=CARTESIAN_POINT('',(50.0,0.0,90.0));
#6218=DIRECTION('',(0.0,-1.0,0.0));
#6219=DIRECTION('',(0.0,0.0,1.0));
#6220=AXIS2_PLACEMENT_3D('orientation',#6217,#6218,#6219);
#6221=REPRESENTATION('',(#2084,#2110),#29);
#6222=PROPERTY_DEFINITION_REPRESENTATION(#6215,#6221);
#6223=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6220),#29);
#6224=PROPERTY_DEFINITION_REPRESENTATION(#6215,#6223);
#6225=FEATURE_COMPONENT_DEFINITION('',$);
#6226=PRODUCT_DEFINITION_SHAPE('',$,#6225);
#6227=SQUARE_U_PROFILE('square u profile','boundary',#6226,.F.);
#6228=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#6229=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#6230=PRECISION_QUALIFIER(4);
#6236=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6228,#6229,#6230)
)REPRESENTATION_ITEM('width'));
#6241=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
#6246=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#6247=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6220,#6236,#6241,#6246),#29);
```

```
#6248=PROPERTY_DEFINITION('',$,#6227);
#6249=PROPERTY_DEFINITION_REPRESENTATION(#6248,#6247);
#6250=FEATURE_COMPONENT_DEFINITION('',$);
#6251=PRODUCT_DEFINITION_SHAPE('',$,#6250);
#6252=PATH_FEATURE_COMPONENT('linear path','linear',#6251,.F.);
#6253=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#6254=STANDARD_UNCERTAINTY('lower limit','minus',-1.0);
#6255=PRECISION_QUALIFIER(4);
#6261=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6253,#6254,#6255)
)REPRESENTATION_ITEM('distance'));
#6262=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6220,#6261),#29);
#6263=PROPERTY_DEFINITION('',$,#6252);
#6264=PROPERTY_DEFINITION_REPRESENTATION(#6263,#6262);
#6265=DIRECTION_SHAPE_REPRESENTATION('',(#6218),#29);
#6266=PROPERTY_DEFINITION_REPRESENTATION(#6263,#6265);
#6267=SHAPE_ASPECT('','open boundary occurrence',#6216,.T.);
#6268=SHAPE_DEFINING_RELATIONSHIP('','profile usage',#6227,#6267);
#6269=SHAPE_ASPECT('','pocket depth occurrence',#6216,.T.);
#6270=SHAPE_DEFINING_RELATIONSHIP('pocket depth','path feature component
usage',#6252,#6269);
#6271=FEATURE_COMPONENT_DEFINITION('',$);
#6272=PRODUCT_DEFINITION_SHAPE('',$,#6271);
#6273=POCKET_BOTTOM('bottom condition','planar',#6272,.F.);
#6274=DESCRIPTIVE_REPRESENTATION_ITEM('pocket bottom orientation','pocket
depth end');
#6275=PROPERTY_DEFINITION('',$,#6273);
#6276=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6274),#29);
#6277=PROPERTY_DEFINITION_REPRESENTATION(#6275,#6276);
#6278=DIRECTION('',(0.0,1.0,0.0));
#6279=DIRECTION_SHAPE_REPRESENTATION('floor normal',(#6278),#29);
#6280=PROPERTY_DEFINITION_REPRESENTATION(#6275,#6279);
#6281=CARTESIAN_POINT('',(50.0,-10.0,90.0));
#6282=LOCATION_SHAPE_REPRESENTATION('floor location',(#6281),#29);
#6283=PROPERTY_DEFINITION_REPRESENTATION(#6275,#6282);
#6284=SHAPE_ASPECT('','bottom condition occurrence',#6216,.T.);
#6285=FEATURE_COMPONENT_RELATIONSHIP('','pocket bottom usage',#6273,#6284);
#6286=DIMENSIONAL_LOCATION('',$,#75,#2111);
#6287=PRECISION_QUALIFIER(4);
#6293=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(81.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6287))REPRESENTAT
ION_ITEM(''));
#6294=SHAPE_DIMENSION_REPRESENTATION('',(#6293),#29);
#6295=PRECISION_QUALIFIER(4);
#6296=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#6297=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#6298=MEASURE_QUALIFICATION('','',#6296,(#6295));
#6299=MEASURE_QUALIFICATION('','',#6297,(#6295));
#6300=TOLERANCE_VALUE(#6297,#6296);
#6301=PLUS_MINUS_TOLERANCE(#6300,#6286);
#6302=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6286,#6294);
#6308=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('',$,#38,.T.)STEP());
#6309=PROPERTY_DEFINITION('',$,#6308);
#6310=PRODUCT_DEFINITION_SHAPE('',$,#6308);
#6311=CARTESIAN_POINT('',(228.0,-40.0,180.0));
```



```
#6312=DIRECTION('',(0.0,0.0,-1.0));
#6313=DIRECTION('',(0.707106781186547,-0.707106781186547,0.0));
#6314=AXIS2_PLACEMENT_3D('orientation',#6311,#6312,#6313);
#6315=REPRESENTATION('',(#1796,#2134,#2153),#29);
#6316=PROPERTY_DEFINITION_REPRESENTATION(#6309,#6315);
#6317=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6314),#29);
#6318=PROPERTY_DEFINITION_REPRESENTATION(#6309,#6317);
#6319=FEATURE_COMPONENT_DEFINITION('',$);
#6320=PRODUCT_DEFINITION_SHAPE('',$,#6319);
#6321=PATH_FEATURE_COMPONENT('linear path','linear',#6320,.F.);
#6322=STANDARD_UNCERTAINTY('upper limit','plus',0.300000000000000);
#6323=STANDARD_UNCERTAINTY('lower limit','minus',-0.300000000000000);
#6324=PRECISION_QUALIFIER(4);
#6330=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(180.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6322,#6323,#6324
))REPRESENTATION_ITEM('distance'));
#6331=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6314,#6330),#29);
#6332=PROPERTY_DEFINITION('',$,#6321);
#6333=PROPERTY_DEFINITION_REPRESENTATION(#6332,#6331);
#6334=DIRECTION_SHAPE_REPRESENTATION('',(#6312),#29);
#6335=PROPERTY_DEFINITION_REPRESENTATION(#6332,#6334);
#6336=FEATURE_COMPONENT_DEFINITION('',$);
#6337=PRODUCT_DEFINITION_SHAPE('',$,#6336);
#6338=VEE_PROFILE('vee profile','v-shape',#6337,.F.);
#6343=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#6348=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-
45.0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#6349=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6314,#6343,#6348),#29);
#6350=PROPERTY_DEFINITION('',$,#6338);
#6351=PROPERTY_DEFINITION_REPRESENTATION(#6350,#6349);
#6352=SHAPE_ASPECT('','removal boundary occurrence',#6310,.T.);
#6353=SHAPE_DEFINING_RELATIONSHIP('vee boundary','profile
usage',#6338,#6352);
#6354=SHAPE_ASPECT('','course of travel occurrence',#6310,.T.);
#6355=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#6321,#6354);
#6356=CARTESIAN_POINT('',(228.0,-40.0,180.0));
#6357=DIRECTION('',(0.0,1.0,0.0));
#6358=DIRECTION('',(1.0,0.0,0.0));
#6359=AXIS2_PLACEMENT_3D('',#6356,#6357,#6358);
#6360=SHAPE_ASPECT('',$,#38,.F.);
#6361=PROPERTY_DEFINITION('',$,#6360);
#6362=REPRESENTATION('',(#6359),#29);
#6363=PROPERTY_DEFINITION_REPRESENTATION(#6361,#6362);
#6364=DIMENSIONAL_LOCATION('',$,#335,#6360);
#6365=PRECISION_QUALIFIER(4);
#6371=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6365))REPRESENTAT
ION_ITEM(''));
#6372=SHAPE_DIMENSION_REPRESENTATION('',(#6371),#29);
#6373=PRECISION_QUALIFIER(4);
#6374=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#6375=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#6376=MEASURE_QUALIFICATION('','',#6374,(#6373));
#6377=MEASURE_QUALIFICATION('','',#6375,(#6373));
```

```
#6378=TOLERANCE_VALUE(#6375,#6374);
#6379=PLUS_MINUS_TOLERANCE(#6378,#6364);
#6380=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6364,#6372);
#6381=DIMENSIONAL_LOCATION('',$,#239,#2135);
#6382=PRECISION_QUALIFIER(4);
#6388=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(228.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6382))REPRESENTA
TION_ITEM(''));
#6389=SHAPE_DIMENSION_REPRESENTATION('',(#6388),#29);
#6390=PRECISION_QUALIFIER(4);
#6391=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#6392=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#6393=MEASURE_QUALIFICATION('','',#6391,(#6390));
#6394=MEASURE_QUALIFICATION('','',#6392,(#6390));
#6395=TOLERANCE_VALUE(#6392,#6391);
#6396=PLUS_MINUS_TOLERANCE(#6395,#6381);
#6397=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6381,#6389);
#6403=(CHARACTERIZED_OBJECT('V_LIN',$)FEATURE_DEFINITION()INSTANCED_FEATURE()
REMOVAL_VOLUME()SHAPE_ASPECT('V_LIN',$,#38,.T.));
#6404=PROPERTY_DEFINITION('',$,#6403);
#6405=PRODUCT_DEFINITION_SHAPE('',$,#6403);
#6406=CARTESIAN_POINT('',(0.0,0.0,0.0));
#6407=DIRECTION('',(0.0,0.0,1.0));
#6408=DIRECTION('',(1.0,0.0,0.0));
#6409=AXIS2_PLACEMENT_3D('orientation',#6406,#6407,#6408);
#6410=REPRESENTATION('',(#1796,#2177,#2196),#29);
#6411=PROPERTY_DEFINITION_REPRESENTATION(#6404,#6410);
#6412=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6409),#29);
#6413=PROPERTY_DEFINITION_REPRESENTATION(#6404,#6412);
#6414=SHAPE_ASPECT('','volume shape',#38,.T.);
#6415=PROPERTY_DEFINITION('',$,#6414);
#6416=PROPERTY_DEFINITION_REPRESENTATION(#6415,#6410);
#6417=SHAPE_ASPECT('','shape volume occurrence',#6405,.T.);
#6418=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#6414,#6417);
#6424=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('',$,#38,.T.)STEP());
#6425=PROPERTY_DEFINITION('',$,#6424);
#6426=PRODUCT_DEFINITION_SHAPE('',$,#6424);
#6427=CARTESIAN_POINT('',(2.0,-40.0,178.0));
#6428=DIRECTION('',(1.0,0.0,0.0));
#6429=DIRECTION('',(0.0,-0.707106781186547,0.707106781186547));
#6430=AXIS2_PLACEMENT_3D('orientation',#6427,#6428,#6429);
#6431=REPRESENTATION('',(#1796,#2220,#2239),#29);
#6432=PROPERTY_DEFINITION_REPRESENTATION(#6425,#6431);
#6433=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6430),#29);
#6434=PROPERTY_DEFINITION_REPRESENTATION(#6425,#6433);
#6435=FEATURE_COMPONENT_DEFINITION('',$);
#6436=PRODUCT_DEFINITION_SHAPE('',$,#6435);
#6437=PATH_FEATURE_COMPONENT('linear path','linear',#6436,.F.);
#6438=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#6439=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#6440=PRECISION_QUALIFIER(4);
#6446=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(230.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6438,#6439,#6440
))REPRESENTATION_ITEM('distance'));
#6447=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6430,#6446),#29);
```



```
#6448=PROPERTY_DEFINITION('',$,#6437);
#6449=PROPERTY_DEFINITION_REPRESENTATION(#6448,#6447);
#6450=DIRECTION_SHAPE_REPRESENTATION('',(#6428),#29);
#6451=PROPERTY_DEFINITION_REPRESENTATION(#6448,#6450);
#6452=FEATURE_COMPONENT_DEFINITION('',$);
#6453=PRODUCT_DEFINITION_SHAPE('',$,#6452);
#6454=VEE_PROFILE('vee profile','v-shape',#6453,.F.);
#6459=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#6464=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-
45.0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#6465=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6430,#6459,#6464),#29);
#6466=PROPERTY_DEFINITION('',$,#6454);
#6467=PROPERTY_DEFINITION_REPRESENTATION(#6466,#6465);
#6468=SHAPE_ASPECT('','removal boundary occurrence',#6426,.T.);
#6469=SHAPE_DEFINING_RELATIONSHIP('vee boundary','profile
usage',#6454,#6468);
#6470=SHAPE_ASPECT('','course of travel occurrence',#6426,.T.);
#6471=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#6437,#6470);
#6472=CARTESIAN_POINT('',(0.0,-40.0,178.0));
#6473=DIRECTION('',(0.0,1.0,0.0));
#6474=DIRECTION('',(0.0,0.0,1.0));
#6475=AXIS2_PLACEMENT_3D('',#6472,#6473,#6474);
#6476=SHAPE_ASPECT('',$,#38,.F.);
#6477=PROPERTY_DEFINITION('',$,#6476);
#6478=REPRESENTATION('',(#6475),#29);
#6479=PROPERTY_DEFINITION_REPRESENTATION(#6477,#6478);
#6480=DIMENSIONAL_LOCATION('',$,#335,#6476);
#6481=PRECISION_QUALIFIER(4);
#6487=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6481))REPRESENTAT
ION_ITEM(''));
#6488=SHAPE_DIMENSION_REPRESENTATION('',(#6487),#29);
#6489=PRECISION_QUALIFIER(4);
#6490=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#6491=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#6492=MEASURE_QUALIFICATION('','',#6490,(#6489));
#6493=MEASURE_QUALIFICATION('','',#6491,(#6489));
#6494=TOLERANCE_VALUE(#6491,#6490);
#6495=PLUS_MINUS_TOLERANCE(#6494,#6480);
#6496=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6480,#6488);
#6497=DIMENSIONAL_LOCATION('',$,#75,#2221);
#6498=PRECISION_QUALIFIER(4);
#6504=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(178.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6498))REPRESENTA
TION_ITEM(''));
#6505=SHAPE_DIMENSION_REPRESENTATION('',(#6504),#29);
#6506=PRECISION_QUALIFIER(4);
#6507=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.300000000000000),#10);
#6508=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.300000000000000),#10);
#6509=MEASURE_QUALIFICATION('','',#6507,(#6506));
#6510=MEASURE_QUALIFICATION('','',#6508,(#6506));
#6511=TOLERANCE_VALUE(#6508,#6507);
#6512=PLUS_MINUS_TOLERANCE(#6511,#6497);
#6513=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6497,#6505);
```

```
#6519=(CHARACTERIZED_OBJECT('V_LIN',$)FEATURE_DEFINITION()INSTANCED_FEATURE()
REMOVAL_VOLUME()SHAPE_ASPECT('V_LIN',$,#38,.T.));
#6520=PROPERTY_DEFINITION('',$,#6519);
#6521=PRODUCT_DEFINITION_SHAPE('',$,#6519);
#6522=CARTESIAN_POINT('',(0.0,0.0,0.0));
#6523=DIRECTION('',(0.0,0.0,1.0));
#6524=DIRECTION('',(1.0,0.0,0.0));
#6525=AXIS2_PLACEMENT_3D('orientation',#6522,#6523,#6524);
#6526=REPRESENTATION('',(#1796,#2263,#2282),#29);
#6527=PROPERTY_DEFINITION_REPRESENTATION(#6520,#6526);
#6528=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6525),#29);
#6529=PROPERTY_DEFINITION_REPRESENTATION(#6520,#6528);
#6530=SHAPE_ASPECT('','volume shape',#38,.T.);
#6531=PROPERTY_DEFINITION('',$,#6530);
#6532=PROPERTY_DEFINITION_REPRESENTATION(#6531,#6526);
#6533=SHAPE_ASPECT('','shape volume occurrence',#6521,.T.);
#6534=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#6530,#6533);
#6540=(CHARACTERIZED_OBJECT('','groove')FEATURE_DEFINITION()INSTANCED_FEATURE
()REVOLVED_PROFILE()SHAPE_ASPECT('','groove',#38,.T.));
#6541=PROPERTY_DEFINITION('',$,#6540);
#6542=PRODUCT_DEFINITION_SHAPE('',$,#6540);
#6543=CARTESIAN_POINT('',(160.0,-20.0,90.0));
#6544=DIRECTION('',(0.0,-1.0,0.0));
#6545=DIRECTION('',(1.0,0.0,0.0));
#6546=AXIS2_PLACEMENT_3D('orientation',#6543,#6544,#6545);
#6547=STANDARD_UNCERTAINTY('upper limit','plus',0.050000000000000);
#6548=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#6549=PRECISION_QUALIFIER(4);
#6555=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(22.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6547,#6548,#6549)
)REPRESENTATION_ITEM('radius'));
#6556=REPRESENTATION('',(#2316,#2364,#2388,#2404),#29);
#6557=PROPERTY_DEFINITION_REPRESENTATION(#6541,#6556);
#6558=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6546,#6555),#29);
#6559=PROPERTY_DEFINITION_REPRESENTATION(#6541,#6558);
#6560=DIRECTION('',(-1.0,0.0,0.0));
#6561=DIRECTION_SHAPE_REPRESENTATION('removal direction',(#6560),#29);
#6562=PROPERTY_DEFINITION_REPRESENTATION(#6541,#6561);
#6563=CARTESIAN_POINT('',(182.0,-20.0,90.0));
#6564=DIRECTION('',(0.0,0.0,-1.0));
#6565=DIRECTION('',(0.0,-1.0,0.0));
#6566=AXIS2_PLACEMENT_3D('orientation',#6563,#6564,#6565);
#6567=FEATURE_COMPONENT_DEFINITION('',$);
#6568=PRODUCT_DEFINITION_SHAPE('',$,#6567);
#6569=SQUARE_U_PROFILE('','sweep',#6568,.F.);
#6570=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#6571=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#6572=PRECISION_QUALIFIER(4);
#6578=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(3.500000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#6570
,#6571,#6572))REPRESENTATION_ITEM('width'));
#6583=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
#6588=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#6589=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6566,#6578,#6583,#6588),#29);
```



```
#6590=PROPERTY_DEFINITION('',$,#6569);
#6591=PROPERTY_DEFINITION_REPRESENTATION(#6590,#6589);
#6592=SHAPE_ASPECT('','sweep occurrence',#6542,.T.);
#6593=SHAPE_DEFINING_RELATIONSHIP('','profile usage',#6569,#6592);
#6594=DIMENSIONAL_LOCATION('',$,#1110,#2317);
#6595=PRECISION_QUALIFIER(4);
#6601=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.2500000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#659
5))REPRESENTATION_ITEM(''));
#6602=SHAPE_DIMENSION_REPRESENTATION('',(#6601),#29);
#6603=PRECISION_QUALIFIER(4);
#6604=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.050000000000000),#10);
#6605=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.050000000000000),#10);
#6606=MEASURE_QUALIFICATION('','',#6604,(#6603));
#6607=MEASURE_QUALIFICATION('','',#6605,(#6603));
#6608=TOLERANCE_VALUE(#6605,#6604);
#6609=PLUS_MINUS_TOLERANCE(#6608,#6594);
#6610=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6594,#6602);
#6617=(CHARACTERIZED_OBJECT('compound feature in
solid','counterbore')COMPOSITE_HOLE()
  COMPOUND_FEATURE()FEATURE_DEFINITION()INSTANCED_FEATURE()
  SHAPE_ASPECT('compound feature in solid','counterbore',#38,.T.));
#6618=PROPERTY_DEFINITION('',$,#6617);
#6619=PRODUCT_DEFINITION_SHAPE('',$,#6617);
#6620=CARTESIAN_POINT('',(160.0,0.0,30.0));
#6621=DIRECTION('',(0.0,-1.0,0.0));
#6622=DIRECTION('',(-1.0,0.0,0.0));
#6623=AXIS2_PLACEMENT_3D('orientation',#6620,#6621,#6622);
#6624=REPRESENTATION('',(#2434,#2468,#2508,#2532,#2548,#2564),#29);
#6625=PROPERTY_DEFINITION_REPRESENTATION(#6618,#6624);
#6626=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6623),#29);
#6627=PROPERTY_DEFINITION_REPRESENTATION(#6618,#6626);

#6633=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#6634=PROPERTY_DEFINITION('',$,#6633);
#6635=PRODUCT_DEFINITION_SHAPE('',$,#6633);
#6636=CARTESIAN_POINT('',(160.0,0.0,30.0));
#6637=DIRECTION('',(0.0,-1.0,0.0));
#6638=DIRECTION('',(-1.0,0.0,0.0));
#6639=AXIS2_PLACEMENT_3D('orientation',#6636,#6637,#6638);
#6640=REPRESENTATION('',(#2434,#2468,#2548),#29);
#6641=PROPERTY_DEFINITION_REPRESENTATION(#6634,#6640);
#6642=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6639),#29);
#6643=PROPERTY_DEFINITION_REPRESENTATION(#6634,#6642);
#6644=FEATURE_COMPONENT_DEFINITION('',$);
#6645=PRODUCT_DEFINITION_SHAPE('',$,#6644);
#6646=CIRCULAR_CLOSED_PROFILE('circular profile',$,#6645,.F.);
#6647=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#6648=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#6649=PRECISION_QUALIFIER(4);
#6655=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0),#10)

QUALIFIED_REPRESENTATION_ITEM((#6647,#6648,#6649))REPRESENTATION_ITEM('diamet
er'));
```

```
#6656=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6639,#6655),#29);
#6657=PROPERTY_DEFINITION('',$,#6646);
#6658=PROPERTY_DEFINITION_REPRESENTATION(#6657,#6656);
#6659=FEATURE_COMPONENT_DEFINITION('',$);
#6660=PRODUCT_DEFINITION_SHAPE('',$,#6659);
#6661=PATH_FEATURE_COMPONENT('linear path','linear',#6660,.F.);
#6662=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#6663=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#6664=PRECISION_QUALIFIER(4);
#6670=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0),#10)

QUALIFIED_REPRESENTATION_ITEM((#6662,#6663,#6664))REPRESENTATION_ITEM('distan
ce'));
#6671=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6639,#6670),#29);
#6672=PROPERTY_DEFINITION('',$,#6661);
#6673=PROPERTY_DEFINITION_REPRESENTATION(#6672,#6671);
#6674=DIRECTION_SHAPE_REPRESENTATION('',(#6637),#29);
#6675=PROPERTY_DEFINITION_REPRESENTATION(#6672,#6674);
#6676=SHAPE_ASPECT('','diameter occurrence',#6635,.T.);
#6677=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#6646,#6676);
#6678=SHAPE_ASPECT('','hole depth occurrence',#6635,.T.);
#6679=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#6661,#6678);

#6680=FEATURE_COMPONENT_DEFINITION('',$);
#6681=PRODUCT_DEFINITION_SHAPE('',$,#6680);
#6682=HOLE_BOTTOM('bottom condition','flat',#6681,.F.);
#6687=SHAPE_ASPECT('','bottom condition occurrence',#6635,.T.);
#6688=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#6682,#6687);

