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

Part 224:

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

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

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



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.

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-224:1999

© ISO 1999

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

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents	Page
1 Scope	1
2 Normative references	2
3 Terms, definitions, and abbreviations	4
4 Information requirements	7
4.1 Units of functionality	7
4.1.1 design_exception	8
4.1.2 feature_definition_item	8
4.1.3 feature_profile	10
4.1.4 manufacturing_feature	10
4.1.5 manufacturing_part_properties	13
4.1.6 manufacturing_process_control_documentation	14
4.1.7 manufacturing_process_requirement_documents	14
4.1.8 measurement_limitations	15
4.1.9 part_administration_data	17
4.1.10 part_model	17
4.1.11 requisitions	18
4.1.12 shape_representation_for_machining	18
4.2 Application objects	19
4.3 Application assertions	170
5 Application interpreted model	204
5.1 Mapping table	204
5.2 AIM EXPRESS short listing	438
6 Conformance requirements	623
Annex A (normative) AIM EXPRESS expanded listing	624
Annex B (normative) AIM short names	766
Annex C (normative) Implementation method specific requirements	779
Annex D (normative) Protocol Implementation Conformance Statement (proforma)	780
Annex E (normative) Information object registration	781
Annex F (informative) Application activity model	782
F.1 Application activity model definitions	783
F.2 Application activity model diagrams	793
Annex G (informative) Application reference model	808
Annex H (informative) AIM EXPRESS-G	831
Annex J (informative) AIM EXPRESS listing	867
Annex K (informative) Application protocol usage guide	868
Bibliography	869
Index	870

Figures

Figure 1 - Data planning model	xix
Figure 2 - Angle_taper	21
Figure 3 - Angular_dimension_tolerance	21
Figure 4 - Angular_dimension_tolerance attribute usage	23
Figure 5 - Angular_size_dimension_tolerance	24
Figure 6 - Angularity_tolerance for a plane surface	25
Figure 7 - Angularity_tolerance for an axis	25
Figure 8 - Block_base_shape	28
Figure 9 - Catalogue_knurl	30
Figure 10 - Catalogue_marking	31
Figure 11 - Catalogue_thread	32
Figure 12 - Chamfer	33
Figure 13 - Circular_boss	34
Figure 14 - Circular_boss with taper	34
Figure 15 - Circular_closed_profile	35
Figure 16 - Circular_offset_pattern	36
Figure 17 - Circular_omit_pattern	37
Figure 18 - Circular_pattern with rotation of base feature	38
Figure 19 - Circular_pattern without rotation of base feature	39
Figure 20 - Circular_runout_tolerance	41
Figure 21 - Circularity_tolerance	42
Figure 22 - Complete_circular_path	43
Figure 23 - Compound_feature	44
Figure 24 - Concentricity_tolerance	46
Figure 25 - Conical_hole_bottom	47
Figure 26 - Constant_radius_edge_round	48
Figure 27 - Constant_radius_fillet	49
Figure 28 - Counterbore_hole	50
Figure 29 - Countersunk_hole	51
Figure 30 - Curved_dimension_tolerance	52
Figure 31 - Cylindrical_base_shape	54
Figure 32 - Cylindricity_tolerance	55
Figure 33 - Defined_marking	58
Figure 34 - Thread and Defined_thread attributes	59
Figure 35 - Diagonal_knurl	61
Figure 36 - Diameter_dimension_tolerance	62
Figure 37 - Diameter_taper	63
Figure 38 - Diamond_knurl	64
Figure 39 - Distance_along_curve_tolerance	66
Figure 40 - Edge_round	67
Figure 41 - Fillet	70
Figure 42 - Flat_hole_bottom and Flat_with_radius_hole_bottom	71
Figure 43 - Flat_slot_end_type	72
Figure 44 - Flatness_tolerance	73
Figure 45 - General_boss	74
Figure 46 - General_closed_profile	75
Figure 47 - General_open_profile	76
Figure 48 - General_outside_profile	77
Figure 49 - General_path	77
Figure 50 - General_pattern	79
Figure 51 - General_pocket	79
Figure 52 - General_removal_volume	80
Figure 53 - General_revolution for outer shape	81

Figure 54 - General_revolution for volume removal	82
Figure 55 - Groove	85
Figure 56 - Implicit_base_shape_representation	87
Figure 57 - Knurl	88
Figure 58 - Linear_path	89
Figure 59 - Linear_profile	89
Figure 60 - Linear_profile_tolerance	91
Figure 61 - Location_dimension_tolerance	91
Figure 62 - Ngon_base_shape	96
Figure 63 - Ngon_profile	98
Figure 64 - Open_slot_end_type	100
Figure 65 - Outer_diameter	102
Figure 66 - Outer_diameter_to_shoulder	103
Figure 67 - Parallelism_tolerance for a plane	105
Figure 68 - Parallelism_tolerance for an axis	105
Figure 69 - Partial_circular_path	110
Figure 70 - Partial_circular_profile	111
Figure 71 - Perpendicularity_tolerance	113
Figure 72 - Planar_face	116
Figure 73 - Position_tolerance	120
Figure 74 - Protrusion	125
Figure 75 - Radial_dimension_tolerance	126
Figure 76 - Radiused_slot_end_type	126
Figure 77 - Rectangular_closed_pocket	127
Figure 78 - Rectangular_closed_profile	128
Figure 79 - Rectangular_offset_pattern	129
Figure 80 - Rectangular_omit_pattern	130
Figure 81 - Rectangular_open_pocket	131
Figure 82 - Rectangular_pattern	132
Figure 83 - Revolved_flat	137
Figure 84 - Revolved_round	137
Figure 85 - Round_hole	138
Figure 86 - Rounded_end	140
Figure 87 - Rounded_U_profile	141
Figure 88 - Slot	145
Figure 89 - Spherical_cap	148
Figure 90 - Spherical_hole_bottom	148
Figure 91 - Square_U_profile	149
Figure 92 - Step	151
Figure 93 - Straight_knurl	152
Figure 94 - Straightness_tolerance	153
Figure 95 - Surface_profile_tolerance	154
Figure 96 - Symmetry_tolerance	155
Figure 97 - Tee_profile	157
Figure 98 - Thread	159
Figure 99 - Through_bottom_condition	162
Figure 100 - Total_runout_tolerance	165
Figure 101 - Turned_knurl	167
Figure 102 - Vee_profile	169
Figure 103 - Slot with Woodruff_slot_end_type at each end	170
Figure 104 - Pattern offset required instances	529
Figure 105 - Pattern omit required instances	533
Figure F.1 - IDEF0 Basic notation	782
Figure F.2 - Mechanical products definition for process planning using machining features	794
Figure F.3 - A0 manufacture mechanical parts	795

Figure F.4 - A1 manage manufacturing process	796
Figure F.5 - A13 manage equipment and materials	797
Figure F.6 - A131 manage inventory	798
Figure F.7 - A2 capture digital product definition	799
Figure F.8 - A21 capture part data definition	800
Figure F.9 - A214 create part model	801
Figure F.10 - A3 generate manufacturing data	802
Figure F.11 - A31 generate process plan	803
Figure F.12 - A311 define resources	804
Figure F.13 - A312 define operator data	805
Figure F.14 - A315 define machine instructions	806
Figure F.15 - A4 operate shop floor	807
Figure G.1 - ARM diagram 1 of 22 in EXPRESS-G	809
Figure G.2 - ARM diagram 2 of 22 in EXPRESS-G	810
Figure G.3 - ARM diagram 3 of 22 in EXPRESS-G	811
Figure G.4 - ARM diagram 4 of 22 in EXPRESS-G	812
Figure G.5 - ARM diagram 5 of 22 in EXPRESS-G	813
Figure G.6 - ARM diagram 6 of 22 in EXPRESS-G	814
Figure G.7 - ARM diagram 7 of 22 in EXPRESS-G	815
Figure G.8 - ARM diagram 8 of 22 in EXPRESS-G	816
Figure G.9 - ARM diagram 9 of 22 in EXPRESS-G	817
Figure G.10 - ARM diagram 10 of 22 in EXPRESS-G	818
Figure G.11 - ARM diagram 11 of 22 in EXPRESS-G	819
Figure G.12 - ARM diagram 12 of 22 in EXPRESS-G	820
Figure G.13 - ARM diagram 13 of 22 in EXPRESS-G	821
Figure G.14 - ARM diagram 14 of 22 in EXPRESS-G	822
Figure G.15 - ARM diagram 15 of 22 in EXPRESS-G	823
Figure G.16 - ARM diagram 16 of 22 in EXPRESS-G	824
Figure G.17 - ARM diagram 17 of 22 in EXPRESS-G	825
Figure G.18 - ARM diagram 18 of 22 in EXPRESS-G	826
Figure G.19 - ARM diagram 19 of 22 in EXPRESS-G	827
Figure G.20 - ARM diagram 20 of 22 in EXPRESS-G	828
Figure G.21 - ARM diagram 21 of 22 in EXPRESS-G	829
Figure G.22 - ARM diagram 22 of 22 in EXPRESS-G	830
Figure H.1 - application context - AIM EXPRESS-G diagram 1 of 35	832
Figure H.2 - product definition - AIM EXPRESS-G diagram 2 of 35	833
Figure H.3 - property definition - AIM EXPRESS-G diagram 3 of 35	834
Figure H.4 - shape representation - AIM EXPRESS-G diagram 4 of 35	835
Figure H.5 - representation - AIM EXPRESS-G diagram 5 of 35	836
Figure H.6 - shape aspect - AIM EXPRESS-G diagram 6 of 35	837
Figure H.7 - machining feature - AIM EXPRESS-G diagram 7 of 35	838
Figure H.8 - replicate feature - AIM EXPRESS-G diagram 8 of 35	839
Figure H.9 - transition feature - AIM EXPRESS-G diagram 9 of 35	840
Figure H.10 - geometry topology - AIM EXPRESS-G diagram 10 of 35	841
Figure H.11 - point - AIM EXPRESS-G diagram 11 of 35	842
Figure H.12 - geometric orientation - AIM EXPRESS-G diagram 12 of 35	843
Figure H.13 - curve - AIM EXPRESS-G diagram 13 of 35	844
Figure H.14 - bounded curve - AIM EXPRESS-G diagram 14 of 35	845
Figure H.15 - surface curve - AIM EXPRESS-G diagram 15 of 35	846
Figure H.16 - surface - AIM EXPRESS-G diagram 16 of 35	847
Figure H.17 - elementary surface - AIM EXPRESS-G diagram 17 of 35	848
Figure H.18 - bounded surface - AIM EXPRESS-G diagram 18 of 35	849
Figure H.19 - topology - AIM EXPRESS-G diagram 19 of 35	850
Figure H.20 - shell - AIM EXPRESS-G diagram 20 of 35	851
Figure H.21 - document - AIM EXPRESS-G diagram 21 of 35	852

Figure H.22 - approval - AIM EXPRESS-G diagram 22 of 35	853
Figure H.23 - person and organization - AIM EXPRESS-G diagram 23 of 35	854
Figure H.24 - person and organization assignment - AIM EXPRESS-G diagram 24 of 35	855
Figure H.25 - date - AIM EXPRESS-G diagram 25 of 35	856
Figure H.26 - action - AIM EXPRESS-G diagram 26 of 35	857
Figure H.27 - security classification - AIM EXPRESS-G diagram 27 of 35	858
Figure H.28 - units - AIM EXPRESS-G diagram 28 of 35	859
Figure H.29 - measure with units - AIM EXPRESS-G diagram 29 of 35	860
Figure H.30 - measures - AIM EXPRESS-G diagram 30 of 35	861
Figure H.31 - datums - AIM EXPRESS-G diagram 31 of 35	862
Figure H.32 - tolerances - AIM EXPRESS-G diagram 32 of 35	863
Figure H.33 - geometric tolerances - AIM EXPRESS-G diagram 33 of 35	864
Figure H.34 - tolerance zone - AIM EXPRESS-G diagram 34 of 35	865
Figure H.35 - material - AIM EXPRESS-G diagram 35 of 35	866

Tables

Table 1 - Mapping table for design_exception UoF	206
Table 2 - Mapping table for feature_definition_item UoF	209
Table 3 - Mapping table for feature_profile UoF	237
Table 4 - Mapping table for manufacturing_feature UoF	267
Table 5 - Mapping table for manufacturing_part_properties UoF	382
Table 6 - Mapping table for manufacturing_process_control_documentation UoF	388
Table 7 - Mapping table for manufacturing_process_requirement_documents UoF	394
Table 8 - Mapping table for measurement_limitations UoF	395
Table 9 - Mapping table for part_administration_data UoF	422
Table 10 - Mapping table for part_model UoF	424
Table 11 - Mapping table for requisitions UoF	429
Table 12 - Mapping table for shape_representation_for_machining UoF	431
Table B.1 - AIM short names of entities	766

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.

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.

International Standard ISO 10303-224 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

ISO 10303 consist of the following parts under the general title *Industrial automation systems and integration* — *Product data representation and exchange*:

- Part 1, Overview and fundamental principles;
- Part 11, Description methods: The EXPRESS language reference manual;
- Part 12, Description method: The EXPRESS-I language reference manual;
- Part 21, Implementation methods: Clear text encoding of the exchange structure;
- Part 22, Implementation method: Standard data access interface specification;
- Part 23, Implementation method: C++ language binding to the standard data access interface;
- Part 24, Implementation method: C language binding to the standard data access interface;
- Part 26, Implementation method: Interface definition language binding to the standard data access;
- Part 31, Conformance testing methodology and framework: General concepts;
- Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;
- Part 34, Conformance testing methodology and framework: Abstract test methods;
- Part 35, Conformance testing methodology and framework: Abstract test methods for SDAI implementations;

- Part 41, Integrated generic resources: Fundamentals of product description and support;
- Part 42, Integrated generic resources: Geometric and topological representation;
- Part 43, Integrated generic resources: Representation structures;
- Part 44, Integrated generic resources: Product structure configuration;
- Part 45, Integrated generic resource: Materials;
- Part 46, Integrated generic resources: Visual presentation;
- Part 47, Integrated generic resource: Shape variation tolerances;
- Part 49, Integrated generic resource: Process structure and properties;
- Part 101, Integrated application resource: Draughting;
- Part 104, Integrated application resource: Finite element analysis;
- Part 105, Integrated application resource: Kinematics;
- Part 106, Integrated application resource: Building construction core model;
- Part 201, Application protocol: Explicit draughting;
- Part 202, Application protocol: Associative draughting;
- Part 203, Application protocol: Configuration controlled design;
- Part 204, Application protocol: Mechanical design using boundary representation;
- Part 205, Application protocol: Mechanical design using surface representation;
- Part 207, Application protocol: Sheet metal die planning and design;
- Part 208, Application protocol: Life cycle management - Change process;
- Part 209, Application protocol: Composite and metallic structural analysis and related design;
- Part 210, Application protocol: Electronic assembly, interconnect, and packaging design;
- Part 212, Application protocol: Electrotechnical design and installation
- Part 213, Application protocol: Numerical control process plans for machined parts;

- Part 214, Application protocol: Core data for automotive mechanical design processes;
- Part 215, Application protocol: Ship arrangement;
- Part 216, Application protocol: Ship moulded forms;
- Part 217, Application protocol: Ship piping;
- Part 218, Application protocol: Ship structures;
- Part 221, Application protocol: Functional data and their schematic representation for process plant;
- Part 222, Application protocol: Exchange of product data for composite structures;
- Part 223, Application protocol: Exchange of design and manufacturing product information for casting parts;
- Part 224, Application protocol: Mechanical product definition for process plans using machining features;
- Part 225, Application protocol: Building elements using explicit shape representation;
- Part 226, Application protocol: Ship mechanical systems;
- Part 227, Application protocol: Plant spatial configuration;
- Part 229, Application protocol: Exchange of design and manufacturing product information for forged parts;
- Part 230, Application protocol: Building structural frame: Steelwork;
- Part 231, Application protocol: Process engineering data: Process design and process specification of major equipment;
- Part 232, Application protocol: Technical data packaging core information and exchange;
- Part 301, Abstract test suite: Explicit draughting;
- Part 302, Abstract test suite: Associative draughting;
- Part 303, Abstract test suite: Configuration controlled design;
- Part 304, Abstract test suite: Mechanical design using boundary representation;
- Part 305, Abstract test suite: Mechanical design using surface representation;

- Part 307, Abstract test suite: Sheet metal die planning and design;
- Part 308, Abstract test suite: Life cycle management - Change process;
- Part 309, Abstract test suite: Composite and metallic structural analysis and related design;
- Part 310, Abstract test suite: Electronic assembly, interconnect, and packaging design;
- Part 312, Abstract test suite: Electrotechnical design and installation;
- Part 313, Abstract test suite: Numerical control process plans for machined parts;
- Part 314, Abstract test suite: Core data for automotive mechanical design processes;
- Part 315, Abstract test suite: Ship arrangement;
- Part 316, Abstract test suite: Ship moulded forms;
- Part 317, Abstract test suite: Ship piping;
- Part 318, Abstract test suite: Ship structures;
- Part 321, Abstract test suite: Functional data and their schematic representation for process plant;
- Part 322, Abstract test suite: Exchange of product data for composite structures;
- Part 323, Abstract test suite: Exchange of design and manufacturing product information for casting parts;
- Part 324, Abstract test suite: Mechanical product definition for process plans using machining features;
- Part 325, Abstract test suite: Building elements using explicit shape representation;
- Part 326, Abstract test suite: Ship mechanical systems;
- Part 327, Abstract test suite: Plant spatial configuration;
- Part 329, Abstract test suite: Exchange of design and manufacturing product information for forged parts;
- Part 330, Abstract test suite: Building structural frame: Steelwork;
- Part 331, Abstract test suite: Process engineering data: Process design and process specification of major equipment;

- Part 332, Abstract test suite: Technical data packaging core information and exchange;
- Part 501, Application interpreted construct: Edge-based wireframe;
- Part 502, Application interpreted construct: Shell-based wireframe;
- Part 503, Application interpreted construct: Geometrically bounded 2D wireframe;
- Part 504, Application interpreted construct: Draughting annotation;
- Part 505, Application interpreted construct: Drawing structure and administration;
- Part 506, Application interpreted construct: Draughting elements;
- Part 507, Application interpreted construct: Geometrically bounded surface;
- Part 508, Application interpreted construct: Non-manifold surface;
- Part 509, Application interpreted construct: Manifold surface;
- Part 510, Application interpreted construct: Geometrically bounded wireframe;
- Part 511, Application interpreted construct: Topologically bounded surface;
- Part 512, Application interpreted construct: Faceted boundary representation;
- Part 513, Application interpreted construct: Elementary boundary representation;
- Part 514, Application interpreted construct: Advanced boundary representation;
- Part 515, Application interpreted construct: Constructive solid geometry;
- Part 517, Application interpreted construct: Mechanical design geometric presentation;
- Part 518, Application interpreted construct: Mechanical design shaded presentation.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of the International Standard reflects its structure:

- Parts 11 to 12 specify the description methods,
- Parts 21 to 26 specify the implementation methods,
- Parts 31 to 35 specify the conformance testing methodology and framework,
- Parts 41 to 49 specify the integrated generic resources,

- Parts 101 to 106 specify the integrated application resources,
- Parts 201 to 232 specify the application protocols,
- Parts 301 to 332 specify the abstract test suites, and
- Parts 501 to 518 specify the application interpreted constructs.

Should further parts be published, they will follow the same numbering pattern.

Annex A, B, C, D, and E form an integral part of this part of ISO 10303. Annexes F, G, H, J, and K are for information only.

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-224:1999

Introduction

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

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

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

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

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

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

Clause 1 defines the scope of the application protocol and summarizes the functionality and data covered by the AP. Clause 3 lists the words defined in this part of ISO 10303 and gives pointers to words defined elsewhere. An application activity model that is the basis for the definition of the scope is provided in annex

F. The information requirements of the application are specified in clause 4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in annex G.

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

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).

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-224:1999



Industrial automation systems and integration — Product data representation and exchange — Part 224: Application protocol: Mechanical product definition for process planning using machining features

1 Scope

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for the representation and exchange of information needed to define product data necessary for manufacturing single piece mechanical parts. The product data is based on existing part designs that have their shapes represented by machining features. This part supports digital representation for computer integrated manufacturing.

NOTE - The application activity model in annex F provides a graphical representation of the process and information flows which are the basis for the definition of the scope of this Part of ISO 10303.

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

- the manufacture of a single piece mechanical part;
- parts that are to be manufactured by either milling or turning processes;
- machining features for defining shapes necessary for manufacturing;

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

- Explicit representation of the 3D shape of machining features through bounded geometry representations;
- implicit representation of machining features through selection of standard parameters;
- machining feature definition elements necessary for creating machining form features;
- customer order administrative data to track receipt of an order for a part to the shop floor, but not including tracking of the order on the shop floor;
- approval data to authorize the manufacture of a part;
- requisition administrative data to identify requirements and track the status of materials and equipment needed to manufacture a part;

- work order data to track and identify the status of a part;
- tracking the state of raw stock for documenting the manufacturing history of a part;
- tracking a design exception notice of a part.

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

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

- results from process planning functions;
- exchange of data within process planning systems;
- feature order or sequence;
- representation of assemblies;
- representation of composite material parts;
- representation of sheet metal parts;
- representation of part pedigree;
- design features of a part;
- schedule for completing a work order through the manufacturing process;
- configuration control for a part.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 1101:1983, *Technical drawings - Geometric tolerancing - Tolerance of form, orientation, location and run-out - Generalities definitions, symbols, indications on drawings*.

ISO 5459:1981, *Technical drawings - Geometric tolerancing - Datums and datum-systems for geometric tolerances.*

ISO/IEC 8824-1:1994, *Information Technology - Open Systems Interconnection - Abstract Syntax Notation One (ASN.1) - Part 1: Specification of Basic Notation.*

ISO 10303-1:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles.*

ISO 10303-11:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual.*

ISO 10303-21:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure.*

ISO 10303-31:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 31: Conformance testing methodology and framework: General concepts.*

ISO 10303-41:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 41: Integrated generic resources: Fundamentals of product description and support.*

ISO 10303-42:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 42: Integrated generic resources: Geometric and topological representation.*

ISO 10303-43:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 43: Integrated generic resources: Representation structures.*

ISO 10303-44:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 44: Integrated generic resources: Product structure configuration.*

ISO 10303-45:1998, *Industrial automation systems and integration - Product data representation and exchange - Part 45: Integrated generic resources: Materials.*

ISO 10303-47:1997, *Industrial automation systems and integration - Product data representation and exchange - Part 47: Integrated generic resources: Shape variation tolerances.*

3 Terms, definitions, and abbreviations

3.1

Terms defined in ISO 1101

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

- dimension;
- tolerance.

3.2

Terms defined in ISO 5459

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

- datum.

3.3

Terms defined in ISO 10303-1

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

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

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

3.4

Terms defined in ISO 10303-31

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

- conformance class;
- conformance testing.

3.5

Terms defined in ISO 10303-42

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

- boundary representation (B-rep) solid model;
- manifold solid boundary representation.

3.6

Terms defined in ISO 10303-213

For the purpose of this part of ISO 10303, the following terms defined in ISO 10303-213 apply.

- process plan;
- workstation.

3.7

Other definitions

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

3.7.1

basic angle

is implied where center lines of features in a pattern or surfaces shown on a drawing are located or defined by the use of basic dimensions and no angle is specified.

3.7.2**cutting tool**

an instrument used to remove material from a part.

3.7.3**dedicated fixture**

a fixture designed for a particular part.

3.7.4**fixture**

a device to hold and locate a part during manufacturing and inspection.

3.7.5**indirect material**

consumable material used during the manufacture of a part.

EXAMPLE - Oil, cutting fluid, pallets, brooms, shop towels, etc.

3.7.6**machine**

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

3.7.7**modular fixture**

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

3.8**Abbreviations**

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

AAM	application activity model
AIC	application interpreted construct
AIM	application interpreted model
AP	application protocol
ARM	application reference model
B-rep	boundary representation
ICOM	input, control, output, mechanism

Ngon	N number of sides polygon
PICS	protocol implementation conformance statement
UoF	unit of functionality

4 Information requirements

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

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

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

NOTE 2 - The information requirements correspond to those of the activities identified as being in the scope of the application protocol in annex F.

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

4.1 Units of functionality

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

- design_exception;
- feature_definition_item;
- feature_profile;
- manufacturing_feature;
- manufacturing_part_properties;
- manufacturing_process_control_documentation;
- manufacturing_process_requirement_documents;

- measurement_limitations;
- part_administration_data;
- part_model;
- requisitions;
- shape_representation_for_machining.

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

4.1.1 design_exception

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

The following application objects are used by the design_exception UoF:

- Design_exception_notice;
- Engineering_change_order;
- Engineering_change_proposal.

4.1.2 feature_definition_item

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

The following application objects are used by the feature_definition_item UoF:

- Angle_taper;
- Blind_bottom_condition;
- Boss_top_condition;
- Chamfer_angle;
- Circular_path;
- Complete_circular_path;

- Conical_hole_bottom;
- Diameter_taper;
- First_offset;
- Flat_hole_bottom;
- Flat_slot_end_type;
- Flat_with_radius_hole_bottom;
- General_path;
- General_pocket_bottom_condition;
- General_top_condition;
- Linear_path;
- Open_slot_end_type;
- Partial_area_definition;
- Partial_circular_path;
- Path;
- Planar_pocket_bottom_condition;
- Planar_top_condition;
- Pocket_bottom_condition;
- Radiused_slot_end_type;
- Second_chamfer_offset;
- Second_offset;
- Slot_end_type;
- Spherical_hole_bottom;
- Through_bottom_condition;

- Through_pocket_bottom_condition;
- Woodruff_slot_end_type.

4.1.3 feature_profile

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

The following application objects are used by the feature_profile UoF:

- Circular_closed_profile;
- Closed_profile;
- General_closed_profile;
- General_open_profile;
- Linear_profile;
- Ngon_profile;
- Open_profile;
- Partial_circular_profile;
- Profile;
- Rectangular_closed_profile;
- Rounded_U_profile;
- Square_U_profile;
- Tee_profile;
- Vee_profile.