#6689=COMPOSITE_SHAPE_ASPECT('compound feature in solid',$,#6619,.T.);
#6695=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#6696=PROPERTY_DEFINITION('',$,#6695);
#6697=PRODUCT_DEFINITION_SHAPE('',$,#6695);
#6698=CARTESIAN_POINT('',(160.0,0.0,30.0));
#6699=DIRECTION('',(0.0,-1.0,0.0));
#6700=DIRECTION('',(-1.0,0.0,0.0));
#6701=AXIS2_PLACEMENT_3D('orientation',#6698,#6699,#6700);
#6702=REPRESENTATION('',(#2508,#2532,#2564),#29);
#6703=PROPERTY_DEFINITION_REPRESENTATION(#6696,#6702);
#6704=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6701),#29);
#6705=PROPERTY_DEFINITION_REPRESENTATION(#6696,#6704);
#6706=FEATURE_COMPONENT_DEFINITION('',$);
#6707=PRODUCT_DEFINITION_SHAPE('',$,#6706);
#6708=CIRCULAR_CLOSED_PROFILE('circular profile',$,#6707,.F.);
#6709=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#6710=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#6711=PRECISION_QUALIFIER(4);
#6717=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6709,#6710,#6711)
)REPRESENTATION_ITEM('diameter'));
#6718=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6701,#6717),#29);
#6719=PROPERTY_DEFINITION('',$,#6708);
```



```
#6720=PROPERTY_DEFINITION_REPRESENTATION(#6719,#6718);
#6721=FEATURE_COMPONENT_DEFINITION('',$);
#6722=PRODUCT_DEFINITION_SHAPE('',$,#6721);
#6723=PATH_FEATURE_COMPONENT('linear path','linear',#6722,.F.);
#6724=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#6725=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#6726=PRECISION_QUALIFIER(4);
#6732=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6724,#6725,#6726)
)REPRESENTATION_ITEM('distance'));
#6733=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6701,#6732),#29);
#6734=PROPERTY_DEFINITION('',$,#6723);
#6735=PROPERTY_DEFINITION_REPRESENTATION(#6734,#6733);
#6736=DIRECTION_SHAPE_REPRESENTATION('',(#6699),#29);
#6737=PROPERTY_DEFINITION_REPRESENTATION(#6734,#6736);
#6738=SHAPE_ASPECT('','diameter occurrence',#6697,.T.);
#6739=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#6708,#6738);
#6740=SHAPE_ASPECT('','hole depth occurrence',#6697,.T.);
#6741=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#6723,#6740);

#6742=FEATURE_COMPONENT_DEFINITION('',$);
#6743=PRODUCT_DEFINITION_SHAPE('',$,#6742);
#6744=HOLE_BOTTOM('bottom condition','flat',#6743,.F.);
#6749=SHAPE_ASPECT('','bottom condition occurrence',#6697,.T.);

#6750=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#6744,#6749);
#6751=FEATURE_COMPONENT_RELATIONSHIP('large hole',$,#6689,#6695);
#6752=FEATURE_COMPONENT_RELATIONSHIP('small hole',$,#6689,#6633);
#6753=CARTESIAN_POINT('',(160.0,-30.0,30.0));
#6754=DIRECTION('',(0.0,1.0,0.0));
#6755=AXIS1_PLACEMENT('',#6753,#6754);
#6756=SHAPE_ASPECT('',$,#38,.F.);
#6757=PROPERTY_DEFINITION('',$,#6756);
#6758=REPRESENTATION('',(#6755),#29);
#6759=PROPERTY_DEFINITION_REPRESENTATION(#6757,#6758);
#6760=DIMENSIONAL_LOCATION('',$,#1261,#6756);
#6761=PRECISION_QUALIFIER(4);
#6767=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(70.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6761))REPRESENTAT
ION_ITEM(''));
#6768=SHAPE_DIMENSION_REPRESENTATION('',(#6767),#29);
#6769=PRECISION_QUALIFIER(4);
#6770=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#6771=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.250000000000000),#10);
#6772=MEASURE_QUALIFICATION('','',#6770,(#6769));
#6773=MEASURE_QUALIFICATION('','',#6771,(#6769));
#6774=TOLERANCE_VALUE(#6771,#6770);
#6775=PLUS_MINUS_TOLERANCE(#6774,#6760);
#6776=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6760,#6768);
#6777=CARTESIAN_POINT('',(160.0,-30.0,30.0));
#6778=DIRECTION('',(0.0,1.0,0.0));
#6779=AXIS1_PLACEMENT('',#6777,#6778);
#6780=SHAPE_ASPECT('',$,#38,.F.);
#6781=PROPERTY_DEFINITION('',$,#6780);
```

```
#6782=REPRESENTATION('',(#6779),#29);
#6783=PROPERTY_DEFINITION_REPRESENTATION(#6781,#6782);
#6784=DIMENSIONAL_LOCATION('',$,#75,#6780);
#6785=PRECISION_QUALIFIER(4);
#6791=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6785))REPRESENTAT
ION_ITEM(''));
#6792=SHAPE_DIMENSION_REPRESENTATION('',(#6791),#29);
#6793=PRECISION_QUALIFIER(4);
#6794=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#6795=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.250000000000000),#10);
#6796=MEASURE_QUALIFICATION('','',#6794,(#6793));
#6797=MEASURE_QUALIFICATION('','',#6795,(#6793));
#6798=TOLERANCE_VALUE(#6795,#6794);
#6799=PLUS_MINUS_TOLERANCE(#6798,#6784);
#6800=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6784,#6792);
#6807=(CHARACTERIZED_OBJECT('compound feature in
solid','counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid','counterbore',#38,.T.));
#6808=PROPERTY_DEFINITION('',$,#6807);
#6809=PRODUCT_DEFINITION_SHAPE('',$,#6807);
#6810=CARTESIAN_POINT('',(160.0,0.0,150.0));
#6811=DIRECTION('',(0.0,-1.0,0.0));
#6812=DIRECTION('',(-1.0,0.0,0.0));
#6813=AXIS2_PLACEMENT_3D('orientation',#6810,#6811,#6812);
#6814=REPRESENTATION('',(#2598,#2638,#2662,#2678,#2694),#29);
#6815=PROPERTY_DEFINITION_REPRESENTATION(#6808,#6814);
#6816=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6813),#29);
#6817=PROPERTY_DEFINITION_REPRESENTATION(#6808,#6816);
#6823=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#6824=PROPERTY_DEFINITION('',$,#6823);
#6825=PRODUCT_DEFINITION_SHAPE('',$,#6823);
#6826=CARTESIAN_POINT('',(160.0,0.0,150.0));
#6827=DIRECTION('',(0.0,-1.0,0.0));
#6828=DIRECTION('',(-1.0,0.0,0.0));
#6829=AXIS2_PLACEMENT_3D('orientation',#6826,#6827,#6828);
#6830=REPRESENTATION('',(#2598,#2678),#29);
#6831=PROPERTY_DEFINITION_REPRESENTATION(#6824,#6830);
#6832=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6829),#29);
#6833=PROPERTY_DEFINITION_REPRESENTATION(#6824,#6832);
#6834=FEATURE_COMPONENT_DEFINITION('',$);
#6835=PRODUCT_DEFINITION_SHAPE('',$,#6834);
#6836=CIRCULAR_CLOSED_PROFILE('circular profile',$,#6835,.F.);
#6837=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#6838=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#6839=PRECISION_QUALIFIER(4);
#6845=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6837,#6838,#6839)
)REPRESENTATION_ITEM('diameter'));
#6846=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6829,#6845),#29);
#6847=PROPERTY_DEFINITION('',$,#6836);
#6848=PROPERTY_DEFINITION_REPRESENTATION(#6847,#6846);
#6849=FEATURE_COMPONENT_DEFINITION('',$);
#6850=PRODUCT_DEFINITION_SHAPE('',$,#6849);
```



```
#6851=PATH_FEATURE_COMPONENT('linear path','linear',#6850,.F.);
#6852=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#6853=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#6854=PRECISION_QUALIFIER(4);
#6860=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(40.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6852,#6853,#6854)
)REPRESENTATION_ITEM('distance'));
#6861=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6829,#6860),#29);
#6862=PROPERTY_DEFINITION('',$,#6851);
#6863=PROPERTY_DEFINITION_REPRESENTATION(#6862,#6861);
#6864=DIRECTION_SHAPE_REPRESENTATION('',(#6827),#29);
#6865=PROPERTY_DEFINITION_REPRESENTATION(#6862,#6864);
#6866=SHAPE_ASPECT('','diameter occurrence',#6825,.T.);
#6867=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#6836,#6866);
#6868=SHAPE_ASPECT('','hole depth occurrence',#6825,.T.);
#6869=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#6851,#6868);

#6870=FEATURE_COMPONENT_DEFINITION('',$);
#6871=PRODUCT_DEFINITION_SHAPE('',$,#6870);
#6872=HOLE_BOTTOM('bottom condition','flat',#6871,.F.);
#6877=SHAPE_ASPECT('','bottom condition occurrence',#6825,.T.);

#6878=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#6872,#6877);
#6879=COMPOSITE_SHAPE_ASPECT('compound feature in solid',$,#6809,.T.);
#6885=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#6886=PROPERTY_DEFINITION('',$,#6885);
#6887=PRODUCT_DEFINITION_SHAPE('',$,#6885);
#6888=CARTESIAN_POINT('',(160.0,0.0,150.0));
#6889=DIRECTION('',(0.0,-1.0,0.0));
#6890=DIRECTION('',(-1.0,0.0,0.0));
#6891=AXIS2_PLACEMENT_3D('orientation',#6888,#6889,#6890);
#6892=REPRESENTATION('',(#2638,#2662,#2694),#29);
#6893=PROPERTY_DEFINITION_REPRESENTATION(#6886,#6892);
#6894=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6891),#29);
#6895=PROPERTY_DEFINITION_REPRESENTATION(#6886,#6894);
#6896=FEATURE_COMPONENT_DEFINITION('',$);
#6897=PRODUCT_DEFINITION_SHAPE('',$,#6896);
#6898=CIRCULAR_CLOSED_PROFILE('circular profile',$,#6897,.F.);
#6899=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#6900=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#6901=PRECISION_QUALIFIER(4);
#6907=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6899,#6900,#6901)
)REPRESENTATION_ITEM('diameter'));
#6908=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6891,#6907),#29);
#6909=PROPERTY_DEFINITION('',$,#6898);
#6910=PROPERTY_DEFINITION_REPRESENTATION(#6909,#6908);
#6911=FEATURE_COMPONENT_DEFINITION('',$);
#6912=PRODUCT_DEFINITION_SHAPE('',$,#6911);
#6913=PATH_FEATURE_COMPONENT('linear path','linear',#6912,.F.);
#6914=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#6915=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#6916=PRECISION_QUALIFIER(4);
```

```
#6922=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6914,#6915,#6916)
)REPRESENTATION_ITEM('distance'));
#6923=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6891,#6922),#29);
#6924=PROPERTY_DEFINITION('',$,#6913);
#6925=PROPERTY_DEFINITION_REPRESENTATION(#6924,#6923);
#6926=DIRECTION_SHAPE_REPRESENTATION('',(#6889),#29);
#6927=PROPERTY_DEFINITION_REPRESENTATION(#6924,#6926);
#6928=SHAPE_ASPECT('','diameter occurrence',#6887,.T.);
#6929=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#6898,#6928);
#6930=SHAPE_ASPECT('','hole depth occurrence',#6887,.T.);
#6931=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#6913,#6930);

#6932=FEATURE_COMPONENT_DEFINITION('',$);
#6933=PRODUCT_DEFINITION_SHAPE('',$,#6932);
#6934=HOLE_BOTTOM('bottom condition','flat',#6933,.F.);
#6939=SHAPE_ASPECT('','bottom condition occurrence',#6887,.T.);

#6940=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#6934,#6939);
#6941=FEATURE_COMPONENT_RELATIONSHIP('large hole',$,#6879,#6885);
#6942=FEATURE_COMPONENT_RELATIONSHIP('small hole',$,#6879,#6823);
#6943=CARTESIAN_POINT('',(160.0,-40.0,150.0));
#6944=DIRECTION('',(0.0,1.0,0.0));
#6945=AXIS1_PLACEMENT('',#6943,#6944);
#6946=SHAPE_ASPECT('',$,#38,.F.);
#6947=PROPERTY_DEFINITION('',$,#6946);
#6948=REPRESENTATION('',(#6945),#29);
#6949=PROPERTY_DEFINITION_REPRESENTATION(#6947,#6948);
#6950=DIMENSIONAL_LOCATION('',$,#1261,#6946);
#6951=PRECISION_QUALIFIER(4);
#6957=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(70.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6951))REPRESENTAT
ION_ITEM(''));
#6958=SHAPE_DIMENSION_REPRESENTATION('',(#6957),#29);
#6959=PRECISION_QUALIFIER(4);
#6960=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#6961=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.250000000000000),#10);
#6962=MEASURE_QUALIFICATION('','',#6960,(#6959));
#6963=MEASURE_QUALIFICATION('','',#6961,(#6959));
#6964=TOLERANCE_VALUE(#6961,#6960);
#6965=PLUS_MINUS_TOLERANCE(#6964,#6950);
#6966=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6950,#6958);
#6967=CARTESIAN_POINT('',(160.0,-40.0,150.0));
#6968=DIRECTION('',(0.0,1.0,0.0));
#6969=AXIS1_PLACEMENT('',#6967,#6968);
#6970=SHAPE_ASPECT('',$,#38,.F.);
#6971=PROPERTY_DEFINITION('',$,#6970);
#6972=REPRESENTATION('',(#6969),#29);
#6973=PROPERTY_DEFINITION_REPRESENTATION(#6971,#6972);
#6974=DIMENSIONAL_LOCATION('',$,#143,#6970);
#6975=PRECISION_QUALIFIER(4);
#6981=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6975))REPRESENTAT
ION_ITEM(''));
```



```
#6982=SHAPE_DIMENSION_REPRESENTATION('',(#6981),#29);
#6983=PRECISION_QUALIFIER(4);
#6984=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#6985=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.250000000000000),#10);
#6986=MEASURE_QUALIFICATION('','',#6984,(#6983));
#6987=MEASURE_QUALIFICATION('','',#6985,(#6983));
#6988=TOLERANCE_VALUE(#6985,#6984);
#6989=PLUS_MINUS_TOLERANCE(#6988,#6974);
#6990=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6974,#6982);
#6997=(CHARACTERIZED_OBJECT('compound feature in
solid','counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid','counterbore',#38,.T.));
#6998=PROPERTY_DEFINITION('',$,#6997);
#6999=PRODUCT_DEFINITION_SHAPE('',$,#6997);
#7000=CARTESIAN_POINT('',(205.0,0.0,27.500000000000000));
#7001=DIRECTION('',(0.0,-1.0,0.0));
#7002=DIRECTION('',(-1.0,0.0,0.0));
#7003=AXIS2_PLACEMENT_3D('orientation',#7000,#7001,#7002);
#7004=REPRESENTATION('',(#2728,#2768,#2792,#2808,#2824),#29);
#7005=PROPERTY_DEFINITION_REPRESENTATION(#6998,#7004);
#7006=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7003),#29);
#7007=PROPERTY_DEFINITION_REPRESENTATION(#6998,#7006);
#7013=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7014=PROPERTY_DEFINITION('',$,#7013);
#7015=PRODUCT_DEFINITION_SHAPE('',$,#7013);
#7016=CARTESIAN_POINT('',(205.0,0.0,27.500000000000000));
#7017=DIRECTION('',(0.0,-1.0,0.0));
#7018=DIRECTION('',(-1.0,0.0,0.0));
#7019=AXIS2_PLACEMENT_3D('orientation',#7016,#7017,#7018);
#7020=REPRESENTATION('',(#2728,#2808),#29);
#7021=PROPERTY_DEFINITION_REPRESENTATION(#7014,#7020);
#7022=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7019),#29);
#7023=PROPERTY_DEFINITION_REPRESENTATION(#7014,#7022);
#7024=FEATURE_COMPONENT_DEFINITION('',$);
#7025=PRODUCT_DEFINITION_SHAPE('',$,#7024);
#7026=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7025,.F.);
#7027=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#7028=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#7029=PRECISION_QUALIFIER(4);
#7035=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7027,#7028,#7029)
)REPRESENTATION_ITEM('diameter'));
#7036=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7019,#7035),#29);
#7037=PROPERTY_DEFINITION('',$,#7026);
#7038=PROPERTY_DEFINITION_REPRESENTATION(#7037,#7036);
#7039=FEATURE_COMPONENT_DEFINITION('',$);
#7040=PRODUCT_DEFINITION_SHAPE('',$,#7039);
#7041=PATH_FEATURE_COMPONENT('linear path','linear',#7040,.F.);
#7042=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#7043=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#7044=PRECISION_QUALIFIER(4);
#7050=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7042,#7043,#7044)
)REPRESENTATION_ITEM('distance'));
```

```
#7051=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7019,#7050),#29);
#7052=PROPERTY_DEFINITION('',$,#7041);
#7053=PROPERTY_DEFINITION_REPRESENTATION(#7052,#7051);
#7054=DIRECTION_SHAPE_REPRESENTATION('',(#7017),#29);
#7055=PROPERTY_DEFINITION_REPRESENTATION(#7052,#7054);
#7056=SHAPE_ASPECT('','diameter occurrence',#7015,.T.);
#7057=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7026,#7056);
#7058=SHAPE_ASPECT('','hole depth occurrence',#7015,.T.);
#7059=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7041,#7058);

#7060=FEATURE_COMPONENT_DEFINITION('',$);
#7061=PRODUCT_DEFINITION_SHAPE('',$,#7060);
#7062=HOLE_BOTTOM('bottom condition','flat',#7061,.F.);
#7067=SHAPE_ASPECT('','bottom condition occurrence',#7015,.T.);

#7068=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7062,#7067);
#7069=COMPOSITE_SHAPE_ASPECT('compound feature in solid',$,#6999,.T.);
#7075=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7076=PROPERTY_DEFINITION('',$,#7075);
#7077=PRODUCT_DEFINITION_SHAPE('',$,#7075);
#7078=CARTESIAN_POINT('',(205.0,0.0,27.5000000000000));
#7079=DIRECTION('',(0.0,-1.0,0.0));
#7080=DIRECTION('',(-1.0,0.0,0.0));
#7081=AXIS2_PLACEMENT_3D('orientation',#7078,#7079,#7080);
#7082=REPRESENTATION('',(#2768,#2792,#2824),#29);
#7083=PROPERTY_DEFINITION_REPRESENTATION(#7076,#7082);
#7084=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7081),#29);
#7085=PROPERTY_DEFINITION_REPRESENTATION(#7076,#7084);
#7086=FEATURE_COMPONENT_DEFINITION('',$);
#7087=PRODUCT_DEFINITION_SHAPE('',$,#7086);
#7088=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7087,.F.);
#7089=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#7090=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#7091=PRECISION_QUALIFIER(4);
#7097=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(26.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7089,#7090,#7091)
)REPRESENTATION_ITEM('diameter'));
#7098=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7081,#7097),#29);
#7099=PROPERTY_DEFINITION('',$,#7088);
#7100=PROPERTY_DEFINITION_REPRESENTATION(#7099,#7098);
#7101=FEATURE_COMPONENT_DEFINITION('',$);
#7102=PRODUCT_DEFINITION_SHAPE('',$,#7101);
#7103=PATH_FEATURE_COMPONENT('linear path','linear',#7102,.F.);
#7104=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#7105=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#7106=PRECISION_QUALIFIER(4);
#7112=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7104,#7105,#7106)
)REPRESENTATION_ITEM('distance'));
#7113=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7081,#7112),#29);
#7114=PROPERTY_DEFINITION('',$,#7103);
#7115=PROPERTY_DEFINITION_REPRESENTATION(#7114,#7113);
#7116=DIRECTION_SHAPE_REPRESENTATION('',(#7079),#29);
```



```
#7117=PROPERTY_DEFINITION_REPRESENTATION(#7114,#7116);
#7118=SHAPE_ASPECT('','diameter occurrence',#7077,.T.);
#7119=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7088,#7118);
#7120=SHAPE_ASPECT('','hole depth occurrence',#7077,.T.);
#7121=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7103,#7120);

#7122=FEATURE_COMPONENT_DEFINITION('',$);
#7123=PRODUCT_DEFINITION_SHAPE('',$,#7122);
#7124=HOLE_BOTTOM('bottom condition','flat',#7123,.F.);
#7129=SHAPE_ASPECT('','bottom condition occurrence',#7077,.T.);

#7130=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7124,#7129);
#7131=FEATURE_COMPONENT_RELATIONSHIP('large hole',$,#7069,#7075);
#7132=FEATURE_COMPONENT_RELATIONSHIP('small hole',$,#7069,#7013);
#7133=CARTESIAN_POINT('',(205.0,-50.0,27.500000000000000));
#7134=DIRECTION('',(0.0,1.0,0.0));
#7135=AXIS1_PLACEMENT('',#7133,#7134);
#7136=SHAPE_ASPECT('',$,#38,.F.);
#7137=PROPERTY_DEFINITION('',$,#7136);
#7138=REPRESENTATION('',(#7135),#29);
#7139=PROPERTY_DEFINITION_REPRESENTATION(#7137,#7138);
#7140=DIMENSIONAL_LOCATION('',$,#1261,#7136);
#7141=PRECISION_QUALIFIER(4);
#7147=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7141))REPRESENTAT
ION_ITEM(''));
#7148=SHAPE_DIMENSION_REPRESENTATION('',(#7147),#29);
#7149=PRECISION_QUALIFIER(4);
#7150=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7151=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7152=MEASURE_QUALIFICATION('','',#7150,(#7149));
#7153=MEASURE_QUALIFICATION('','',#7151,(#7149));
#7154=TOLERANCE_VALUE(#7151,#7150);
#7155=PLUS_MINUS_TOLERANCE(#7154,#7140);
#7156=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7140,#7148);
#7157=CARTESIAN_POINT('',(205.0,-50.0,27.500000000000000));
#7158=DIRECTION('',(0.0,1.0,0.0));
#7159=AXIS1_PLACEMENT('',#7157,#7158);
#7160=SHAPE_ASPECT('',$,#38,.F.);
#7161=PROPERTY_DEFINITION('',$,#7160);
#7162=REPRESENTATION('',(#7159),#29);
#7163=PROPERTY_DEFINITION_REPRESENTATION(#7161,#7162);
#7164=DIMENSIONAL_LOCATION('',$,#75,#7160);
#7165=PRECISION_QUALIFIER(4);
#7171=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(27.500000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#716
5))REPRESENTATION_ITEM(''));
#7172=SHAPE_DIMENSION_REPRESENTATION('',(#7171),#29);
#7173=PRECISION_QUALIFIER(4);
#7174=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7175=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7176=MEASURE_QUALIFICATION('','',#7174,(#7173));
#7177=MEASURE_QUALIFICATION('','',#7175,(#7173));
#7178=TOLERANCE_VALUE(#7175,#7174);
```