4.1.4 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:

- Boss;
- Catalogue_knurl;
- Catalogue_marking;
- Catalogue_thread;
- Chamfer;
- Circular_boss;
- Circular_offset_pattern;
- Circular_omit_pattern;
- Circular_pattern;
- Compound_feature;
- Compound_feature_element;
- Compound_feature_relationship;
- Constant_radius_edge_round;
- Constant_radius_fillet;
- Counterbore_hole;
- Countersunk_hole;
- Defined_marking;
- Defined_thread;
- Diagonal_knurl;
- Diamond_knurl;
- Edge_round;
- Fillet;

- General_boss;
- General_outside_profile;
- General_pattern;
- General_pocket;
- General_removal_volume;
- General_revolution;
- Groove;
- Hole;
- Knurl;
- Machining_feature;
- Manufacturing_feature;
- Marking;
- Outer_diameter;
- Outer_diameter_to_shoulder;
- Outer_round;
- Planar_face;
- Pocket;
- Protrusion;
- Rectangular_closed_pocket;
- Rectangular_offset_pattern;
- Rectangular_omit_pattern;
- Rectangular_open_pocket;
- Rectangular_pattern;

- Replicate_base;
- Replicate_feature;
- Revolved_feature;
- Revolved_flat;
- Revolved_round;
- Round_hole;
- Rounded_end;
- Slot;
- Spherical_cap;
- Step;
- Straight_knurl;
- Thread;
- Transition_feature;
- Turned_knurl.

4.1.5 manufacturing_part_properties

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

The following application objects are used by the manufacturing_part_properties UoF:

- Alternate_material;
- Descriptive_parameter;
- Hardness;
- Material;

- Material_property;
- Numeric_parameter;
- Numeric_parameter_with_tolerance;
- Part_property;
- Process_property;
- Property;
- Property_parameter;
- Surface_property.

4.1.6 manufacturing_process_control_documentation

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

The following application objects are used by the manufacturing_process_control_documentation UoF:

- Customer_order;
- Digital_technical_data_package_work_order;
- Ordered_part;
- Pedigree_creation_order;
- Project_order;
- Resource_acquisition_order;
- Shop_work_order.

4.1.7 manufacturing_process_requirement_documents

The manufacturing_process_requirement_documents UoF provides the ability to specify documents that are directly related to product data as they support the definition of the product. These documents may be specific to an operation on the part being manufactured or a property of the part at a particular stage in the

manufacturing process. The documents may, but need not be, company, customer, military, national, or international standards.

The following application objects are used by the manufacturing_process_requirement_documents UoF:

- Specification;
- Specification_usage_constraint.

4.1.8 measurement_limitations

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

The following application objects are used by the measurement_limitations UoF:

- Angular_dimension_tolerance;
- Angular_size_dimension_tolerance;
- Angularity_tolerance;
- Circular_runout_tolerance;
- Circularity_tolerance;
- Compound_datum;
- Concentricity_tolerance;
- Curved_dimension_tolerance;
- Cylindricity_tolerance;
- Datum;
- Datum_feature;
- Datum_target;
- Datum_target_set;
- Diameter_dimension_tolerance;

- Dimensional_tolerance;
- Distance_along_curve_tolerance;
- Externally_defined_tolerance_value;
- Flatness_tolerance;
- Geometric_tolerance;
- Geometric_tolerance_precedence_relationship;
- Linear_profile_tolerance;
- Location_tolerance;
- Location_dimension_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;
- Tolerance_limit;
- Tolerance_range;
- Tolerance_value;
- Tolerance_zone;
- Tolerance_zone_definition;
- Total_runout_tolerance.

4.1.9 part_administration_data

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

The following application objects are used by the part_administration_data UoF:

- Approval;
- Organization;
- Person;
- Person_in_organization.

4.1.10 part_model

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

The following application object is used by the part_model UoF:

- Part.

4.1.11 requisitions

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

The following application objects are used by the requisitions UoF:

- Cutting_tool_requisition;
- Dedicated_fixture_requisition;
- Indirect_stock_requisition;
- Machine_requisition;
- Material_requisition;
- Modular_fixture_requisition;
- Requisition.

4.1.12 shape_representation_for_machining

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

The following application objects are used by the shape_representation_for_machining UoF:

- Base_shape;
- Block_base_shape;
- B-rep_model;
- B-rep_model_element;
- B-rep_shape_aspect_representation;
- B-rep_shape_representation;
- Cylindrical_base_shape;
- Direction_element;

- Explicit_base_shape_representation;
- Face_shape_element;
- Implicit_base_shape_representation;
- Location_element;
- Ngon_base_shape;
- Orientation;
- Path_element;
- Shape;
- Shape_aspect;
- Shape_element.

4.2 Application objects

This subclause specifies the application objects for the mechanical product definition for process planning using machining features application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

4.2.1 Alternate_material

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

- alternate_ranking;
- material_substitute.

4.2.1.1 alternate_ranking

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

4.2.1.2 material_substitute

The material_substitute specifies the material to be used as the substitute. See 4.3.1 for the application assertion.

4.2.2 Angle_taper

An Angle_taper is a constant change in shape of a feature for a part. An Angle_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 Angle_taper.

NOTE - Figure 2 illustrates a Round_hole that is applying an Angle_taper.

The data associated with an Angle_taper are the following: angle. The angle specifies the amount of slope from the start of an Angle_taper to the end of the Angle_taper. See 4.3.2 for the application assertion. The angle shall be greater than -90 degrees and less than 90 degrees. A negative angle indicates a taper which decreases along the length of the taper, and a positive angle indicates a taper which increases along the length of the taper.

4.2.3 Angular_dimension_tolerance

An Angular_dimension_tolerance is a type of Location_tolerance (see 4.2.99) that defines the allowable variation in the angle between two elements of the shape of a part. Each Angular_dimension_tolerance shall have an origin shape and a termination shape.

NOTE - Figure 3 illustrates the Angular_dimension_tolerance and Figure 4 illustrates use of the attributes.

The data associated with an Angular_dimension_tolerance are the following:

- major_angle;
- mirror_origin;
- mirror_termination.

4.2.3.1 major_angle

The major_angle specifies a boolean value that indicates the size of the angle which defines a 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 (origin and termination) that are related to the Angular_dimension_tolerance. This attribute is used in combination with the other attributes to determine the angle that is being limited.

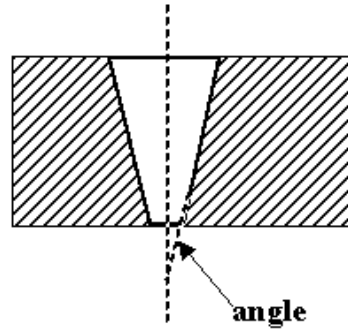


Figure 2 - Angle_taper

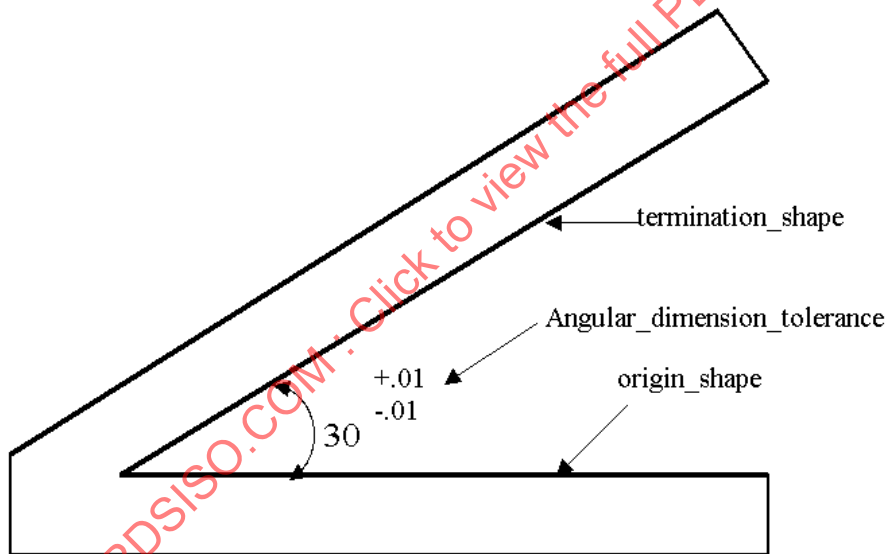


Figure 3 - Angular_dimension_tolerance

4.2.3.2 mirror_origin

The mirror_origin specifies a boolean value that indicates whether the element of the part's shape defining the origin of the limitation is to be mirrored in order to define the angle whose allowable variation is being specified. A value of true indicates that the origin element is mirrored about a plane that it is orthogonal to.

4.2.3.3 mirror_termination

The mirror_termination specifies a boolean value that indicates whether the element of the part's shape that defines the termination of the limitation is to be mirrored in order to define the angle whose allowable variation is being specified. A value of true indicates that the termination element is mirrored about a plane that it is orthogonal to.

4.2.4 Angular_size_dimension_tolerance

An Angular_size_dimension_tolerance is a type of Size_tolerance (see 4.2.172) that specifies the allowable variation on the size or gap formed by two angular elements of the shape of a part.

NOTE - Figure 5 illustrates the Angular_size_dimension_tolerance.

The data associated with an Angular_size_dimension_tolerance are the following:

- full_or_half;
- major_angle.

4.2.4.1 full_or_half

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

4.2.4.2 major_angle

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

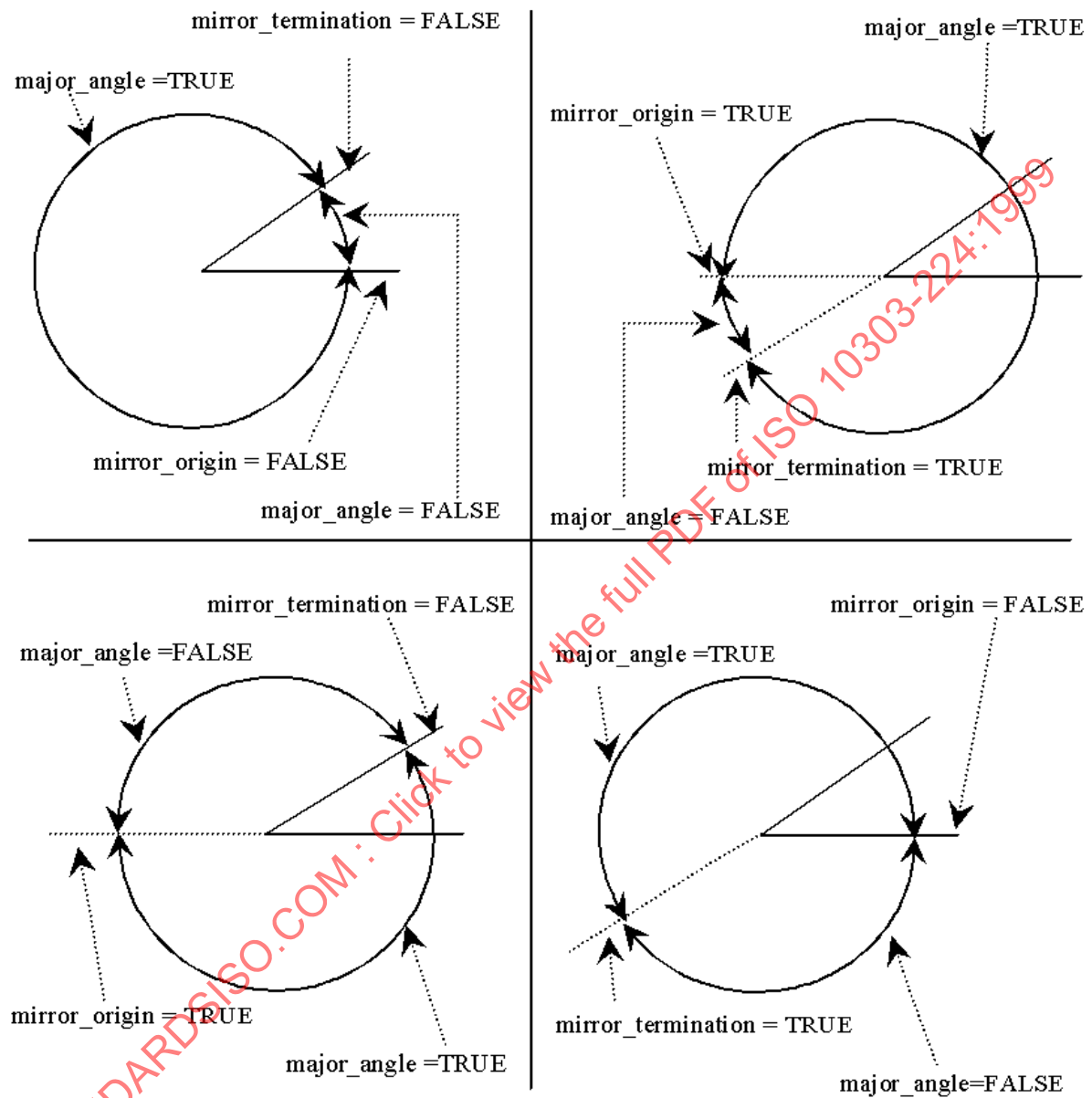


Figure 4 - Angular_dimension_tolerance attribute usage

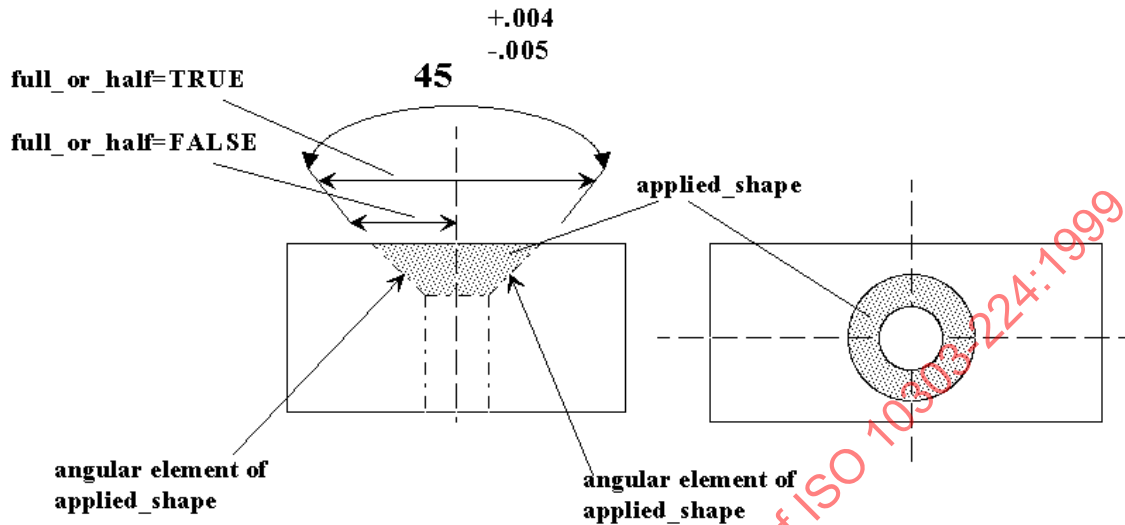


Figure 5 - Angular_size_dimension_tolerance

4.2.5 Angularity_tolerance

An Angularity_tolerance is a type of Geometric_tolerance (see 4.2.86) that is the allowable variation of a surface or axis at a specified angle (other than 90 degrees) from a datum plane or axis. An Angularity_tolerance constrains a shape that is one of the following:

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

NOTE 1 - Figure 6 illustrates the Angularity_tolerance for a plane surface.

NOTE 2 - Figure 7 illustrates Angularity_tolerance for an axis.

NOTE 3 - The Angularity_tolerance definition is derived from paragraph 14.9 of ISO 1101.

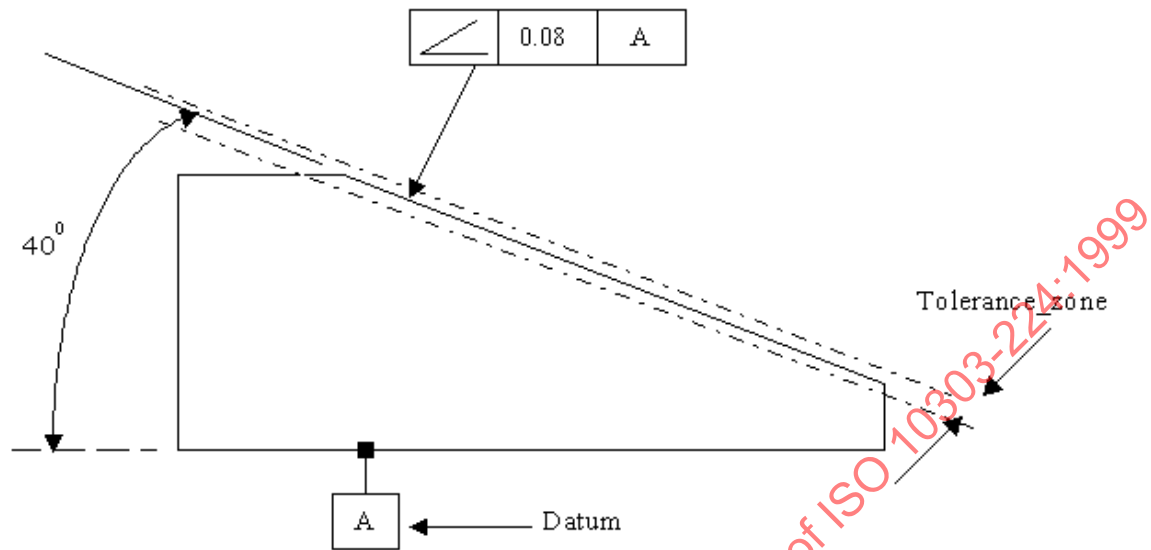


Figure 6 - Angularity_tolerance for a plane surface

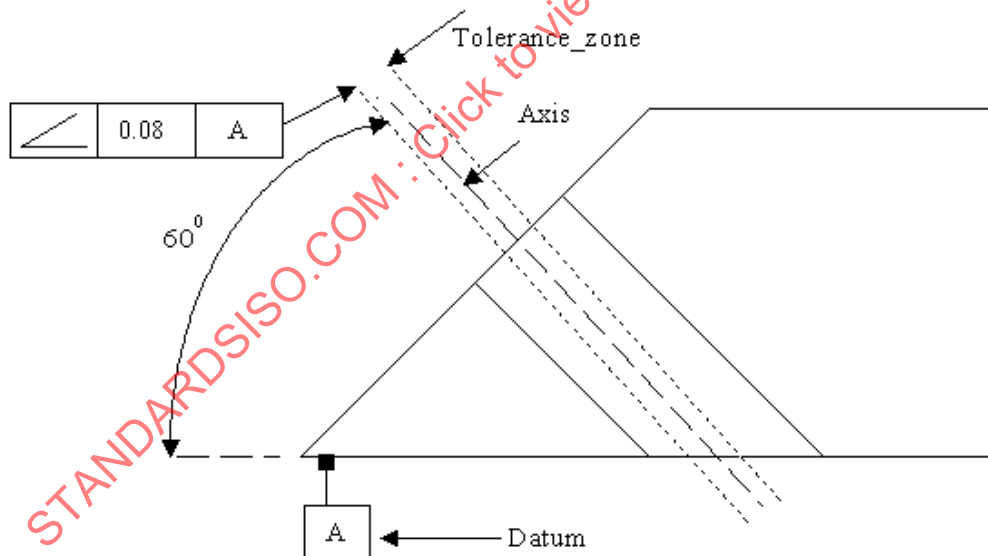


Figure 7 - Angularity_tolerance for an axis

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.

4.2.5.2 segment_size

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

4.2.6 Approval

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

- approval_authority;
- approval_date;
- status.

4.2.6.1 approval_authority

The approval_authority specifies the responsible person and company for making the approval. There may be more than one approval_authority for an Approval. See 4.3.4 for the application assertion.

4.2.6.2 approval_date

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

4.2.6.3 status

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

- approved;
- disapproved;

- not_yet_approved;
- withdrawn.

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

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

4.2.6.3.3 not_yet_approved: a concurrence has not yet been given.

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

4.2.7 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 4.2.66) or an Implicit_base_shape_representation (see 4.2.91).

4.2.8 Blind_bottom_condition

A Blind_bottom_condition is a selection type of Hole_bottom_condition_select that has a finite depth and does not go through the entire part. Each Blind_bottom_condition is either a Conical_hole_bottom (see 4.2.36), Flat_hole_bottom (see 4.2.71), Flat_with_radius_hole_bottom (see 4.2.73), or a Spherical_hole_bottom (see 4.2.178).

The data associated with a Blind_bottom_condition are the following: start_or_end. The start_or_end specifies a boolean value of TRUE if the Blind_bottom_condition is positioned at the bottom 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 4.2.91) that describes the initial shape of the material as a rectangular cross section of some determined length.

NOTE - Figure 8 illustrates a Block_base_shape.

The data associated with a Block_base_shape are the following:

- height;
- width.

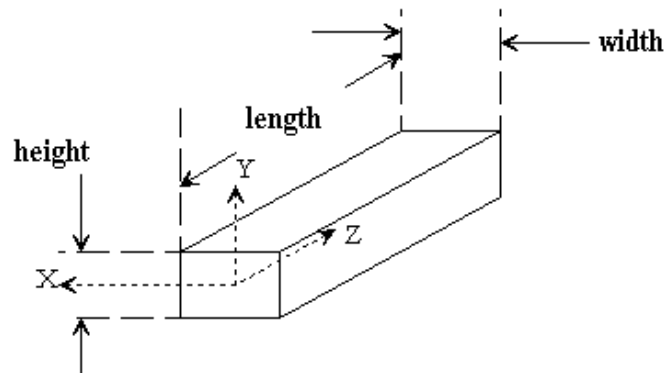


Figure 8 - Block_base_shape

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.

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 Machining_feature (see 4.2.101) that is a closed shape that protrudes from a base 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 4.2.21), or a General_boss (see 4.2.75). The Boss may be positioned on the face of a part with the Z-axis in the direction away from the part, or at the top of the Boss with the Z-axis in the direction toward the part face. The data associated with a Boss are the following:

- boss_height;
- fillet_radius;
- top_condition.

4.2.10.1 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 of a Boss 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 4.2.85) or a Planar_top_condition (see 4.2.136). The data associated with a Boss_top_condition are the following: start_or_end. The start_or_end specifies a boolean value of TRUE if the Boss_top_condition is located at the start of the Boss, FALSE if it is located at the end of the Boss.

4.2.12 B-rep_model

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

4.2.13 B-rep_model_element

A B-rep_model_element is a portion of a boundary representation. The data associated with a B-rep_model_element are the following: element. The element specifies the portion of the B-rep_model that defines the B-rep_model_element. See 4.3.9 for the application assertion.

4.2.14 B-rep_shape_aspect_representation

A B-rep_shape_aspect_representation is a point of interest with respect to the boundary representation of a part. The data associated with a B-rep_shape_aspect_representation are the following: shape_definition. The shape_definition specifies the shape that is the boundary representation. See 4.3.10 for the application assertion.

4.2.15 B-rep_shape_representation

A B-rep_shape_representation is a grouping of boundary representations to define a shape. The data associated with a B-rep_shape_representation are the following: shape_definition. The shape_definition specifies the shape that is the boundary representation. See 4.3.11 for the application assertion.

4.2.16 Catalogue_knurl

A Catalogue_knurl is a type of Knurl (see 4.2.93) that is a reference to a document containing the information to create a knurl on a part.

NOTE - Figure 9 illustrates a Catalogue_knurl that is a diamond knurl with further definition found in a specification called ANSI B94.6.

The data associated with a Catalogue_knurl are the following: documentation. The documentation specifies the document that defines information pertaining to a Knurl feature. See 4.3.12 for the application assertion.

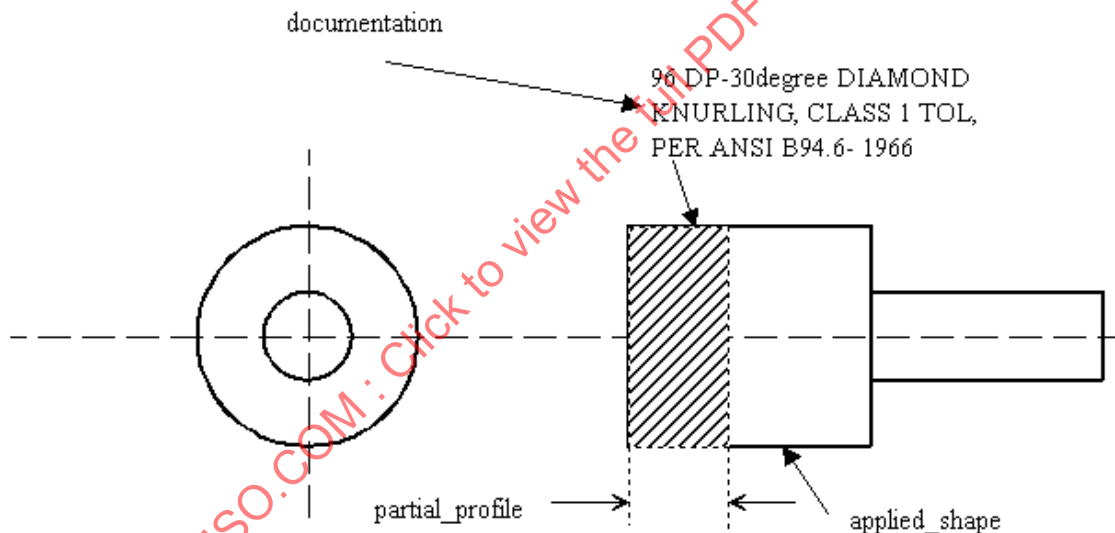


Figure 9 - Catalogue_knurl

4.2.17 Catalogue_marking

A Catalogue_marking is a type of Marking (see 4.2.103) that is a reference to a document containing the information for marking on a surface of a part.

NOTE - Figure 10 illustrates a Catalogue_marking with text of '1928-12345ED34' and further definition found in a specification called ANSI/SAE-AS478-2.

The data associated with a Catalogue_marking are the following: documentation. The documentation specifies the document that defines information pertaining to a Marking feature. See 4.3.13 for the application assertion.

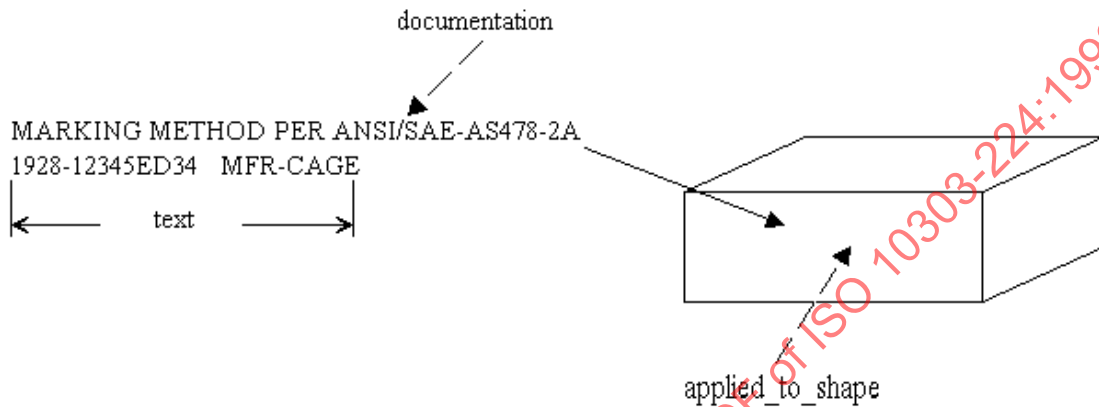


Figure 10 - Catalogue_marking

4.2.18 Catalogue_thread

A Catalogue_thread is a type of Thread (see 4.2.192) that is a reference to a document containing the information to create threads on a part.

NOTE - Figure 11 illustrates a Catalogue_thread that is an outer thread.

The data associated with a Catalogue_thread are the following: documentation. The documentation specifies the document that defines information pertaining to a Thread feature. See 4.3.14 for the application assertion.

4.2.19 Chamfer

A Chamfer is a type of Transition_feature (see 4.2.201) that is a transition between corresponding edges of two joining non-coplanar surfaces, having a flat cross section. A Chamfer feature requires an offset length from one face, and either an angular amount from the same surface or an offset length from a second face.

NOTE - Figure 12 illustrates a Chamfer applied to a rectangular block.

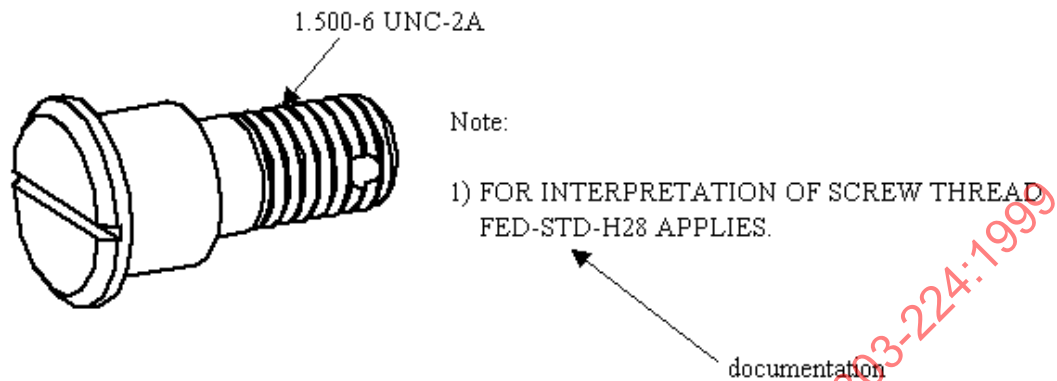


Figure 11 - Catalogue_thread

The data associated with a Chamfer are the following:

- chamfer_face;
- first_face_offset;
- second_face_offset.

4.2.19.1 chamfer_face

The chamfer_face specifies the flat transition face between the two edges of two faces. See 4.3.15 for the application assertion.

4.2.19.2 first_face_offset

The first_face_offset specifies a face and offset amount for one of the edges of the Chamfer. See 4.3.16 for the application assertion.

4.2.19.3 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 4.3.17 for the application assertion.

4.2.20 Chamfer_angle

A Chamfer_angle is a type of Second_chamfer_offset (see 4.2.166) that is the angular amount about from the Second_offset face to the Chamfer face for creating a Chamfer feature.

NOTE - Figure 12 illustrates a Chamfer that has the angle defined by a Chamfer_angle .

The data associated with a Chamfer_angle are the following: angle_amount. The Angle_amount specifies the angular measurement from a face for creating a Chamfer feature. See 4.3.18 for the application assertion.

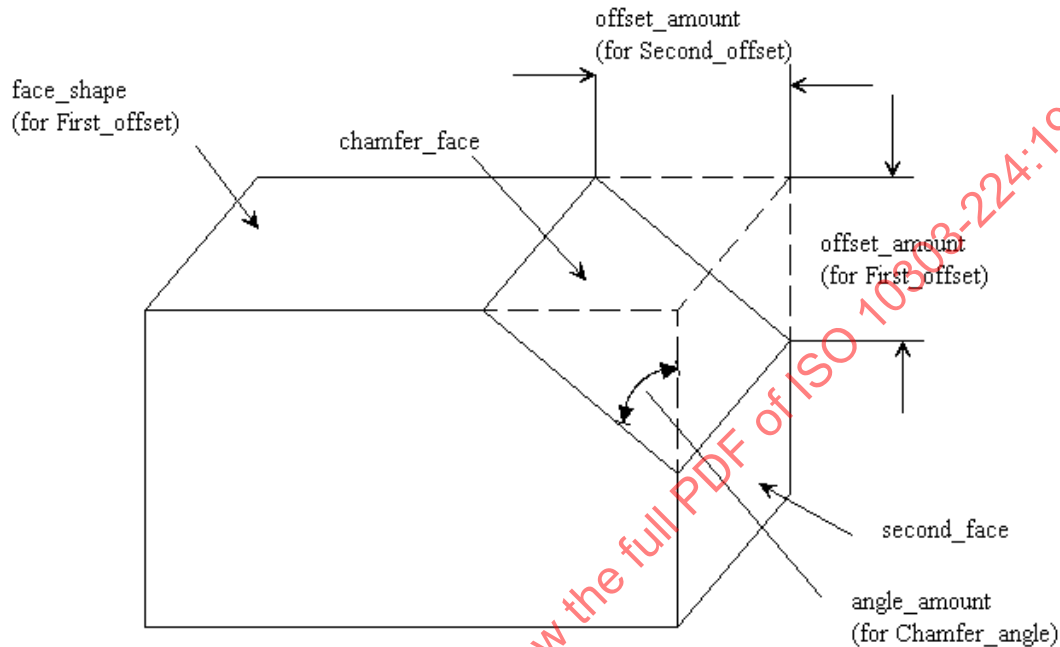


Figure 12 - Chamfer

4.2.21 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 - Figure 13 illustrates a Circular_boss with a planar top and a fillet radius, and Figure 14 illustrates a Circular_boss with taper.

The data associated with a Circular_boss are the following:

- change_in_diameter;
- circular_profile.

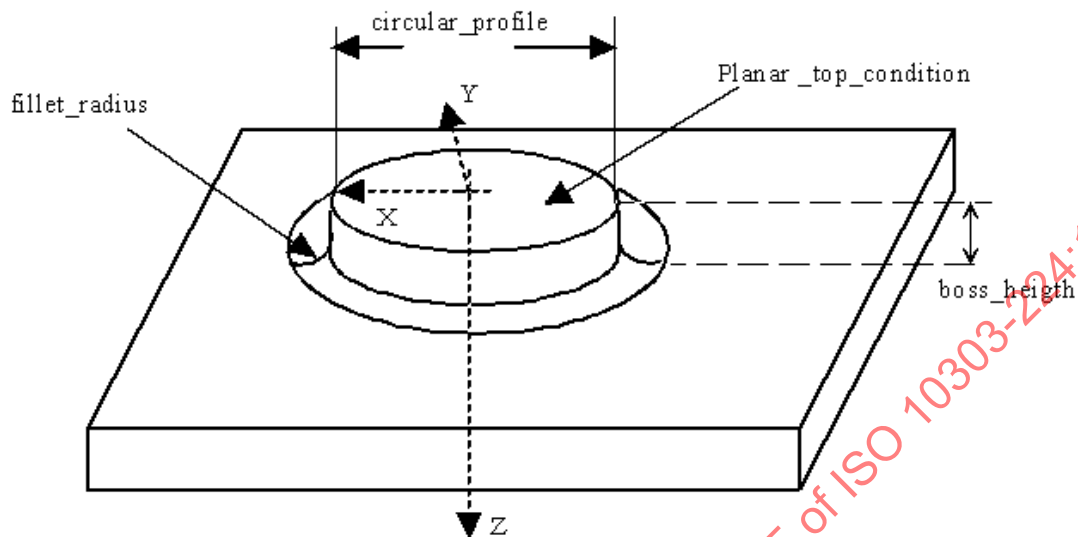


Figure 13 - Circular boss

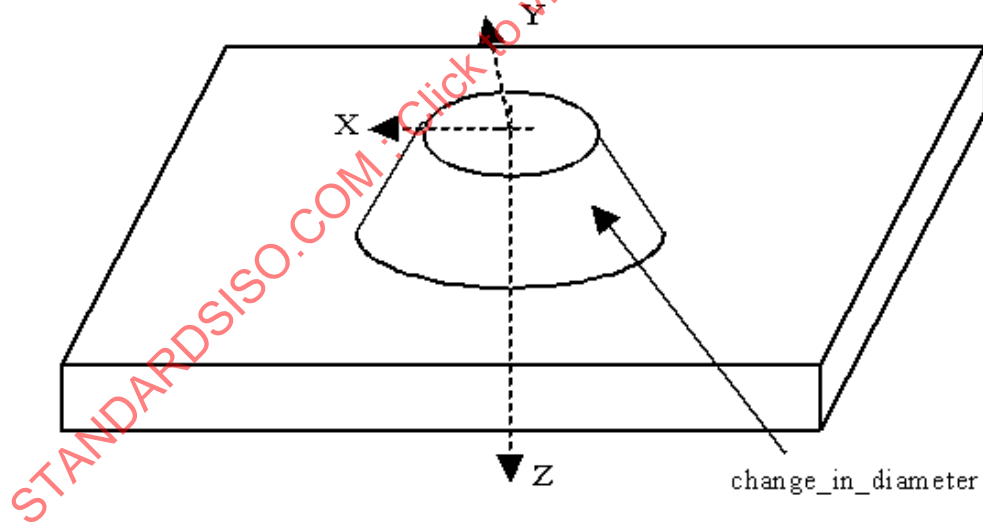


Figure 14 - Circular boss with taper

4.2.21.1 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 4.3.21 and 4.3.20 for the application assertion.

4.2.21.2 circular_profile

The `Circular_closed_profile` specifies the diameter required by a `Circular_boss`. The diameter is the distance across the `Circular_boss`. This placement of the profile shall be with the origin of the `Circular_closed_profile` at the origin of the `Circular_boss`. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the `Circular_boss`. See 4.3.19 for the application assertion.

4.2.22 Circular_closed_profile

A `Circular_closed_profile` is a type of `Closed_profile` (see 4.2.29) that is an enclosed area bounded by a circle. The orientation is at the center of the circle.

NOTE - Figure 15 illustrates a `Circular_closed_profile`.

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

— diameter.

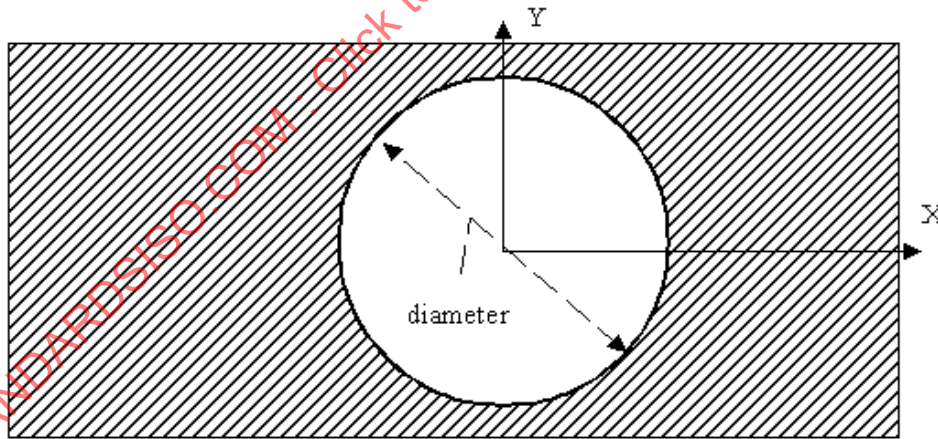


Figure 15 - `Circular_closed_profile`

4.2.22.1 diameter

The diameter specifies the distance across the `Circular_closed_profile`. See 4.3.22 for the application assertion.

4.2.23 Circular_offset_pattern

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

NOTE - Figure 16 illustrates a `Circular_offset_pattern`. The rectangular base shape is offset from index number seven.

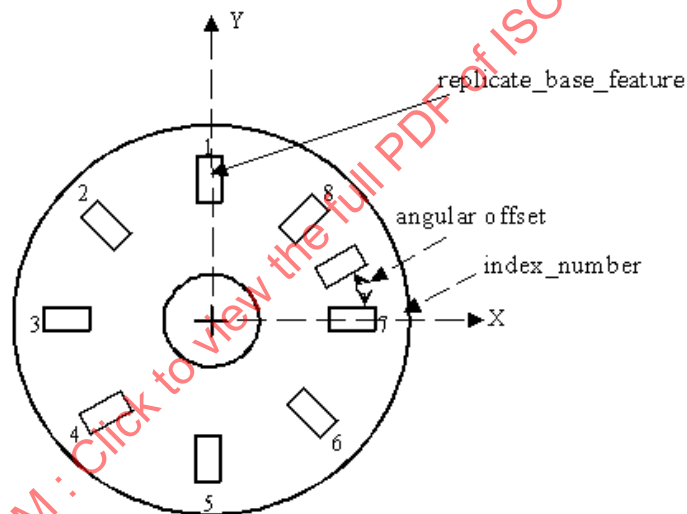


Figure 16 - Circular_offset_pattern

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

- angular_offset;
- index_number.

4.2.23.1 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 4.3.23 for the application assertion.

4.2.23.2 index_number

The `index_number` specifies the value for unique identification of a location of a base feature used in a `Circular_pattern`. See 4.3.23 for the application assertion.

4.2.24 Circular_omit_pattern

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

NOTE - Figure 17 illustrates a `Circular_omit_pattern` with the omission of the rectangular base shape from index number 7.

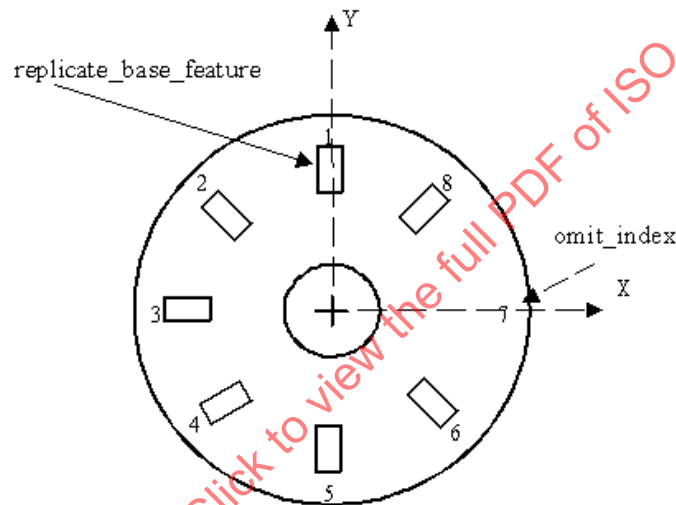


Figure 17 - Circular_omit_pattern

The data associated with a `Circular_omit_pattern` are the following: `omit_index`. The `omit_index` specifies the value for unique identification of the base feature to be omitted in a `Circular_pattern`. See 4.3.24 for the application assertion.

4.2.25 Circular_path

A `Circular_path` is a type of `Path` (see 4.2.127) 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 4.2.30) or a `Partial_circular_path` (see 4.2.125).

The data associated with a `Circular_path` are the following: `radius`. The `radius` specifies the constant distance from an axis for the `Circular_path`. See 4.3.25 for the application assertion.

4.2.26 Circular_pattern

A Circular_pattern is a type of Replicate_feature (see 4.2.157) that is a base feature arranged in a pattern around a circular arc, equally spaced about an axis. Instances of base features need not be rotated. When a base feature rotation is required, the first feature instance is not rotated. The second through the Nth feature instance are rotated by the same angular amount measured from the placement and orientation of the preceding feature instance. (This does not imply the use of chain dimensioning.)

NOTE - Figure 18 illustrates a Circular_pattern with rotation of a base feature, and Figure 19 illustrates a Circular_pattern without rotation.

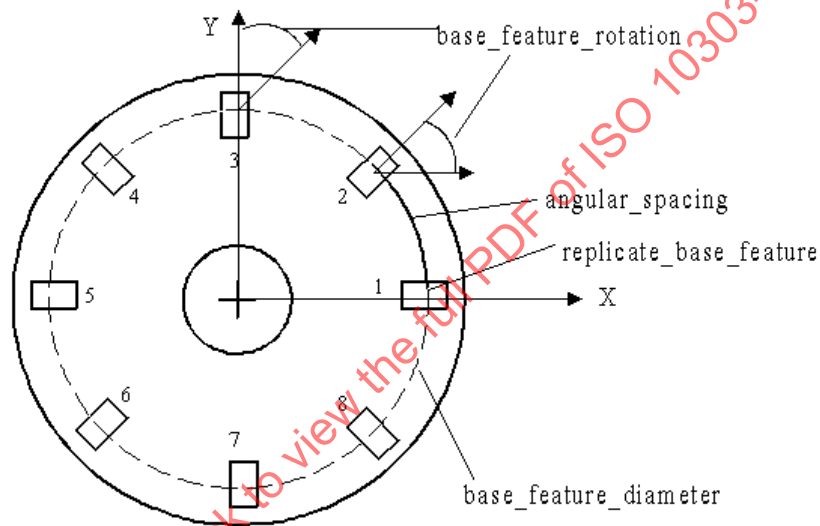


Figure 18 - Circular_pattern with rotation of base feature

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.

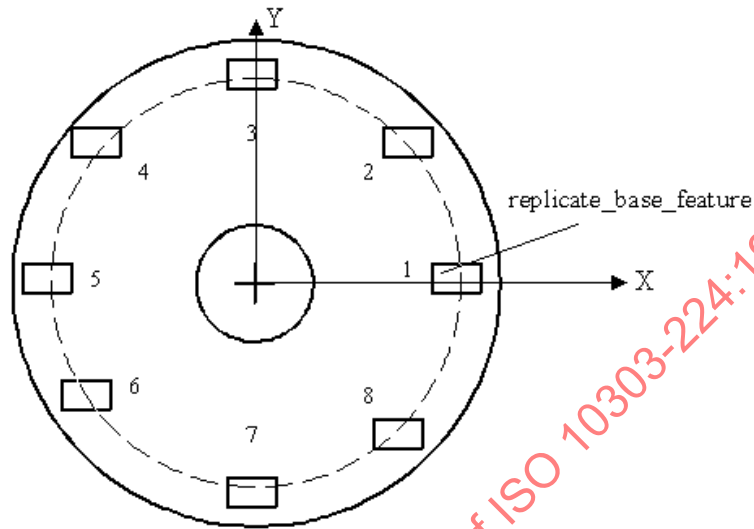


Figure 19 - Circular_pattern without rotation of base feature

4.2.26.1 angular_spacing

The `angular_spacing` specifies the angle amount between features in a `Circular_pattern`. See 4.3.28 for the application assertion.

4.2.26.2 base_feature_diameter

The `base_feature_diameter` specifies the size of the circle for placement of features in a `Circular_pattern`. See 4.3.28 for the application assertion.

4.2.26.3 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 4.3.28 for the application assertion.

4.2.26.4 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 4.3.27 for the application assertion.

4.2.26.5 number_of_features

The number_of_features specifies the quantity of a base feature to be used in a Circular_pattern. See 4.3.28 for the application assertion.

4.2.26.6 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 4.3.26 for the application assertion.

4.2.27 Circular_runout_tolerance

A Circular_runout_tolerance is a type of Geometric_tolerance (see 4.2.86) that is a tolerance to control circular elements of a part to a datum axis the part is rotated about. The tolerance is applied independent of any circular position as the part is rotated 360 degrees. Where applied to surfaces around a datum axis, Circular_runout_tolerance may be used to control the cumulative variations of circularity and coaxiality. Where applied to surfaces at right angles to the datum axis, it controls circular elements of a plane surface.

NOTE - Figure 20 illustrates a Circular_runout_tolerance.

The data associated with a Circular_runout_tolerance are the following:

- geometric_reference;
- runout.

4.2.27.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.29 for the application assertion.

4.2.27.2 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 Circular_runout_tolerance.

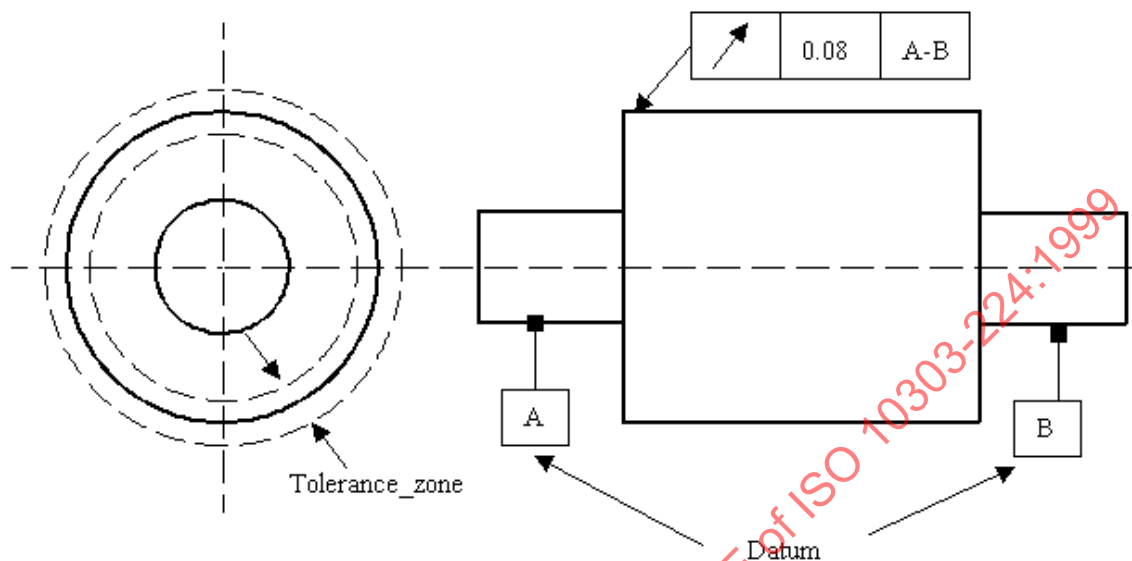


Figure 20 - Circular_runout_tolerance

4.2.28 Circularity_tolerance

A Circularity_tolerance is a type of Geometric_tolerance (see 4.2.86) that describes the allowable deviation of a surface from round. The actual surface shall lie within a tolerance zone defined by two concentric circles.

NOTE 1 - Figure 21 illustrates Circularity_tolerance.

NOTE 2 - The Circularity_tolerance definition is derived from paragraph 14.3 of ISO 1101.

4.2.29 Closed_profile

A Closed_profile is a type of Profile (see 4.2.142) that is an outline or shape that bounds an enclosed area with no opening. Each Closed_profile is either a Circular_closed_profile (see 4.2.22), General_closed_profile (see 4.2.76), Ngon_profile (see 4.2.110), or a Rectangular_closed_profile (see 4.2.151).

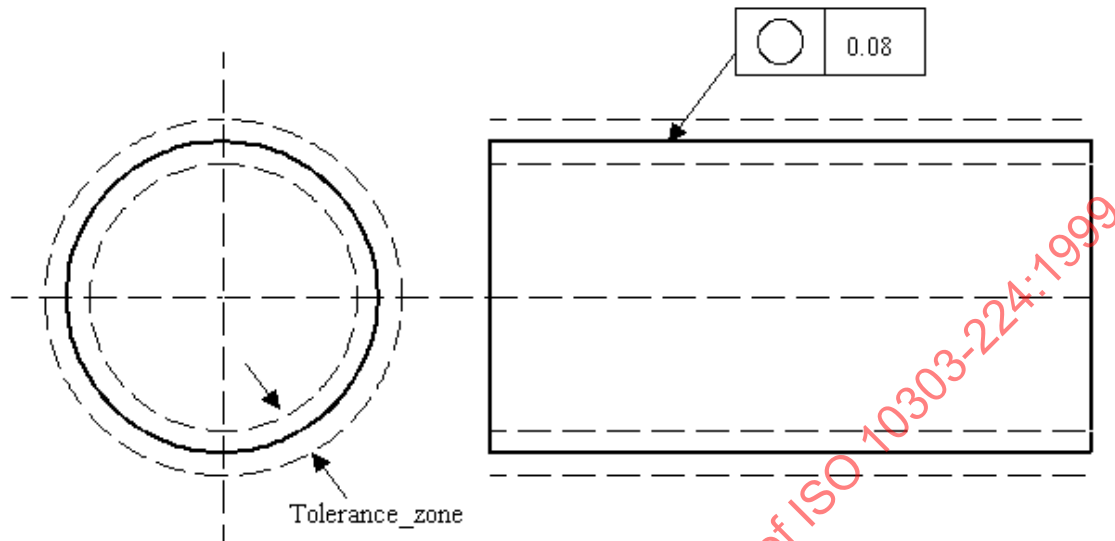


Figure 21 - Circularity tolerance

4.2.30 Complete_circular_path

A Complete_circular_path is a type of Circular_path (see 4.2.25) that is a direction of travel that begins and ends at the same point on the arc.

NOTE - Figure 22 illustrates a Slot feature with a Square_u_profile and a Complete_circular_path.

4.2.31 Compound_datum

A Compound_datum is a type of Datum (see 4.2.46) that is a set of two or more Datum_feature objects which are for establishing a single datum plane or axis. The data associated with a Compound_datum are the following: element. The element specifies the list of Datum_feature objects for defining the Compound_datum. See 4.3.30 for the application assertion.

4.2.32 Compound_feature

A Compound_feature is a type of Machining_feature (see 4.2.101) 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.