```
#7179=PLUS_MINUS_TOLERANCE(#7178,#7164);
#7180=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7164,#7172);
#7187=(CHARACTERIZED_OBJECT('compound feature in
solid','counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid','counterbore',#38,.T.));
#7188=PROPERTY_DEFINITION('',$,#7187);
#7189=PRODUCT_DEFINITION_SHAPE('',$,#7187);
#7190=CARTESIAN_POINT('',(205.0,0.0,152.500000000000000));
#7191=DIRECTION('',(0.0,-1.0,0.0));
#7192=DIRECTION('',(-1.0,0.0,0.0));
#7193=AXIS2_PLACEMENT_3D('orientation',#7190,#7191,#7192);
#7194=REPRESENTATION('',(#2858,#2898,#2922,#2938,#2954),#29);
#7195=PROPERTY_DEFINITION_REPRESENTATION(#7188,#7194);
#7196=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7193),#29);
#7197=PROPERTY_DEFINITION_REPRESENTATION(#7188,#7196);
#7203=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7204=PROPERTY_DEFINITION('',$,#7203);
#7205=PRODUCT_DEFINITION_SHAPE('',$,#7203);
#7206=CARTESIAN_POINT('',(205.0,0.0,152.500000000000000));
#7207=DIRECTION('',(0.0,-1.0,0.0));
#7208=DIRECTION('',(-1.0,0.0,0.0));
#7209=AXIS2_PLACEMENT_3D('orientation',#7206,#7207,#7208);
#7210=REPRESENTATION('',(#2858,#2938),#29);
#7211=PROPERTY_DEFINITION_REPRESENTATION(#7204,#7210);
#7212=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7209),#29);
#7213=PROPERTY_DEFINITION_REPRESENTATION(#7204,#7212);
#7214=FEATURE_COMPONENT_DEFINITION('',$);
#7215=PRODUCT_DEFINITION_SHAPE('',$,#7214);
#7216=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7215,.F.);
#7217=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#7218=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#7219=PRECISION_QUALIFIER(4);
#7225=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7217,#7218,#7219)
)REPRESENTATION_ITEM('diameter'));
#7226=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7209,#7225),#29);
#7227=PROPERTY_DEFINITION('',$,#7216);
#7228=PROPERTY_DEFINITION_REPRESENTATION(#7227,#7226);
#7229=FEATURE_COMPONENT_DEFINITION('',$);
#7230=PRODUCT_DEFINITION_SHAPE('',$,#7229);
#7231=PATH_FEATURE_COMPONENT('linear path','linear',#7230,.F.);
#7232=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#7233=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#7234=PRECISION_QUALIFIER(4);
#7240=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7232,#7233,#7234)
)REPRESENTATION_ITEM('distance'));
#7241=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7209,#7240),#29);
#7242=PROPERTY_DEFINITION('',$,#7231);
#7243=PROPERTY_DEFINITION_REPRESENTATION(#7242,#7241);
#7244=DIRECTION_SHAPE_REPRESENTATION('',(#7207),#29);
#7245=PROPERTY_DEFINITION_REPRESENTATION(#7242,#7244);
#7246=SHAPE_ASPECT('','diameter occurrence',#7205,.T.);
#7247=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7216,#7246);
```



```
#7248=SHAPE_ASPECT('','hole depth occurrence',#7205,.T.);
#7249=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7231,#7248);

#7250=FEATURE_COMPONENT_DEFINITION('',$);
#7251=PRODUCT_DEFINITION_SHAPE('',$,#7250);
#7252=HOLE_BOTTOM('bottom condition','flat',#7251,.F.);
#7257=SHAPE_ASPECT('','bottom condition occurrence',#7205,.T.);

#7258=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7252,#7257);
#7259=COMPOSITE_SHAPE_ASPECT('compound feature in solid',$,#7189,.T.);
#7265=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7266=PROPERTY_DEFINITION('',$,#7265);
#7267=PRODUCT_DEFINITION_SHAPE('',$,#7265);
#7268=CARTESIAN_POINT('',(205.0,0.0,152.500000000000000));
#7269=DIRECTION('',(0.0,-1.0,0.0));
#7270=DIRECTION('',(-1.0,0.0,0.0));
#7271=AXIS2_PLACEMENT_3D('orientation',#7268,#7269,#7270);
#7272=REPRESENTATION('',(#2898,#2922,#2954),#29);
#7273=PROPERTY_DEFINITION_REPRESENTATION(#7266,#7272);
#7274=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7271),#29);
#7275=PROPERTY_DEFINITION_REPRESENTATION(#7266,#7274);
#7276=FEATURE_COMPONENT_DEFINITION('',$);
#7277=PRODUCT_DEFINITION_SHAPE('',$,#7276);
#7278=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7277,.F.);
#7279=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#7280=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#7281=PRECISION_QUALIFIER(4);
#7287=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(26.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7279,#7280,#7281)
)REPRESENTATION_ITEM('diameter'));
#7288=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7271,#7287),#29);
#7289=PROPERTY_DEFINITION('',$,#7278);
#7290=PROPERTY_DEFINITION_REPRESENTATION(#7289,#7288);
#7291=FEATURE_COMPONENT_DEFINITION('',$);
#7292=PRODUCT_DEFINITION_SHAPE('',$,#7291);
#7293=PATH_FEATURE_COMPONENT('linear path','linear',#7292,.F.);
#7294=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#7295=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#7296=PRECISION_QUALIFIER(4);
#7302=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7294,#7295,#7296)
)REPRESENTATION_ITEM('distance'));
#7303=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7271,#7302),#29);
#7304=PROPERTY_DEFINITION('',$,#7293);
#7305=PROPERTY_DEFINITION_REPRESENTATION(#7304,#7303);
#7306=DIRECTION_SHAPE_REPRESENTATION('',(#7269),#29);
#7307=PROPERTY_DEFINITION_REPRESENTATION(#7304,#7306);
#7308=SHAPE_ASPECT('','diameter occurrence',#7267,.T.);
#7309=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7278,#7308);
#7310=SHAPE_ASPECT('','hole depth occurrence',#7267,.T.);
#7311=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7293,#7310);
```

```
#7312=FEATURE_COMPONENT_DEFINITION('',$);
#7313=PRODUCT_DEFINITION_SHAPE('',$,#7312);
#7314=HOLE_BOTTOM('bottom condition','flat',#7313,.F.);
#7319=SHAPE_ASPECT('','bottom condition occurrence',#7267,.T.);

#7320=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7314,#7319);
#7321=FEATURE_COMPONENT_RELATIONSHIP('large hole',$,#7259,#7265);
#7322=FEATURE_COMPONENT_RELATIONSHIP('small hole',$,#7259,#7203);
#7323=CARTESIAN_POINT('',(205.0,-50.0,152.500000000000000));
#7324=DIRECTION('',(0.0,1.0,0.0));
#7325=AXIS1_PLACEMENT('',#7323,#7324);
#7326=SHAPE_ASPECT('',$,#38,.F.);
#7327=PROPERTY_DEFINITION('',$,#7326);
#7328=REPRESENTATION('',(#7325),#29);
#7329=PROPERTY_DEFINITION_REPRESENTATION(#7327,#7328);
#7330=DIMENSIONAL_LOCATION('',$,#1261,#7326);
#7331=PRECISION_QUALIFIER(4);
#7337=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7331))REPRESENTAT
ION_ITEM(''));
#7338=SHAPE_DIMENSION_REPRESENTATION('',(#7337),#29);
#7339=PRECISION_QUALIFIER(4);
#7340=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7341=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7342=MEASURE_QUALIFICATION('','',#7340,(#7339));
#7343=MEASURE_QUALIFICATION('','',#7341,(#7339));
#7344=TOLERANCE_VALUE(#7341,#7340);
#7345=PLUS_MINUS_TOLERANCE(#7344,#7330);
#7346=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7330,#7338);
#7347=CARTESIAN_POINT('',(205.0,-50.0,152.500000000000000));
#7348=DIRECTION('',(0.0,1.0,0.0));
#7349=AXIS1_PLACEMENT('',#7347,#7348);
#7350=SHAPE_ASPECT('',$,#38,.F.);
#7351=PROPERTY_DEFINITION('',$,#7350);
#7352=REPRESENTATION('',(#7349),#29);
#7353=PROPERTY_DEFINITION_REPRESENTATION(#7351,#7352);
#7354=DIMENSIONAL_LOCATION('',$,#143,#7350);
#7355=PRECISION_QUALIFIER(4);
#7361=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(27.500000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#735
5))REPRESENTATION_ITEM(''));
#7362=SHAPE_DIMENSION_REPRESENTATION('',(#7361),#29);
#7363=PRECISION_QUALIFIER(4);
#7364=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7365=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7366=MEASURE_QUALIFICATION('','',#7364,(#7363));
#7367=MEASURE_QUALIFICATION('','',#7365,(#7363));
#7368=TOLERANCE_VALUE(#7365,#7364);
#7369=PLUS_MINUS_TOLERANCE(#7368,#7354);
#7370=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7354,#7362);
#7377=(CHARACTERIZED_OBJECT('compound feature in
solid','counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid','counterbore',#38,.T.));
#7378=PROPERTY_DEFINITION('',$,#7377);
```



```
#7379=PRODUCT_DEFINITION_SHAPE('',$,#7377);
#7380=CARTESIAN_POINT('',(25.0,0.0,27.500000000000000));
#7381=DIRECTION('',(0.0,-1.0,0.0));
#7382=DIRECTION('',(-1.0,0.0,0.0));
#7383=AXIS2_PLACEMENT_3D('orientation',#7380,#7381,#7382);
#7384=REPRESENTATION('',(#2988,#3028,#3052,#3068,#3084),#29);
#7385=PROPERTY_DEFINITION_REPRESENTATION(#7378,#7384);
#7386=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7383),#29);
#7387=PROPERTY_DEFINITION_REPRESENTATION(#7378,#7386);
#7393=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7394=PROPERTY_DEFINITION('',$,#7393);
#7395=PRODUCT_DEFINITION_SHAPE('',$,#7393);
#7396=CARTESIAN_POINT('',(25.0,0.0,27.500000000000000));
#7397=DIRECTION('',(0.0,-1.0,0.0));
#7398=DIRECTION('',(-1.0,0.0,0.0));
#7399=AXIS2_PLACEMENT_3D('orientation',#7396,#7397,#7398);
#7400=REPRESENTATION('',(#2988,#3068),#29);
#7401=PROPERTY_DEFINITION_REPRESENTATION(#7394,#7400);
#7402=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7399),#29);
#7403=PROPERTY_DEFINITION_REPRESENTATION(#7394,#7402);
#7404=FEATURE_COMPONENT_DEFINITION('',$);
#7405=PRODUCT_DEFINITION_SHAPE('',$,#7404);
#7406=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7405,.F.);
#7407=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#7408=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#7409=PRECISION_QUALIFIER(4);
#7415=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7407,#7408,#7409)
)REPRESENTATION_ITEM('diameter'));
#7416=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7399,#7415),#29);
#7417=PROPERTY_DEFINITION('',$,#7406);
#7418=PROPERTY_DEFINITION_REPRESENTATION(#7417,#7416);
#7419=FEATURE_COMPONENT_DEFINITION('',$);
#7420=PRODUCT_DEFINITION_SHAPE('',$,#7419);
#7421=PATH_FEATURE_COMPONENT('linear path','linear',#7420,.F.);
#7422=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#7423=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#7424=PRECISION_QUALIFIER(4);
#7430=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7422,#7423,#7424)
)REPRESENTATION_ITEM('distance'));
#7431=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7399,#7430),#29);
#7432=PROPERTY_DEFINITION('',$,#7421);
#7433=PROPERTY_DEFINITION_REPRESENTATION(#7432,#7431);
#7434=DIRECTION_SHAPE_REPRESENTATION('',(#7397),#29);
#7435=PROPERTY_DEFINITION_REPRESENTATION(#7432,#7434);
#7436=SHAPE_ASPECT('','diameter occurrence',#7395,.T.);
#7437=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7406,#7436);
#7438=SHAPE_ASPECT('','hole depth occurrence',#7395,.T.);
#7439=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7421,#7438);

#7440=FEATURE_COMPONENT_DEFINITION('',$);
#7441=PRODUCT_DEFINITION_SHAPE('',$,#7440);
#7442=HOLE_BOTTOM('bottom condition','flat',#7441,.F.);
```

```
#7447=SHAPE_ASPECT('','bottom condition occurrence',#7395,.T.);

#7448=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7442,#7447);
#7449=COMPOSITE_SHAPE_ASPECT('compound feature in solid',$,#7379,.T.);
#7455=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7456=PROPERTY_DEFINITION('',$,#7455);
#7457=PRODUCT_DEFINITION_SHAPE('',$,#7455);
#7458=CARTESIAN_POINT('',(25.0,0.0,27.500000000000000));
#7459=DIRECTION('',(0.0,-1.0,0.0));
#7460=DIRECTION('',(-1.0,0.0,0.0));
#7461=AXIS2_PLACEMENT_3D('orientation',#7458,#7459,#7460);
#7462=REPRESENTATION('',(#3028,#3052,#3084),#29);
#7463=PROPERTY_DEFINITION_REPRESENTATION(#7456,#7462);
#7464=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7461),#29);
#7465=PROPERTY_DEFINITION_REPRESENTATION(#7456,#7464);
#7466=FEATURE_COMPONENT_DEFINITION('',$);
#7467=PRODUCT_DEFINITION_SHAPE('',$,#7466);
#7468=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7467,.F.);
#7469=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#7470=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#7471=PRECISION_QUALIFIER(4);
#7477=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(26.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7469,#7470,#7471)
)REPRESENTATION_ITEM('diameter'));
#7478=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7461,#7477),#29);
#7479=PROPERTY_DEFINITION('',$,#7468);
#7480=PROPERTY_DEFINITION_REPRESENTATION(#7479,#7478);
#7481=FEATURE_COMPONENT_DEFINITION('',$);
#7482=PRODUCT_DEFINITION_SHAPE('',$,#7481);
#7483=PATH_FEATURE_COMPONENT('linear path','linear',#7482,.F.);
#7484=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#7485=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#7486=PRECISION_QUALIFIER(4);
#7492=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7484,#7485,#7486)
)REPRESENTATION_ITEM('distance'));
#7493=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7461,#7492),#29);
#7494=PROPERTY_DEFINITION('',$,#7483);
#7495=PROPERTY_DEFINITION_REPRESENTATION(#7494,#7493);
#7496=DIRECTION_SHAPE_REPRESENTATION('',(#7459),#29);
#7497=PROPERTY_DEFINITION_REPRESENTATION(#7494,#7496);
#7498=SHAPE_ASPECT('','diameter occurrence',#7457,.T.);
#7499=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7468,#7498);
#7500=SHAPE_ASPECT('','hole depth occurrence',#7457,.T.);
#7501=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7483,#7500);

#7502=FEATURE_COMPONENT_DEFINITION('',$);
#7503=PRODUCT_DEFINITION_SHAPE('',$,#7502);
#7504=HOLE_BOTTOM('bottom condition','flat',#7503,.F.);
#7509=SHAPE_ASPECT('','bottom condition occurrence',#7457,.T.);

#7510=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7504,#7509);
```



```
#7511=FEATURE_COMPONENT_RELATIONSHIP('large hole',$,#7449,#7455);
#7512=FEATURE_COMPONENT_RELATIONSHIP('small hole',$,#7449,#7393);
#7513=CARTESIAN_POINT('',(25.0,-50.0,27.500000000000000));
#7514=DIRECTION('',(0.0,1.0,0.0));
#7515=AXIS1_PLACEMENT('',#7513,#7514);
#7516=SHAPE_ASPECT('',$,#38,.F.);
#7517=PROPERTY_DEFINITION('',$,#7516);
#7518=REPRESENTATION('',(#7515),#29);
#7519=PROPERTY_DEFINITION_REPRESENTATION(#7517,#7518);
#7520=DIMENSIONAL_LOCATION('',$,#239,#7516);
#7521=PRECISION_QUALIFIER(4);
#7527=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7521))REPRESENTAT
ION_ITEM(''));
#7528=SHAPE_DIMENSION_REPRESENTATION('',(#7527),#29);
#7529=PRECISION_QUALIFIER(4);
#7530=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7531=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7532=MEASURE_QUALIFICATION('','',#7530,(#7529));
#7533=MEASURE_QUALIFICATION('','',#7531,(#7529));
#7534=TOLERANCE_VALUE(#7531,#7530);
#7535=PLUS_MINUS_TOLERANCE(#7534,#7520);
#7536=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7520,#7528);
#7537=CARTESIAN_POINT('',(25.0,-50.0,27.500000000000000));
#7538=DIRECTION('',(0.0,1.0,0.0));
#7539=AXIS1_PLACEMENT('',#7537,#7538);
#7540=SHAPE_ASPECT('',$,#38,.F.);
#7541=PROPERTY_DEFINITION('',$,#7540);
#7542=REPRESENTATION('',(#7539),#29);
#7543=PROPERTY_DEFINITION_REPRESENTATION(#7541,#7542);
#7544=DIMENSIONAL_LOCATION('',$,#75,#7540);
#7545=PRECISION_QUALIFIER(4);
#7551=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(27.500000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#754
5))REPRESENTATION_ITEM(''));
#7552=SHAPE_DIMENSION_REPRESENTATION('',(#7551),#29);
#7553=PRECISION_QUALIFIER(4);
#7554=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7555=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7556=MEASURE_QUALIFICATION('','',#7554,(#7553));
#7557=MEASURE_QUALIFICATION('','',#7555,(#7553));
#7558=TOLERANCE_VALUE(#7555,#7554);
#7559=PLUS_MINUS_TOLERANCE(#7558,#7544);
#7560=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7544,#7552);
#7567=(CHARACTERIZED_OBJECT('compound feature in
solid','counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid','counterbore',#38,.T.));
#7568=PROPERTY_DEFINITION('',$,#7567);
#7569=PRODUCT_DEFINITION_SHAPE('',$,#7567);
#7570=CARTESIAN_POINT('',(25.0,0.0,152.500000000000000));
#7571=DIRECTION('',(0.0,-1.0,0.0));
#7572=DIRECTION('',(-1.0,0.0,0.0));
#7573=AXIS2_PLACEMENT_3D('orientation',#7570,#7571,#7572);
#7574=REPRESENTATION('',(#3118,#3158,#3182,#3198,#3214),#29);
#7575=PROPERTY_DEFINITION_REPRESENTATION(#7568,#7574);
```

```
#7576=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7573),#29);
#7577=PROPERTY_DEFINITION_REPRESENTATION(#7568,#7576);
#7583=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7584=PROPERTY_DEFINITION('',$,#7583);
#7585=PRODUCT_DEFINITION_SHAPE('',$,#7583);
#7586=CARTESIAN_POINT('',(25.0,0.0,152.500000000000));
#7587=DIRECTION('',(0.0,-1.0,0.0));
#7588=DIRECTION('',(-1.0,0.0,0.0));
#7589=AXIS2_PLACEMENT_3D('orientation',#7586,#7587,#7588);
#7590=REPRESENTATION('',(#3118,#3198),#29);
#7591=PROPERTY_DEFINITION_REPRESENTATION(#7584,#7590);
#7592=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7589),#29);
#7593=PROPERTY_DEFINITION_REPRESENTATION(#7584,#7592);
#7594=FEATURE_COMPONENT_DEFINITION('',$);
#7595=PRODUCT_DEFINITION_SHAPE('',$,#7594);
#7596=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7595,.F.);
#7597=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#7598=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#7599=PRECISION_QUALIFIER(4);
#7605=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7597,#7598,#7599)
)REPRESENTATION_ITEM('diameter'));
#7606=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7589,#7605),#29);
#7607=PROPERTY_DEFINITION('',$,#7596);
#7608=PROPERTY_DEFINITION_REPRESENTATION(#7607,#7606);
#7609=FEATURE_COMPONENT_DEFINITION('',$);
#7610=PRODUCT_DEFINITION_SHAPE('',$,#7609);
#7611=PATH_FEATURE_COMPONENT('linear path','linear',#7610,.F.);
#7612=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#7613=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#7614=PRECISION_QUALIFIER(4);
#7620=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7612,#7613,#7614)
)REPRESENTATION_ITEM('distance'));
#7621=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7589,#7620),#29);
#7622=PROPERTY_DEFINITION('',$,#7611);
#7623=PROPERTY_DEFINITION_REPRESENTATION(#7622,#7621);
#7624=DIRECTION_SHAPE_REPRESENTATION('',(#7587),#29);
#7625=PROPERTY_DEFINITION_REPRESENTATION(#7622,#7624);
#7626=SHAPE_ASPECT('','diameter occurrence',#7585,.T.);
#7627=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7596,#7626);
#7628=SHAPE_ASPECT('','hole depth occurrence',#7585,.T.);
#7629=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7611,#7628);

#7630=FEATURE_COMPONENT_DEFINITION('',$);
#7631=PRODUCT_DEFINITION_SHAPE('',$,#7630);
#7632=HOLE_BOTTOM('bottom condition','flat',#7631,.F.);
#7637=SHAPE_ASPECT('','bottom condition occurrence',#7585,.T.);

#7638=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7632,#7637);
#7639=COMPOSITE_SHAPE_ASPECT('compound feature in solid',$,#7569,.T.);
#7645=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
```



```
#7646=PROPERTY_DEFINITION('',$,#7645);
#7647=PRODUCT_DEFINITION_SHAPE('',$,#7645);
#7648=CARTESIAN_POINT('',(25.0,0.0,152.500000000000000));
#7649=DIRECTION('',(0.0,-1.0,0.0));
#7650=DIRECTION('',(-1.0,0.0,0.0));
#7651=AXIS2_PLACEMENT_3D('orientation',#7648,#7649,#7650);
#7652=REPRESENTATION('',(#3158,#3182,#3214),#29);
#7653=PROPERTY_DEFINITION_REPRESENTATION(#7646,#7652);
#7654=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7651),#29);
#7655=PROPERTY_DEFINITION_REPRESENTATION(#7646,#7654);
#7656=FEATURE_COMPONENT_DEFINITION('',$);
#7657=PRODUCT_DEFINITION_SHAPE('',$,#7656);
#7658=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7657,.F.);
#7659=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#7660=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#7661=PRECISION_QUALIFIER(4);
#7667=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(26.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7659,#7660,#7661)
)REPRESENTATION_ITEM('diameter'));
#7668=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7651,#7667),#29);
#7669=PROPERTY_DEFINITION('',$,#7658);
#7670=PROPERTY_DEFINITION_REPRESENTATION(#7669,#7668);
#7671=FEATURE_COMPONENT_DEFINITION('',$);
#7672=PRODUCT_DEFINITION_SHAPE('',$,#7671);
#7673=PATH_FEATURE_COMPONENT('linear path','linear',#7672,.F.);
#7674=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#7675=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#7676=PRECISION_QUALIFIER(4);
#7682=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7674,#7675,#7676)
)REPRESENTATION_ITEM('distance'));
#7683=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7651,#7682),#29);
#7684=PROPERTY_DEFINITION('',$,#7673);
#7685=PROPERTY_DEFINITION_REPRESENTATION(#7684,#7683);
#7686=DIRECTION_SHAPE_REPRESENTATION('',(#7649),#29);
#7687=PROPERTY_DEFINITION_REPRESENTATION(#7684,#7686);
#7688=SHAPE_ASPECT('','diameter occurrence',#7647,.T.);
#7689=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7658,#7688);
#7690=SHAPE_ASPECT('','hole depth occurrence',#7647,.T.);
#7691=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7673,#7690);

#7692=FEATURE_COMPONENT_DEFINITION('',$);
#7693=PRODUCT_DEFINITION_SHAPE('',$,#7692);
#7694=HOLE_BOTTOM('bottom condition','flat',#7693,.F.);
#7699=SHAPE_ASPECT('','bottom condition occurrence',#7647,.T.);

#7700=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7694,#7699);
#7701=FEATURE_COMPONENT_RELATIONSHIP('large hole',$,#7639,#7645);
#7702=FEATURE_COMPONENT_RELATIONSHIP('small hole',$,#7639,#7583);
#7703=CARTESIAN_POINT('',(25.0,-50.0,152.500000000000000));
#7704=DIRECTION('',(0.0,1.0,0.0));
#7705=AXIS1_PLACEMENT('',#7703,#7704);
#7706=SHAPE_ASPECT('',$,#38,.F.);
#7707=PROPERTY_DEFINITION('',$,#7706);
```