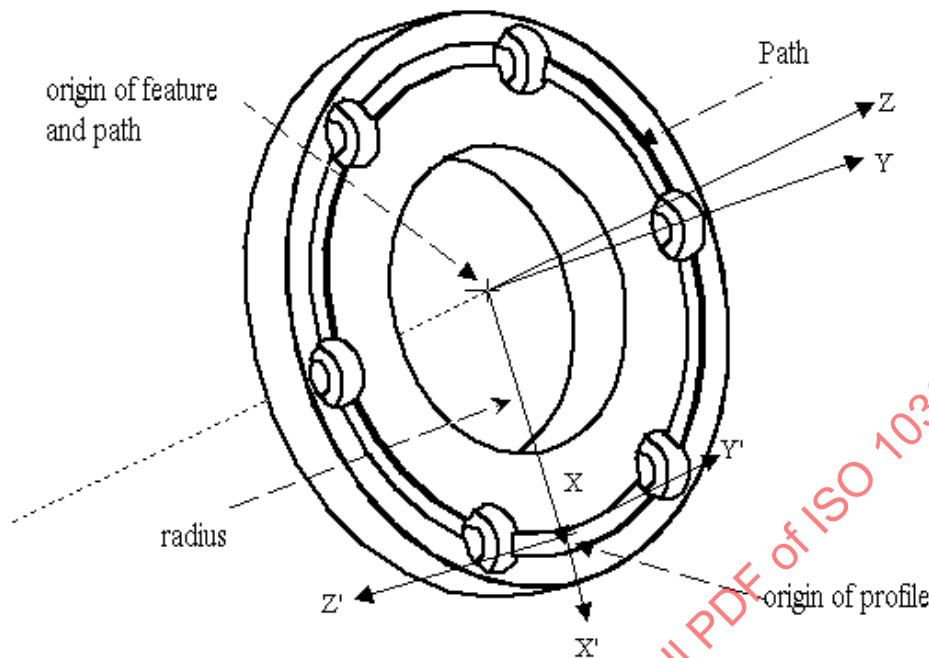


Figure 22 - Complete_circular_path

NOTE - Figure 23 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.

4.2.32.1 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 4.3.31 for the application assertion.

4.2.32.2 feature_description

The feature_description specifies a user defined explanation of the Compound_feature.

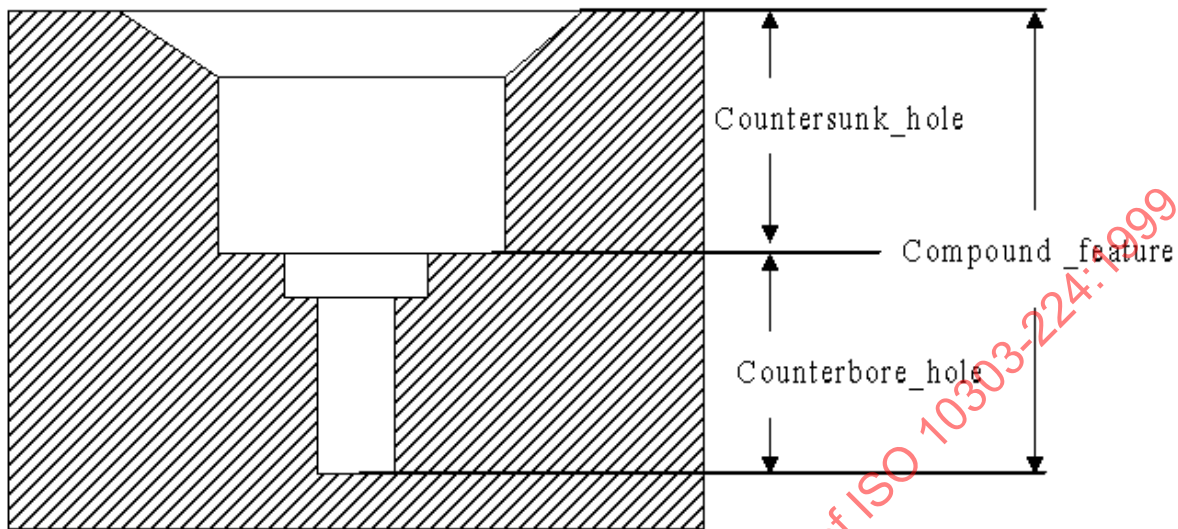


Figure 23 - Compound_feature

4.2.32.3 feature_name

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

4.2.33 Compound_feature_element

A Compound_feature_element specifies the type of Machining_feature (see 4.2.101) or Transition_feature (see 4.2.201) to be used for a particular element of a Compound_feature. A Compound_feature consists of two or more Compound_feature_element objects and need not require Compound_feature_element objects to be ordered. The data associated with a Compound_feature_element are the following: element. The element specifies the base feature to be used as one of the components for the Compound_feature. See 4.3.32 and 4.3.33 for the application assertions.

4.2.34 Compound_feature_relationship

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

The data associated with a Compound_feature_relationship are the following:

- predecessor;
- successor.

4.2.34.1 predecessor

The predecessor specifies the Compound_feature_element with the highest precedence. See 4.3.34 for the application assertion.

4.2.34.2 successor

The successor specifies Compound_feature_element with a lesser precedence. See 4.3.34 for the application assertion.

4.2.35 Concentricity_tolerance

A Concentricity_tolerance is a type of Geometric_tolerance (see 4.2.86) that is a cylindrical or conical feature of a part which shall be fundamentally concentric. When the part is rotated about the datum axis, the axis of the feature shall be within the cylindrical tolerance zone.

NOTE 1 - Figure 24 illustrates the Concentricity_tolerance

NOTE 2 - The Concentricity_tolerance definition is derived from paragraph 14.11 of ISO 1101.

The data associated with a Concentricity_tolerance are the following:

- geometric_reference;
- value_qualifier.

4.2.35.1 geometric_reference

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

4.2.35.2 value_qualifier

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

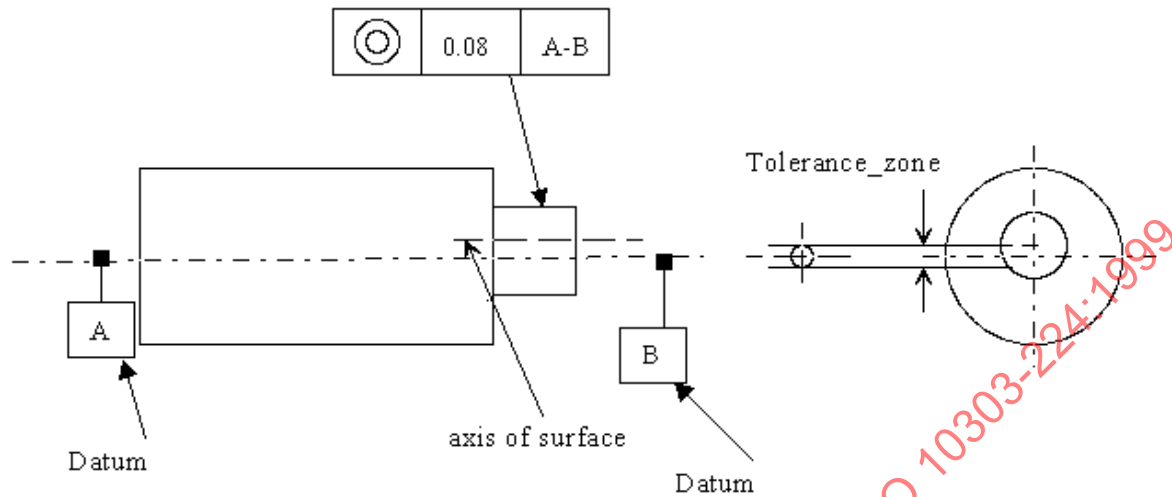


Figure 24 - Concentricity tolerance

4.2.36 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 - Figure 25 illustrates the Conical_hole_bottom for a Round_hole.

The data associated with a Conical_hole_bottom are the following:

— tip_angle;

— tip_radius.

4.2.36.1 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 4.3.36 for the application assertion.

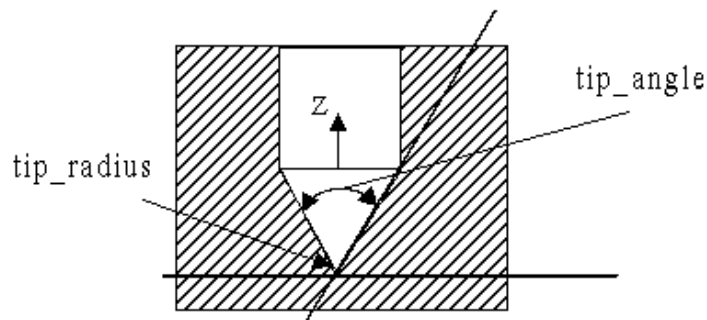


Figure 25 - Conical_hole_bottom

4.2.36.2 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 4.3.36 for the application assertion.

4.2.37 Constant_radius_edge_round

A `Constant_radius_edge_round` is a type of `Edge_round` (see 4.2.63) that is defined with a constant radius value.

NOTE - Figure 26 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`.

4.2.37.1 first_face_offset

The `first_face_offset` specifies the amount of offset from the `first_face_shape` (see 4.2.63.2) to the center of the `edge_round_face` (see 4.2.63.1). See 4.3.37 for the application assertion.

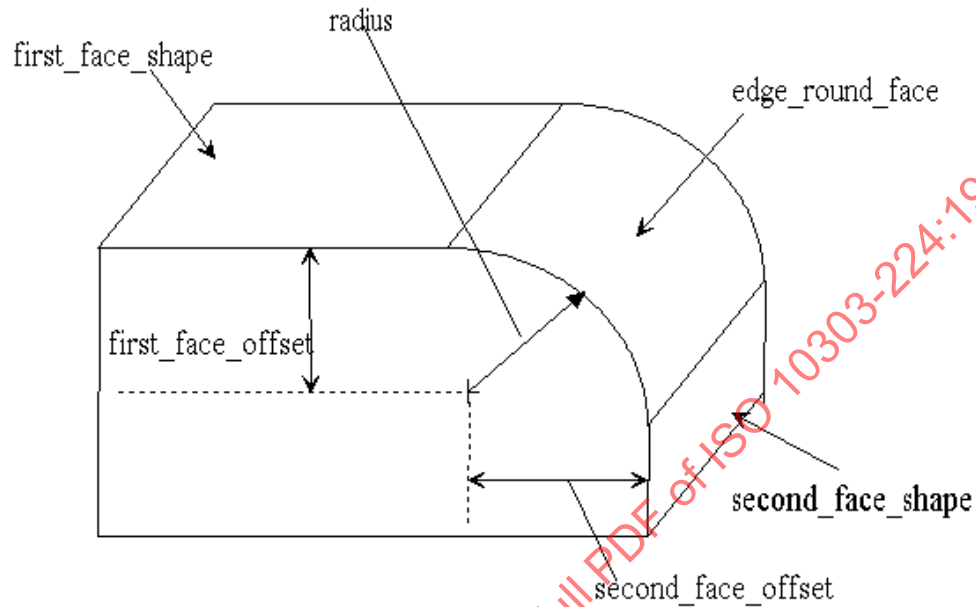


Figure 26 - Constant_radius_edge_round

4.2.37.2 second_face_offset

The second_face_offset specifies the amount of offset from the second_face_shape (see 4.2.63.3) to the center of the edge_round_face (see 4.2.63.1). See 4.3.37 for the application assertion.

4.2.37.3 radius

The radius specifies the amount of curvature for a convex transition between the two faces of a Constant_radius_edge_round. See 4.3.37 for the application assertion.

4.2.38 Constant_radius_fillet

A Constant_radius_fillet is a type of Fillet (see 4.2.69) that is defined with a constant radius value.

NOTE - Figure 27 illustrates the Constant_radius_fillet. The Fillet surface is not tangent to the other surfaces because the offsets.

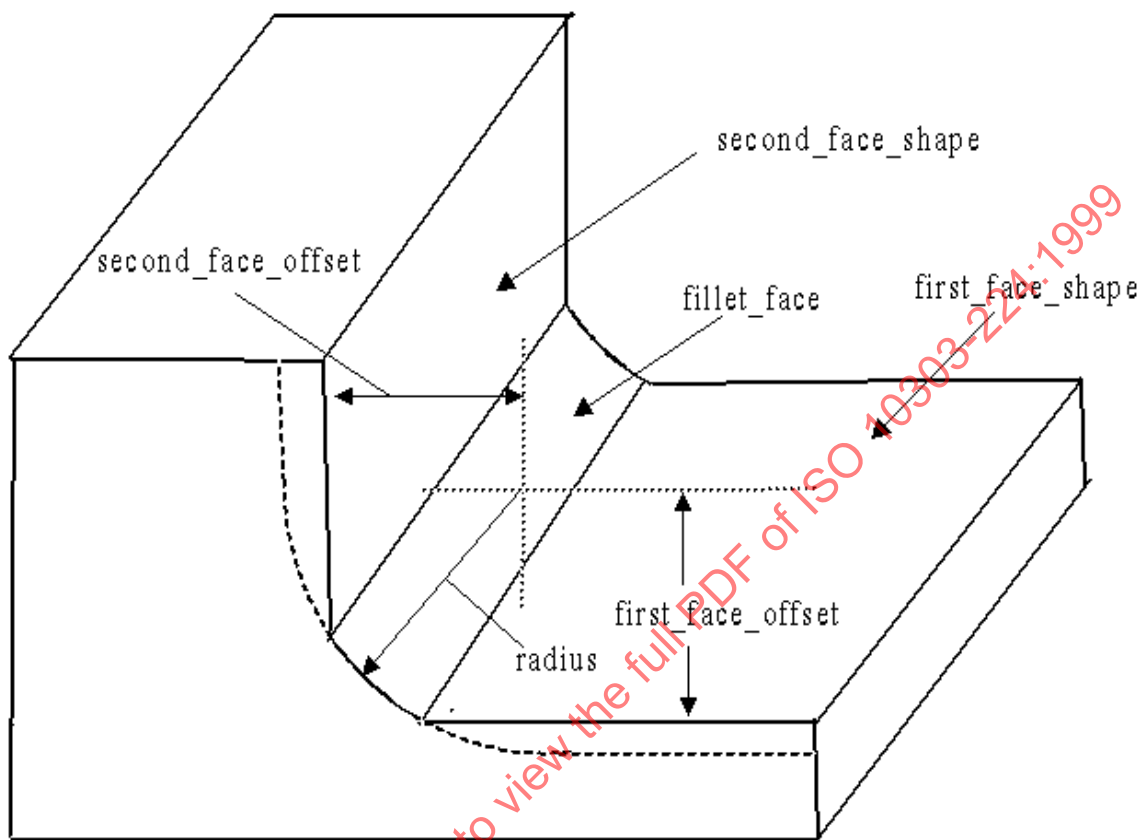


Figure 27 - Constant_radius_fillet

The data associated with a Constant_radius_fillet are the following:

- first_face_offset;
- second_face_offset;
- radius.

4.2.38.1 first_face_offset

The first_face_offset specifies the amount of offset from the first_face_shape (see 4.2.69.2) to the center of the fillet_face (see 4.2.69.1). See 4.3.38 for the application assertion.

4.2.38.2 second_face_offset

The `second_face_offset` specifies the amount of offset from the `second_face_shape` (see 4.2.69.3) to the center of the `fillet_face` (4.2.69.1). See 4.3.38 for the application assertion.

4.2.38.3 radius

The `radius` specifies the amount of curvature for a concave transition between two surfaces. See 4.3.38 for the application assertion.

4.2.39 Counterbore_hole

A `Counterbore_hole` is a type of `Hole` (see 4.2.90) that is a combination of two `Round_holes`. The first `Round_hole` shall have either a `Through_bottom_condition` or `Blind_bottom_condition`, the second shall have a `Blind_bottom_condition`, and a larger diameter than the first `Round_hole`. The top of the first `Round_hole` shall mate with the bottom of the second `Round_hole`. The `Counterbore_hole` orientation shall be the same as the orientation of the first `Round_hole`. Both `Round_holes` shall be co-axial.

NOTE - Figure 28 illustrates the `Counterbore_hole` with a `Blind_bottom_condition` and a `Through_bottom_condition`.

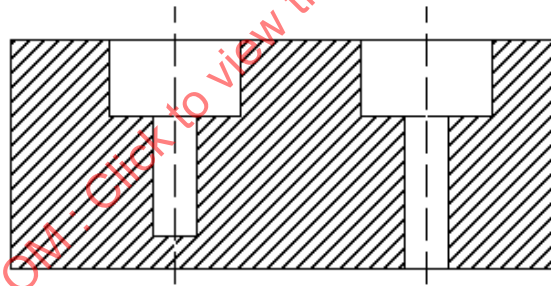


Figure 28 - Counterbore_hole

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

- `larger_hole`;
- `smaller_hole`.

4.2.39.1 larger_hole

The `larger_hole` specifies the `Round_hole` that will be used as the larger hole for the `Counterbore_hole`. See 4.3.39 for the application assertion.

4.2.39.2 smaller_hole

The smaller_hole specifies the Round_hole that will be used as the smaller hole for the Counterbore_hole. See 4.3.39 for the application assertion.

4.2.40 Countersunk_hole

A Countersunk_hole is a type of Hole (see 4.2.90) 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 coaxial.

NOTE - Figure 29 illustrates the Countersunk_hole with a Blind_bottom_condition and a Through-bottom_condition.

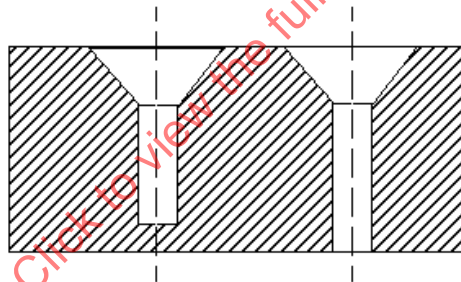


Figure 29 - Countersunk_hole

The data associated with a Countersunk_hole are the following:

- constant_diameter_hole;
- tapered_hole.

4.2.40.1 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 4.3.40 for the application assertion.

4.2.40.2 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 4.3.40 for the application assertion.

4.2.41 Curved_dimension_tolerance

A Curved_dimension_tolerance is a type of Size_tolerance (see 4.2.172) that is the tolerance on a dimension for a curve measured along the entire path of the curve.

NOTE - Figure 30 illustrates the Curved_dimension_tolerance.

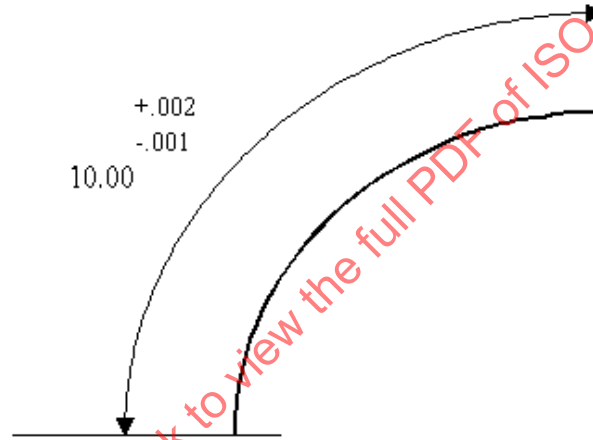


Figure 30 - Curved_dimension_tolerance

4.2.42 Customer_order

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

- customer;
- delivery_date;
- initiated_order;
- material_disposition;
- order_number;

- order_status;
- quantity_ordered;
- special_instructions.

4.2.42.1 customer

The customer specifies the person and company requesting the manufacture of the part. See 4.3.42 for the application assertion.

4.2.42.2 delivery_date

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

4.2.42.3 initiated_order

The initiated_order specifies Project_order that is initiated by the Customer_order. There may be more than one initiated_order for a Customer_order. See 4.3.43 for the application assertion.

4.2.42.4 material_disposition

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

4.2.42.5 order_number

The order_number specifies a unique identification of the Customer_order.

4.2.42.6 order_status

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

4.2.42.7 quantity_ordered

The quantity_ordered specifies number of parts to be manufactured. There may be more than one quantity_ordered for a Customer_order. See 4.3.41 for the application assertion.

4.2.42.8 special_instructions

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

4.2.43 Cutting_tool_requisition

A Cutting_tool_requisition is a type of Requisition (see 4.2.158) that is an organization's requirements for the purchase of tools required to support part manufacturing.

4.2.44 Cylindrical_base_shape

A Cylindrical_base_shape is a type of Implicit_base_shape_representation (see 4.2.91) that is the initial shape of the material which is cylindrical.

NOTE - Figure 31 illustrates a Cylindrical_base_shape.

The data associated with a Cylindrical_base_shape are the following: diameter. The diameter specifies the distance across a Cylindrical_base_shape. See 4.3.44 for the application assertion.

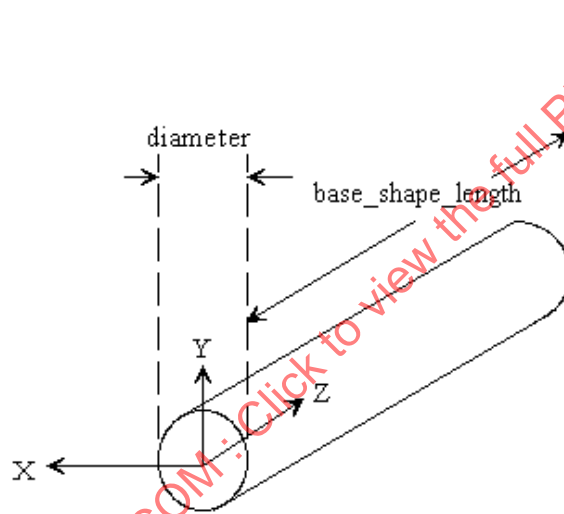


Figure 31 - Cylindrical_base_shape

4.2.45 Cylindricity_tolerance

A Cylindricity_tolerance is a type of Geometric_tolerance (see 4.2.86) that describes the amount of deviation a feature may have from being truly cylindrical. The feature shall be contained between two coaxial cylinders. The distance between the two coaxial cylinders defines the allowable tolerance deviation.

NOTE 1 - Figure 32 illustrates Cylindricity_tolerance.

NOTE 2 - The Cylindricity_tolerance definition is derived from paragraph 14.4 of ISO 1101.

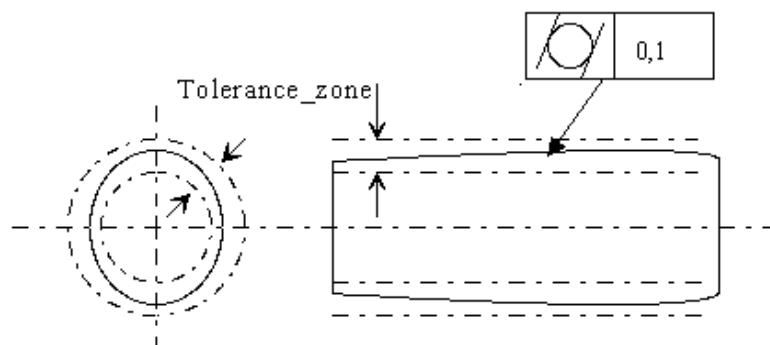


Figure 32 - Cylindricity_tolerance

4.2.46 Datum

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

- name;
- precedence.

4.2.46.1 name

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

4.2.46.2 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.

4.2.47 Datum_feature

A Datum_feature is a type of Datum (see 4.2.46) that is a feature on a part used to establish a Datum. The data associated with a Datum_feature are the following:

- datum_representation;
- modifier.

4.2.47.1 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 4.3.46 and 4.3.47 for the application assertions.

4.2.47.2 modifier

The modifier specifies the tolerance value applied to the Datum_feature. The modifier need not be specified for a particular Datum_feature. See 4.3.45 for the application assertion.

4.2.48 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 4.2.133) or a Target_area (see 4.2.186). The data associated with a Datum_target are the following: identifier. The identifier specifies a unique identification for the Datum_target.

4.2.49 Datum_target_set

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

- rule_description;
- target_shape.

4.2.49.1 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.

4.2.49.2 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 4.3.48 for the application assertion.

4.2.50 Dedicated_fixture_requisition

A Dedicated_fixture_requisition is a type of Requisition (see 4.2.158) for dedicated fixtures required to support part manufacturing.

4.2.51 Defined_marking

A Defined_marking is a type of Marking (see 4.2.103) that is specified explicitly, all attributes for a Marking are declared and defined.

NOTE - Figure 33 illustrates the Defined_marking on a rectangular block shape and with text of 'TEXT FOR A MARKING'.

The data associated with a Defined_marking are the following:

- character_height;
- character_spacing;
- font_name;
- special_instructions.

4.2.51.1 character_height

The character_height specifies the size of the text used for a Defined_marking feature. See 4.3.49 for the application assertion.

4.2.51.2 character_spacing

The character_spacing specifies the amount of space between text letters used for a Defined_marking feature. See 4.3.50 for the application assertion.

4.2.51.3 font_name

The font_name specifies the appearance of the characters. A font consists of typeface, treatment, and size. See 4.3.49 for the application assertion.

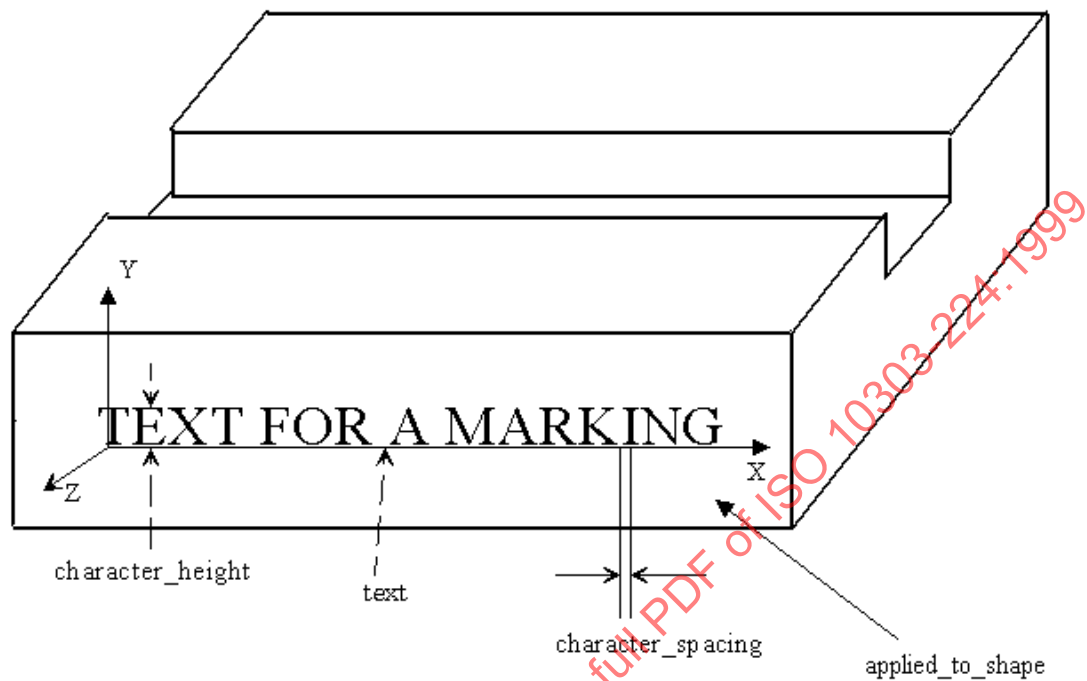


Figure 33 - Defined_marking

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.

4.2.51.4 special_instructions

The special_instructions specifies a description on how to apply the text given by the Defined_marking entity. See 4.3.49 for the application assertion.

4.2.52 Defined_thread

A Defined_thread is a type of Thread (see 4.2.192) that is specified explicitly.

NOTE - Figure 34 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.

The data associated with a Defined_thread are the following:

- minor_diameter;
- pitch_diameter.

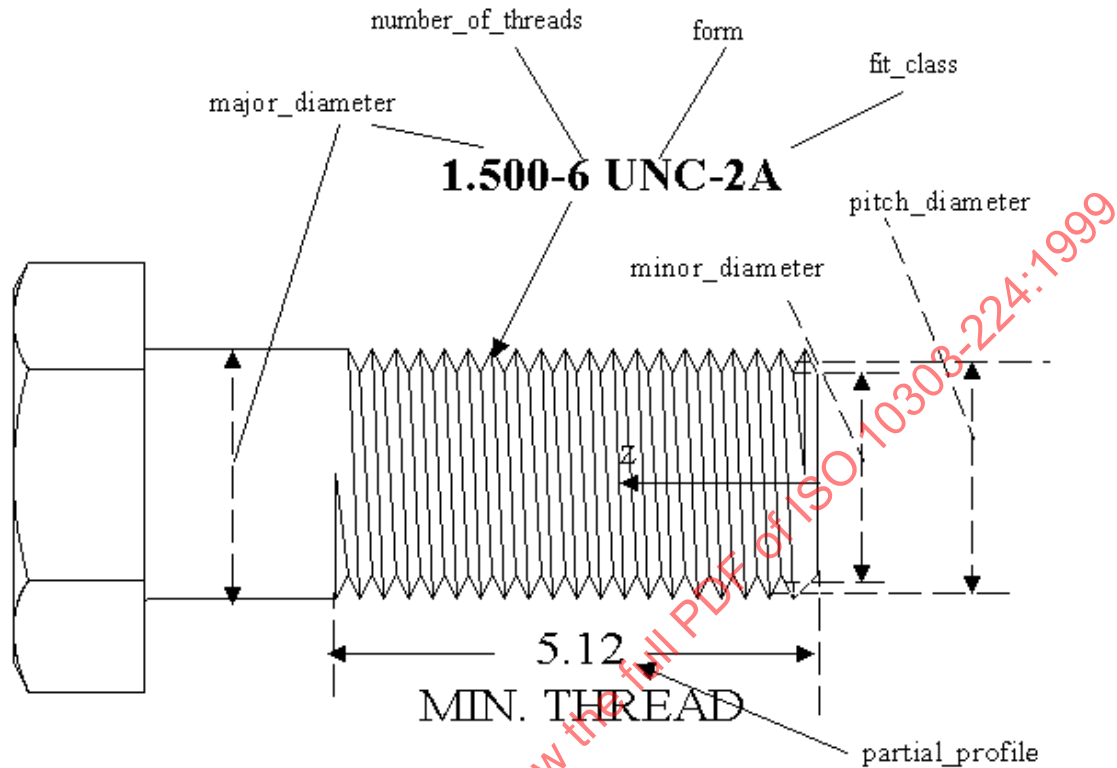


Figure 34 - Thread and Defined_thread attributes

4.2.52.1 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 4.3.51 for the application assertion.

4.2.52.2 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. The `pitch_diameter` need not be specified for a particular `Defined_thread`. See 4.3.51 for the application assertion.

4.2.53 Descriptive_parameter

A `Descriptive_parameter` is a type of `Property_parameter` (see 4.2.146) that is an explanation of the property being defined by a specification.

The data associated with a Descriptive_parameter are the following: descriptive_string. The descriptive_string specifies a word or group of words by which a Descriptive_parameter is explained.

4.2.54 Design_exception_notice

A Design_exception_notice is a notification of a design discrepancy discovered during the creation of the process plan for a given part. Process planning cannot continue until a technical recommendation is made to correct the problem. The data associated with a Design_exception_notice are the following:

- discrepant_part;
- issuing_date;
- notice_description;
- notice_number;
- result;
- technical_recommendation.

4.2.54.1 discrepant_part

The discrepant_part specifies the set of parts that has a design discrepancy. There may be more than one discrepant_part for a Design_exception_notice. See 4.3.53 for the application assertion.

4.2.54.2 issuing_date

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

4.2.54.3 notice_description

The notice_description specifies the kind of problem or non-conformance machining condition causing a rejection of a part.

EXAMPLE - If two holes were drilled simultaneously and the drill bits would run together, a recommendation would be needed to either change the depth of the hole or change the machining process.

4.2.54.4 notice_number

The notice_number specifies a unique identification for each Design_exception_notice.

4.2.54.5 result

The result specifies the set of change proposals to modify a Part. The result need not be specified for a particular Design_exception_notice. There may be more than one result for a Design_exception_notice. See 4.3.52 for the application assertion.

4.2.54.6 technical_recommendation

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

4.2.55 Diagonal_knurl

A Diagonal_knurl is a type of Turned_knurl (see 4.2.202) with helical cuts at an angle about the axis of a surface.

NOTE - Figure 35 illustrates a Diagonal_knurl with a right hand helix.

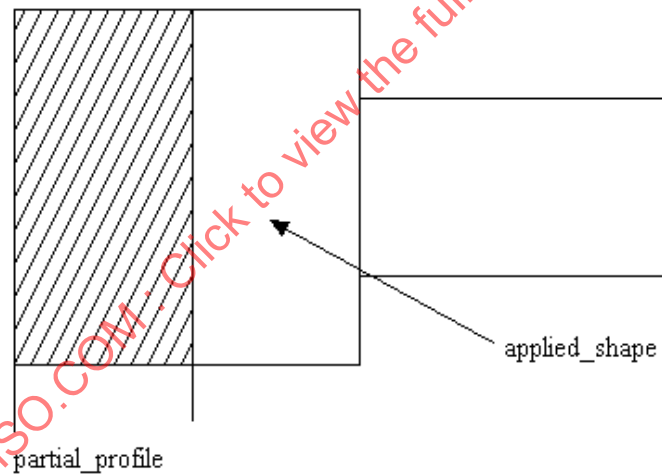


Figure 35 - Diagonal_knurl

The data associated with a Diagonal_knurl are the following:

- helix_angle;
- helix_hand.

4.2.55.1 helix_angle

The `helix_angle` specifies the angle the knurl pattern makes with the orientation axis of an applied to surface. See 4.3.55 for the application assertion.

4.2.55.2 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 4.3.54 for the application assertion.

4.2.56 Diameter_dimension_tolerance

A `Diameter_dimension_tolerance` is a type of `Size_tolerance` (see 4.2.172) that is the allowable variation of the size of a hole in a surface.

NOTE - Figure 36 illustrates the `Diameter_dimension_tolerance`.

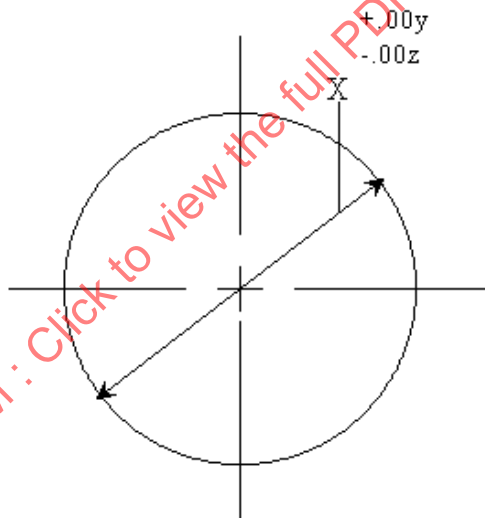


Figure 36 - Diameter_dimension_tolerance

4.2.57 Diameter_taper

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

NOTE - Figure 37 illustrates the `Diameter_taper`.

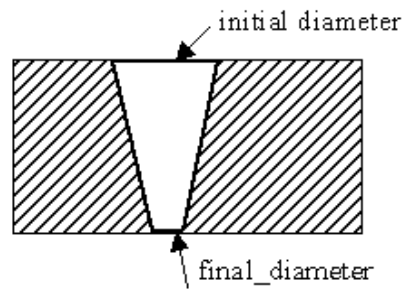


Figure 37 - Diameter_taper

The data associated with a Diameter_taper are the following: 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 4.3.56 for the application assertion.

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

4.2.58 Diamond_knurl

A Diamond_knurl is a type of Turned_knurl (see 4.2.202) that is a knurl ridge that is doubly helical, a left hand and a right hand helix, about the axis of a surface, with equal spacing of the two.

NOTE - Figure 38 illustrates a Diamond_knurl.

The data associated with a Diamond_knurl are the following: helix_angle. The helix_angle specifies the angle the knurl pattern makes with the orientation axis of an applied to surface. See 4.3.57 for the application assertion.

4.2.59 Digital_technical_data_package_work_order

A Digital_technical_data_package_work_order is a document that defines the need to create and track the digital product data definition for a part. The data associated with a Digital_technical_data_package_work_order are the following: order_id. The order_id specifies a unique identifier for the Digital_technical_data_package_work_order.

4.2.60 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 4.2.99) or a Size_tolerance (see 4.2.172).

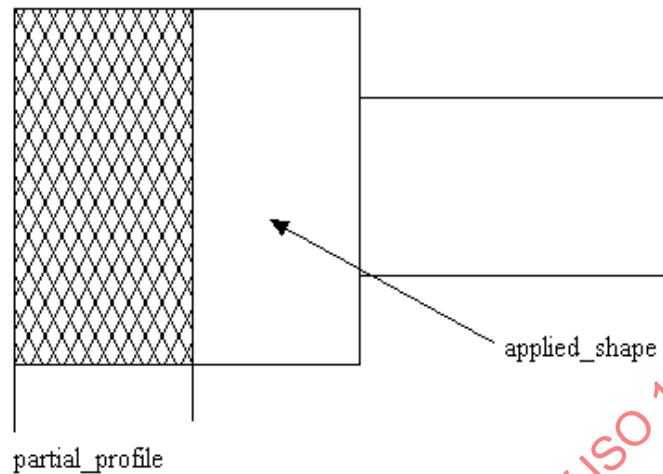


Figure 38 - Diamond knurl

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_value;
- limit;
- significant_digits;
- unit_of_measure.

4.2.60.1 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.

4.2.60.2 dimension_value

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

4.2.60.3 limit

The limit specifies the tolerance value applied to the Dimension_tolerance. The limit need not be specified for a particular Dimensional_tolerance. See 4.3.58 for the application assertion.

4.2.60.4 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.

4.2.60.5 unit_of_measure

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

4.2.61 Direction_element

A Direction_element is a type of Shape_element (see 4.2.170) that is a Shape_aspect definition for a direction.

4.2.62 Distance_along_curve_tolerance

A Distance_along_curve_tolerance is a type of Location_tolerance (see 4.2.99) that is the distance calculated between two elements along a path defined by a third element of geometry.

NOTE - Figure 39 illustrates the Distance_along_curve_tolerance.

The data associated with a Distance_along_curve_tolerance are the following:

- path;
- with_curve_direction.

4.2.62.1 path

The path specifies the shape that the tolerance applies to. See 4.3.59 for the application assertion.

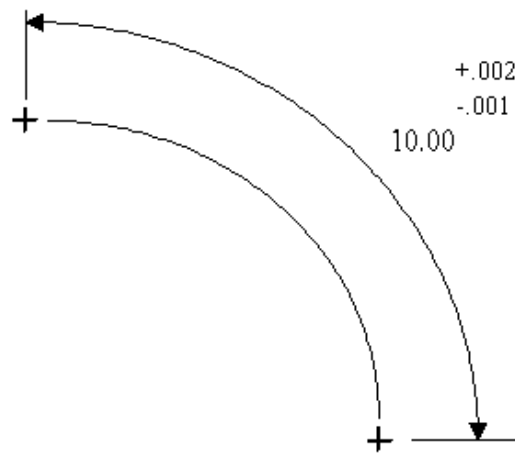


Figure 39 - Distance_along_curve_tolerance

4.2.62.2 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.

4.2.63 Edge_round

An `Edge_round` is a type of `Transition_feature` (see 4.2.201) 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 4.2.37).

NOTE - Figure 40 illustrates the `Edge_round` with an `edge_round_face` that is not constant radius.

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

- `edge_round_face`;
- `first_face_shape`;
- `second_face_shape`.

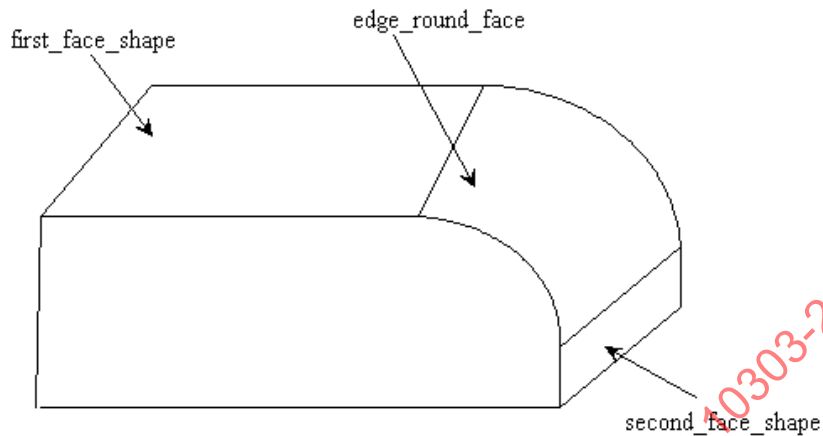


Figure 40 - Edge_round

4.2.63.1 edge_round_face

The `edge_round_face` specifies the circular transition surface between the two edges of two surfaces. See 4.3.60 for the application assertion.

4.2.63.2 first_face_shape

The `first_face_shape` specifies one of two surfaces the `Edge_round` feature will transition between. See 4.3.60 for the application assertion.

4.2.63.3 second_face_shape

The `second_face_shape` specifies the other surfaces the `Edge_round` feature will transition between. See 4.3.60 for the application assertion.

4.2.64 Engineering_change_order

An `Engineering_change_order` is an authorization for modification of the product data that will result in a new process plan for a part. The data associated with an `Engineering_change_order` are the following:

- `change_order_number`;
- `incorporated_proposal`;
- `new_version`.

4.2.64.1 change_order_number

The change_order_number specifies a unique identification of the Engineering_change_order.

4.2.64.2 incorporated_proposal

The incorporated_proposal specifies the change proposals that describe the modifications to the Part. There may be more than one incorporated_proposal for an Engineering_change_order. See 4.3.61 for the application assertion.

4.2.64.3 new_version

The new_version specifies the version of a Part which has an effected change made by the Engineering_change_order. There may be more than one new_version of an Engineering_change_order. See 4.3.62 for the application assertion.

4.2.65 Engineering_change_proposal

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

4.2.66 Explicit_base_shape_representation

An Explicit_base_shape_representation is a type of Base_shape (see 4.2.7) that is the geometric representation needed to define the shape of the initial material.

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

The data associated with an Explicit_base_shape_representation are the following: B-rep_form. The B-rep_form specifies the shape that is the representation of the initial material. See 4.3.63 for the application assertion.

4.2.67 Externally_defined_tolerance_value

An Externally_defined_tolerance_value is a reference to a document containing the information necessary to create a tolerance expression.

EXAMPLE - sample values for an Externally_defined_tolerance_value may be 30h6.

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

- `documentation`;
- `tolerance_value`.

4.2.67.1 documentation

The `documentation` specifies the document reference that contains the tolerance information. See 4.3.64 for the application assertion.

4.2.67.2 tolerance_value

The `tolerance_value` specifies the tolerance expression used for the required accuracy and for proper fit.

4.2.68 Face_shape_element

A `Face_shape_element` is a type of `Shape_element` (see 4.2.170) that is a `Shape_aspect` definition for a `Face`.

4.2.69 Fillet

A `Fillet` is a type of `Transition_feature` (see 4.2.201) 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 4.2.38).

NOTE - Figure 41 illustrates a `Fillet` with a `fillet_face` that is not constant radius.

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

- `fillet_face`;
- `first_face_shape`;
- `second_face_shape`.

4.2.69.1 fillet_face

The `fillet_face` specifies the circular transition surface between the two edges of two surfaces. See 4.3.65 for the application assertion.

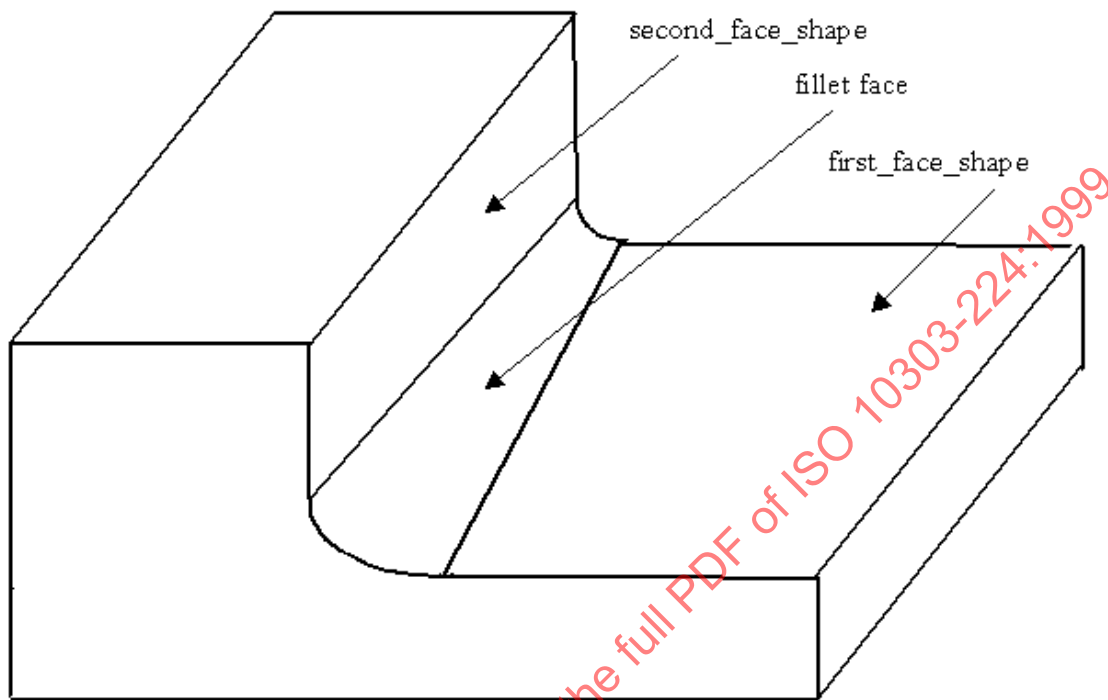


Figure 41 - Fillet

4.2.69.2 first_face_shape

The first_face_shape specifies one of two surfaces the Fillet feature will transition between. See 4.3.65 for the application assertion.

4.2.69.3 second_face_shape

The second_face_shape specifies the second of two surfaces the Fillet feature will transition between. See 4.3.65 for the application assertion.

4.2.70 First_offset

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

NOTE - Figure 12 illustrates a First_offset for a Chamfer feature.

The data associated with a First_offset are the following:

- face_shape;
- offset_amount.

4.2.70.1 face_shape

The first_face specifies a shape for one of two faces the Chamfer feature will transition between. See 4.3.66 for the application assertion.

4.2.70.2 offset_amount

The offset_amount specifies a distance from the edge of a face to the start of the Chamfer. See 4.3.67 for the application assertion.

4.2.71 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 - Figure 42 illustrates a Flat_hole_bottom for a Round_hole.

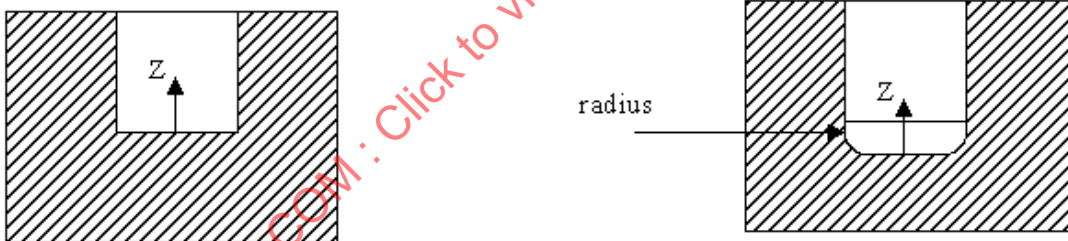


Figure 42 - Flat_hole_bottom and Flat_with_radius_hole_bottom

4.2.72 Flat_slot_end_type

A Flat_slot_end_type is a type of Slot_end_type (see 4.2.174) that is an end condition of a slot that shall be a planar shape perpendicular to both of the adjacent Slot wall surfaces. The intersection of the Slot wall surfaces and the end planar shape need not be blended by a radius.

NOTE - Figure 43 illustrates a Flat_slot_end_type for a Slot.

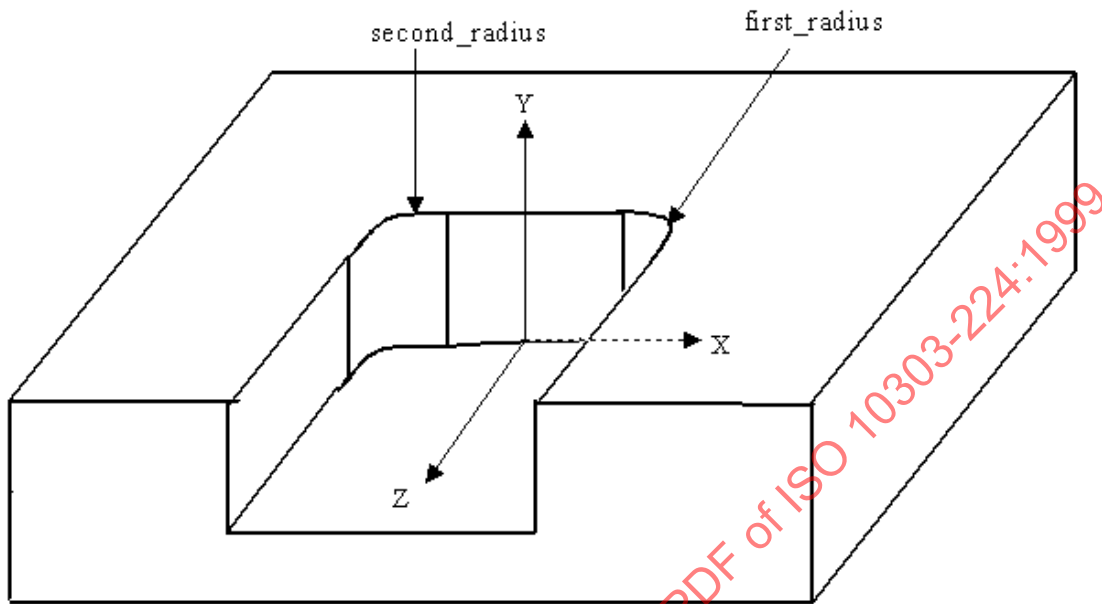


Figure 43 - Flat_slot_end_type

The data associated with a Flat_slot_end_type are the following:

- first_radius;
- second_radius.

4.2.72.1 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 4.3.68 for the application assertion.

4.2.72.2 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 4.3.68 for the application assertion.

4.2.73 Flat_with_radius_hole_bottom

A Flat_with_radius_hole_bottom is a type of Blind_bottom_condition (see 4.2.8) that the bottom of a Round_hole is flat and has corner radius that are smaller than the diameter of the Round_hole.

NOTE - Figure 42 illustrates a Flat_with_radius_hole_bottom for a Round_hole.

The data associated with a Flat_with_radius_hole_bottom are the following: corner_radius. The corner_radius specifies the radius between the side and the floor of a Round_hole. See 4.3.69 for the application assertion.

4.2.74 Flatness_tolerance

A Flatness_tolerance is a type of Geometric_tolerance (see 4.2.86) that is a tolerance for how much a surface is allowed to deviate from being flat. All points of the actual surface shall lie between two parallel planes that are a distance apart equal to the specified tolerance.

NOTE 1 - Figure 44 illustrates a Flatness_tolerance.

NOTE 2 - The Flatness_tolerance definition is derived from paragraph 14.2 of ISO 1101.

The data associated with a Flatness_tolerance are the following: 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.

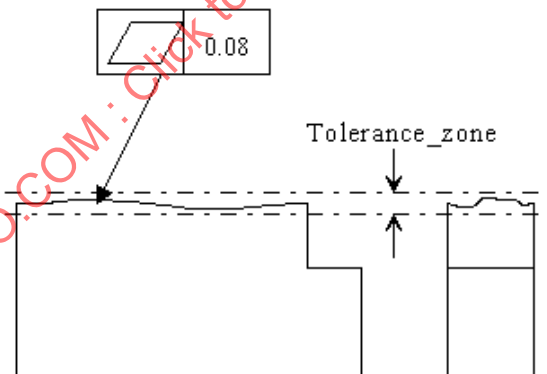


Figure 44 - Flatness_tolerance

4.2.75 GeneralBoss

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

NOTE - Figure 45 illustrates the GeneralBoss.

The data associated with a General_boss are the following:

- change_in_boundary;
- enclosed_boundary.

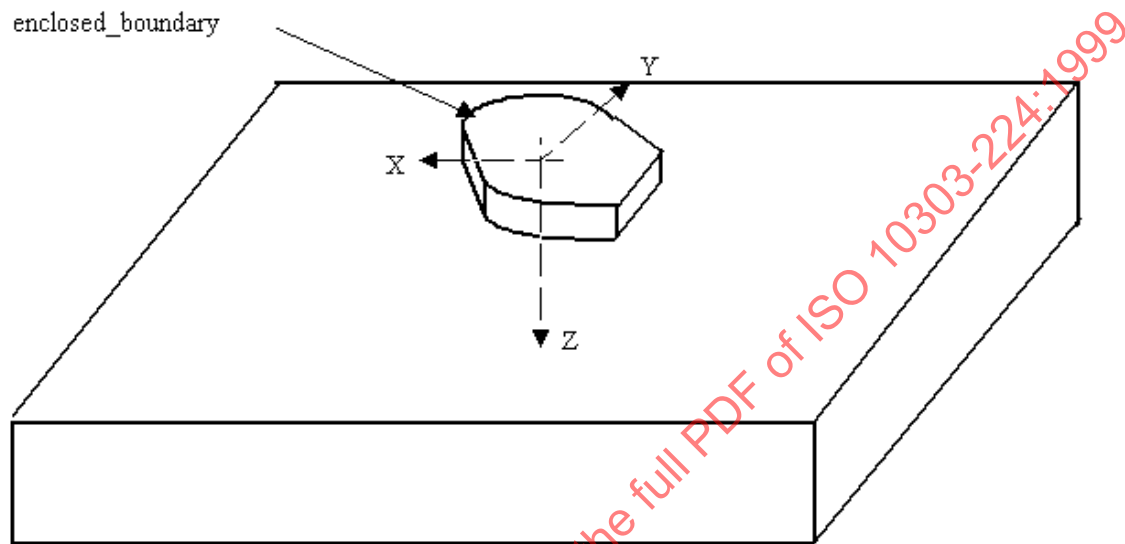


Figure 45. General_boss

4.2.75.1 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 4.3.71 for the application assertion.