```
#7708=REPRESENTATION('',(#7705),#29);
#7709=PROPERTY_DEFINITION_REPRESENTATION(#7707,#7708);
#7710=DIMENSIONAL_LOCATION('',$,#239,#7706);
#7711=PRECISION_QUALIFIER(4);
#7717=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7711))REPRESENTAT
ION_ITEM(''));
#7718=SHAPE_DIMENSION_REPRESENTATION('',(#7717),#29);
#7719=PRECISION_QUALIFIER(4);
#7720=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7721=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7722=MEASURE_QUALIFICATION('','',#7720,(#7719));
#7723=MEASURE_QUALIFICATION('','',#7721,(#7719));
#7724=TOLERANCE_VALUE(#7721,#7720);
#7725=PLUS_MINUS_TOLERANCE(#7724,#7710);
#7726=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7710,#7718);
#7727=CARTESIAN_POINT('',(25.0,-50.0,152.500000000000));
#7728=DIRECTION('',(0.0,1.0,0.0));
#7729=AXIS1_PLACEMENT('',#7727,#7728);
#7730=SHAPE_ASPECT('',$,#38,.F.);
#7731=PROPERTY_DEFINITION('',$,#7730);
#7732=REPRESENTATION('',(#7729),#29);
#7733=PROPERTY_DEFINITION_REPRESENTATION(#7731,#7732);
#7734=DIMENSIONAL_LOCATION('',$,#143,#7730);
#7735=PRECISION_QUALIFIER(4);
#7741=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(27.5000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#773
5))REPRESENTATION_ITEM(''));
#7742=SHAPE_DIMENSION_REPRESENTATION('',(#7741),#29);
#7743=PRECISION_QUALIFIER(4);
#7744=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0100000000000000),#10);
#7745=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.0100000000000000),#10);
#7746=MEASURE_QUALIFICATION('','',#7744,(#7743));
#7747=MEASURE_QUALIFICATION('','',#7745,(#7743));
#7748=TOLERANCE_VALUE(#7745,#7744);
#7749=PLUS_MINUS_TOLERANCE(#7748,#7734);
#7750=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7734,#7742);
#7756=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('',$,#38,.T.)SLOT());
#7757=PROPERTY_DEFINITION('',$,#7756);
#7758=PRODUCT_DEFINITION_SHAPE('',$,#7756);
#7759=CARTESIAN_POINT('',(205.0,-10.0,90.0));
#7760=DIRECTION('',(1.0,0.0,0.0));
#7761=DIRECTION('',(0.0,0.0,-1.0));
#7762=AXIS2_PLACEMENT_3D('orientation',#7759,#7760,#7761);
#7763=REPRESENTATION('',(#3240,#3268,#3288,#3304),#29);
#7764=PROPERTY_DEFINITION_REPRESENTATION(#7757,#7763);
#7765=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7762),#29);
#7766=PROPERTY_DEFINITION_REPRESENTATION(#7757,#7765);
#7767=FEATURE_COMPONENT_DEFINITION('',$);
#7768=PRODUCT_DEFINITION_SHAPE('',$,#7767);
#7769=SQUARE_U_PROFILE('',$,#7768,.F.);
#7770=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#7771=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#7772=PRECISION_QUALIFIER(4);
```



```
#7778=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7770,#7771,#7772)
)REPRESENTATION_ITEM('width'));
#7783=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
#7788=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#7789=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7762,#7778,#7783,#7788),#29);
#7790=PROPERTY_DEFINITION('',$,#7769);
#7791=PROPERTY_DEFINITION_REPRESENTATION(#7790,#7789);
#7792=SHAPE_ASPECT('','swept shape occurrence',#7758,.T.);
#7793=SHAPE_DEFINING_RELATIONSHIP('','profile usage',#7769,#7792);
#7794=FEATURE_COMPONENT_DEFINITION('',$);
#7795=PRODUCT_DEFINITION_SHAPE('',$,#7794);
#7796=PATH_FEATURE_COMPONENT('linear path','linear',#7795,.F.);
#7797=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#7798=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#7799=PRECISION_QUALIFIER(4);
#7805=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7797,#7798,#7799)
)REPRESENTATION_ITEM('distance'));
#7806=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7762,#7805),#29);
#7807=PROPERTY_DEFINITION('',$,#7796);
#7808=PROPERTY_DEFINITION_REPRESENTATION(#7807,#7806);
#7809=DIRECTION_SHAPE_REPRESENTATION('',(#7760),#29);
#7810=PROPERTY_DEFINITION_REPRESENTATION(#7807,#7809);
#7811=SHAPE_ASPECT('','course of travel occurrence',#7758,.T.);
#7812=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#7796,#7811);
#7813=FEATURE_COMPONENT_DEFINITION('',$);
#7814=PRODUCT_DEFINITION_SHAPE('',$,#7813);
#7815=SLOT_END('','radiused',#7814,.F.);
#7816=FEATURE_COMPONENT_DEFINITION('',$);
#7817=PRODUCT_DEFINITION_SHAPE('',$,#7816);
#7818=SLOT_END('','open',#7817,.F.);
#7819=SHAPE_ASPECT('','end condition occurrence',#7758,.T.);
#7820=FEATURE_COMPONENT_RELATIONSHIP('course of travel start','slot end
usage',#7815,#7819);
#7821=SHAPE_ASPECT('','end condition occurrence',#7758,.T.);
#7822=FEATURE_COMPONENT_RELATIONSHIP('course of travel end','slot end
usage',#7818,#7821);
#7823=DIMENSIONAL_LOCATION('',$,#1110,#3269);
#7824=PRECISION_QUALIFIER(4);
#7830=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7824))REPRESENTAT
ION_ITEM(''));
#7831=SHAPE_DIMENSION_REPRESENTATION('',(#7830),#29);
#7832=PRECISION_QUALIFIER(4);
#7833=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7834=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7835=MEASURE_QUALIFICATION('','',#7833,(#7832));
#7836=MEASURE_QUALIFICATION('','',#7834,(#7832));
#7837=TOLERANCE_VALUE(#7834,#7833);
#7838=PLUS_MINUS_TOLERANCE(#7837,#7823);
#7839=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7823,#7831);
#7840=DIMENSIONAL_LOCATION('',$,#5503,#3241);
```

```
#7841=PRECISION_QUALIFIER(4);
#7847=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7841))REPRESENTATI
ON_ITEM(''));
#7848=SHAPE_DIMENSION_REPRESENTATION('',(#7847),#29);
#7849=PRECISION_QUALIFIER(4);
#7850=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#7851=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7852=MEASURE_QUALIFICATION('','',#7850,(#7849));
#7853=MEASURE_QUALIFICATION('','',#7851,(#7849));
#7854=TOLERANCE_VALUE(#7851,#7850);
#7855=PLUS_MINUS_TOLERANCE(#7854,#7840);
#7856=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7840,#7848);
#7862=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('',$,#38,.T.)SLOT());
#7863=PROPERTY_DEFINITION('',$,#7862);
#7864=PRODUCT_DEFINITION_SHAPE('',$,#7862);
#7865=CARTESIAN_POINT('',(80.0,-10.0,90.0));
#7866=DIRECTION('',(1.0,0.0,0.0));
#7867=DIRECTION('',(0.0,0.0,-1.0));
#7868=AXIS2_PLACEMENT_3D('orientation',#7865,#7866,#7867);
#7869=REPRESENTATION('',(#3348,#3372,#3392,#3412,#3428),#29);
#7870=PROPERTY_DEFINITION_REPRESENTATION(#7863,#7869);
#7871=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7868),#29);
#7872=PROPERTY_DEFINITION_REPRESENTATION(#7863,#7871);
#7873=FEATURE_COMPONENT_DEFINITION('',$);
#7874=PRODUCT_DEFINITION_SHAPE('',$,#7873);
#7875=SQUARE_U_PROFILE('',$,#7874,.F.);
#7876=STANDARD_UNCERTAINTY('upper limit','plus',0.200000000000000);
#7877=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#7878=PRECISION_QUALIFIER(4);
#7884=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7876,#7877,#7878)
)REPRESENTATION_ITEM('width'));
#7889=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
#7894=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#7895=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7868,#7884,#7889,#7894),#29);
#7896=PROPERTY_DEFINITION('',$,#7875);
#7897=PROPERTY_DEFINITION_REPRESENTATION(#7896,#7895);
#7898=SHAPE_ASPECT('','swept shape occurrence',#7864,.T.);
#7899=SHAPE_DEFINING_RELATIONSHIP('','profile usage',#7875,#7898);
#7900=FEATURE_COMPONENT_DEFINITION('',$);
#7901=PRODUCT_DEFINITION_SHAPE('',$,#7900);
#7902=PATH_FEATURE_COMPONENT('linear path','linear',#7901,.F.);
#7907=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(35.0),#10)REPRESENTATION_ITEM('distance'));
#7908=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7868,#7907),#29);
#7909=PROPERTY_DEFINITION('',$,#7902);
#7910=PROPERTY_DEFINITION_REPRESENTATION(#7909,#7908);
#7911=DIRECTION_SHAPE_REPRESENTATION('',(#7866),#29);
#7912=PROPERTY_DEFINITION_REPRESENTATION(#7909,#7911);
#7913=SHAPE_ASPECT('','course of travel occurrence',#7864,.T.);
#7914=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#7902,#7913);
```



```
#7915=FEATURE_COMPONENT_DEFINITION('',$);
#7916=PRODUCT_DEFINITION_SHAPE('',$,#7915);
#7917=SLOT_END('','radiused',#7916,.F.);
#7918=FEATURE_COMPONENT_DEFINITION('',$);
#7919=PRODUCT_DEFINITION_SHAPE('',$,#7918);
#7920=SLOT_END('','radiused',#7919,.F.);
#7921=SHAPE_ASPECT('','end condition occurrence',#7864,.T.);
#7922=FEATURE_COMPONENT_RELATIONSHIP('course of travel start','slot end
usage',#7917,#7921);
#7923=SHAPE_ASPECT('','end condition occurrence',#7864,.T.);
#7924=FEATURE_COMPONENT_RELATIONSHIP('course of travel end','slot end
usage',#7920,#7923);
#7925=CARTESIAN_POINT('',(260.0,0.0,0.0));
#7926=DIRECTION('',(0.0,1.0,0.0));
#7927=DIRECTION('',(0.0,0.0,1.0));
#7928=AXIS2_PLACEMENT_3D('',#7925,#7926,#7927);
#7929=SHAPE_ASPECT('',$,#38,.F.);
#7930=PROPERTY_DEFINITION('',$,#7929);
#7931=REPRESENTATION('',(#7928),#29);
#7932=PROPERTY_DEFINITION_REPRESENTATION(#7930,#7931);
#7933=DIMENSIONAL_LOCATION('',$,#7929,#3349);
#7934=PRECISION_QUALIFIER(4);
#7940=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7934))REPRESENTAT
ION_ITEM(''));
#7941=SHAPE_DIMENSION_REPRESENTATION('',(#7940),#29);
#7942=PRECISION_QUALIFIER(4);
#7943=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#7944=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.250000000000000),#10);
#7945=MEASURE_QUALIFICATION('','',#7943,(#7942));
#7946=MEASURE_QUALIFICATION('','',#7944,(#7942));
#7947=TOLERANCE_VALUE(#7944,#7943);
#7948=PLUS_MINUS_TOLERANCE(#7947,#7933);
#7949=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7933,#7941);
#7950=DIMENSIONAL_LOCATION('',$,#3429,#3393);
#7951=PRECISION_QUALIFIER(4);
#7957=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7951))REPRESENTAT
ION_ITEM(''));
#7958=SHAPE_DIMENSION_REPRESENTATION('',(#7957),#29);
#7959=PRECISION_QUALIFIER(4);
#7960=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.400000000000000),#10);
#7961=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0),#10);
#7962=MEASURE_QUALIFICATION('','',#7960,(#7959));
#7963=MEASURE_QUALIFICATION('','',#7961,(#7959));
#7964=TOLERANCE_VALUE(#7961,#7960);
#7965=PLUS_MINUS_TOLERANCE(#7964,#7950);
#7966=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7950,#7958);
#7967=DIMENSIONAL_LOCATION('',$,#1261,#3429);
#7968=PRECISION_QUALIFIER(4);
#7974=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(120.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7968))REPRESENTA
TION_ITEM(''));
#7975=SHAPE_DIMENSION_REPRESENTATION('',(#7974),#29);
#7976=PRECISION_QUALIFIER(4);
#7977=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
```



```
#7978=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#7979=MEASURE_QUALIFICATION('','',#7977,(#7976));
#7980=MEASURE_QUALIFICATION('','',#7978,(#7976));
#7981=TOLERANCE_VALUE(#7978,#7977);
#7982=PLUS_MINUS_TOLERANCE(#7981,#7967);
#7983=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7967,#7975);
#7989=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7990=PROPERTY_DEFINITION('',$,#7989);
#7991=PRODUCT_DEFINITION_SHAPE('',$,#7989);
#7992=CARTESIAN_POINT('',(190.0,0.0,120.0));
#7993=DIRECTION('',(0.0,-1.0,0.0));
#7994=DIRECTION('',(1.0,0.0,0.0));
#7995=AXIS2_PLACEMENT_3D('orientation',#7992,#7993,#7994);
#7996=REPRESENTATION('',(#3458,#3482,#3498),#29);
#7997=PROPERTY_DEFINITION_REPRESENTATION(#7990,#7996);
#7998=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7995),#29);
#7999=PROPERTY_DEFINITION_REPRESENTATION(#7990,#7998);
#8000=FEATURE_COMPONENT_DEFINITION('',$);
#8001=PRODUCT_DEFINITION_SHAPE('',$,#8000);
#8002=CIRCULAR_CLOSED_PROFILE('circular profile',$,#8001,.F.);
#8003=STANDARD_UNCERTAINTY('upper limit','plus',0.120100000000000);
#8004=STANDARD_UNCERTAINTY('lower limit','minus',-0.120100000000000);
#8005=PRECISION_QUALIFIER(4);
#8011=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.13770000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#8003
,#8004,#8005))REPRESENTATION_ITEM('diameter'));
#8012=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7995,#8011),#29);
#8013=PROPERTY_DEFINITION('',$,#8002);
#8014=PROPERTY_DEFINITION_REPRESENTATION(#8013,#8012);
#8015=FEATURE_COMPONENT_DEFINITION('',$);
#8016=PRODUCT_DEFINITION_SHAPE('',$,#8015);
#8017=PATH_FEATURE_COMPONENT('linear path','linear',#8016,.F.);
#8018=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#8019=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#8020=PRECISION_QUALIFIER(4);
#8026=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.0500000000000001),#10)QUALIFIED_REPRESENTATION_ITEM((#801
8,#8019,#8020))REPRESENTATION_ITEM('distance'));
#8027=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7995,#8026),#29);
#8028=PROPERTY_DEFINITION('',$,#8017);
#8029=PROPERTY_DEFINITION_REPRESENTATION(#8028,#8027);
#8030=DIRECTION_SHAPE_REPRESENTATION('',(#7993),#29);
#8031=PROPERTY_DEFINITION_REPRESENTATION(#8028,#8030);
#8032=SHAPE_ASPECT('','diameter occurrence',#7991,.T.);
#8033=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#8002,#8032);
#8034=SHAPE_ASPECT('','hole depth occurrence',#7991,.T.);
#8035=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#8017,#8034);

#8036=FEATURE_COMPONENT_DEFINITION('',$);
#8037=PRODUCT_DEFINITION_SHAPE('',$,#8036);
#8038=HOLE_BOTTOM('bottom condition','flat',#8037,.F.);
#8043=SHAPE_ASPECT('','bottom condition occurrence',#7991,.T.);
```



```
#8044=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#8038,#8043);
#8045=CARTESIAN_POINT('',(190.0,0.0,120.0));
#8046=DIRECTION('',(0.0,-1.0,0.0));
#8047=AXIS1_PLACEMENT('',#8045,#8046);
#8048=SHAPE_ASPECT('',$,#38,.F.);
#8049=PROPERTY_DEFINITION('',$,#8048);
#8050=REPRESENTATION('',(#8047),#29);
#8051=PROPERTY_DEFINITION_REPRESENTATION(#8049,#8050);
#8052=DIMENSIONAL_LOCATION('',$,#1261,#8048);
#8053=PRECISION_QUALIFIER(4);
#8059=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(40.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8053))REPRESENTAT
ION_ITEM(''));
#8060=SHAPE_DIMENSION_REPRESENTATION('',(#8059),#29);
#8061=PRECISION_QUALIFIER(4);
#8062=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8063=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8064=MEASURE_QUALIFICATION('','',#8062,(#8061));
#8065=MEASURE_QUALIFICATION('','',#8063,(#8061));
#8066=TOLERANCE_VALUE(#8063,#8062);
#8067=PLUS_MINUS_TOLERANCE(#8066,#8052);
#8068=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8052,#8060);
#8069=CARTESIAN_POINT('',(190.0,0.0,120.0));
#8070=DIRECTION('',(0.0,-1.0,0.0));
#8071=AXIS1_PLACEMENT('',#8069,#8070);
#8072=SHAPE_ASPECT('',$,#38,.F.);
#8073=PROPERTY_DEFINITION('',$,#8072);
#8074=REPRESENTATION('',(#8071),#29);
#8075=PROPERTY_DEFINITION_REPRESENTATION(#8073,#8074);
#8076=DIMENSIONAL_LOCATION('',$,#143,#8072);
#8077=PRECISION_QUALIFIER(4);
#8083=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(60.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8077))REPRESENTAT
ION_ITEM(''));
#8084=SHAPE_DIMENSION_REPRESENTATION('',(#8083),#29);
#8085=PRECISION_QUALIFIER(4);
#8086=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8087=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8088=MEASURE_QUALIFICATION('','',#8086,(#8085));
#8089=MEASURE_QUALIFICATION('','',#8087,(#8085));
#8090=TOLERANCE_VALUE(#8087,#8086);
#8091=PLUS_MINUS_TOLERANCE(#8090,#8076);
#8092=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8076,#8084);
#8098=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#8099=PROPERTY_DEFINITION('',$,#8098);
#8100=PRODUCT_DEFINITION_SHAPE('',$,#8098);
#8101=CARTESIAN_POINT('',(190.0,0.0,60.0));
#8102=DIRECTION('',(0.0,-1.0,0.0));
#8103=DIRECTION('',(1.0,0.0,0.0));
#8104=AXIS2_PLACEMENT_3D('orientation',#8101,#8102,#8103);
#8105=REPRESENTATION('',(#3528,#3552,#3568),#29);
#8106=PROPERTY_DEFINITION_REPRESENTATION(#8099,#8105);
#8107=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8104),#29);
#8108=PROPERTY_DEFINITION_REPRESENTATION(#8099,#8107);
```

```
#8109=FEATURE_COMPONENT_DEFINITION('',$);
#8110=PRODUCT_DEFINITION_SHAPE('',$,#8109);
#8111=CIRCULAR_CLOSED_PROFILE('circular profile',$,#8110,.F.);
#8112=STANDARD_UNCERTAINTY('upper limit','plus',0.120100000000000);
#8113=STANDARD_UNCERTAINTY('lower limit','minus',-0.120100000000000);
#8114=PRECISION_QUALIFIER(4);
#8120=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.13770000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#8112
,#8113,#8114))REPRESENTATION_ITEM('diameter'));
#8121=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8104,#8120),#29);
#8122=PROPERTY_DEFINITION('',$,#8111);
#8123=PROPERTY_DEFINITION_REPRESENTATION(#8122,#8121);
#8124=FEATURE_COMPONENT_DEFINITION('',$);
#8125=PRODUCT_DEFINITION_SHAPE('',$,#8124);
#8126=PATH_FEATURE_COMPONENT('linear path','linear',#8125,.F.);
#8127=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#8128=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#8129=PRECISION_QUALIFIER(4);
#8135=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.0500000000000001),#10)QUALIFIED_REPRESENTATION_ITEM((#812
7,#8128,#8129))REPRESENTATION_ITEM('distance'));
#8136=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8104,#8135),#29);
#8137=PROPERTY_DEFINITION('',$,#8126);
#8138=PROPERTY_DEFINITION_REPRESENTATION(#8137,#8136);
#8139=DIRECTION_SHAPE_REPRESENTATION('',(#8102),#29);
#8140=PROPERTY_DEFINITION_REPRESENTATION(#8137,#8139);
#8141=SHAPE_ASPECT('','diameter occurrence',#8100,.T.);
#8142=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#8111,#8141);
#8143=SHAPE_ASPECT('','hole depth occurrence',#8100,.T.);
#8144=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#8126,#8143);

#8145=FEATURE_COMPONENT_DEFINITION('',$);
#8146=PRODUCT_DEFINITION_SHAPE('',$,#8145);
#8147=HOLE_BOTTOM('bottom condition','flat',#8146,.F.);
#8152=SHAPE_ASPECT('','bottom condition occurrence',#8100,.T.);

#8153=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#8147,#8152);
#8154=CARTESIAN_POINT('',(190.0,0.0,60.0));
#8155=DIRECTION('',(0.0,-1.0,0.0));
#8156=AXIS1_PLACEMENT('',#8154,#8155);
#8157=SHAPE_ASPECT('',$,#38,.F.);
#8158=PROPERTY_DEFINITION('',$,#8157);
#8159=REPRESENTATION('',(#8156),#29);
#8160=PROPERTY_DEFINITION_REPRESENTATION(#8158,#8159);
#8161=DIMENSIONAL_LOCATION('',$,#1261,#8157);
#8162=PRECISION_QUALIFIER(4);
#8168=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(40.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8162))REPRESENTAT
ION_ITEM(''));
#8169=SHAPE_DIMENSION_REPRESENTATION('',(#8168),#29);
#8170=PRECISION_QUALIFIER(4);
#8171=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8172=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8173=MEASURE_QUALIFICATION('','',#8171,(#8170));
```



```
#8174=MEASURE_QUALIFICATION('','',#8172,(#8170));
#8175=TOLERANCE_VALUE(#8172,#8171);
#8176=PLUS_MINUS_TOLERANCE(#8175,#8161);
#8177=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8161,#8169);
#8178=CARTESIAN_POINT('',(190.0,0.0,60.0));
#8179=DIRECTION('',(0.0,-1.0,0.0));
#8180=AXIS1_PLACEMENT('',#8178,#8179);
#8181=SHAPE_ASPECT('',$,#38,.F.);
#8182=PROPERTY_DEFINITION('',$,#8181);
#8183=REPRESENTATION('',(#8180),#29);
#8184=PROPERTY_DEFINITION_REPRESENTATION(#8182,#8183);
#8185=DIMENSIONAL_LOCATION('',$,#143,#8181);
#8186=PRECISION_QUALIFIER(4);
#8192=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(120.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8186))REPRESENTA
TION_ITEM(''));
#8193=SHAPE_DIMENSION_REPRESENTATION('',(#8192),#29);
#8194=PRECISION_QUALIFIER(4);
#8195=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8196=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8197=MEASURE_QUALIFICATION('','',#8195,(#8194));
#8198=MEASURE_QUALIFICATION('','',#8196,(#8194));
#8199=TOLERANCE_VALUE(#8196,#8195);
#8200=PLUS_MINUS_TOLERANCE(#8199,#8185);
#8201=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8185,#8193);
#8207=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#8208=PROPERTY_DEFINITION('',$,#8207);
#8209=PRODUCT_DEFINITION_SHAPE('',$,#8207);
#8210=CARTESIAN_POINT('',(130.0,0.0,120.0));
#8211=DIRECTION('',(0.0,-1.0,0.0));
#8212=DIRECTION('',(1.0,0.0,0.0));
#8213=AXIS2_PLACEMENT_3D('orientation',#8210,#8211,#8212);
#8214=REPRESENTATION('',(#3598,#3622,#3638),#29);
#8215=PROPERTY_DEFINITION_REPRESENTATION(#8208,#8214);
#8216=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8213),#29);
#8217=PROPERTY_DEFINITION_REPRESENTATION(#8208,#8216);
#8218=FEATURE_COMPONENT_DEFINITION('',$);
#8219=PRODUCT_DEFINITION_SHAPE('',$,#8218);
#8220=CIRCULAR_CLOSED_PROFILE('circular profile',$,#8219,.F.);
#8221=STANDARD_UNCERTAINTY('upper limit','plus',0.120100000000000);
#8222=STANDARD_UNCERTAINTY('lower limit','minus',-0.120100000000000);
#8223=PRECISION_QUALIFIER(4);
#8229=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.137700000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#8221
,#8222,#8223))REPRESENTATION_ITEM('diameter'));
#8230=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8213,#8229),#29);
#8231=PROPERTY_DEFINITION('',$,#8220);
#8232=PROPERTY_DEFINITION_REPRESENTATION(#8231,#8230);
#8233=FEATURE_COMPONENT_DEFINITION('',$);
#8234=PRODUCT_DEFINITION_SHAPE('',$,#8233);
#8235=PATH_FEATURE_COMPONENT('linear path','linear',#8234,.F.);
#8236=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#8237=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#8238=PRECISION_QUALIFIER(4);
```