4.2.75.2 enclosed_boundary

The enclosed_boundary specifies an outline or shape that bounds an enclosed area with no opening for a General_boss. The General_profile specifies the area required by a General_boss. This placement of the profile shall be with the origin of the Closed_profile at the origin of the General_boss. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the General_boss. See 4.3.70 for the application assertion.

4.2.76 General_closed_profile

A General_closed_profile is a type of Closed_profile (see 4.2.29) that is an enclosed area bounded by a arbitrary shape. The orientation is defined by the explicit geometry of the shape.

NOTE - Figure 46 illustrates a General_closed_profile.

The data associated with a General_closed_profile are the following: closed_profile_shape. The closed_profile_shape specifies a closed curve that defines the arbitrary shape of the profile. See 4.3.72 for the application assertion.

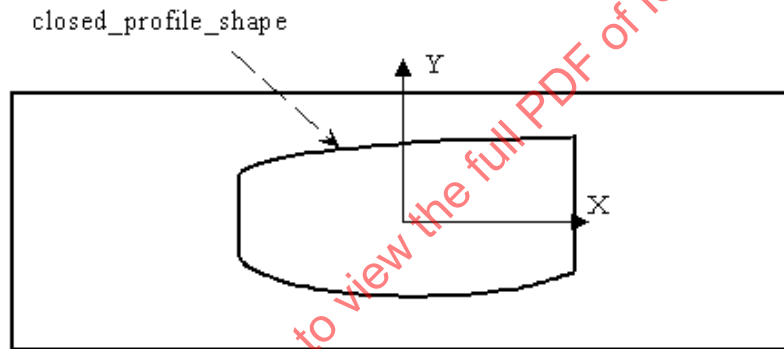


Figure 46 - General_closed_profile

4.2.77 General_open_profile

A General_open_profile is a type of Open_profile (see 4.2.113) that is specified by a shape bounded by an arbitrary planar shape. The orientation is defined by the explicit geometry of the shape.

NOTE - Figure 47 illustrates a General_open_profile.

The data associated with a General_open_profile are the following: enclosed_boundary. The enclosed_boundary specifies a curve with no enclosing bounds that defines the arbitrary planar shape of the profile. See 4.3.73 for the application assertion.

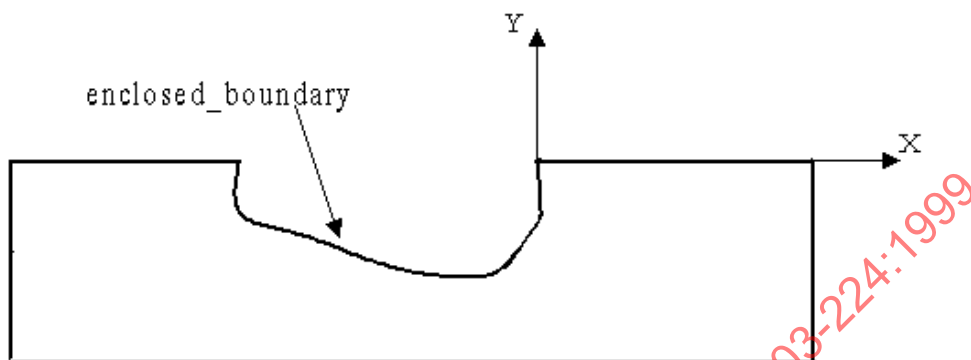


Figure 47 - General_open_profile

4.2.78 General_outside_profile

A General_outside_profile is a type of Machining_feature (see 4.2.101) that is specified by a volume removal of raw stock 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 native coordinate systems of the General_outside_profile objects pointing in any direction.

NOTE - Figure 48 illustrates a General_outside_profile.

The data associated with a General_outside_profile are the following: boundary. The boundary specifies a profile that identifies the outside shape of the part. This placement of the profile 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 4.3.74 for the application assertion.

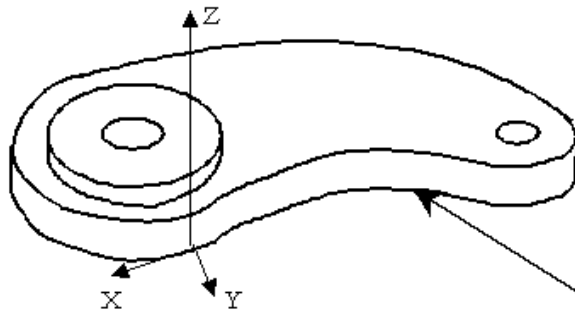
4.2.79 General_path

A General_path is a type of Path (see 4.2.127) that is a direction of travel along an arbitrary curve.

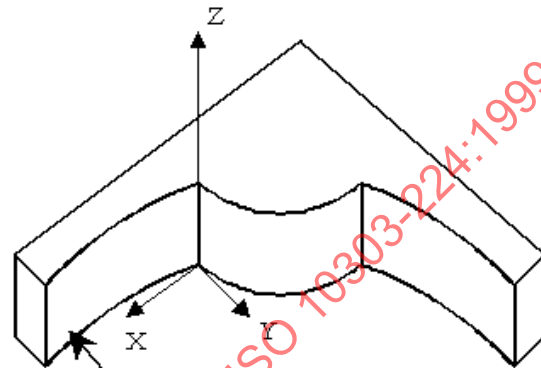
NOTE - Figure 49 illustrates a Slot feature with a Square_u_profile and a General_path. The path has the same orientation as the Slot feature. The orientation of the profile positions it to the orientation of the Slot feature. The Square_U_profile has orientation to position it at the same position as the Slot feature.

The data associated with a General_path are the following: sweep_path. The sweep_path specifies a continuous set of curves that define an arbitrary direction of travel. See 4.3.75 for the application assertion.

General_outside_profile
using a Closed_profile



General_outside_profile
using an Open_profile



boundary

Figure 48 - General_outside_profile

sweep_path

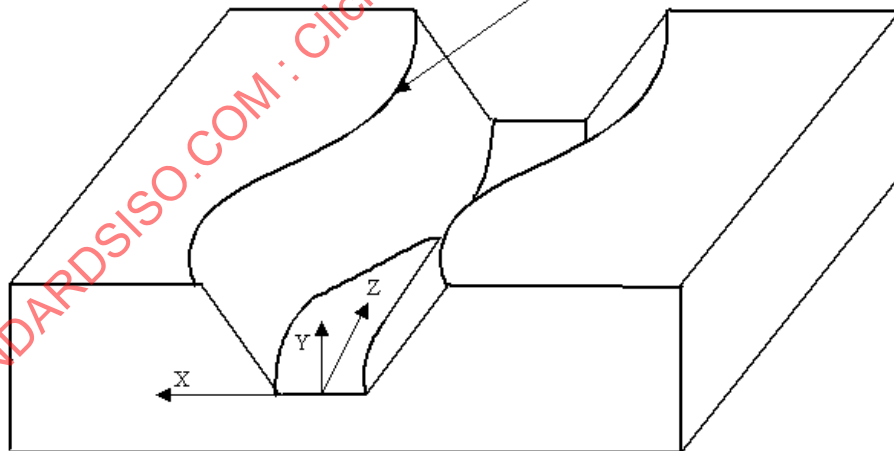


Figure 49 - General_path

4.2.80 General_pattern

A General_pattern is a type of Replicate_feature (see 4.2.157) that is a base shape component and a list of arbitrary placements to arrange identical copies of the base feature. The placement of all instances of the base feature are relative to the Replicate_feature coordinate system.

NOTE -Figure 50 illustrates a General_pattern of Round_hole features.

The data associated with a General_pattern are the following: feature_placement. The feature_placement specifies a set of axis and positions to place a base feature in the General_pattern. A General_pattern is defined as one base feature and many placements allowing the base feature to be placed several times in a random pattern. There may be more than one feature_placement for a General_pattern. See 4.3.76 for the application assertion.

4.2.81 General_pocket

A General_pocket is a type of Pocket (see 4.2.138) that is a volume of arbitrary shape removed from the part.

NOTE -Figure 51 illustrates a General_pocket with a Through_pocket_bottom_condition and a pocket depth that defines the highest point of the pocket.

The data associated with a General_pocket are the following: boundary. The boundary specifies an outline or shape that is an enclosed area that may be partially open. The profile specifies the area required by a General_pocket. This placement of the profile shall be with the origin of the Profile at the origin of the General_pocket. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the General_pocket. See 4.3.77 for the application assertion.

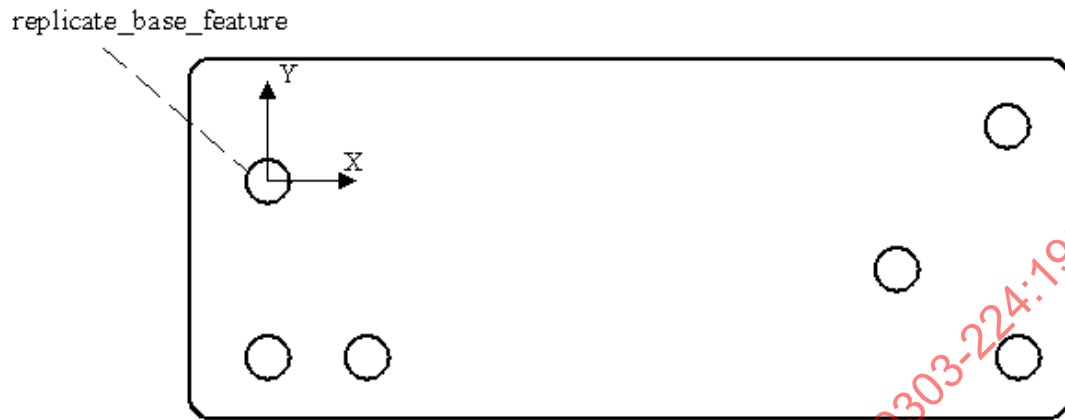
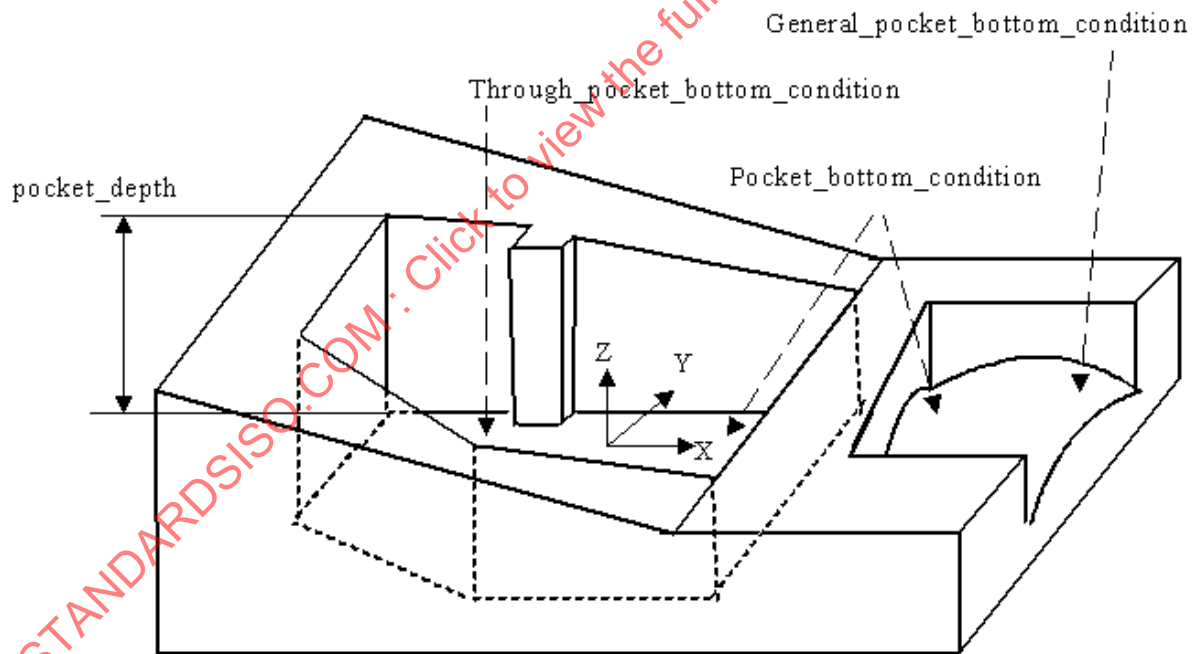
4.2.82 General_pocket_bottom_condition

A General_pocket_bottom_condition is a type of Pocket_bottom_condition (see 4.2.139) that specifies an enclosed area bounded by an arbitrary shape.

NOTE -Figure 51 illustrates the General_pocket_bottom_condition applied to a General_pocket with an Open_profile.

The data associated with a General_pocket_bottom_condition are the following:

- floor;
- floor_radius.

**Figure 50 - General_pattern****Figure 51 - General_pocket**

4.2.82.1 floor

The floor specifies the face at the bottom of a Pocket feature, adjacent to all the pocket walls. See 4.3.78 for the application assertion.

4.2.82.2 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 4.3.79 for the application assertion.

4.2.83 General_removal_volume

A General_removal_volume is a type of Machining_feature (see 4.2.101) that is an enclosed volume of arbitrary shape that shall be removed from the part. The position and orientation shall be determined from the shape defining geometry.

NOTE -Figure 52 illustrates a General_removal_volume.

The data associated with a General_removal_volume are the following: removal_volume. The removal_volume specifies the arbitrary shape to be removed. See 4.3.80 for the application assertion.

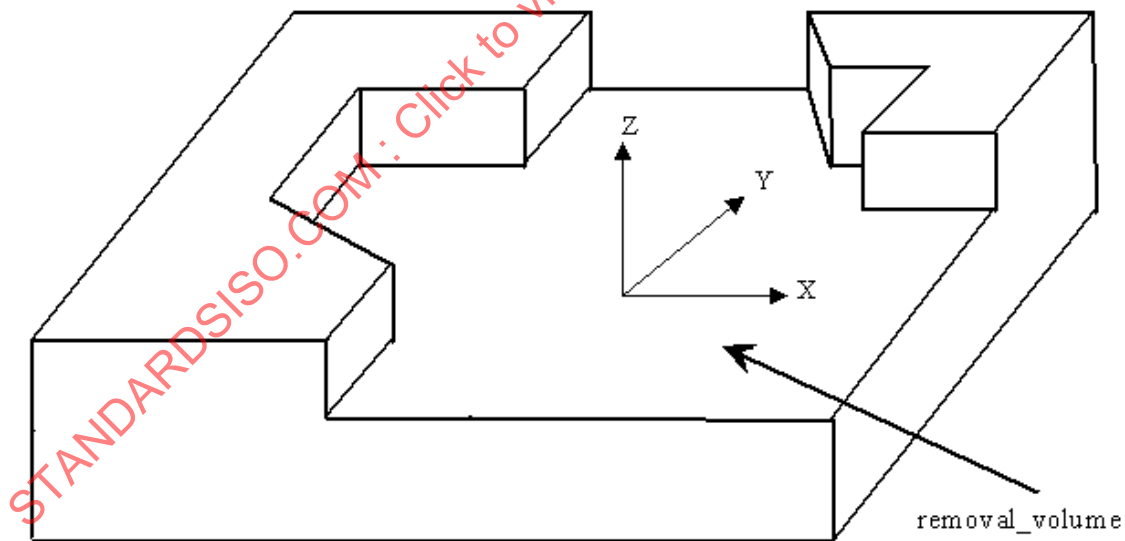


Figure 52 - General_removal_volume

4.2.84 General_revolution

A General_revolution is a type of Revolved_feature (see 4.2.160) that is an arbitrary planar shape swept one complete revolution about an axis. The arbitrary planar shape shall be finite in length, coplanar with the axis of revolution, and shall not intersect the axis of revolution. The General_revolution may be either an outer shape of a part or a volume removal, depending on the material direction.

NOTE - Figure 53 illustrates a General_revolution for an outer shape, the material direction points to the material side. Figure 54 illustrates a General_revolution for a volume removal.

The data associated with a General_revolution are the following: outer_edge_shape. The outer_edge_shape specifies an outline or shape that shall be revolved about an axis. The General_open_profile specifies the outer edge shape required by a General_revolution. This placement of the profile shall be on the Y-axis of the General_revolution at a specified distance away from the origin of the General_revolution. The X-axis and Y-axis of the profile shall be the same as the Y-axis and Z-axis of the General_revolution. See 4.3.81 for the application assertion.

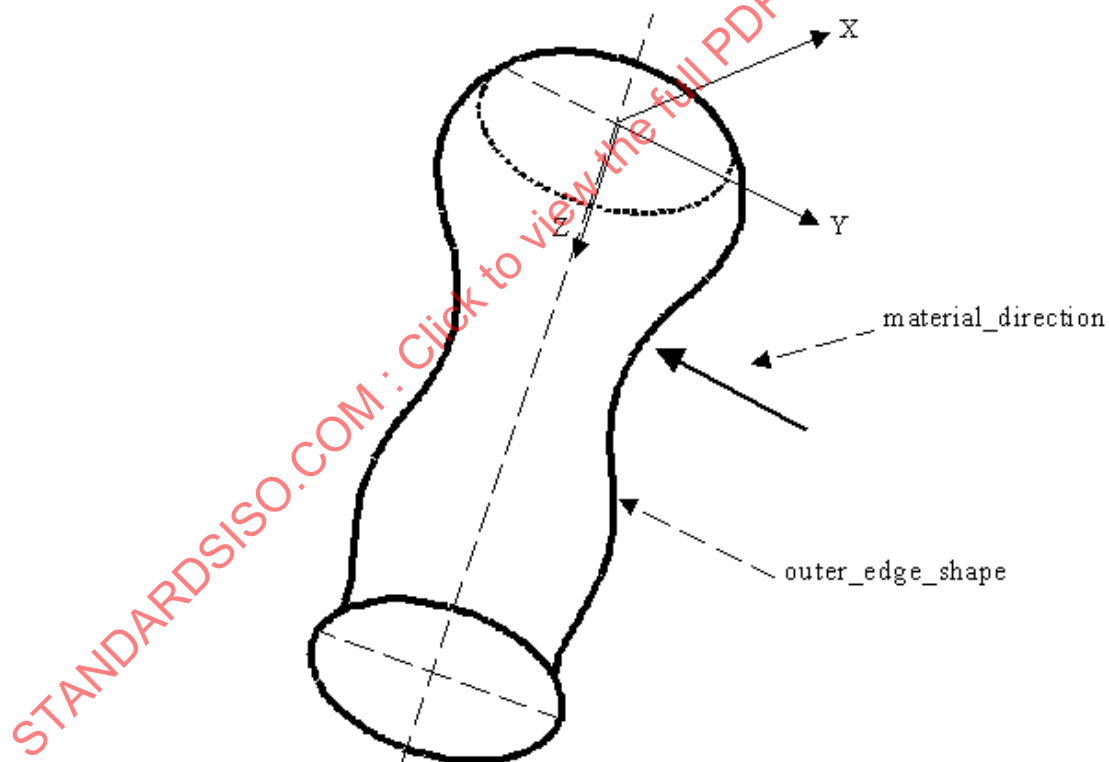


Figure 53 - General_revolution for outer shape

4.2.85 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. The data associated with a **General_top_condition** are the following: **top_face**. The **top_face** specifies a face at the top of a Boss feature, adjacent to the Boss sides. See 4.3.82 for the application assertion.

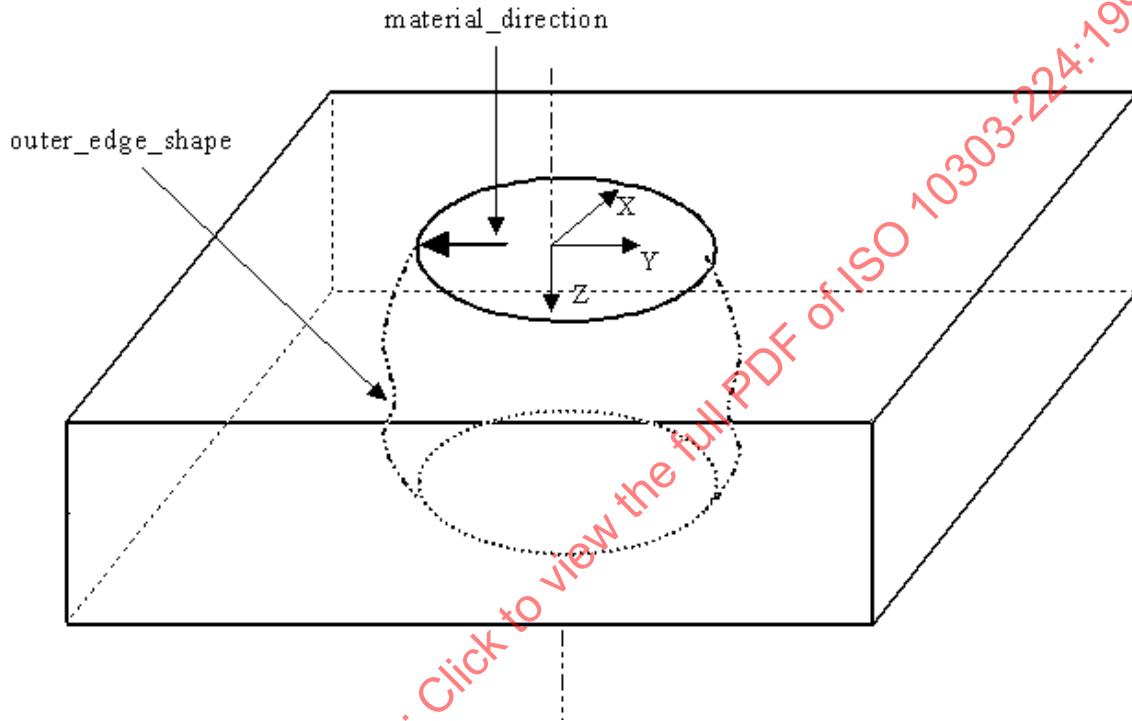


Figure 54 - General_revolution for volume removal

4.2.86 Geometric_tolerance

A **Geometric_tolerance** is the maximum or minimum variation from true geometric form or position that may be permitted in manufacture. **Geometric_tolerance** should be employed only for those requirements of a part critical to its functioning or interchangeability. Each **Geometric_tolerance** is one of the following: **Angularity_tolerance** (see 4.2.5), **Circular_runout_tolerance** (see 4.2.27), **Circularity_tolerance** (see 4.2.28), **Concentricity_tolerance** (see 4.2.35), **Cylindricity_tolerance** (see 4.2.45), **Flatness_tolerance** (see 4.2.74), **Linear_profile_tolerance** (see 4.2.96), **Parallelism_tolerance** (see 4.2.121), **Perpendicularity_tolerance** (see 4.2.130), **Position_tolerance** (see 4.2.140), **Straightness_tolerance** (see 4.2.182), **Surface_profile_tolerance** (see 4.2.183), **Symmetry_tolerance** (see 4.2.185), or a **Total_runout_tolerance** (see 4.2.200).

NOTE - Geometric tolerance definitions are derived from ISO 1101.

The data associated with a Geometric_tolerance are the following:

- applied_shape;
- geometric_tolerance_value;
- modifier_control;
- significant_digits;
- unit_of_measure;
- zone_definition.

4.2.86.1 applied_shape

The applied_shape specifies the shape on a Part that is being tolerated by a Geometric_tolerance. There may be more than one applied_shape for a Geometric_tolerance. See 4.3.84 for the application assertion.

4.2.86.2 geometric_tolerance_value

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

4.2.86.3 modifier_control

The modifier_control specifies the material condition which is applied to the shape being tolerated by the Geometric_tolerance. The modifier_control need not be specified for a particular Geometric_tolerance. See 4.3.83 for the application assertion.

4.2.86.4 significant_digits

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

4.2.86.5 unit_of_measure

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

4.2.86.6 zone_definition

The zone_definition specifies the tolerance zone that restricts the Geometric_tolerance. The zone_definition need not be specified for a particular Geometric_tolerance. See 4.3.85 for the application assertion.

4.2.87 Geometric_tolerance_precedence_relationship

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

- base_shape_tolerance;
- pattern_shape_tolerance.

4.2.87.1 base_shape_tolerance

The base_shape_tolerance specifies the Geometric_tolerance which is applied to a shape on a Part. See 4.3.86 for the application assertion.

4.2.87.2 pattern_shape_tolerance

The pattern_shape_tolerance specifies the Geometric_tolerance which is applied to a pattern on a Part. See 4.3.86 for the application assertion.

4.2.88 Groove

A Groove is a type of Revolved_feature (see 4.2.160) that is a narrow channel or depression that is swept through one complete revolution about an axis.

NOTE - Figure 55 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. The sweep specifies an outline or shape that shall be revolved about an axis. The Open_profile specifies the sweep shape required by a Groove. The placement of the profile shall be along the X-axis of the Groove at a specified distance away from the origin. The orientation of the Open_profile is independent of the orientation of the Groove feature. The Groove feature may be defined on different faces of a part depending on the orientation of the profile. See 4.3.87 for the application assertion.

4.2.89 Hardness

A Hardness is the resistance of a material surface to deformation by external forces.

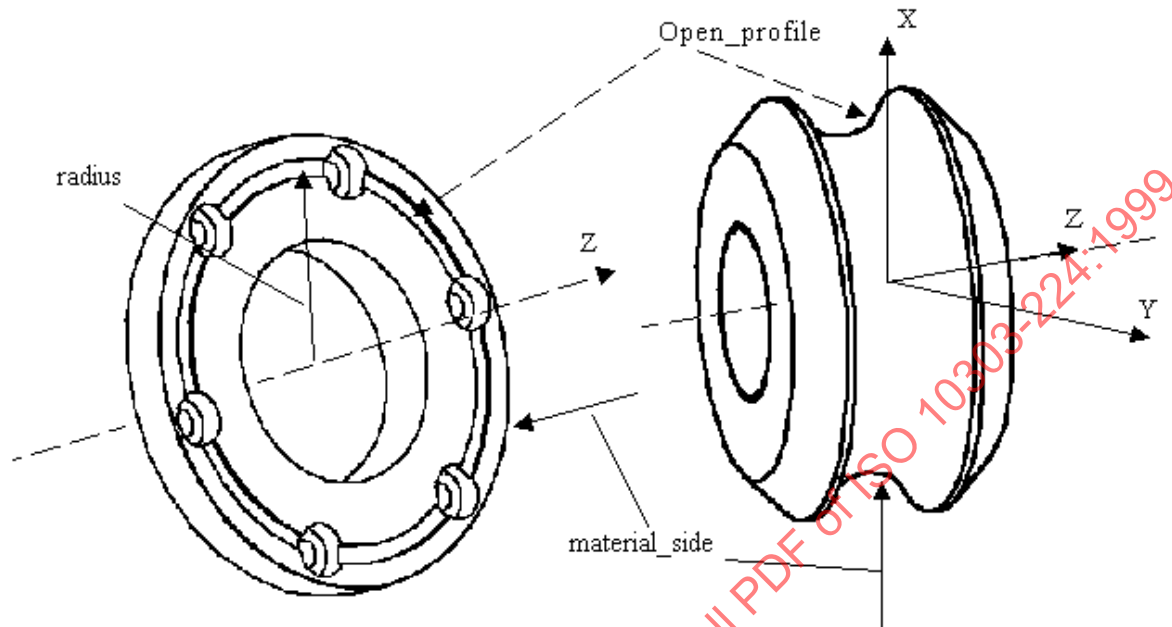


Figure 55 - Groove

The data associated with a Hardness are the following:

- high_value;
- low_value;
- nominal;
- scale.

4.2.89.1 high_value

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

4.2.89.2 low_value

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

4.2.89.3 nominal

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

4.2.89.4 scale

The scale specifies the method of determining hardness.

EXAMPLE - Rockwell and Brinell are examples of scale.

4.2.90 Hole

A Hole is a type of Machining_feature (see 4.2.101) that is the removal of a cylindrical volume from a part. Each Hole is either a Counterbore_hole (see 4.2.39), Countersunk_hole (see 4.2.40), or a Round_hole (see 4.2.163). The Hole may be positioned at its bottom with the Z-axis in the direction out of the Hole or at a position at the top of the hole with the Z-axis in the direction into the part.

4.2.91 Implicit_base_shape_representation

An Implicit_base_shape_representation is a type of Base_shape (see 4.2.7) that is the type of representation needed to define the shape 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 4.2.44), or a Ngon_base_shape (see 4.2.109).

NOTE - Figure 56 illustrates types of Implicit_base_shape_representations.

The data associated with an Implicit_base_shape_representation are the following:

- base_shape_length;
- placement.

4.2.91.1 base_shape_length

The base_shape_length is the size of the length of a Implicit_base_shape_representation. See 4.3.88 for the application assertion.

4.2.91.2 placement

A placement specifies the positioning of the part with respect to basic material stock. The positioning will be different for different types of Implicit_base_shape_representation. See 4.3.89 for the application assertion.

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

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

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

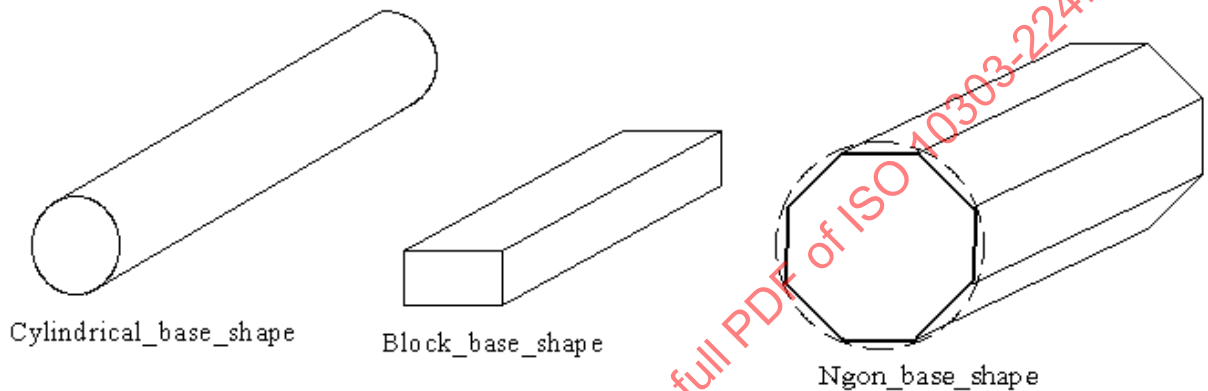


Figure 56 - Implicit_base_shape_representation

4.2.92 Indirect_stock_requisition

An `Indirect_stock_requisition` is a type of `Requisition` (see 4.2.158) that defines indirect items required to support part manufacturing.

4.2.93 Knurl

A `Knurl` is a type of `Machining_feature` (see 4.2.101) 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 4.2.16) or a `Turned_knurl` (see 4.2.202).

NOTE - Figure 57 illustrates a `Knurl` types.

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

- `applied_shape`;
- `partial_profile`.

4.2.93.1 applied_shape

The `applied_shape` specifies a base shape for applying the Knurl feature. The `applied_shape` need not be specified for a particular Knurl. See 4.3.90 for the application assertion.

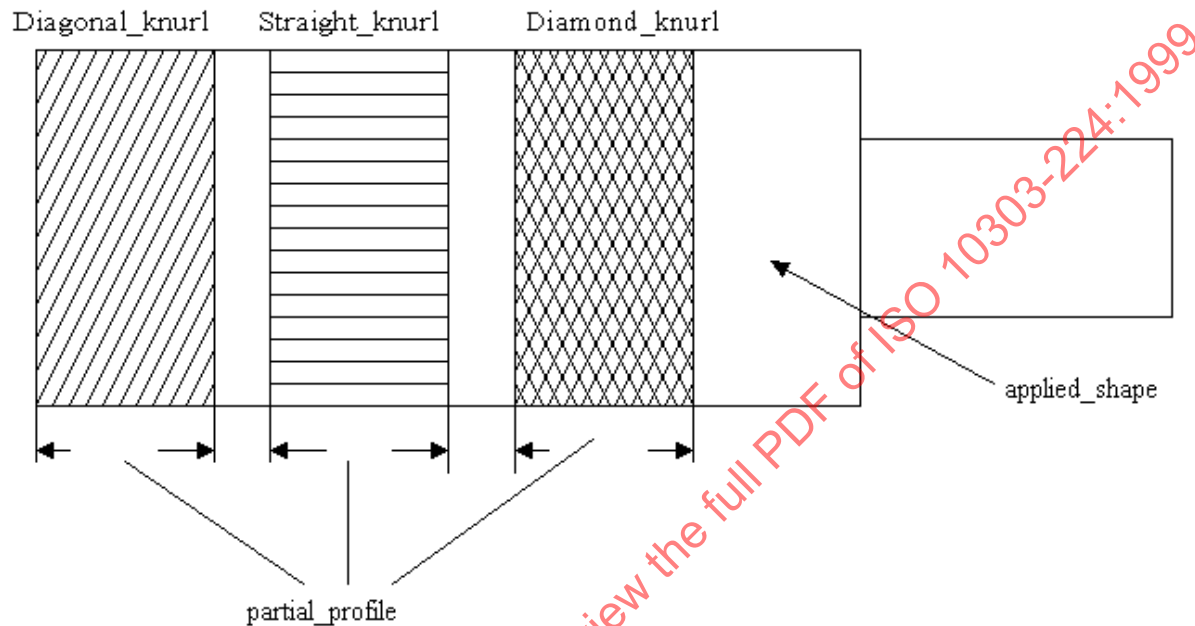


Figure 57 - Knurl

4.2.93.2 partial_profile

The `partial_profile` specifies the placement and length of a surface to apply a Knurl feature. See 4.3.91 for the application assertion.

4.2.94 Linear_path

A `Linear_path` is a type of `Path` (see 4.2.127) that is a direction of travel along a line.

NOTE - Figure 58 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: distance. The distance specifies the length of the path. See 4.3.92 for the application assertion.

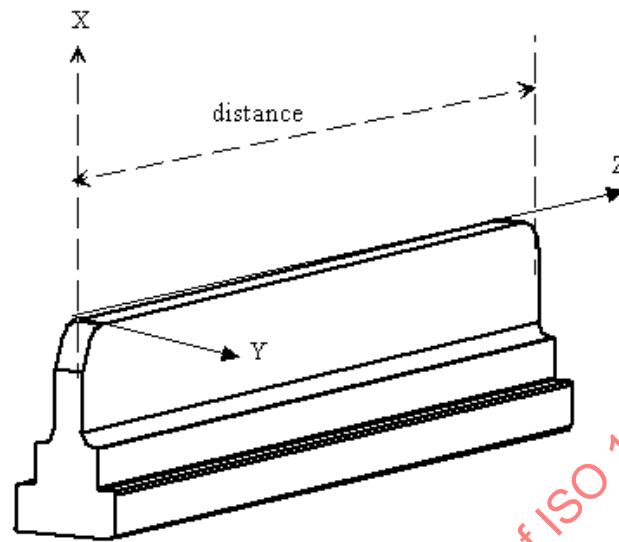


Figure 58 - Linear_path

4.2.95 Linear_profile

A Linear_profile is a type of Open_profile (see 4.2.113) that is a straight line of a specified length. The Linear_profile shall have orientation parallel to the X-axis.

NOTE - Figure 59 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. The profile_length specifies the length of the profile. See 4.3.93 for the application assertion.

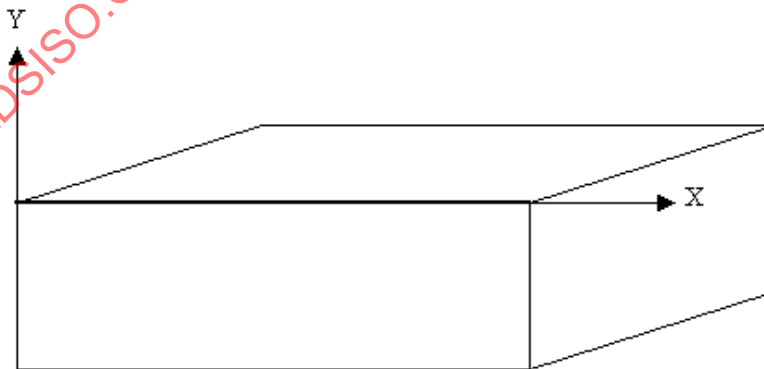


Figure 59 - Linear_profile

4.2.96 Linear_profile_tolerance

A Linear_profile_tolerance is a type of Geometric_tolerance (see 4.2.86) that is a uniform boundary or zone along the true profile within which all elements of the surface shall lie.

NOTES

1 - Figure 60 illustrates the Linear_profile_tolerance

2 - The Linear_profile_tolerance definition is derived from paragraph 14.5 of ISO 1101.

The data associated with a Linear_profile_tolerance are the following:

- affected_plane;
- geometric_reference.

4.2.96.1 affected_plane

The affected_plane specifies the plane to apply the Linear_profile_tolerance. The affected_plane need not be specified for a particular Linear_profile_tolerance. See 4.3.95 for the application assertion.

4.2.96.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.94 for the application assertion.

4.2.97 Location_dimension_tolerance

A Location_dimension_tolerance is a type of Location_tolerance (see 4.2.99) that is the allowable variation in locating one feature of a part with respect to another.

NOTE - Figure 61 illustrates a Location_dimension_tolerance.

The data associated with a Location_dimension_tolerance are the following:

- ← directed;
- plane_and_direction.

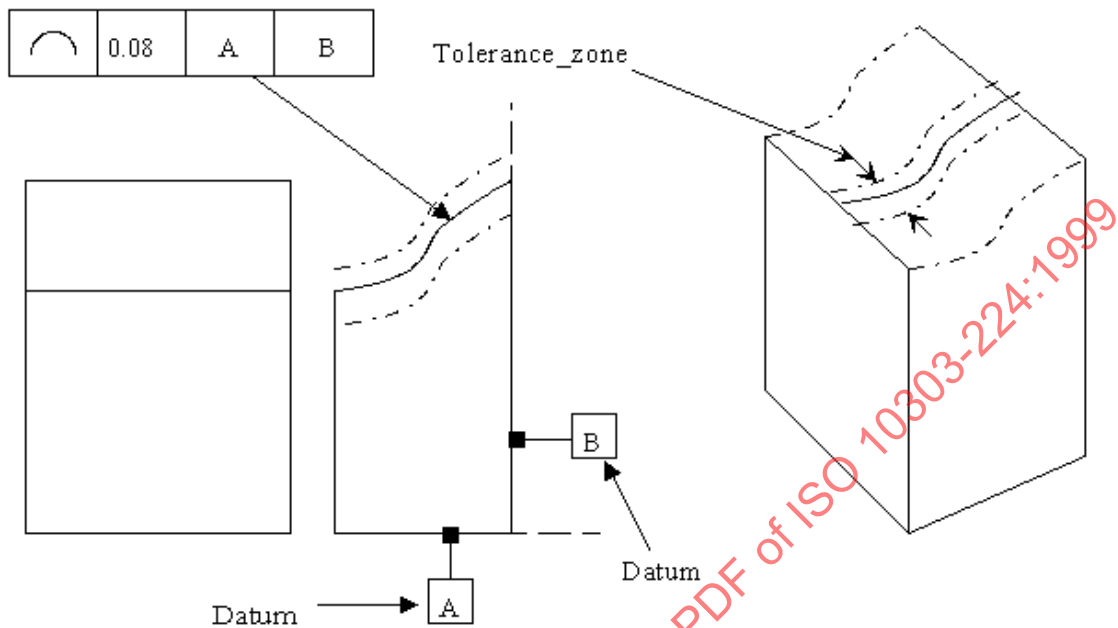


Figure 60 - Linear_profile_tolerance

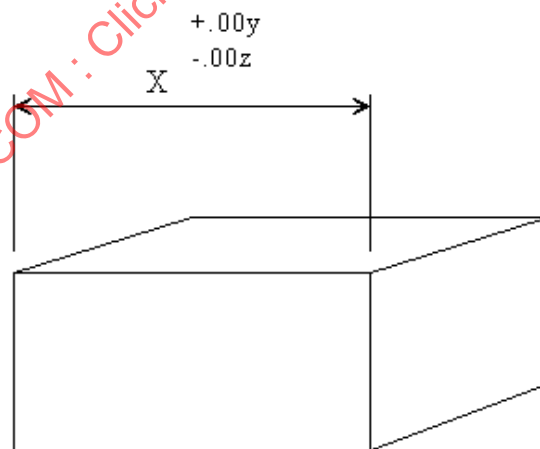


Figure 61 - Location_dimension_tolerance

4.2.97.1 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.

4.2.97.2 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 4.3.96 for the application assertion.

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

4.2.98 Location_element

A Location_element is a type of Shape_element (see 4.2.170) that is a Shape_aspect representation for a reference point.

4.2.99 Location_tolerance

A Location_tolerance is a type of Dimensional_tolerance (see 4.2.60) 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 4.2.3.1), Location_dimension_tolerance (see 4.2.97), or Distance_along_curve_tolerance (see 4.2.62). The data associated with a Location_tolerance are the following:

- origin_shape;
- termination_shape.

4.2.99.1 origin_shape

The origin_shape specifies the shape on the Part that defines the starting position for a Location _tolerance. See 4.3.97 for the application assertion.

4.2.99.2 termination_shape

The termination_shape specifies the shape on the Part that defines the ending position for a Location_tolerance. See 4.3.97 for the application assertion.

4.2.100 Machine_requisition

A Machine_requisition is a type of Requisition (see 4.2.158) that describes machines required to support part manufacturing.

4.2.101 Machining_feature

A Machining_feature is a type of Manufacturing_feature (see 4.2.102) 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: Boss (see 4.2.10), General_outside_profile (see 4.2.78), General_removal_volume (see 4.2.83), Hole (see 4.2.90), Knurl (see 4.2.93), Outer_round (see 4.2.120), Revolved_feature (see 4.2.160), Rounded_end (see 4.2.164), Planar_face (see 4.2.134), Pocket (see 4.2.138), Protrusion (see 4.2.147), Slot (see 4.2.173), Step (see 4.2.180), Thread (see 4.2.192), Marking (see 4.2.103), Spherical_cap (see 4.2.177), or a Compound_feature (see 4.2.32). The data associated with a Machining_feature are the following:

- placement;
- usage_name.

4.2.101.1 placement

The placement specifies the position and orientation of a Machining_feature relative to the base shape for a part. See 4.3.98 for the application assertion.

4.2.101.2 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.

4.2.102 Manufacturing_feature

A Manufacturing_feature is a type of Shape_element (see 4.2.170) that identifies the types of features necessary to manufacture a machined part. Each Manufacturing_feature is either a Machining_feature (see 4.2.101), a Replicate_feature (see 4.2.157), or a Transition_feature (see 4.2.201).

4.2.103 Marking

A Marking is a type of Machining_feature (see 4.2.101) that is one or more text characters on a surface of a part. Each Marking is either a Defined_marking (see 4.2.51) or a Catalogue_marking (see 4.2.17).

The data associated with a Marking are the following:

- applied_to_shape;
- text.

4.2.103.1 applied_to_shape

The applied_to_shape specifies a base shape for applying the Marking feature. See 4.3.100 for the application assertion.

4.2.103.2 text

The text specifies the characters that will be applied to the part. See 4.3.99 for the application assertion.

4.2.104 Material

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

- material_characteristics;
- material_description;
- material_id;
- material_specification;
- stock_size.

4.2.104.1 material_characteristics

The material_characteristics specifies the properties which define the Material. The material_characteristics need not be specified for a particular Material. There may be more than one material_characteristics for a Material. See 4.3.101 for the application assertion.

4.2.104.2 material_description

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

4.2.104.3 material_id

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

4.2.104.4 material_specification

The material_specification specifies the documentation which contains additional information about Material. The material_specification need not be specified for a particular Material. There may be more than one material_specification for a Material. See 4.3.102 for the application assertion.

4.2.104.5 stock_size

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

4.2.105 Material_condition_modifier

A Material_condition_modifier is a description of a condition for a part surface wherein it contains the amount of material. Tolerances of position are determined by a material condition of a mating surfaces. The data associated with a Material_condition_modifier are the following: 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).

4.2.106 Material_property

A Material_property is the information about the material that describes its shape characteristic for a part. The data associated with a Material_property are the following:

- material_hardness;
- property_characteristic.

4.2.106.1 material_hardness

The material_hardness specifies additional information to define hardness properties. The material_hardness need not be specified for a particular Material_property. There may be more than one material_hardness for a Material_property. See 4.3.103 for the application assertion.

4.2.106.2 property_characteristic

The property_characteristic specifies the parameter for the description of Material_property. The property_characteristics need not be specified for a particular Material_property. There may be more than one property_characteristics for a Material_property. See 4.3.104 for the application assertion.

4.2.107 Material_requisition

A Material_requisition is a type of Requisition (see 4.2.158) which describes an order for purchase of raw materials by the manufacturing organization.

4.2.108 Modular_fixture_requisition

A Modular_fixture_requisition is a type of Requisition (see 4.2.158) which describes an order for purchase of modular fixtures by the manufacturing organization.

4.2.109 Ngon_base_shape

An Ngon_base_shape is a type of Implicit_base_shape_representation (see 4.2.91) that specifies the initial shape of the material is a polygon with any number of sides.

NOTE - Figure 62 illustrates a Ngon_base_shape.

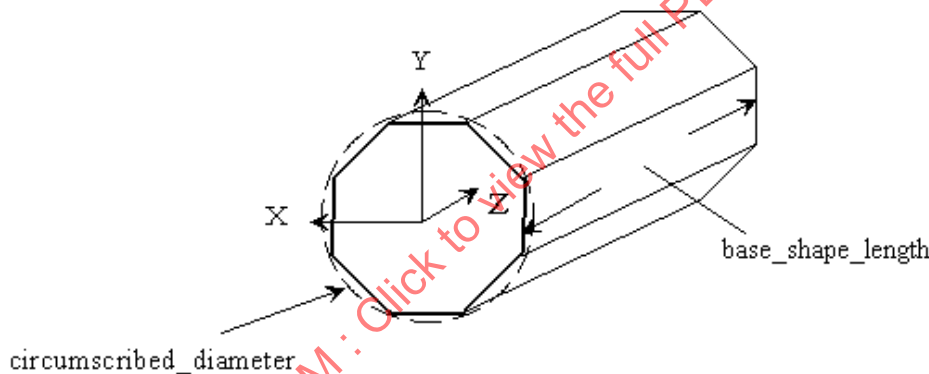


Figure 62 - Ngon_base_shape

The data associated with a Ngon_base_shape are the following:

- circumscribed_diameter;
- corner_radius;
- number_of_sides.

4.2.109.1 circumscribed_diameter

The `circumscribed_diameter` specifies the size of the circle which is circumscribed about the ngon. See 4.3.105 for the application assertion.

4.2.109.2 corner_radius

The `corner_radius` specifies the size of an arc blend between two sides of the ngon. See 4.3.105 for the application assertion.

4.2.109.3 number_of_sides

The `number_of_sides` specifies how many sides are needed for the Ngon. See 4.3.105 for the application assertion.

4.2.110 Ngon_profile

An `Ngon_profile` is a type of `Closed_profile` (see 4.2.29) that is an enclosed area bounded by three or more connected straight line sides. The orientation is at the center of the profile with one side of the ngon parallel to the X-axis crossing the Y-axis at a negative value.

NOTE - Figure 63 illustrates the `Ngon_profile`.

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

- `circumscribed_diameter`;
- `corner_radius`;
- `number_of_sides`.

4.2.110.1 circumscribed_diameter

The `circumscribed_diameter` specifies the size of the circle which surrounds the ngon. See 4.3.106 for the application assertion.

4.2.110.2 corner_radius

The `corner_radius` specifies the size of an arc blend between two sides of the ngon. See 4.3.106 for the application assertion.

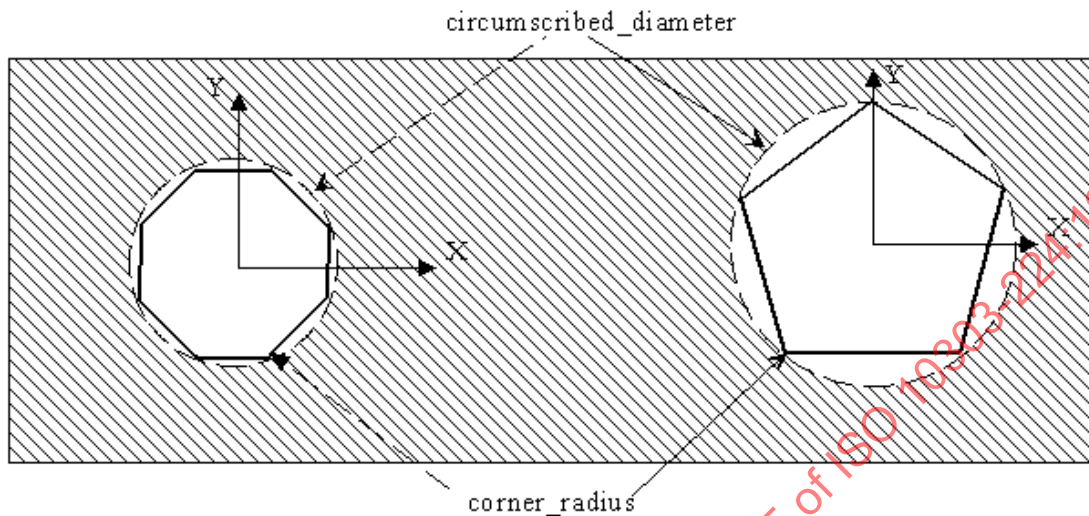


Figure 63 - Ngon_profile

4.2.110.3 number_of_sides

The `number_of_sides` specifies how many sides are needed for the Ngon. See 4.3.106 for the application assertion.

4.2.111 Numeric_parameter

A `Numeric_parameter` is a type of `Property_parameter` (see 4.2.146) that a numeric value with units of the property being defined. A `Numeric_parameter` is either a `Numeric_parameter` or a `Numeric_parameter_with_tolerance` (see 4.2.112). The data associated with a `Numeric_parameter` are the following:

- `parameter_units`,
- `parameter_value`.

4.2.111.1 parameter_units

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

EXAMPLE - watt, meters, degrees, etc.

4.2.111.2 parameter_value

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

4.2.112 Numeric_parameter_with_tolerance

A Numeric_parameter_with_tolerance is a type of Numeric_parameter (see 4.2.111) with an implied tolerance value.

NOTE - A thread has a implicit definition for the minor_diameter attribute. This attribute has no explicit geometry definition, so the dimensional tolerance of this attribute is represented with Numeric_parameter_with_tolerance.

The data associated with a Numeric_parameter_with_tolerance are the following: implicit_tolerance. The implicit_tolerance specifies the type of tolerance to apply to a numeric parameter value. See 4.3.107 and 4.3.108 for the application assertion.

4.2.113 Open_profile

An Open_profile is a type of Profile (see 4.2.142) that is an outline or shape with no enclosing or confining bounds. The open ends of the profile may extend infinitely. Each Open_profile is either a General_open_profile (see 4.2.77), Linear_profile (see 4.2.95), Partial_circular_profile (see 4.2.126), Rounded_U_profile (see 4.2.165), Square_U_profile (see 4.2.179), Tee_profile (see 4.2.191), or a Vee_profile (see 4.2.203).

4.2.114 Open_slot_end_type

An Open_slot_end_type is a type of Slot_end_type (see 4.2.174) that is an end condition of a slot that shall pass through the end of the part.

NOTE - Figure 64 illustrates a slot with two Open_slot_end_type objects.

4.2.115 Ordered_part

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

- quantity_required;
- quantity_unit_of_measure.

4.2.115.1 quantity_required

The quantity_required specifies the number of items to be manufactured.