```
#8244=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.050000000000001),#10)QUALIFIED_REPRESENTATION_ITEM((#823
6,#8237,#8238))REPRESENTATION_ITEM('distance'));
#8245=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8213,#8244),#29);
#8246=PROPERTY_DEFINITION('',$,#8235);
#8247=PROPERTY_DEFINITION_REPRESENTATION(#8246,#8245);
#8248=DIRECTION_SHAPE_REPRESENTATION('',(#8211),#29);
#8249=PROPERTY_DEFINITION_REPRESENTATION(#8246,#8248);
#8250=SHAPE_ASPECT('','diameter occurrence',#8209,.T.);
#8251=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#8220,#8250);
#8252=SHAPE_ASPECT('','hole depth occurrence',#8209,.T.);
#8253=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#8235,#8252);

#8254=FEATURE_COMPONENT_DEFINITION('',$);
#8255=PRODUCT_DEFINITION_SHAPE('',$,#8254);
#8256=HOLE_BOTTOM('bottom condition','flat',#8255,.F.);
#8261=SHAPE_ASPECT('','bottom condition occurrence',#8209,.T.);

#8262=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#8256,#8261);
#8263=CARTESIAN_POINT('',(130.0,0.0,120.0));
#8264=DIRECTION('',(0.0,-1.0,0.0));
#8265=AXIS1_PLACEMENT('',#8263,#8264);
#8266=SHAPE_ASPECT('',$,#38,.F.);
#8267=PROPERTY_DEFINITION('',$,#8266);
#8268=REPRESENTATION('',(#8265),#29);
#8269=PROPERTY_DEFINITION_REPRESENTATION(#8267,#8268);
#8270=DIMENSIONAL_LOCATION('',$,#1261,#8266);
#8271=PRECISION_QUALIFIER(4);
#8277=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(100.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8271))REPRESENTA
TION_ITEM(''));
#8278=SHAPE_DIMENSION_REPRESENTATION('',(#8277),#29);
#8279=PRECISION_QUALIFIER(4);
#8280=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8281=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8282=MEASURE_QUALIFICATION('','',#8280,(#8279));
#8283=MEASURE_QUALIFICATION('','',#8281,(#8279));
#8284=TOLERANCE_VALUE(#8281,#8280);
#8285=PLUS_MINUS_TOLERANCE(#8284,#8270);
#8286=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8270,#8278);
#8287=CARTESIAN_POINT('',(130.0,0.0,120.0));
#8288=DIRECTION('',(0.0,-1.0,0.0));
#8289=AXIS1_PLACEMENT('',#8287,#8288);
#8290=SHAPE_ASPECT('',$,#38,.F.);
#8291=PROPERTY_DEFINITION('',$,#8290);
#8292=REPRESENTATION('',(#8289),#29);
#8293=PROPERTY_DEFINITION_REPRESENTATION(#8291,#8292);
#8294=DIMENSIONAL_LOCATION('',$,#143,#8290);
#8295=PRECISION_QUALIFIER(4);
#8301=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(60.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8295))REPRESENTAT
ION_ITEM(''));
#8302=SHAPE_DIMENSION_REPRESENTATION('',(#8301),#29);
#8303=PRECISION_QUALIFIER(4);
```



```
#8304=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8305=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8306=MEASURE_QUALIFICATION('','',#8304,(#8303));
#8307=MEASURE_QUALIFICATION('','',#8305,(#8303));
#8308=TOLERANCE_VALUE(#8305,#8304);
#8309=PLUS_MINUS_TOLERANCE(#8308,#8294);
#8310=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8294,#8302);
#8316=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#8317=PROPERTY_DEFINITION('',$,#8316);
#8318=PRODUCT_DEFINITION_SHAPE('',$,#8316);
#8319=CARTESIAN_POINT('',(130.0,0.0,60.0));
#8320=DIRECTION('',(0.0,-1.0,0.0));
#8321=DIRECTION('',(1.0,0.0,0.0));
#8322=AXIS2_PLACEMENT_3D('orientation',#8319,#8320,#8321);
#8323=REPRESENTATION('',(#3668,#3692,#3708),#29);
#8324=PROPERTY_DEFINITION_REPRESENTATION(#8317,#8323);
#8325=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8322),#29);
#8326=PROPERTY_DEFINITION_REPRESENTATION(#8317,#8325);
#8327=FEATURE_COMPONENT_DEFINITION('',$);
#8328=PRODUCT_DEFINITION_SHAPE('',$,#8327);
#8329=CIRCULAR_CLOSED_PROFILE('circular profile',$,#8328,.F.);
#8330=STANDARD_UNCERTAINTY('upper limit','plus',0.120100000000000);
#8331=STANDARD_UNCERTAINTY('lower limit','minus',-0.120100000000000);
#8332=PRECISION_QUALIFIER(4);
#8338=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.13770000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#8330
,#8331,#8332))REPRESENTATION_ITEM('diameter'));
#8339=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8322,#8338),#29);
#8340=PROPERTY_DEFINITION('',$,#8329);
#8341=PROPERTY_DEFINITION_REPRESENTATION(#8340,#8339);
#8342=FEATURE_COMPONENT_DEFINITION('',$);
#8343=PRODUCT_DEFINITION_SHAPE('',$,#8342);
#8344=PATH_FEATURE_COMPONENT('linear path','linear',#8343,.F.);
#8345=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#8346=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#8347=PRECISION_QUALIFIER(4);
#8353=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.0500000000000001),#10)QUALIFIED_REPRESENTATION_ITEM((#834
5,#8346,#8347))REPRESENTATION_ITEM('distance'));
#8354=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8322,#8353),#29);
#8355=PROPERTY_DEFINITION('',$,#8344);
#8356=PROPERTY_DEFINITION_REPRESENTATION(#8355,#8354);
#8357=DIRECTION_SHAPE_REPRESENTATION('',(#8320),#29);
#8358=PROPERTY_DEFINITION_REPRESENTATION(#8355,#8357);
#8359=SHAPE_ASPECT('','diameter occurrence',#8318,.T.);
#8360=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#8329,#8359);
#8361=SHAPE_ASPECT('','hole depth occurrence',#8318,.T.);
#8362=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#8344,#8361);

#8363=FEATURE_COMPONENT_DEFINITION('',$);
#8364=PRODUCT_DEFINITION_SHAPE('',$,#8363);
#8365=HOLE_BOTTOM('bottom condition','flat',#8364,.F.);
#8370=SHAPE_ASPECT('','bottom condition occurrence',#8318,.T.);
```

```
#8371=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#8365,#8370);
#8372=CARTESIAN_POINT('',(130.0,0.0,60.0));
#8373=DIRECTION('',(0.0,-1.0,0.0));
#8374=AXIS1_PLACEMENT('',#8372,#8373);
#8375=SHAPE_ASPECT('',$,#38,.F.);
#8376=PROPERTY_DEFINITION('',$,#8375);
#8377=REPRESENTATION('',(#8374),#29);
#8378=PROPERTY_DEFINITION_REPRESENTATION(#8376,#8377);
#8379=DIMENSIONAL_LOCATION('',$,#1261,#8375);
#8380=PRECISION_QUALIFIER(4);
#8386=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(100.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8380))REPRESENTA
TION_ITEM(''));
#8387=SHAPE_DIMENSION_REPRESENTATION('',(#8386),#29);
#8388=PRECISION_QUALIFIER(4);
#8389=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8390=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8391=MEASURE_QUALIFICATION('','',#8389,(#8388));
#8392=MEASURE_QUALIFICATION('','',#8390,(#8388));
#8393=TOLERANCE_VALUE(#8390,#8389);
#8394=PLUS_MINUS_TOLERANCE(#8393,#8379);
#8395=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8379,#8387);
#8396=CARTESIAN_POINT('',(130.0,0.0,60.0));
#8397=DIRECTION('',(0.0,-1.0,0.0));
#8398=AXIS1_PLACEMENT('',#8396,#8397);
#8399=SHAPE_ASPECT('',$,#38,.F.);
#8400=PROPERTY_DEFINITION('',$,#8399);
#8401=REPRESENTATION('',(#8398),#29);
#8402=PROPERTY_DEFINITION_REPRESENTATION(#8400,#8401);
#8403=DIMENSIONAL_LOCATION('',$,#143,#8399);
#8404=PRECISION_QUALIFIER(4);
#8410=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(120.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8404))REPRESENTA
TION_ITEM(''));
#8411=SHAPE_DIMENSION_REPRESENTATION('',(#8410),#29);
#8412=PRECISION_QUALIFIER(4);
#8413=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8414=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8415=MEASURE_QUALIFICATION('','',#8413,(#8412));
#8416=MEASURE_QUALIFICATION('','',#8414,(#8412));
#8417=TOLERANCE_VALUE(#8414,#8413);
#8418=PLUS_MINUS_TOLERANCE(#8417,#8403);
#8419=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8403,#8411);
#8425=(CHARACTERIZED_OBJECT('ROUND',$)FEATURE_DEFINITION()INSTANCED_FEATURE()
REMOVAL_VOLUME()SHAPE_ASPECT('ROUND',$,#38,.T.));
#8426=PROPERTY_DEFINITION('',$,#8425);
#8427=PRODUCT_DEFINITION_SHAPE('',$,#8425);
#8428=CARTESIAN_POINT('',(0.0,0.0,0.0));
#8429=DIRECTION('',(0.0,0.0,1.0));
#8430=DIRECTION('',(1.0,0.0,0.0));
#8431=AXIS2_PLACEMENT_3D('orientation',#8428,#8429,#8430);
#8432=REPRESENTATION('',(#4214,#4242,#4264,#4286,#4308,#4323,#4338,#4360,#438
8,#4410,#4432,#4454,#4469,#4484),#29);
#8433=PROPERTY_DEFINITION_REPRESENTATION(#8426,#8432);
#8434=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8431),#29);
```



```
#8435=PROPERTY_DEFINITION_REPRESENTATION(#8426,#8434);
#8436=SHAPE_ASPECT('','volume shape',#38,.T.);
#8437=PROPERTY_DEFINITION('',$,#8436);
#8438=PROPERTY_DEFINITION_REPRESENTATION(#8437,#8432);
#8439=SHAPE_ASPECT('','shape volume occurrence',#8427,.T.);
#8440=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#8436,#8439);
#8446=(CHARACTERIZED_OBJECT('CUT',$)FEATURE_DEFINITION()INSTANCED_FEATURE()RE
MOVAL_VOLUME()SHAPE_ASPECT('CUT',$,#38,.T.));
#8447=PROPERTY_DEFINITION('',$,#8446);
#8448=PRODUCT_DEFINITION_SHAPE('',$,#8446);
#8449=CARTESIAN_POINT('',(0.0,0.0,0.0));
#8450=DIRECTION('',(0.0,0.0,1.0));
#8451=DIRECTION('',(1.0,0.0,0.0));
#8452=AXIS2_PLACEMENT_3D('orientation',#8449,#8450,#8451);
#8453=REPRESENTATION('',(#8452),#29);
#8454=PROPERTY_DEFINITION_REPRESENTATION(#8447,#8453);
#8455=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8452),#29);
#8456=PROPERTY_DEFINITION_REPRESENTATION(#8447,#8455);
#8457=SHAPE_ASPECT('','volume shape',#38,.T.);
#8458=PROPERTY_DEFINITION('',$,#8457);
#8459=PROPERTY_DEFINITION_REPRESENTATION(#8458,#8453);
#8460=SHAPE_ASPECT('','shape volume occurrence',#8448,.T.);
#8461=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#8457,#8460);
#8467=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#8468=PROPERTY_DEFINITION('',$,#8467);
#8469=PRODUCT_DEFINITION_SHAPE('',$,#8467);
#8470=CARTESIAN_POINT('',(50.0,0.0,90.0));
#8471=DIRECTION('',(0.0,-1.0,0.0));
#8472=DIRECTION('',(1.0,0.0,0.0));
#8473=AXIS2_PLACEMENT_3D('orientation',#8470,#8471,#8472);
#8474=REPRESENTATION('',(#4506),#29);
#8475=PROPERTY_DEFINITION_REPRESENTATION(#8468,#8474);
#8476=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8473),#29);
#8477=PROPERTY_DEFINITION_REPRESENTATION(#8468,#8476);
#8478=FEATURE_COMPONENT_DEFINITION('',$);
#8479=PRODUCT_DEFINITION_SHAPE('',$,#8478);
#8480=CIRCULAR_CLOSED_PROFILE('circular profile',$,#8479,.F.);
#8481=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#8482=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#8483=PRECISION_QUALIFIER(4);
#8489=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8481,#8482,#8483)
)REPRESENTATION_ITEM('diameter'));
#8490=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8473,#8489),#29);
#8491=PROPERTY_DEFINITION('',$,#8480);
#8492=PROPERTY_DEFINITION_REPRESENTATION(#8491,#8490);
#8493=FEATURE_COMPONENT_DEFINITION('',$);
#8494=PRODUCT_DEFINITION_SHAPE('',$,#8493);
#8495=PATH_FEATURE_COMPONENT('linear path','linear',#8494,.F.);
#8496=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#8497=STANDARD_UNCERTAINTY('lower limit','minus',-1.0);
#8498=PRECISION_QUALIFIER(4);
#8504=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8496,#8497,#8498)
)REPRESENTATION_ITEM('distance'));
```

```
#8505=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8473,#8504),#29);
#8506=PROPERTY_DEFINITION('',$,#8495);
#8507=PROPERTY_DEFINITION_REPRESENTATION(#8506,#8505);
#8508=DIRECTION_SHAPE_REPRESENTATION('',(#8471),#29);
#8509=PROPERTY_DEFINITION_REPRESENTATION(#8506,#8508);
#8510=SHAPE_ASPECT('','diameter occurrence',#8469,.T.);
#8511=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#8480,#8510);
#8512=SHAPE_ASPECT('','hole depth occurrence',#8469,.T.);
#8513=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#8495,#8512);

#8514=FEATURE_COMPONENT_DEFINITION('',$);
#8515=PRODUCT_DEFINITION_SHAPE('',$,#8514);
#8516=HOLE_BOTTOM('bottom condition','flat',#8515,.F.);
#8521=SHAPE_ASPECT('','bottom condition occurrence',#8469,.T.);

#8522=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#8516,#8521);
#8523=CARTESIAN_POINT('',(50.0,0.0,90.0));
#8524=DIRECTION('',(0.0,-1.0,0.0));
#8525=AXIS1_PLACEMENT('',#8523,#8524);
#8526=SHAPE_ASPECT('',$,#38,.F.);
#8527=PROPERTY_DEFINITION('',$,#8526);
#8528=REPRESENTATION('',(#8525),#29);
#8529=PROPERTY_DEFINITION_REPRESENTATION(#8527,#8528);
#8530=DIMENSIONAL_LOCATION('',$,#239,#8526);
#8531=PRECISION_QUALIFIER(4);
#8537=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8531))REPRESENTAT
ION_ITEM(''));
#8538=SHAPE_DIMENSION_REPRESENTATION('',(#8537),#29);
#8539=PRECISION_QUALIFIER(4);
#8540=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.010000000000000),#10);
#8541=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.010000000000000),#10);
#8542=MEASURE_QUALIFICATION('','',#8540,(#8539));
#8543=MEASURE_QUALIFICATION('','',#8541,(#8539));
#8544=TOLERANCE_VALUE(#8541,#8540);
#8545=PLUS_MINUS_TOLERANCE(#8544,#8530);
#8546=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8530,#8538);
#8547=CARTESIAN_POINT('',(50.0,0.0,90.0));
#8548=DIRECTION('',(0.0,-1.0,0.0));
#8549=AXIS1_PLACEMENT('',#8547,#8548);
#8550=SHAPE_ASPECT('',$,#38,.F.);
#8551=PROPERTY_DEFINITION('',$,#8550);
#8552=REPRESENTATION('',(#8549),#29);
#8553=PROPERTY_DEFINITION_REPRESENTATION(#8551,#8552);
#8554=DIMENSIONAL_LOCATION('',$,#143,#8550);
#8555=PRECISION_QUALIFIER(4);
#8561=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(90.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8555))REPRESENTAT
ION_ITEM(''));
#8562=SHAPE_DIMENSION_REPRESENTATION('',(#8561),#29);
#8563=PRECISION_QUALIFIER(4);
#8564=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.010000000000000),#10);
#8565=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.010000000000000),#10);
#8566=MEASURE_QUALIFICATION('','',#8564,(#8563));
```



```
#8567=MEASURE_QUALIFICATION('','',#8565,(#8563));
#8568=TOLERANCE_VALUE(#8565,#8564);
#8569=PLUS_MINUS_TOLERANCE(#8568,#8554);
#8570=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8554,#8562);
#8576=(CHARACTERIZED_OBJECT('CUT',$)FEATURE_DEFINITION()INSTANCED_FEATURE()RE
MOVAL_VOLUME()SHAPE_ASPECT('CUT',$,#38,.T.));
#8577=PROPERTY_DEFINITION('',$,#8576);
#8578=PRODUCT_DEFINITION_SHAPE('',$,#8576);
#8579=CARTESIAN_POINT('',(0.0,0.0,0.0));
#8580=DIRECTION('',(0.0,0.0,1.0));
#8581=DIRECTION('',(1.0,0.0,0.0));
#8582=AXIS2_PLACEMENT_3D('orientation',#8579,#8580,#8581);
#8583=REPRESENTATION('',(#4622,#4642,#4662,#4682,#4702,#4718,#4742,#4762,#478
2,#4802,#4822,#4842,#4862,#4878),#29);
#8584=PROPERTY_DEFINITION_REPRESENTATION(#8577,#8583);
#8585=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8582),#29);
#8586=PROPERTY_DEFINITION_REPRESENTATION(#8577,#8585);
#8587=SHAPE_ASPECT('','volume shape',#38,.T.);
#8588=PROPERTY_DEFINITION('',$,#8587);
#8589=PROPERTY_DEFINITION_REPRESENTATION(#8588,#8583);
#8590=SHAPE_ASPECT('','shape volume occurrence',#8578,.T.);
#8591=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#8587,#8590);
#8592=DIMENSIONAL_SIZE(#4663,'radius');
#8593=PRECISION_QUALIFIER(4);
#8599=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8593))REPRESENTATI
ON_ITEM(''));
#8600=SHAPE_DIMENSION_REPRESENTATION('',(#8599),#29);
#8601=PRECISION_QUALIFIER(4);
#8602=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8603=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8604=MEASURE_QUALIFICATION('','',#8602,(#8601));
#8605=MEASURE_QUALIFICATION('','',#8603,(#8601));
#8606=TOLERANCE_VALUE(#8603,#8602);
#8607=PLUS_MINUS_TOLERANCE(#8606,#8592);
#8608=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8592,#8600);
#8609=DIMENSIONAL_SIZE(#4703,'radius');
#8610=PRECISION_QUALIFIER(4);
#8616=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8610))REPRESENTATI
ON_ITEM(''));
#8617=SHAPE_DIMENSION_REPRESENTATION('',(#8616),#29);
#8618=PRECISION_QUALIFIER(4);
#8619=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8620=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8621=MEASURE_QUALIFICATION('','',#8619,(#8618));
#8622=MEASURE_QUALIFICATION('','',#8620,(#8618));
#8623=TOLERANCE_VALUE(#8620,#8619);
#8624=PLUS_MINUS_TOLERANCE(#8623,#8609);
#8625=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8609,#8617);
#8626=DIMENSIONAL_SIZE(#4879,'radius');
#8627=PRECISION_QUALIFIER(4);
#8633=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8627))REPRESENTATI
ON_ITEM(''));
#8634=SHAPE_DIMENSION_REPRESENTATION('',(#8633),#29);
```

```
#8635=PRECISION_QUALIFIER(4);
#8636=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8637=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8638=MEASURE_QUALIFICATION('','',#8636,(#8635));
#8639=MEASURE_QUALIFICATION('','',#8637,(#8635));
#8640=TOLERANCE_VALUE(#8637,#8636);
#8641=PLUS_MINUS_TOLERANCE(#8640,#8626);
#8642=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8626,#8634);
#8643=DIMENSIONAL_SIZE(#4763,'radius');
#8644=PRECISION_QUALIFIER(4);
#8650=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8644))REPRESENTATI
ON_ITEM(''));
#8651=SHAPE_DIMENSION_REPRESENTATION('',(#8650),#29);
#8652=PRECISION_QUALIFIER(4);
#8653=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8654=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8655=MEASURE_QUALIFICATION('','',#8653,(#8652));
#8656=MEASURE_QUALIFICATION('','',#8654,(#8652));
#8657=TOLERANCE_VALUE(#8654,#8653);
#8658=PLUS_MINUS_TOLERANCE(#8657,#8643);
#8659=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8643,#8651);
#8660=DIMENSIONAL_SIZE(#4803,'radius');
#8661=PRECISION_QUALIFIER(4);
#8667=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8661))REPRESENTATI
ON_ITEM(''));
#8668=SHAPE_DIMENSION_REPRESENTATION('',(#8667),#29);
#8669=PRECISION_QUALIFIER(4);
#8670=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8671=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8672=MEASURE_QUALIFICATION('','',#8670,(#8669));
#8673=MEASURE_QUALIFICATION('','',#8671,(#8669));
#8674=TOLERANCE_VALUE(#8671,#8670);
#8675=PLUS_MINUS_TOLERANCE(#8674,#8660);
#8676=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8660,#8668);
#8677=DIMENSIONAL_SIZE(#4843,'radius');
#8678=PRECISION_QUALIFIER(4);
#8684=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8678))REPRESENTATI
ON_ITEM(''));
#8685=SHAPE_DIMENSION_REPRESENTATION('',(#8684),#29);
#8686=PRECISION_QUALIFIER(4);
#8687=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8688=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8689=MEASURE_QUALIFICATION('','',#8687,(#8686));
#8690=MEASURE_QUALIFICATION('','',#8688,(#8686));
#8691=TOLERANCE_VALUE(#8688,#8687);
#8692=PLUS_MINUS_TOLERANCE(#8691,#8677);
#8693=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8677,#8685);
#8694=CARTESIAN_POINT('',(47.999999999999979,0.0,180.0));
#8695=DIRECTION('',(0.0,0.0,1.0));
#8696=DIRECTION('',(1.0,0.0,0.0));
#8697=AXIS2_PLACEMENT_3D('',#8694,#8695,#8696);
#8698=SHAPE_ASPECT('',$,#38,.F.);
#8699=PROPERTY_DEFINITION('',$,#8698);
```



```
#8700=REPRESENTATION('',(#8697),#29);
#8701=PROPERTY_DEFINITION_REPRESENTATION(#8699,#8700);
#8702=DIMENSIONAL_LOCATION('',$,#4743,#4823);
#8703=PRECISION_QUALIFIER(4);
#8709=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.000000000000007),#10)QUALIFIED_REPRESENTATION_ITEM((#870
3))REPRESENTATION_ITEM(''));
#8710=SHAPE_DIMENSION_REPRESENTATION('',(#8709),#29);
#8711=PRECISION_QUALIFIER(4);
#8712=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8713=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8714=MEASURE_QUALIFICATION('','',#8712,(#8711));
#8715=MEASURE_QUALIFICATION('','',#8713,(#8711));
#8716=TOLERANCE_VALUE(#8713,#8712);
#8717=PLUS_MINUS_TOLERANCE(#8716,#8702);
#8718=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8702,#8710);
#8719=DIMENSIONAL_LOCATION('',$,#4783,#4863);
#8720=PRECISION_QUALIFIER(4);
#8726=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8720))REPRESENTAT
ION_ITEM(''));
#8727=SHAPE_DIMENSION_REPRESENTATION('',(#8726),#29);
#8728=PRECISION_QUALIFIER(4);
#8729=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8730=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8731=MEASURE_QUALIFICATION('','',#8729,(#8728));
#8732=MEASURE_QUALIFICATION('','',#8730,(#8728));
#8733=TOLERANCE_VALUE(#8730,#8729);
#8734=PLUS_MINUS_TOLERANCE(#8733,#8719);
#8735=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8719,#8727);
#8736=CARTESIAN_POINT('',(47.999999999999979,0.0,180.0));
#8737=DIRECTION('',(0.0,0.0,1.0));
#8738=DIRECTION('',(1.0,0.0,0.0));
#8739=AXIS2_PLACEMENT_3D('',#8736,#8737,#8738);
#8740=SHAPE_ASPECT('',$,#38,.F.);
#8741=PROPERTY_DEFINITION('',$,#8740);
#8742=REPRESENTATION('',(#8739),#29);
#8743=PROPERTY_DEFINITION_REPRESENTATION(#8741,#8742);
#8744=DIMENSIONAL_LOCATION('',$,#4683,#8740);
#8745=PRECISION_QUALIFIER(4);
#8751=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(49.999999999999993),#10)QUALIFIED_REPRESENTATION_ITEM((#874
5))REPRESENTATION_ITEM(''));
#8752=SHAPE_DIMENSION_REPRESENTATION('',(#8751),#29);
#8753=PRECISION_QUALIFIER(4);
#8754=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8755=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8756=MEASURE_QUALIFICATION('','',#8754,(#8753));
#8757=MEASURE_QUALIFICATION('','',#8755,(#8753));
#8758=TOLERANCE_VALUE(#8755,#8754);
#8759=PLUS_MINUS_TOLERANCE(#8758,#8744);
#8760=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8744,#8752);
#8761=CARTESIAN_POINT('',(47.999999999999979,0.0,180.0));
#8762=DIRECTION('',(0.0,0.0,1.0));
#8763=DIRECTION('',(1.0,0.0,0.0));
#8764=AXIS2_PLACEMENT_3D('',#8761,#8762,#8763);
```

```
#8765=SHAPE_ASPECT('',$,#38,.F.);
#8766=PROPERTY_DEFINITION('',$,#8765);
#8767=REPRESENTATION('',(#8764),#29);
#8768=PROPERTY_DEFINITION_REPRESENTATION(#8766,#8767);
#8769=DIMENSIONAL_LOCATION('',$,#8765,#4823);
#8770=PRECISION_QUALIFIER(4);
#8776=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.000000000000002),#10)QUALIFIED_REPRESENTATION_ITEM((#877
0))REPRESENTATION_ITEM(''));
#8777=SHAPE_DIMENSION_REPRESENTATION('',(#8776),#29);
#8778=PRECISION_QUALIFIER(4);
#8779=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8780=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8781=MEASURE_QUALIFICATION('','',#8779,(#8778));
#8782=MEASURE_QUALIFICATION('','',#8780,(#8778));
#8783=TOLERANCE_VALUE(#8780,#8779);
#8784=PLUS_MINUS_TOLERANCE(#8783,#8769);
#8785=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8769,#8777);
#8786=DIMENSIONAL_LOCATION('',$,#4643,#4783);
#8787=PRECISION_QUALIFIER(4);
#8793=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.999999999999972),#10)QUALIFIED_REPRESENTATION_ITEM((#878
7))REPRESENTATION_ITEM(''));
#8794=SHAPE_DIMENSION_REPRESENTATION('',(#8793),#29);
#8795=PRECISION_QUALIFIER(4);
#8796=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8797=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8798=MEASURE_QUALIFICATION('','',#8796,(#8795));
#8799=MEASURE_QUALIFICATION('','',#8797,(#8795));
#8800=TOLERANCE_VALUE(#8797,#8796);
#8801=PLUS_MINUS_TOLERANCE(#8800,#8786);
#8802=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8786,#8794);
#8803=DIMENSIONAL_LOCATION('',$,#5534,#4719);
#8804=PRECISION_QUALIFIER(4);
#8810=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(47.999999999999979),#10)QUALIFIED_REPRESENTATION_ITEM((#880
4))REPRESENTATION_ITEM(''));
#8811=SHAPE_DIMENSION_REPRESENTATION('',(#8810),#29);
#8812=PRECISION_QUALIFIER(4);
#8813=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8814=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8815=MEASURE_QUALIFICATION('','',#8813,(#8812));
#8816=MEASURE_QUALIFICATION('','',#8814,(#8812));
#8817=TOLERANCE_VALUE(#8814,#8813);
#8818=PLUS_MINUS_TOLERANCE(#8817,#8803);
#8819=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8803,#8811);
#8820=CARTESIAN_POINT('',(0.0,0.0,180.0));
#8821=DIRECTION('',(0.0,1.0,0.0));
#8822=DIRECTION('',(1.0,0.0,0.0));
#8823=AXIS2_PLACEMENT_3D('',#8820,#8821,#8822);
#8824=SHAPE_ASPECT('',$,#38,.F.);
#8825=PROPERTY_DEFINITION('',$,#8824);
#8826=REPRESENTATION('',(#8823),#29);
#8827=PROPERTY_DEFINITION_REPRESENTATION(#8825,#8826);
#8828=DIMENSIONAL_LOCATION('',$,#4623,#8824);
#8829=PRECISION_QUALIFIER(4);
```



```
#8835=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8829))REPRESENTAT
ION_ITEM(''));
#8836=SHAPE_DIMENSION_REPRESENTATION('',(#8835),#29);
#8837=PRECISION_QUALIFIER(4);
#8838=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8839=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8840=MEASURE_QUALIFICATION('','',#8838,(#8837));
#8841=MEASURE_QUALIFICATION('','',#8839,(#8837));
#8842=TOLERANCE_VALUE(#8839,#8838);
#8843=PLUS_MINUS_TOLERANCE(#8842,#8828);
#8844=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8828,#8836);
#8850=(CHARACTERIZED_OBJECT('CUT',$)FEATURE_DEFINITION()INSTANCED_FEATURE()RE
MOVAL_VOLUME()SHAPE_ASPECT('CUT',$,#38,.T.));
#8851=PROPERTY_DEFINITION('',$,#8850);
#8852=PRODUCT_DEFINITION_SHAPE('',$,#8850);
#8853=CARTESIAN_POINT('',(0.0,0.0,0.0));
#8854=DIRECTION('',(0.0,0.0,1.0));
#8855=DIRECTION('',(1.0,0.0,0.0));
#8856=AXIS2_PLACEMENT_3D('orientation',#8853,#8854,#8855);
#8857=REPRESENTATION('',(#1796,#4902,#4922,#4942,#4966,#4986,#5006,#5030,#505
0,#5070,#5094,#5114,#5134),#29);
#8858=PROPERTY_DEFINITION_REPRESENTATION(#8851,#8857);
#8859=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8856),#29);
#8860=PROPERTY_DEFINITION_REPRESENTATION(#8851,#8859);
#8861=SHAPE_ASPECT('','volume shape',#38,.T.);
#8862=PROPERTY_DEFINITION('',$,#8861);
#8863=PROPERTY_DEFINITION_REPRESENTATION(#8862,#8857);
#8864=SHAPE_ASPECT('','shape volume occurrence',#8852,.T.);
#8865=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#8861,#8864);
#8866=DIMENSIONAL_SIZE(#5051,'radius');
#8867=PRECISION_QUALIFIER(4);
#8873=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8867))REPRESENTATI
ON_ITEM(''));
#8874=SHAPE_DIMENSION_REPRESENTATION('',(#8873),#29);
#8875=PRECISION_QUALIFIER(4);
#8876=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8877=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8878=MEASURE_QUALIFICATION('','',#8876,(#8875));
#8879=MEASURE_QUALIFICATION('','',#8877,(#8875));
#8880=TOLERANCE_VALUE(#8877,#8876);
#8881=PLUS_MINUS_TOLERANCE(#8880,#8866);
#8882=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8866,#8874);
#8883=DIMENSIONAL_SIZE(#4923,'radius');
#8884=PRECISION_QUALIFIER(4);
#8890=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8884))REPRESENTATI
ON_ITEM(''));
#8891=SHAPE_DIMENSION_REPRESENTATION('',(#8890),#29);
#8892=PRECISION_QUALIFIER(4);
#8893=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8894=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8895=MEASURE_QUALIFICATION('','',#8893,(#8892));
#8896=MEASURE_QUALIFICATION('','',#8894,(#8892));
#8897=TOLERANCE_VALUE(#8894,#8893);
```

```
#8898=PLUS_MINUS_TOLERANCE(#8897,#8883);
#8899=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8883,#8891);
#8900=DIMENSIONAL_SIZE(#4987,'radius');
#8901=PRECISION_QUALIFIER(4);
#8907=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8901))REPRESENTATI
ON_ITEM(''));
#8908=SHAPE_DIMENSION_REPRESENTATION('',(#8907),#29);
#8909=PRECISION_QUALIFIER(4);
#8910=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8911=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8912=MEASURE_QUALIFICATION('','',#8910,(#8909));
#8913=MEASURE_QUALIFICATION('','',#8911,(#8909));
#8914=TOLERANCE_VALUE(#8911,#8910);
#8915=PLUS_MINUS_TOLERANCE(#8914,#8900);
#8916=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8900,#8908);
#8917=DIMENSIONAL_SIZE(#5115,'radius');
#8918=PRECISION_QUALIFIER(4);
#8924=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8918))REPRESENTATI
ON_ITEM(''));
#8925=SHAPE_DIMENSION_REPRESENTATION('',(#8924),#29);
#8926=PRECISION_QUALIFIER(4);
#8927=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8928=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8929=MEASURE_QUALIFICATION('','',#8927,(#8926));
#8930=MEASURE_QUALIFICATION('','',#8928,(#8926));
#8931=TOLERANCE_VALUE(#8928,#8927);
#8932=PLUS_MINUS_TOLERANCE(#8931,#8917);
#8933=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8917,#8925);
#8934=CARTESIAN_POINT('',(228.0,-50.0,50.0));
#8935=DIRECTION('',(1.0,0.0,0.0));
#8936=DIRECTION('',(0.0,0.0,1.0));
#8937=AXIS2_PLACEMENT_3D('',#8934,#8935,#8936);
#8938=SHAPE_ASPECT('',$,#38,.F.);
#8939=PROPERTY_DEFINITION('',$,#8938);
#8940=REPRESENTATION('',(#8937),#29);
#8941=PROPERTY_DEFINITION_REPRESENTATION(#8939,#8940);
#8942=CARTESIAN_POINT('',(179.999999999999972,-50.0,178.0));
#8943=DIRECTION('',(0.0,0.0,1.0));
#8944=DIRECTION('',(-1.0,0.0,0.0));
#8945=AXIS2_PLACEMENT_3D('',#8942,#8943,#8944);
#8946=SHAPE_ASPECT('',$,#38,.F.);
#8947=PROPERTY_DEFINITION('',$,#8946);
#8948=REPRESENTATION('',(#8945),#29);
#8949=PROPERTY_DEFINITION_REPRESENTATION(#8947,#8948);
#8950=CARTESIAN_POINT('',(2.0,-50.0,129.999999999999972));
#8951=DIRECTION('',(-1.0,0.0,0.0));
#8952=DIRECTION('',(0.0,0.0,-1.0));
#8953=AXIS2_PLACEMENT_3D('',#8950,#8951,#8952);
#8954=SHAPE_ASPECT('',$,#38,.F.);
#8955=PROPERTY_DEFINITION('',$,#8954);
#8956=REPRESENTATION('',(#8953),#29);
#8957=PROPERTY_DEFINITION_REPRESENTATION(#8955,#8956);
#8958=CARTESIAN_POINT('',(50.0,-50.0,2.0));
#8959=DIRECTION('',(0.0,0.0,-1.0));
```



```
#8960=DIRECTION('',(1.0,0.0,0.0));
#8961=AXIS2_PLACEMENT_3D('',#8958,#8959,#8960);
#8962=SHAPE_ASPECT('',$,#38,.F.);
#8963=PROPERTY_DEFINITION('',$,#8962);
#8964=REPRESENTATION('',(#8961),#29);
#8965=PROPERTY_DEFINITION_REPRESENTATION(#8963,#8964);
#8966=DIMENSIONAL_LOCATION('',$,#5135,#5007);
#8967=PRECISION_QUALIFIER(4);
#8973=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(79.999999999999972),#10)QUALIFIED_REPRESENTATION_ITEM((#896
7))REPRESENTATION_ITEM(''));
#8974=SHAPE_DIMENSION_REPRESENTATION('',(#8973),#29);
#8975=PRECISION_QUALIFIER(4);
#8976=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8977=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8978=MEASURE_QUALIFICATION('','',#8976,(#8975));
#8979=MEASURE_QUALIFICATION('','',#8977,(#8975));
#8980=TOLERANCE_VALUE(#8977,#8976);
#8981=PLUS_MINUS_TOLERANCE(#8980,#8966);
#8982=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8966,#8974);
#8983=CARTESIAN_POINT('',(228.0,-50.0,50.0));
#8984=DIRECTION('',(1.0,0.0,0.0));
#8985=DIRECTION('',(0.0,0.0,1.0));
#8986=AXIS2_PLACEMENT_3D('',#8983,#8984,#8985);
#8987=SHAPE_ASPECT('',$,#38,.F.);
#8988=PROPERTY_DEFINITION('',$,#8987);
#8989=REPRESENTATION('',(#8986),#29);
#8990=PROPERTY_DEFINITION_REPRESENTATION(#8988,#8989);
#8991=DIMENSIONAL_LOCATION('',$,#5095,#8987);
#8992=PRECISION_QUALIFIER(4);
#8998=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(178.000000000000028),#10)QUALIFIED_REPRESENTATION_ITEM((#89
92))REPRESENTATION_ITEM(''));
#8999=SHAPE_DIMENSION_REPRESENTATION('',(#8998),#29);
#9000=PRECISION_QUALIFIER(4);
#9001=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#9002=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#9003=MEASURE_QUALIFICATION('','',#9001,(#9000));
#9004=MEASURE_QUALIFICATION('','',#9002,(#9000));
#9005=TOLERANCE_VALUE(#9002,#9001);
#9006=PLUS_MINUS_TOLERANCE(#9005,#8991);
#9007=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8991,#8999);
#9008=DIMENSIONAL_LOCATION('',$,#2264,#4903);
#9009=PRECISION_QUALIFIER(4);
#9015=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(48.000000000000007),#10)QUALIFIED_REPRESENTATION_ITEM((#900
9))REPRESENTATION_ITEM(''));
#9016=SHAPE_DIMENSION_REPRESENTATION('',(#9015),#29);
#9017=PRECISION_QUALIFIER(4);
#9018=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#9019=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#9020=MEASURE_QUALIFICATION('','',#9018,(#9017));
#9021=MEASURE_QUALIFICATION('','',#9019,(#9017));
#9022=TOLERANCE_VALUE(#9019,#9018);
#9023=PLUS_MINUS_TOLERANCE(#9022,#9008);
#9024=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#9008,#9016);
```

```
#9025=DIMENSIONAL_LOCATION('',$,#2178,#4943);
#9026=PRECISION_QUALIFIER(4);
#9032=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(48.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9026))REPRESENTAT
ION_ITEM(''));
#9033=SHAPE_DIMENSION_REPRESENTATION('',(#9032),#29);
#9034=PRECISION_QUALIFIER(4);
#9035=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#9036=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#9037=MEASURE_QUALIFICATION('','',#9035,(#9034));
#9038=MEASURE_QUALIFICATION('','',#9036,(#9034));
#9039=TOLERANCE_VALUE(#9036,#9035);
#9040=PLUS_MINUS_TOLERANCE(#9039,#9025);
#9041=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#9025,#9033);
#9042=CARTESIAN_POINT('',(0.0,-40.0,0.0));
#9043=DIRECTION('',(0.0,-1.0,0.0));
#9044=DIRECTION('',(1.0,0.0,0.0));
#9045=AXIS2_PLACEMENT_3D('',#9042,#9043,#9044);
#9046=SHAPE_ASPECT('',$,#38,.F.);
#9047=PROPERTY_DEFINITION('',$,#9046);
#9048=REPRESENTATION('',(#9045),#29);
#9049=PROPERTY_DEFINITION_REPRESENTATION(#9047,#9048);
#9050=CARTESIAN_POINT('',(0.0,-50.0,0.0));
#9051=DIRECTION('',(0.0,-1.0,0.0));
#9052=DIRECTION('',(1.0,0.0,0.0));
#9053=AXIS2_PLACEMENT_3D('',#9050,#9051,#9052);
#9054=SHAPE_ASPECT('',$,#38,.F.);
#9055=PROPERTY_DEFINITION('',$,#9054);
#9056=REPRESENTATION('',(#9053),#29);
#9057=PROPERTY_DEFINITION_REPRESENTATION(#9055,#9056);
#9058=DIMENSIONAL_LOCATION('',$,#9046,#9054);
#9059=PRECISION_QUALIFIER(4);
#9065=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9059))REPRESENTAT
ION_ITEM(''));
#9066=SHAPE_DIMENSION_REPRESENTATION('',(#9065),#29);
#9067=PRECISION_QUALIFIER(4);
#9068=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#9069=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#9070=MEASURE_QUALIFICATION('','',#9068,(#9067));
#9071=MEASURE_QUALIFICATION('','',#9069,(#9067));
#9072=TOLERANCE_VALUE(#9069,#9068);
#9073=PLUS_MINUS_TOLERANCE(#9072,#9058);
#9074=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#9058,#9066);
#9080=(CHARACTERIZED_OBJECT('','open
rectangular')FEATURE_DEFINITION()INSTANCED_FEATURE()POCKET()SHAPE_ASPECT('','
open rectangular',#38,.T.));
#9081=PROPERTY_DEFINITION('',$,#9080);
#9082=PRODUCT_DEFINITION_SHAPE('',$,#9080);
#9083=CARTESIAN_POINT('',(66.0,-10.0,90.0));
#9084=DIRECTION('',(0.0,-1.0,0.0));
#9085=DIRECTION('',(0.0,0.0,1.0));
#9086=AXIS2_PLACEMENT_3D('orientation',#9083,#9084,#9085);
#9087=REPRESENTATION('',(#5316,#5366,#5386,#5406,#5426,#5446,#5462),#29);
#9088=PROPERTY_DEFINITION_REPRESENTATION(#9081,#9087);
#9089=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9086),#29);
```



```
#9090=PROPERTY_DEFINITION_REPRESENTATION(#9081,#9089);
#9091=FEATURE_COMPONENT_DEFINITION('',$);
#9092=PRODUCT_DEFINITION_SHAPE('',$,#9091);
#9093=SQUARE_U_PROFILE('square u profile','boundary',#9092,.F.);
#9094=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#9095=STANDARD_UNCERTAINTY('lower limit','minus',-1.0);
#9096=PRECISION_QUALIFIER(4);
#9102=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(32.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9094,#9095,#9096)
)REPRESENTATION_ITEM('width'));
#9107=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
#9112=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#9113=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#9114=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#9115=PRECISION_QUALIFIER(4);
#9121=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.9000000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#911
3,#9114,#9115))REPRESENTATION_ITEM('first radius'));
#9122=STANDARD_UNCERTAINTY('upper limit','plus',0.500000000000000);
#9123=STANDARD_UNCERTAINTY('lower limit','minus',-0.500000000000000);
#9124=PRECISION_QUALIFIER(4);
#9130=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.9000000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#912
2,#9123,#9124))REPRESENTATION_ITEM('second radius'));
#9131=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9086,#9102,#9107,#9112,#9121,
#9130),#29);
#9132=PROPERTY_DEFINITION('',$,#9093);
#9133=PROPERTY_DEFINITION_REPRESENTATION(#9132,#9131);
#9134=FEATURE_COMPONENT_DEFINITION('',$);
#9135=PRODUCT_DEFINITION_SHAPE('',$,#9134);
#9136=PATH_FEATURE_COMPONENT('linear path','linear',#9135,.F.);
#9137=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#9138=STANDARD_UNCERTAINTY('lower limit','minus',-1.0);
#9139=PRECISION_QUALIFIER(4);
#9145=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9137,#9138,#9139)
)REPRESENTATION_ITEM('distance'));
#9146=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9086,#9145),#29);
#9147=PROPERTY_DEFINITION('',$,#9136);
#9148=PROPERTY_DEFINITION_REPRESENTATION(#9147,#9146);
#9149=DIRECTION_SHAPE_REPRESENTATION('',(#9084),#29);
#9150=PROPERTY_DEFINITION_REPRESENTATION(#9147,#9149);
#9151=SHAPE_ASPECT('','open boundary occurrence',#9082,.T.);
#9152=SHAPE_DEFINING_RELATIONSHIP('','profile usage',#9093,#9151);
#9153=SHAPE_ASPECT('','pocket depth occurrence',#9082,.T.);
#9154=SHAPE_DEFINING_RELATIONSHIP('pocket depth','path feature component
usage',#9136,#9153);
#9155=FEATURE_COMPONENT_DEFINITION('',$);
#9156=PRODUCT_DEFINITION_SHAPE('',$,#9155);
#9157=POCKET_BOTTOM('bottom condition','planar',#9156,.F.);
#9158=DESCRIPTIVE_REPRESENTATION_ITEM('pocket bottom orientation','pocket
depth end');
#9159=PROPERTY_DEFINITION('',$,#9157);
#9160=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9158),#29);
```

```
#9161=PROPERTY_DEFINITION_REPRESENTATION(#9159,#9160);
#9162=DIRECTION('',(0.0,1.0,0.0));
#9163=DIRECTION_SHAPE_REPRESENTATION('floor normal',(#9162),#29);
#9164=PROPERTY_DEFINITION_REPRESENTATION(#9159,#9163);
#9165=CARTESIAN_POINT('',(66.0,-30.0,90.0));
#9166=LOCATION_SHAPE_REPRESENTATION('floor location',(#9165),#29);
#9167=PROPERTY_DEFINITION_REPRESENTATION(#9159,#9166);
#9168=SHAPE_ASPECT('','bottom condition occurrence',#9082,.T.);
#9169=FEATURE_COMPONENT_RELATIONSHIP('','pocket bottom usage',#9157,#9168);
#9170=DIMENSIONAL_LOCATION('',$,#75,#5463);
#9171=PRECISION_QUALIFIER(4);
#9177=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(74.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9171))REPRESENTAT
ION_ITEM(''));
#9178=SHAPE_DIMENSION_REPRESENTATION('',(#9177),#29);
#9179=PRECISION_QUALIFIER(4);
#9180=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#9181=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#9182=MEASURE_QUALIFICATION('','',#9180,(#9179));
#9183=MEASURE_QUALIFICATION('','',#9181,(#9179));
#9184=TOLERANCE_VALUE(#9181,#9180);
#9185=PLUS_MINUS_TOLERANCE(#9184,#9170);
#9186=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#9170,#9178);

#9187=EDGE_ROUND('','constant radius',#38,.T.);
#9188=PROPERTY_DEFINITION('',$,#9187);
#9189=FACE_SHAPE_REPRESENTATION('edge round face',(#3724),#29);
#9190=PROPERTY_DEFINITION_REPRESENTATION(#9188,#9189);
#9191=FACE_SHAPE_REPRESENTATION('first face shape',(#1260),#29);
#9192=PROPERTY_DEFINITION_REPRESENTATION(#9188,#9191);
#9193=FACE_SHAPE_REPRESENTATION('second face shape',(#74),#29);
#9194=PROPERTY_DEFINITION_REPRESENTATION(#9188,#9193);
#9195=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9196=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9197=PRECISION_QUALIFIER(4);
#9203=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9195,#9196,#9197))
REPRESENTATION_ITEM('radius'));
#9208=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9213=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9214=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9203,#9208,#9213),#29);
#9215=PROPERTY_DEFINITION_REPRESENTATION(#9188,#9214);


#9220=EDGE_ROUND('','constant radius',#38,.T.);
#9221=PROPERTY_DEFINITION('',$,#9220);
#9222=FACE_SHAPE_REPRESENTATION('edge round face',(#3740),#29);
#9223=PROPERTY_DEFINITION_REPRESENTATION(#9221,#9222);
#9224=FACE_SHAPE_REPRESENTATION('first face shape',(#1260),#29);
#9225=PROPERTY_DEFINITION_REPRESENTATION(#9221,#9224);
#9226=FACE_SHAPE_REPRESENTATION('second face shape',(#142),#29);
#9227=PROPERTY_DEFINITION_REPRESENTATION(#9221,#9226);
#9228=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9229=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
```



```
#9230=PRECISION_QUALIFIER(4);
#9236=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9228,#9229,#9230))
REPRESENTATION_ITEM('radius'));
#9241=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9246=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9247=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9236,#9241,#9246),#29);
#9248=PROPERTY_DEFINITION_REPRESENTATION(#9221,#9247);


#9253=EDGE_ROUND('','constant radius',#38,.T.);
#9254=PROPERTY_DEFINITION('',$,#9253);
#9255=FACE_SHAPE_REPRESENTATION('edge round face',(#3756),#29);
#9256=PROPERTY_DEFINITION_REPRESENTATION(#9254,#9255);
#9257=FACE_SHAPE_REPRESENTATION('first face shape',(#142),#29);
#9258=PROPERTY_DEFINITION_REPRESENTATION(#9254,#9257);
#9259=FACE_SHAPE_REPRESENTATION('second face shape',(#238),#29);
#9260=PROPERTY_DEFINITION_REPRESENTATION(#9254,#9259);
#9261=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9262=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9263=PRECISION_QUALIFIER(4);
#9269=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9261,#9262,#9263))
REPRESENTATION_ITEM('radius'));
#9274=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9279=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9280=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9269,#9274,#9279),#29);
#9281=PROPERTY_DEFINITION_REPRESENTATION(#9254,#9280);

#9286=EDGE_ROUND('','constant radius',#38,.T.);
#9287=PROPERTY_DEFINITION('',$,#9286);
#9288=FACE_SHAPE_REPRESENTATION('edge round face',(#3772),#29);
#9289=PROPERTY_DEFINITION_REPRESENTATION(#9287,#9288);
#9290=FACE_SHAPE_REPRESENTATION('first face shape',(#74),#29);
#9291=PROPERTY_DEFINITION_REPRESENTATION(#9287,#9290);
#9292=FACE_SHAPE_REPRESENTATION('second face shape',(#238),#29);
#9293=PROPERTY_DEFINITION_REPRESENTATION(#9287,#9292);
#9294=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9295=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9296=PRECISION_QUALIFIER(4);
#9302=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9294,#9295,#9296))
REPRESENTATION_ITEM('radius'));
#9307=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9312=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9313=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9302,#9307,#9312),#29);
#9314=PROPERTY_DEFINITION_REPRESENTATION(#9287,#9313);

#9319=EDGE_ROUND('','constant radius',#38,.T.);
#9320=PROPERTY_DEFINITION('',$,#9319);
```

```
#9321=FACE_SHAPE_REPRESENTATION('edge round face',(#3788),#29);
#9322=PROPERTY_DEFINITION_REPRESENTATION(#9320,#9321);
#9323=FACE_SHAPE_REPRESENTATION('first face shape',(#647),#29);
#9324=PROPERTY_DEFINITION_REPRESENTATION(#9320,#9323);
#9325=FACE_SHAPE_REPRESENTATION('second face shape',(#1876),#29);
#9326=PROPERTY_DEFINITION_REPRESENTATION(#9320,#9325);
#9327=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9328=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9329=PRECISION_QUALIFIER(4);
#9335=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9327,#9328,#9329))
REPRESENTATION_ITEM('radius'));
#9340=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9345=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9346=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9335,#9340,#9345),#29);
#9347=PROPERTY_DEFINITION_REPRESENTATION(#9320,#9346);

#9352=EDGE_ROUND('','constant radius',#38,.T.);
#9353=PROPERTY_DEFINITION('',$,#9352);
#9354=FACE_SHAPE_REPRESENTATION('edge round face',(#3804),#29);
#9355=PROPERTY_DEFINITION_REPRESENTATION(#9353,#9354);
#9356=FACE_SHAPE_REPRESENTATION('first face shape',(#647),#29);
#9357=PROPERTY_DEFINITION_REPRESENTATION(#9353,#9356);
#9358=FACE_SHAPE_REPRESENTATION('second face shape',(#1836),#29);
#9359=PROPERTY_DEFINITION_REPRESENTATION(#9353,#9358);
#9360=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9361=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9362=PRECISION_QUALIFIER(4);
#9368=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9360,#9361,#9362))
REPRESENTATION_ITEM('radius'));
#9373=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9378=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9379=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9368,#9373,#9378),#29);
#9380=PROPERTY_DEFINITION_REPRESENTATION(#9353,#9379);

#9385=EDGE_ROUND('','constant radius',#38,.T.);
#9386=PROPERTY_DEFINITION('',$,#9385);
#9387=FACE_SHAPE_REPRESENTATION('edge round face',(#3820),#29);
#9388=PROPERTY_DEFINITION_REPRESENTATION(#9386,#9387);
#9389=FACE_SHAPE_REPRESENTATION('first face shape',(#2153),#29);
#9390=PROPERTY_DEFINITION_REPRESENTATION(#9386,#9389);
#9391=FACE_SHAPE_REPRESENTATION('second face shape',(#2220),#29);
#9392=PROPERTY_DEFINITION_REPRESENTATION(#9386,#9391);
#9393=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9394=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9395=PRECISION_QUALIFIER(4);
#9401=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9393,#9394,#9395))
REPRESENTATION_ITEM('radius'));
#9406=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
```



```
#9411=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9412=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9401,#9406,#9411),#29);
#9413=PROPERTY_DEFINITION_REPRESENTATION(#9386,#9412);

#9418=EDGE_ROUND('','constant radius',#38,.T.);
#9419=PROPERTY_DEFINITION('',$,#9418);
#9420=FACE_SHAPE_REPRESENTATION('edge round face',(#3836),#29);
#9421=PROPERTY_DEFINITION_REPRESENTATION(#9419,#9420);
#9422=FACE_SHAPE_REPRESENTATION('first face shape',(#2134),#29);
#9423=PROPERTY_DEFINITION_REPRESENTATION(#9419,#9422);
#9424=FACE_SHAPE_REPRESENTATION('second face shape',(#2282),#29);
#9425=PROPERTY_DEFINITION_REPRESENTATION(#9419,#9424);
#9426=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9427=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9428=PRECISION_QUALIFIER(4);
#9434=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9426,#9427,#9428))
REPRESENTATION_ITEM('radius'));
#9439=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9444=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9445=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9434,#9439,#9444),#29);
#9446=PROPERTY_DEFINITION_REPRESENTATION(#9419,#9445);

#9451=EDGE_ROUND('','constant radius',#38,.T.);
#9452=PROPERTY_DEFINITION('',$,#9451);
#9453=FACE_SHAPE_REPRESENTATION('edge round face',(#3852),#29);
#9454=PROPERTY_DEFINITION_REPRESENTATION(#9452,#9453);
#9455=FACE_SHAPE_REPRESENTATION('first face shape',(#2196),#29);
#9456=PROPERTY_DEFINITION_REPRESENTATION(#9452,#9455);
#9457=FACE_SHAPE_REPRESENTATION('second face shape',(#2239),#29);
#9458=PROPERTY_DEFINITION_REPRESENTATION(#9452,#9457);
#9459=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#9460=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9461=PRECISION_QUALIFIER(4);
#9467=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9459,#9460,#9461))
REPRESENTATION_ITEM('radius'));
#9472=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9477=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9478=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9467,#9472,#9477),#29);
#9479=PROPERTY_DEFINITION_REPRESENTATION(#9452,#9478);

#9484=EDGE_ROUND('','constant radius',#38,.T.);
#9485=PROPERTY_DEFINITION('',$,#9484);
#9486=FACE_SHAPE_REPRESENTATION('edge round face',(#3868),#29);
#9487=PROPERTY_DEFINITION_REPRESENTATION(#9485,#9486);
#9488=FACE_SHAPE_REPRESENTATION('first face shape',(#2177),#29);
#9489=PROPERTY_DEFINITION_REPRESENTATION(#9485,#9488);
#9490=FACE_SHAPE_REPRESENTATION('second face shape',(#2263),#29);
#9491=PROPERTY_DEFINITION_REPRESENTATION(#9485,#9490);
#9492=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
```

```
#9493=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#9494=PRECISION_QUALIFIER(4);
#9500=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9492,#9493,#9494))
REPRESENTATION_ITEM('radius'));
#9505=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9510=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9511=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9500,#9505,#9510),#29);
#9512=PROPERTY_DEFINITION_REPRESENTATION(#9485,#9511);

#9517=EDGE_ROUND('','constant radius',#38,.T.);
#9518=PROPERTY_DEFINITION('',$,#9517);
#9519=FACE_SHAPE_REPRESENTATION('edge round face',(#5150),#29);
#9520=PROPERTY_DEFINITION_REPRESENTATION(#9518,#9519);
#9521=FACE_SHAPE_REPRESENTATION('first face shape',(#2282),#29);
#9522=PROPERTY_DEFINITION_REPRESENTATION(#9518,#9521);
#9523=FACE_SHAPE_REPRESENTATION('second face shape',(#4966),#29);
#9524=PROPERTY_DEFINITION_REPRESENTATION(#9518,#9523);
#9525=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#9526=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#9527=PRECISION_QUALIFIER(4);
#9533=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9525,#9526,#9527))
REPRESENTATION_ITEM('radius'));
#9538=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9543=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9544=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9533,#9538,#9543),#29);
#9545=PROPERTY_DEFINITION_REPRESENTATION(#9518,#9544);

#9550=EDGE_ROUND('','constant radius',#38,.T.);
#9551=PROPERTY_DEFINITION('',$,#9550);
#9552=FACE_SHAPE_REPRESENTATION('edge round face',(#5166),#29);
#9553=PROPERTY_DEFINITION_REPRESENTATION(#9551,#9552);
#9554=FACE_SHAPE_REPRESENTATION('first face shape',(#2220),#29);
#9555=PROPERTY_DEFINITION_REPRESENTATION(#9551,#9554);
#9556=FACE_SHAPE_REPRESENTATION('second face shape',(#5070),#29);
#9557=PROPERTY_DEFINITION_REPRESENTATION(#9551,#9556);
#9558=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#9559=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#9560=PRECISION_QUALIFIER(4);
#9566=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9558,#9559,#9560))
REPRESENTATION_ITEM('radius'));
#9571=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9576=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9577=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9566,#9571,#9576),#29);
#9578=PROPERTY_DEFINITION_REPRESENTATION(#9551,#9577);

#9583=EDGE_ROUND('','constant radius',#38,.T.);
#9584=PROPERTY_DEFINITION('',$,#9583);
```

```
#9585=FACE_SHAPE_REPRESENTATION('edge round face',(#5182),#29);
#9586=PROPERTY_DEFINITION_REPRESENTATION(#9584,#9585);
#9587=FACE_SHAPE_REPRESENTATION('first face shape',(#2177),#29);
#9588=PROPERTY_DEFINITION_REPRESENTATION(#9584,#9587);
#9589=FACE_SHAPE_REPRESENTATION('second face shape',(#4902),#29);
#9590=PROPERTY_DEFINITION_REPRESENTATION(#9584,#9589);
#9591=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#9592=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#9593=PRECISION_QUALIFIER(4);
#9599=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9591,#9592,#9593))
REPRESENTATION_ITEM('radius'));
#9604=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9609=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9610=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9599,#9604,#9609),#29);
#9611=PROPERTY_DEFINITION_REPRESENTATION(#9584,#9610);

#9616=EDGE_ROUND('','constant radius',#38,.T.);
#9617=PROPERTY_DEFINITION('',$,#9616);
#9618=FACE_SHAPE_REPRESENTATION('edge round face',(#5198),#29);
#9619=PROPERTY_DEFINITION_REPRESENTATION(#9617,#9618);
#9620=FACE_SHAPE_REPRESENTATION('first face shape',(#2134),#29);
#9621=PROPERTY_DEFINITION_REPRESENTATION(#9617,#9620);
#9622=FACE_SHAPE_REPRESENTATION('second face shape',(#5006),#29);
#9623=PROPERTY_DEFINITION_REPRESENTATION(#9617,#9622);
#9624=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#9625=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#9626=PRECISION_QUALIFIER(4);
#9632=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9624,#9625,#9626))
REPRESENTATION_ITEM('radius'));
#9637=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9642=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9643=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9632,#9637,#9642),#29);
#9644=PROPERTY_DEFINITION_REPRESENTATION(#9617,#9643);

#9649=EDGE_ROUND('','constant radius',#38,.T.);
#9650=PROPERTY_DEFINITION('',$,#9649);
#9651=FACE_SHAPE_REPRESENTATION('edge round face',(#5214),#29);
#9652=PROPERTY_DEFINITION_REPRESENTATION(#9650,#9651);
#9653=FACE_SHAPE_REPRESENTATION('first face shape',(#2153),#29);
#9654=PROPERTY_DEFINITION_REPRESENTATION(#9650,#9653);
#9655=FACE_SHAPE_REPRESENTATION('second face shape',(#5030),#29);
#9656=PROPERTY_DEFINITION_REPRESENTATION(#9650,#9655);
#9657=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#9658=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#9659=PRECISION_QUALIFIER(4);
#9665=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9657,#9658,#9659))
REPRESENTATION_ITEM('radius'));
#9670=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
```

```
#9675=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9676=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9665,#9670,#9675),#29);
#9677=PROPERTY_DEFINITION_REPRESENTATION(#9650,#9676);

#9682=EDGE_ROUND('','constant radius',#38,.T.);
#9683=PROPERTY_DEFINITION('',$,#9682);
#9684=FACE_SHAPE_REPRESENTATION('edge round face',(#5230),#29);
#9685=PROPERTY_DEFINITION_REPRESENTATION(#9683,#9684);
#9686=FACE_SHAPE_REPRESENTATION('first face shape',(#2263),#29);
#9687=PROPERTY_DEFINITION_REPRESENTATION(#9683,#9686);
#9688=FACE_SHAPE_REPRESENTATION('second face shape',(#4942),#29);
#9689=PROPERTY_DEFINITION_REPRESENTATION(#9683,#9688);
#9690=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#9691=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#9692=PRECISION_QUALIFIER(4);
#9698=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9690,#9691,#9692))
REPRESENTATION_ITEM('radius'));
#9703=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9708=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9709=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9698,#9703,#9708),#29);
#9710=PROPERTY_DEFINITION_REPRESENTATION(#9683,#9709);

#9715=EDGE_ROUND('','constant radius',#38,.T.);
#9716=PROPERTY_DEFINITION('',$,#9715);
#9717=FACE_SHAPE_REPRESENTATION('edge round face',(#5246),#29);
#9718=PROPERTY_DEFINITION_REPRESENTATION(#9716,#9717);
#9719=FACE_SHAPE_REPRESENTATION('first face shape',(#2239),#29);
#9720=PROPERTY_DEFINITION_REPRESENTATION(#9716,#9719);
#9721=FACE_SHAPE_REPRESENTATION('second face shape',(#5094),#29);
#9722=PROPERTY_DEFINITION_REPRESENTATION(#9716,#9721);
#9723=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
#9724=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#9725=PRECISION_QUALIFIER(4);
#9731=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9723,#9724,#9725))
REPRESENTATION_ITEM('radius'));
#9736=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9741=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9742=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9731,#9736,#9741),#29);
#9743=PROPERTY_DEFINITION_REPRESENTATION(#9716,#9742);

#9748=EDGE_ROUND('','constant radius',#38,.T.);
#9749=PROPERTY_DEFINITION('',$,#9748);
#9750=FACE_SHAPE_REPRESENTATION('edge round face',(#5262),#29);
#9751=PROPERTY_DEFINITION_REPRESENTATION(#9749,#9750);
#9752=FACE_SHAPE_REPRESENTATION('first face shape',(#2196),#29);
#9753=PROPERTY_DEFINITION_REPRESENTATION(#9749,#9752);
#9754=FACE_SHAPE_REPRESENTATION('second face shape',(#5134),#29);
#9755=PROPERTY_DEFINITION_REPRESENTATION(#9749,#9754);
#9756=STANDARD_UNCERTAINTY('upper limit','plus',0.010000000000000);
```



```
#9757=STANDARD_UNCERTAINTY('lower limit','minus',-0.010000000000000);
#9758=PRECISION_QUALIFIER(4);
#9764=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9756,#9757,#9758))
REPRESENTATION_ITEM('radius'));
#9769=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9774=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9775=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9764,#9769,#9774),#29);
#9776=PROPERTY_DEFINITION_REPRESENTATION(#9749,#9775);

#9786=(CHARACTERIZED_OBJECT('','thread')FEATURE_DEFINITION()INSTANCED_FEATURE
()SHAPE_ASPECT('','thread',#38,.T.)THREAD());
#9787=PROPERTY_DEFINITION('',$,#9786);
#9788=CARTESIAN_POINT('',(190.0,0.0,120.0));
#9789=DIRECTION('',(0.0,-1.0,0.0));
#9790=DIRECTION('',(1.0,0.0,0.0));
#9791=AXIS2_PLACEMENT_3D('orientation',#9788,#9789,#9790);
#9792=DESCRIPTIVE_REPRESENTATION_ITEM('thread side','internal');
#9793=DESCRIPTIVE_REPRESENTATION_ITEM('form','UNC');
#9794=DESCRIPTIVE_REPRESENTATION_ITEM('fit class','2B');
#9795=DESCRIPTIVE_REPRESENTATION_ITEM('hand','right');
#9796=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0),#11
)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('number of threads'));
#9797=STANDARD_UNCERTAINTY('upper limit','plus',0.002400000000000);
#9798=STANDARD_UNCERTAINTY('lower limit','minus',0.002400000000000);
#9799=PRECISION_QUALIFIER(4);
#9805=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.219900000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9797
,#9798,#9799))REPRESENTATION_ITEM('pitch diameter'));
#9810=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.250000000000000),#10)REPRESENTATION_ITEM('major
diameter'));
#9811=STANDARD_UNCERTAINTY('upper limit','plus',0.005500000000000);
#9812=STANDARD_UNCERTAINTY('lower limit','minus',0.005500000000000);
#9813=PRECISION_QUALIFIER(4);
#9819=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.201500000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9811
,#9812,#9813))REPRESENTATION_ITEM('minor diameter'));
#9820=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9791,#9792,#9793,#9794,#9795,
#9796,#9805,#9810,#9819),#29);
#9821=PROPERTY_DEFINITION_REPRESENTATION(#9787,#9820);
#9822=SHAPE_DEFINING_RELATIONSHIP('','applied shape',#3499,#9833);
#9823=APPLIED_AREA('',$,#38,.T.);
#9824=PROPERTY_DEFINITION('',$,#9823);
#9829=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(12.699999999999999),#10)REPRESENTATION_ITEM('length'));
#9830=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9791,#9829),#29);
#9831=PROPERTY_DEFINITION_REPRESENTATION(#9824,#9830);
#9832=PRODUCT_DEFINITION_SHAPE('',$,#9786);
#9833=SHAPE_ASPECT('','partial area occurrence',#9832,.T.);
#9834=SHAPE_DEFINING_RELATIONSHIP('','applied area usage',#9823,#9833);
```

```
#9840=(CHARACTERIZED_OBJECT('','thread')FEATURE_DEFINITION()INSTANCED_FEATURE
()SHAPE_ASPECT('','thread',#38,.T.)THREAD());
#9841=PROPERTY_DEFINITION('',$,#9840);
#9842=CARTESIAN_POINT('',(190.0,0.0,60.0));
#9843=DIRECTION('',(0.0,-1.0,0.0));
#9844=DIRECTION('',(1.0,0.0,0.0));
#9845=AXIS2_PLACEMENT_3D('orientation',#9842,#9843,#9844);
#9846=DESCRIPTIVE_REPRESENTATION_ITEM('thread side','internal');
#9847=DESCRIPTIVE_REPRESENTATION_ITEM('form','UNC');
#9848=DESCRIPTIVE_REPRESENTATION_ITEM('fit class','2B');
#9849=DESCRIPTIVE_REPRESENTATION_ITEM('hand','right');
#9850=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0),#11
)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('number of threads'));
#9851=STANDARD_UNCERTAINTY('upper limit','plus',0.002400000000000);
#9852=STANDARD_UNCERTAINTY('lower limit','minus',0.002400000000000);
#9853=PRECISION_QUALIFIER(4);
#9859=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.219900000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9851
,#9852,#9853))REPRESENTATION_ITEM('pitch diameter'));
#9864=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.250000000000000),#10)REPRESENTATION_ITEM('major
diameter'));
#9865=STANDARD_UNCERTAINTY('upper limit','plus',0.005500000000000);
#9866=STANDARD_UNCERTAINTY('lower limit','minus',0.005500000000000);
#9867=PRECISION_QUALIFIER(4);
#9873=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.201500000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9865
,#9866,#9867))REPRESENTATION_ITEM('minor diameter'));
#9874=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9845,#9846,#9847,#9848,#9849,
#9850,#9859,#9864,#9873),#29);
#9875=PROPERTY_DEFINITION_REPRESENTATION(#9841,#9874);
#9876=SHAPE_DEFINING_RELATIONSHIP('','applied shape',#3569,#9887);
#9877=APPLIED_AREA('',$,#38,.T.);
#9878=PROPERTY_DEFINITION('',$,#9877);
#9883=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(12.699999999999999),#10)REPRESENTATION_ITEM('length'));
#9884=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9845,#9883),#29);
#9885=PROPERTY_DEFINITION_REPRESENTATION(#9878,#9884);
#9886=PRODUCT_DEFINITION_SHAPE('',$,#9840);
#9887=SHAPE_ASPECT('','partial area occurrence',#9886,.T.);
#9888=SHAPE_DEFINING_RELATIONSHIP('','applied area usage',#9877,#9887);
#9894=(CHARACTERIZED_OBJECT('','thread')FEATURE_DEFINITION()INSTANCED_FEATURE
()SHAPE_ASPECT('','thread',#38,.T.)THREAD());
#9895=PROPERTY_DEFINITION('',$,#9894);
#9896=CARTESIAN_POINT('',(130.0,0.0,60.0));
#9897=DIRECTION('',(0.0,-1.0,0.0));
#9898=DIRECTION('',(1.0,0.0,0.0));
#9899=AXIS2_PLACEMENT_3D('orientation',#9896,#9897,#9898);
#9900=DESCRIPTIVE_REPRESENTATION_ITEM('thread side','internal');
#9901=DESCRIPTIVE_REPRESENTATION_ITEM('form','UNC');
#9902=DESCRIPTIVE_REPRESENTATION_ITEM('fit class','2B');
#9903=DESCRIPTIVE_REPRESENTATION_ITEM('hand','right');
#9904=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0),#11
)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('number of threads'));
```

```
#9905=STANDARD_UNCERTAINTY('upper limit','plus',0.002400000000000);
#9906=STANDARD_UNCERTAINTY('lower limit','minus',0.002400000000000);
#9907=PRECISION_QUALIFIER(4);
#9913=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.219900000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9905
,#9906,#9907))REPRESENTATION_ITEM('pitch diameter'));
#9918=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.250000000000000),#10)REPRESENTATION_ITEM('major
diameter'));
#9919=STANDARD_UNCERTAINTY('upper limit','plus',0.005500000000000);
#9920=STANDARD_UNCERTAINTY('lower limit','minus',0.005500000000000);
#9921=PRECISION_QUALIFIER(4);
#9927=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.201500000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9919
,#9920,#9921))REPRESENTATION_ITEM('minor diameter'));
#9928=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9899,#9900,#9901,#9902,#9903,
#9904,#9913,#9918,#9927),#29);
#9929=PROPERTY_DEFINITION_REPRESENTATION(#9895,#9928);
#9930=SHAPE_DEFINING_RELATIONSHIP('','applied shape',#3709,#9941);
#9931=APPLIED_AREA('',$,#38,.T.);
#9932=PROPERTY_DEFINITION('',$,#9931);
#9937=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(12.699999999999999),#10)REPRESENTATION_ITEM('length'));
#9938=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9899,#9937),#29);
#9939=PROPERTY_DEFINITION_REPRESENTATION(#9932,#9938);
#9940=PRODUCT_DEFINITION_SHAPE('',$,#9894);
#9941=SHAPE_ASPECT('','partial area occurrence',#9940,.T.);
#9942=SHAPE_DEFINING_RELATIONSHIP('','applied area usage',#9931,#9941);
#9948=(CHARACTERIZED_OBJECT('','thread')FEATURE_DEFINITION()INSTANCED_FEATURE
()SHAPE_ASPECT('','thread',#38,.T.)THREAD());
#9949=PROPERTY_DEFINITION('',$,#9948);
#9950=CARTESIAN_POINT('',(130.0,0.0,120.0));
#9951=DIRECTION('',(0.0,-1.0,0.0));
#9952=DIRECTION('',(1.0,0.0,0.0));
#9953=AXIS2_PLACEMENT_3D('orientation',#9950,#9951,#9952);
#9954=DESCRIPTIVE_REPRESENTATION_ITEM('thread side','internal');
#9955=DESCRIPTIVE_REPRESENTATION_ITEM('form','UNC');
#9956=DESCRIPTIVE_REPRESENTATION_ITEM('fit class','2B');
#9957=DESCRIPTIVE_REPRESENTATION_ITEM('hand','right');
#9958=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0),#11
)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('number of threads'));
#9959=STANDARD_UNCERTAINTY('upper limit','plus',0.002400000000000);
#9960=STANDARD_UNCERTAINTY('lower limit','minus',0.002400000000000);
#9961=PRECISION_QUALIFIER(4);
#9967=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.219900000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9959
,#9960,#9961))REPRESENTATION_ITEM('pitch diameter'));
#9972=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.250000000000000),#10)REPRESENTATION_ITEM('major
diameter'));
#9973=STANDARD_UNCERTAINTY('upper limit','plus',0.005500000000000);
#9974=STANDARD_UNCERTAINTY('lower limit','minus',0.005500000000000);
#9975=PRECISION_QUALIFIER(4);
```

```
#9981=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.201500000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9973
,#9974,#9975))REPRESENTATION_ITEM('minor diameter'));
#9982=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9953,#9954,#9955,#9956,#9957,
#9958,#9967,#9972,#9981),#29);
#9983=PROPERTY_DEFINITION_REPRESENTATION(#9949,#9982);
#9984=SHAPE_DEFINING_RELATIONSHIP('','applied shape',#3639,#9988);
#9985=APPLIED_AREA('',$,#38,.T.);
#9986=PROPERTY_DEFINITION('',$,#9985);
#9988=SHAPE_ASPECT('','',#9994,.T.);
#9991=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(12.6999999999999999),#10)REPRESENTATION_ITEM('length'));
#9992=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9953,#9991),#29);
#9993=PROPERTY_DEFINITION_REPRESENTATION(#9986,#9992);
#9994=PRODUCT_DEFINITION_SHAPE('',$,#9948);
#9995=SHAPE_ASPECT('','partial area occurrence',#9994,.T.);
#9996=SHAPE_DEFINING_RELATIONSHIP('','applied area usage',#9985,#9995);
#9997=DATUM_FEATURE('',$,#38,.T.);
#9998=PROPERTY_DEFINITION('',$,#9997);
#9999=REPRESENTATION('',(#1990),#29);
#10000=PROPERTY_DEFINITION_REPRESENTATION(#9998,#9999);
#10001=DATUM('B',$,#38,.F.,'B');
#10002=SHAPE_ASPECT_RELATIONSHIP('',$,#9997,#10001);
#10003=REFERENCED_MODIFIED_DATUM(2,#10001,.MAXIMUM_MATERIAL_CONDITION.);
#10004=REFERENCED_MODIFIED_DATUM(3,#5534,.MAXIMUM_MATERIAL_CONDITION.);
#10005=PRECISION_QUALIFIER(3);
#10006=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10007=MEASURE_QUALIFICATION('','',#10006,(#10005));
#10012=(GEOMETRIC_TOLERANCE('position','',#10006,#7862)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10013=REFERENCED_MODIFIED_DATUM(2,#5534,.MAXIMUM_MATERIAL_CONDITION.);
#10014=DATUM_REFERENCE(3,#5555);
#10015=PRECISION_QUALIFIER(3);
#10016=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10017=MEASURE_QUALIFICATION('','',#10016,(#10015));
#10021=(GEOMETRIC_TOLERANCE('surface
profile','',#10016,#3289)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#100
13,#10014))SURFACE_PROFILE_TOLERANCE());
#10022=PRECISION_QUALIFIER(3);
#10023=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.500000000000000),#10);
#10024=MEASURE_QUALIFICATION('','',#10023,(#10022));
#10025=CONCENTRICITY_TOLERANCE('concentricity','',#10023,#2405,(#5536));
#10026=TOLERANCE_ZONE_FORM('diametral');
#10027=TOLERANCE_ZONE('',$,#38,.F.,(#10025),#10026);
#10028=PRECISION_QUALIFIER(3);
#10029=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.500000000000000),#10);
#10030=MEASURE_QUALIFICATION('','',#10029,(#10028));
#10035=(GEOMETRIC_TOLERANCE('position','',#10029,#6540)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10036=TOLERANCE_ZONE_FORM('diametral');
#10037=TOLERANCE_ZONE('',$,#38,.F.,(#10035),#10036);
#10038=PRECISION_QUALIFIER(3);
#10039=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10040=MEASURE_QUALIFICATION('','',#10039,(#10038));
```



```
#10045=(GEOMETRIC_TOLERANCE('position','',#10039,#2469)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10046=TOLERANCE_ZONE_FORM('diametral');
#10047=TOLERANCE_ZONE('',$,#38,.F.,(#10045),#10046);
#10048=PRECISION_QUALIFIER(3);
#10049=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10050=MEASURE_QUALIFICATION('','',#10049,(#10048));
#10055=(GEOMETRIC_TOLERANCE('position','',#10049,#2533)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10056=TOLERANCE_ZONE_FORM('diametral');
#10057=TOLERANCE_ZONE('',$,#38,.F.,(#10055),#10056);
#10058=PRECISION_QUALIFIER(3);
#10059=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10060=MEASURE_QUALIFICATION('','',#10059,(#10058));
#10065=(GEOMETRIC_TOLERANCE('position','',#10059,#2599)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10066=TOLERANCE_ZONE_FORM('diametral');
#10067=TOLERANCE_ZONE('',$,#38,.F.,(#10065),#10066);
#10068=PRECISION_QUALIFIER(3);
#10069=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10070=MEASURE_QUALIFICATION('','',#10069,(#10068));
#10075=(GEOMETRIC_TOLERANCE('position','',#10069,#2663)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10076=TOLERANCE_ZONE_FORM('diametral');
#10077=TOLERANCE_ZONE('',$,#38,.F.,(#10075),#10076);
#10078=PRECISION_QUALIFIER(3);
#10079=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10080=MEASURE_QUALIFICATION('','',#10079,(#10078));
#10085=(GEOMETRIC_TOLERANCE('position','',#10079,#2859)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10086=TOLERANCE_ZONE_FORM('diametral');
#10087=TOLERANCE_ZONE('',$,#38,.F.,(#10085),#10086);
#10088=PRECISION_QUALIFIER(3);
#10089=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10090=MEASURE_QUALIFICATION('','',#10089,(#10088));
#10095=(GEOMETRIC_TOLERANCE('position','',#10089,#2859)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10096=TOLERANCE_ZONE_FORM('diametral');
#10097=TOLERANCE_ZONE('',$,#38,.F.,(#10095),#10096);
#10098=PRECISION_QUALIFIER(3);
#10099=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10100=MEASURE_QUALIFICATION('','',#10099,(#10098));
#10105=(GEOMETRIC_TOLERANCE('position','',#10099,#2989)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10106=TOLERANCE_ZONE_FORM('diametral');
#10107=TOLERANCE_ZONE('',$,#38,.F.,(#10105),#10106);
#10108=PRECISION_QUALIFIER(3);
#10109=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10110=MEASURE_QUALIFICATION('','',#10109,(#10108));
```

```
#10115=(GEOMETRIC_TOLERANCE('position','',#10109,#3119)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10116=TOLERANCE_ZONE_FORM('diametral');
#10117=TOLERANCE_ZONE('',$,#38,.F.,(#10115),#10116);
#10118=PRECISION_QUALIFIER(3);
#10119=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10120=MEASURE_QUALIFICATION('','',#10119,(#10118));
#10125=(GEOMETRIC_TOLERANCE('position','',#10119,#2989)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10126=TOLERANCE_ZONE_FORM('diametral');
#10127=TOLERANCE_ZONE('',$,#38,.F.,(#10125),#10126);
#10128=PRECISION_QUALIFIER(3);
#10129=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10130=MEASURE_QUALIFICATION('','',#10129,(#10128));
#10135=(GEOMETRIC_TOLERANCE('position','',#10129,#2989)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10136=TOLERANCE_ZONE_FORM('diametral');
#10137=TOLERANCE_ZONE('',$,#38,.F.,(#10135),#10136);
#10138=DATUM_REFERENCE(3,#5534);
#10139=PRECISION_QUALIFIER(3);
#10140=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10141=MEASURE_QUALIFICATION('','',#10140,(#10139));
#10146=(GEOMETRIC_TOLERANCE('position','',#10140,#3569)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10138))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10147=TOLERANCE_ZONE_FORM('diametral');
#10148=TOLERANCE_ZONE('',$,#38,.F.,(#10146),#10147);
#10149=PRECISION_QUALIFIER(3);
#10150=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10151=MEASURE_QUALIFICATION('','',#10150,(#10149));
#10156=(GEOMETRIC_TOLERANCE('position','',#10150,#3709)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10138))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10157=TOLERANCE_ZONE_FORM('diametral');
#10158=TOLERANCE_ZONE('',$,#38,.F.,(#10156),#10157);
#10159=PRECISION_QUALIFIER(3);
#10160=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10161=MEASURE_QUALIFICATION('','',#10160,(#10159));
#10166=(GEOMETRIC_TOLERANCE('position','',#10160,#3499)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10138))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10167=TOLERANCE_ZONE_FORM('diametral');
#10168=TOLERANCE_ZONE('',$,#38,.F.,(#10166),#10167);
#10169=PRECISION_QUALIFIER(3);
#10170=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10171=MEASURE_QUALIFICATION('','',#10170,(#10169));
#10176=(GEOMETRIC_TOLERANCE('position','',#10170,#3639)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10138))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10177=TOLERANCE_ZONE_FORM('diametral');
#10178=TOLERANCE_ZONE('',$,#38,.F.,(#10176),#10177);
#10179=PRECISION_QUALIFIER(3);
#10180=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
```



```
#10181=MEASURE_QUALIFICATION('','',#10180,(#10179));
#10186=(GEOMETRIC_TOLERANCE('position','',#10180,#3569)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10187=TOLERANCE_ZONE_FORM('diametral');
#10188=TOLERANCE_ZONE('',$,#38,.F.,(#10186),#10187);
#10189=PRECISION_QUALIFIER(3);
#10190=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10191=MEASURE_QUALIFICATION('','',#10190,(#10189));
#10196=(GEOMETRIC_TOLERANCE('position','',#10190,#3709)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10197=TOLERANCE_ZONE_FORM('diametral');
#10198=TOLERANCE_ZONE('',$,#38,.F.,(#10196),#10197);
#10199=PRECISION_QUALIFIER(3);
#10200=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10201=MEASURE_QUALIFICATION('','',#10200,(#10199));
#10206=(GEOMETRIC_TOLERANCE('position','',#10200,#3499)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10207=TOLERANCE_ZONE_FORM('diametral');
#10208=TOLERANCE_ZONE('',$,#38,.F.,(#10206),#10207);
#10209=PRECISION_QUALIFIER(3);
#10210=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10211=MEASURE_QUALIFICATION('','',#10210,(#10209));
#10216=(GEOMETRIC_TOLERANCE('position','',#10210,#3639)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10217=TOLERANCE_ZONE_FORM('diametral');
#10218=TOLERANCE_ZONE('',$,#38,.F.,(#10216),#10217);
#10219=PRECISION_QUALIFIER(3);
#10220=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10221=MEASURE_QUALIFICATION('','',#10220,(#10219));
#10226=(GEOMETRIC_TOLERANCE('position','',#10220,#2729)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10227=TOLERANCE_ZONE_FORM('diametral');
#10228=TOLERANCE_ZONE('',$,#38,.F.,(#10226),#10227);
#10229=PRECISION_QUALIFIER(3);
#10230=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.250000000000000),#10);
#10231=MEASURE_QUALIFICATION('','',#10230,(#10229));
#10236=(GEOMETRIC_TOLERANCE('position','',#10230,#2729)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10237=TOLERANCE_ZONE_FORM('diametral');
#10238=TOLERANCE_ZONE('',$,#38,.F.,(#10236),#10237);
#10239=PRECISION_QUALIFIER(3);
#10240=MEASURE_WITH_UNIT(LENGTH_MEASURE(12.0),#10);
#10241=MEASURE_QUALIFICATION('','',#10240,(#10239));
#10242=PARALLELISM_TOLERANCE('parallelism','',#10240,#1361,(#5543));
#10243=PRECISION_QUALIFIER(3);
#10244=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#10245=MEASURE_QUALIFICATION('','',#10244,(#10243));
#10246=PERPENDICULARITY_TOLERANCE('perpendicularity','',#10244,#6113,(#5529))
;
#10247=PRECISION_QUALIFIER(3);
```

```
#10248=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.500000000000000),#10);
#10249=MEASURE_QUALIFICATION('','',#10248,(#10247));
#10250=POSITION_TOLERANCE('position','',#10248,#1957,(#5543,#10013,#10014));
#10251=TOLERANCE_ZONE_FORM('diametral');
#10252=TOLERANCE_ZONE('',$,#38,.F.,(#10250),#10251);
#10253=PRECISION_QUALIFIER(3);
#10254=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.500000000000000),#10);
#10255=MEASURE_QUALIFICATION('','',#10254,(#10253));
#10256=POSITION_TOLERANCE('position','',#10254,#1917,(#5543,#10013,#10014));
#10257=TOLERANCE_ZONE_FORM('diametral');
#10258=TOLERANCE_ZONE('',$,#38,.F.,(#10256),#10257);
#10259=PRECISION_QUALIFIER(3);
#10260=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.010000000000000),#10);
#10261=MEASURE_QUALIFICATION('','',#10260,(#10259));
#10262=GEOMETRIC_TOLERANCE('flatness','',#10260,#335);
#10263=PRECISION_QUALIFIER(3);
#10264=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.300000000000000),#10);
#10265=MEASURE_QUALIFICATION('','',#10264,(#10263));
#10270=(GEOMETRIC_TOLERANCE('position','',#10264,#143)GEOMETRIC_TOLERANCE_WIT
H_DATUM_REFERENCE((#5529,#10003))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERI
AL_CONDITION.)POSITION_TOLERANCE());
#10271=PRECISION_QUALIFIER(3);
#10272=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.300000000000000),#10);
#10273=MEASURE_QUALIFICATION('','',#10272,(#10271));
#10274=GEOMETRIC_TOLERANCE('flatness','',#10272,#1110);
#10275=PRECISION_QUALIFIER(3);
#10276=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.300000000000000),#10);
#10277=MEASURE_QUALIFICATION('','',#10276,(#10275));
#10282=(GEOMETRIC_TOLERANCE('position','',#10276,#6214)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10283=PRECISION_QUALIFIER(3);
#10284=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10285=MEASURE_QUALIFICATION('','',#10284,(#10283));
#10289=(GEOMETRIC_TOLERANCE('surface
profile','',#10284,#3789)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#100
13,#10014))SURFACE_PROFILE_TOLERANCE());
#10290=PRECISION_QUALIFIER(3);
#10291=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10292=MEASURE_QUALIFICATION('','',#10291,(#10290));
#10296=(GEOMETRIC_TOLERANCE('surface
profile','',#10291,#3805)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#100
13,#10014))SURFACE_PROFILE_TOLERANCE());
#10297=PRECISION_QUALIFIER(3);
#10298=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10299=MEASURE_QUALIFICATION('','',#10298,(#10297));
#10303=(GEOMETRIC_TOLERANCE('surface
profile','',#10298,#2264)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#100
03,#10004))SURFACE_PROFILE_TOLERANCE());
#10304=PRECISION_QUALIFIER(3);
#10305=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10306=MEASURE_QUALIFICATION('','',#10305,(#10304));
#10310=(GEOMETRIC_TOLERANCE('surface
profile','',#10305,#2264)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#100
03,#10004))SURFACE_PROFILE_TOLERANCE());
#10311=PRECISION_QUALIFIER(3);
```

```
#10312=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10313=MEASURE_QUALIFICATION('','',#10312,(#10311));
#10317=(GEOMETRIC_TOLERANCE('surface
profile','',#10312,#4309)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#100
13,#10014))SURFACE_PROFILE_TOLERANCE());
#10318=PRECISION_QUALIFIER(3);
#10319=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10320=MEASURE_QUALIFICATION('','',#10319,(#10318));
#10324=(GEOMETRIC_TOLERANCE('surface
profile','',#10319,#2221)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#100
03,#10004))SURFACE_PROFILE_TOLERANCE());
#10325=PRECISION_QUALIFIER(3);
#10326=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10327=MEASURE_QUALIFICATION('','',#10326,(#10325));
#10331=(GEOMETRIC_TOLERANCE('surface
profile','',#10326,#2221)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#100
03,#10004))SURFACE_PROFILE_TOLERANCE());
#10332=PRECISION_QUALIFIER(3);
#10333=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10334=MEASURE_QUALIFICATION('','',#10333,(#10332));
#10338=(GEOMETRIC_TOLERANCE('surface
profile','',#10333,#2264)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#100
03,#10004))SURFACE_PROFILE_TOLERANCE());
#10339=PRECISION_QUALIFIER(3);
#10340=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10341=MEASURE_QUALIFICATION('','',#10340,(#10339));
#10345=(GEOMETRIC_TOLERANCE('surface
profile','',#10340,#2264)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#100
03,#10004))SURFACE_PROFILE_TOLERANCE());
#10346=PRECISION_QUALIFIER(3);
#10347=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.500000000000000),#10);
#10348=MEASURE_QUALIFICATION('','',#10347,(#10346));
#10353=(GEOMETRIC_TOLERANCE('position','',#10347,#7862)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10354=PRECISION_QUALIFIER(3);
#10355=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10356=MEASURE_QUALIFICATION('','',#10355,(#10354));
#10360=(GEOMETRIC_TOLERANCE('surface
profile','',#10355,#4643)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#100
13,#10014))SURFACE_PROFILE_TOLERANCE());
#10361=PRECISION_QUALIFIER(3);
#10362=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10363=MEASURE_QUALIFICATION('','',#10362,(#10361));
#10367=(GEOMETRIC_TOLERANCE('surface
profile','',#10362,#4783)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#100
13,#10014))SURFACE_PROFILE_TOLERANCE());
#10368=PRECISION_QUALIFIER(3);
#10369=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.010000000000000),#10);
#10370=MEASURE_QUALIFICATION('','',#10369,(#10368));
#10371=GEOMETRIC_TOLERANCE('flatness','',#10369,#5527);
#10372=PRECISION_QUALIFIER(3);
#10373=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10374=MEASURE_QUALIFICATION('','',#10373,(#10372));
```



```
#10378=(GEOMETRIC_TOLERANCE('surface
profile','',#10373,#4309)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#100
13,#10014))SURFACE_PROFILE_TOLERANCE());
#10379=PRECISION_QUALIFIER(3);
#10380=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.800000000000000),#10);
#10381=MEASURE_QUALIFICATION('','',#10380,(#10379));
#10382=SYMMETRY_TOLERANCE('symmetry','',#10380,#9080,(#5550));

ENDSEC;
END-ISO-10303-21;
```

Bibliography

[1] *IDEF0 Federal Information Processing Standards Publication 183, Integration Definition for Functional Modeling (IDEF0)*, FIPS PUB 183, National Institute of Standards and Technology, December 1993.

[2] *ANSI/CAMI 104.0-2001, Part 1, Dimensional Measuring Interface Standard (DMIS) 4.0* - (CAM-I , Inc., Bedford, Texas, 2001).

[3] ANSI B89.3.1-1972 (R2003), *Measurement of Out-Of-Roundness* (Amer. Soc. Mech. Engrs., New York, 1972).

[4] ISO 22093, *Industrial automation systems and integration — Physical device control — Dimensional Measuring Interface Standard (DMIS)*.

Index

ICS 25.040.40

Price based on 310 pages