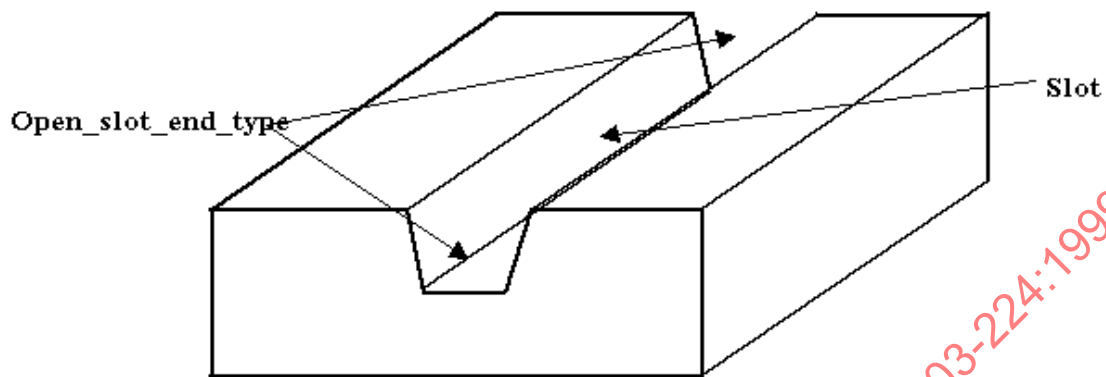


Figure 64 - Open_slot_end_type

4.2.115.2 quantity_unit_of_measure

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

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

4.2.116 Organization

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

- organization_address;
- organization_id;
- organization_name.

4.2.116.1 organization_address

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

4.2.116.2 organization_id

The organization_id specifies the unique identification of an organization.

4.2.116.3 organization_name

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

4.2.117 Orientation

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

- axis;
- location.

4.2.117.1 axis

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

4.2.117.2 location

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

4.2.118 Outer_diameter

An Outer_diameter is a type of Outer_round (see 4.2.120) that is a sweeping of an outline specified by a line segment one complete revolution about an axis. The line is finite in length, coplanar with the axis. An Outer_diameter may have a constant diameter around the axis of rotation, or it may be tapered.

NOTE - Figure 65 illustrates the Outer_diameter with and without a taper.

The data associated with an Outer_diameter are the following:

- diameter;
- feature_length;
- reduced_size.

4.2.118.1 diameter

The diameter specifies the maximum diametric size of an Outer_diameter feature. See 4.3.109 for the application assertion.

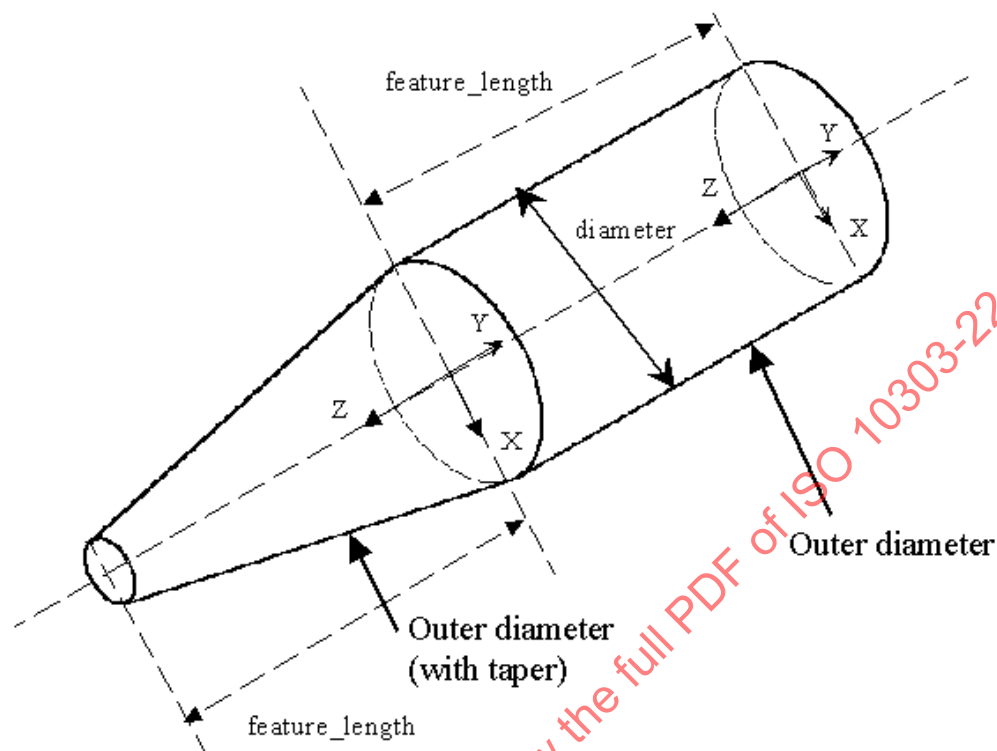


Figure 65. Outer_diameter

4.2.118.2 feature_length

The length specifies the size of a Outer_diameter feature, measured along the feature's axis. See 4.3.109 for the application assertion.

4.2.118.3 reduced_size

The reduced_size specifies the constant change in the Outer_diameter along the feature length. See 4.3.110 and 4.3.111 for the application assertion.

4.2.119 Outer_diameter_to_should

An Outer_diameter_to_should is a type of Outer_round (see 4.2.120) that is a sweeping of a shape one complete revolution about an axis. The shape shall be specified by two lines that connect at a point and extend infinitely. The enclosed angle shall be smaller than a straight angle. The intersection of the two lines need not be blended with a radius.

NOTE - Figure 66 illustrates the Outer_diameter_to_should.

The data associated with an Outer_diameter_to_shoulder are the following:

- diameter;
- v_shape_boundary.

4.2.119.1 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 4.3.112 for the application assertion.

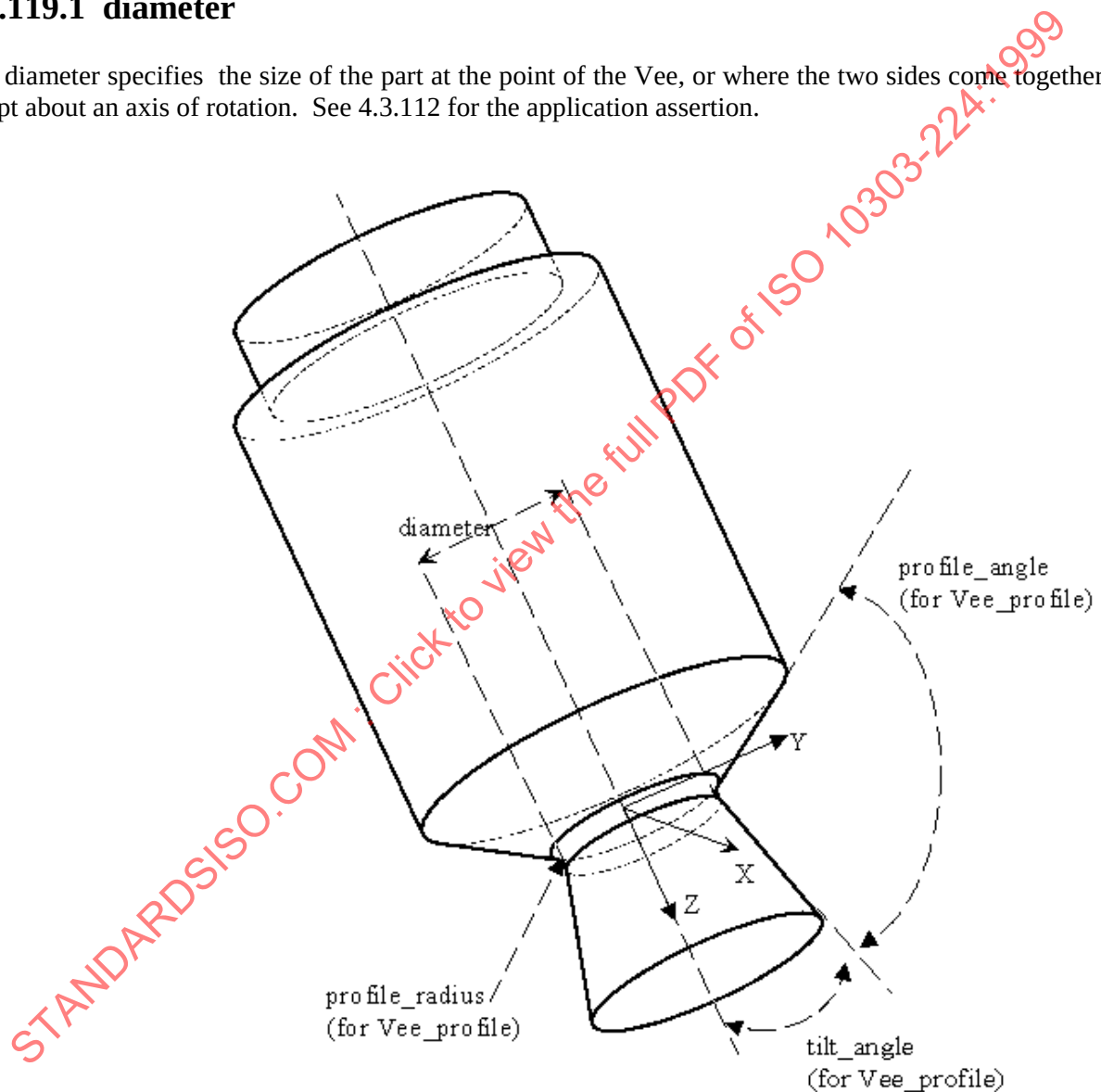


Figure 66 - Outer_diameter_to_shoulder

4.2.119.2 v_shape_boundary

The v_shape_boundary specifies an outline or shape that shall be revolved about an axis. The Vee_profile specifies the revolved shape required by an Outer_diameter_to_shoulder. The placement of the profile shall be along the X-axis of the Outer_diameter_to_shoulder at a specified distance away from the origin. The orientation of the Y-axis of the Vee_profile shall be the same as the Y-axis of the Outer_diameter_to_shoulder and the X-axis of the Vee_profile shall be the same as the Z-axis of the Outer_diameter_to_shoulder. See 4.3.113 for the application assertion.

4.2.120 Outer_round

An Outer_round is a type of Machining_feature (see 4.2.101) 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 4.2.118) or an Outer_diameter_to_shoulder (see 4.2.119). The axis of revolution shall be the same as the Z-axis of the feature.

4.2.121 Parallelism_tolerance

A Parallelism_tolerance is a type of Geometric_tolerance (see 4.2.86) 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 67 illustrates Parallelism_tolerance for a plane.

NOTE 2 - Figure 68 illustrates Parallelism_tolerance for an axis.

NOTE 3 - The Parallelism_tolerance definition is derived from paragraph 14.7 of ISO 1101.

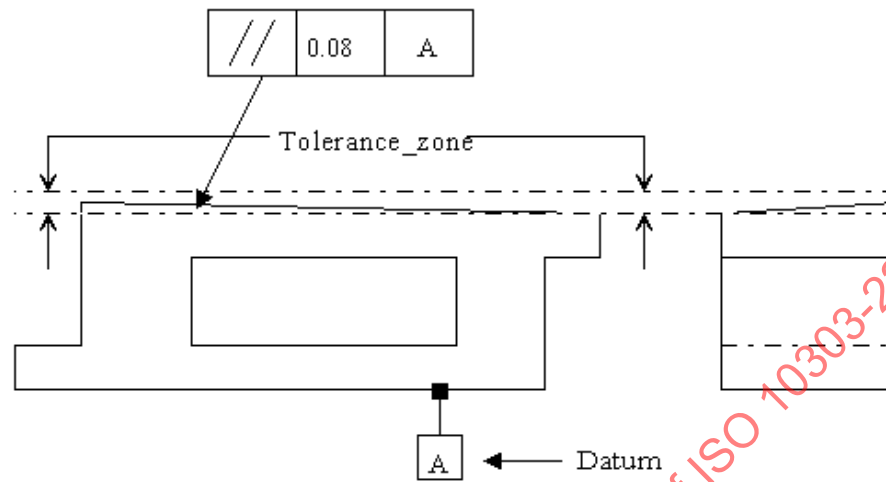


Figure 67 - Parallelism_tolerance for a plane

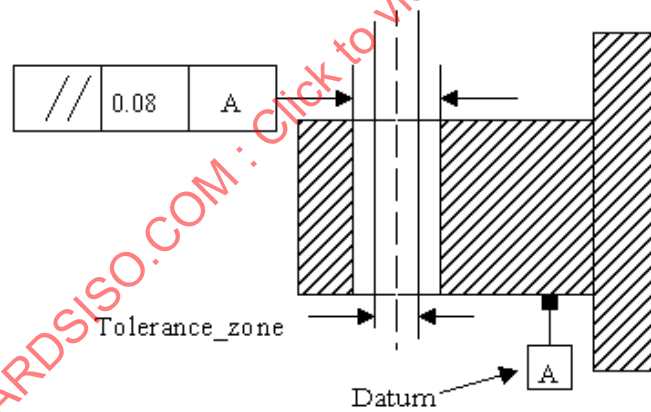


Figure 68 - Parallelism_tolerance for an axis

The data associated with a Parallelism_tolerance are the following:

- affected_plane;
- geometric_reference;
- segment_size.

4.2.121.1 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 4.3.115 for the application assertion.

4.2.121.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.114 for the application assertion.

4.2.121.3 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.

4.2.122 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:

- alternate_material_definition;
- manufacture_authorization;
- manufactured_by_organization;
- manufactured_by_person;
- material_definition;
- owned_by_organization;
- owned_by_person;
- part_description;

- part_id;
- part_name;
- part_revision_id;
- physical_form;
- property characteristics;
- quantity_ordered;
- security_classification.

4.2.122.1 alternate_material_definition

The `alternate_material_definition` specifies the secondary material choices of raw stock for producing the Part. There may be more than one `alternate_material_definition` for a Part. See 4.3.116 for the application assertion.

4.2.122.2 manufacture_authorization

The `manufacture_authorization` specifies the approval description for the manufacture of a Part. The `manufacture_authorization` need not be specified for a particular Part. There may be more than one `manufacture_authorization` for a Part. See 4.3.117 for the application assertion.

4.2.122.3 manufactured_by_organization

The `manufactured_by_organization` specifies the organization responsible for the manufacture of the Part. There may be more than one `manufactured_by_organization` for a Part. See 4.3.120 for the application assertion.

4.2.122.4 manufactured_by_person

The `manufactured_by_person` specifies the person responsible for the manufacture of the Part. There may be more than one `manufactured_by_person` for a Part. See 4.3.121 for the application assertion.

4.2.122.5 material_definition

The `material_definition` specifies primary material choice of raw stock for producing the Part. There may be more than one `material_definition` for a Part. See 4.3.118 for the application assertion.

4.2.122.6 owned_by_organization

The `owned_by_organization` specifies the organization that will own the completed manufactured Part. There may be more than one `owned_by_organization` for a Part. See 4.3.120 for the application assertion.

4.2.122.7 owned_by_person

The `owned_by_person` specifies the person that will own the completed manufactured Part. There may be more than one `owned_by_person` for a Part. See 4.3.121 for the application assertion.

4.2.122.8 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`.

4.2.122.9 part_id

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

4.2.122.10 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.

4.2.122.11 part_revision_id

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

4.2.122.12 physical_form

The `physical_form` specifies the shape of the Part. See 4.3.123 for the application assertion.

4.2.122.13 property_characteristic

The `property_characteristic` specifies the associated data about the physical structure. The `property_characteristic` need not be specified for a particular Part. There may be more than one `property_characteristic` for a Part. See 4.3.122 for the application assertion.

4.2.122.14 quantity_ordered

The `quantity_ordered` specifies the number of Parts to be manufactured. There may be more than one `quantity_ordered` for a Part. See 4.3.119 for the application assertion.

4.2.122.15 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.

4.2.123 Part_property

A `Part_property` is a specific characteristic about the form, fit, or function of a part. The data associated with a `Part_property` are the following: `property_characteristic`. The `property_characteristic` specifies the parameter to describe the `Part_property`. The `property_characteristic` need not be specified for a particular `Part_property`. There may be more than one `property_characteristic` for a `Part_property`. See 4.3.124 for the application assertion.

4.2.124 Partial_area_definition

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

NOTE - Figure 34 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:

- `applied_length`;
- `placement`.

4.2.124.1 applied_length

The `applied_length` specifies the dimension along a surface to apply a feature. Thread and Knurl are types of `Machining_features` that are applied to surfaces. The length limits the dimension along the cylindrical surface axis for defining these feature objects. See 4.3.125 for the application assertion.

4.2.124.2 placement

The `placement` specifies where to locate the `Partial_area_definition`. See 4.3.126 for the application assertion.

4.2.125 Partial_circular_path

A **Partial_circular_path** is a type of **Circular_path** (see 4.2.25) that is a direction of travel along an arc of constant radius around an axis. The path shall begin and end at different points on the arc.

NOTE - Figure 69 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**. The **sweep_angle** specifies the size of the angle to define an arc shaped path. See 4.3.127 for the application assertion.

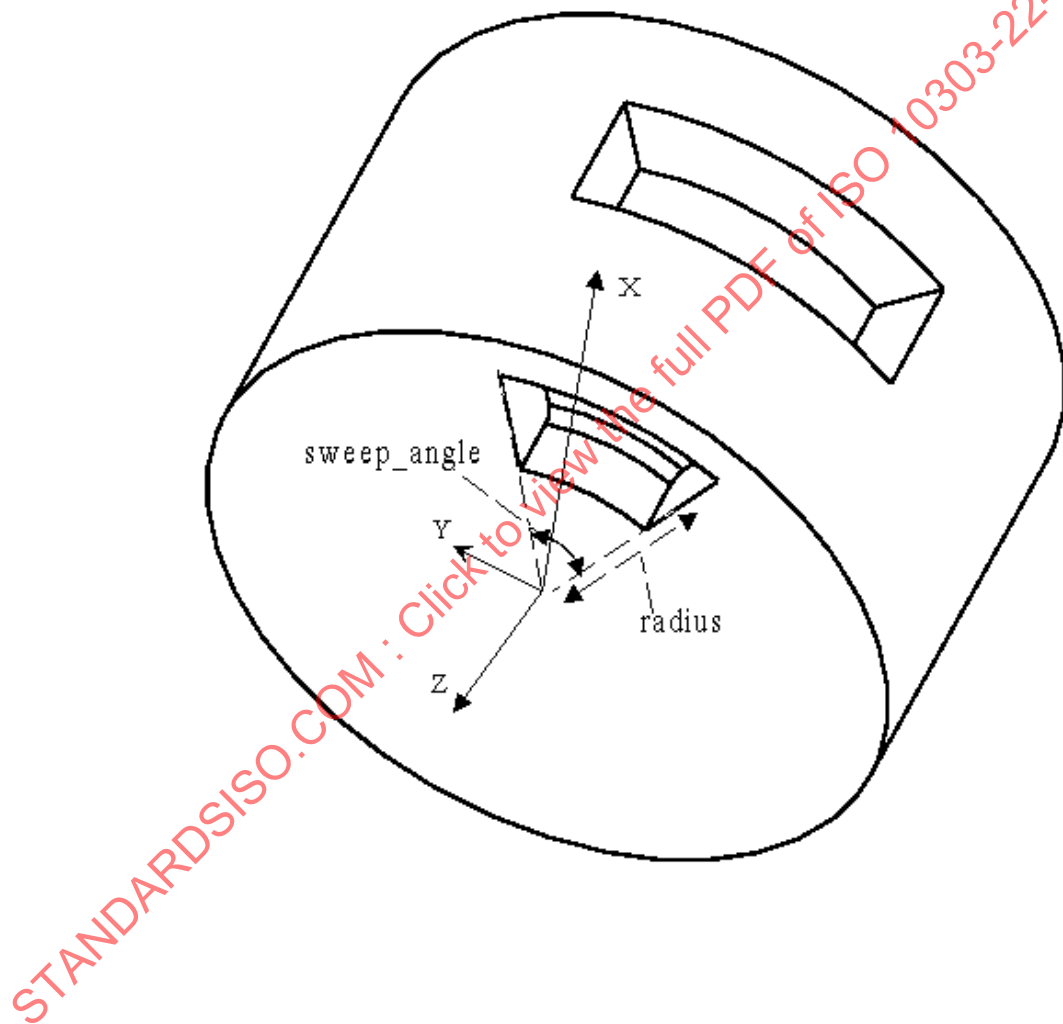


Figure 69 - Partial_circular_path

4.2.126 Partial_circular_profile

A Partial_circular_profile is a type of Open_profile (see 4.2.113) that is specified by an arc. The arc shall be a constant radius swept about a point. The orientation of the profile shall be positioned at the origin of the arc, with one end point of the arc on the X-axis.

NOTE - Figure 69 illustrates a Partial_circular_profile.

The data associated with a Partial_circular_profile are the following:

- radius;
- sweep_angle.

4.2.126.1 radius

The radius specifies the size of the arc to define a Partial_circular_profile. See 4.3.128 for the application assertion.

4.2.126.2 sweep_angle

The sweep_angle specifies the size of the angle to define an circular shaped profile. See 4.3.128 for the application assertion.

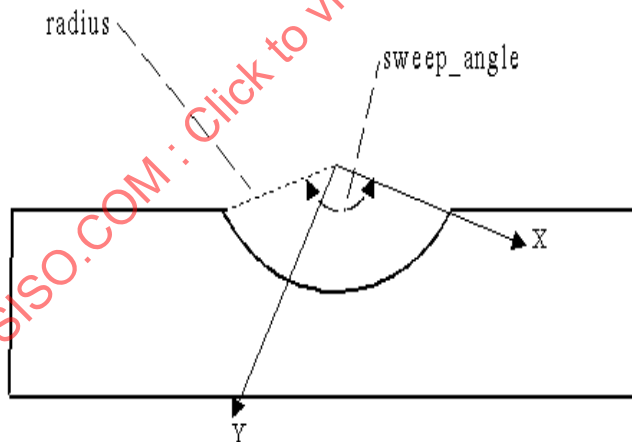


Figure 70 - Partial_circular_profile

4.2.127 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 4.2.25), General_path (see 4.2.79), or a Linear_path (see 4.2.94).

The data associated with a Path are the following: placement. The placement specifies where to locate the Path. See 4.3.129 for the application assertion.

4.2.128 Path_element

A Path_element is a type of Shape_element (see 4.2.170) that is a continuous set of geometric curve that represent the path for a particular Machining_feature.

4.2.129 Pedigree_creation_order

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

4.2.130 Perpendicularity_tolerance

A Perpendicularity_tolerance is a type of Geometric_tolerance (see 4.2.86) that is the amount of deviation of a surface to a perpendicular to a datum. The actual surface shall lie within two planes a tolerance apart and perpendicular to a datum surface.

NOTE 1 - Figure 71 illustrates a Perpendicularity_tolerance.

NOTE 2 - The Perpendicularity_tolerance definition is derived from paragraph 14.8 of ISO 1101.

The data associated with a Perpendicularity_tolerance are the following:

- affected_plane;
- geometric_reference;
- segment_size.

4.2.130.1 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 4.3.131 for the application assertion.

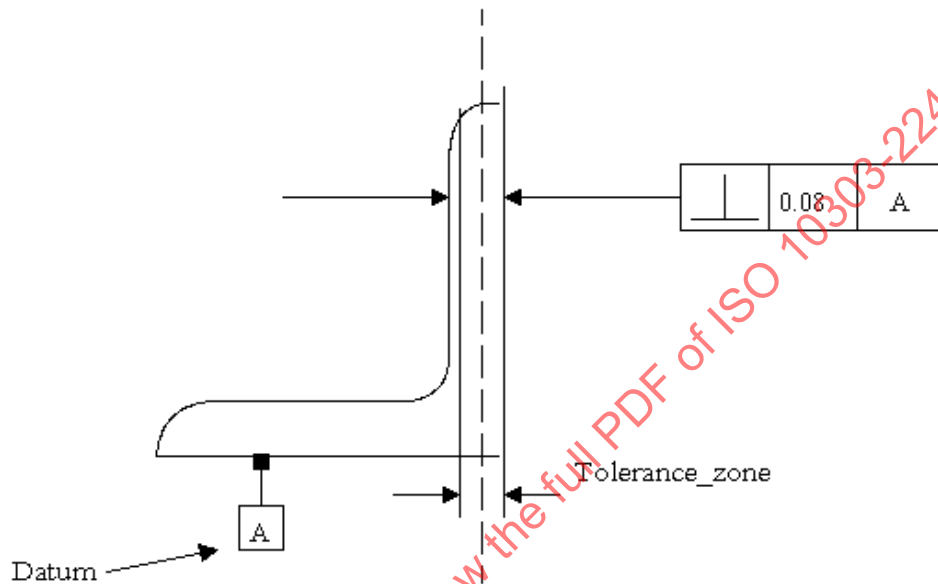


Figure 71 - Perpendicularity_tolerance

4.2.130.2 geometric_reference

The `geometric_reference` specifies the datum to which the tolerance is related. See 4.3.130 for the application assertion.

4.2.130.3 segment_size

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

4.2.131 Person

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

- `person_address`;
- `person_id`;

- person_name;
- person_phone_number.

4.2.131.1 person_address

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

4.2.131.2 person_id

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

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

4.2.131.3 person_name

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

4.2.131.4 person_phone_number

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

4.2.132 Person_in_organization

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

- company;
- employee;
- role.

4.2.132.1 company

The company specifies the functional group that employ's the Person. See 4.3.132 for the application assertion.

4.2.132.2 employee

The employee specifies Person employed by the Organization. See 4.3.133 for the application assertion.

4.2.132.3 role

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

4.2.133 Placed_target

A Placed_target is a type of Datum_target (see 4.2.48) that is the implicit definition of a Datum_target. A Placed_target is either a Target_point (see 4.2.189), Target_line (see 4.2.188), Target_rectangle (see 4.2.190), or a Target_circle (see 4.2.187). The data associated with Placed_target are the following: placement. The placement specifies location and orientation for the implicit definitions of the types of Placed_target. See 4.3.134 for the application assertion.

4.2.134 Planar_face

A Planar_face is a type of Machining_feature (see 4.2.101) that is an unbounded planar cut of a part. The Planar_face shall have an orientation such that the Z-axis is the direction away from the part.

NOTE - Figure 72 illustrates the Planar_face.

The data associated with a Planar_face are the following:

- course_of_travel;
- removal_boundary;
- removal_direction.

4.2.134.1 course_of_travel

The course_of_travel specifies a straight line with magnitude and direction. The placement and orientation of the Linear_path shall be the same as the Planar_face feature. See 4.3.136 for the application assertion.

4.2.134.2 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 4.3.137 for the application assertion.

4.2.134.3 removal_direction

The `removal_direction` specifies the direction of material removal from the `Planar_face` feature. See 4.3.135 for the application assertion.

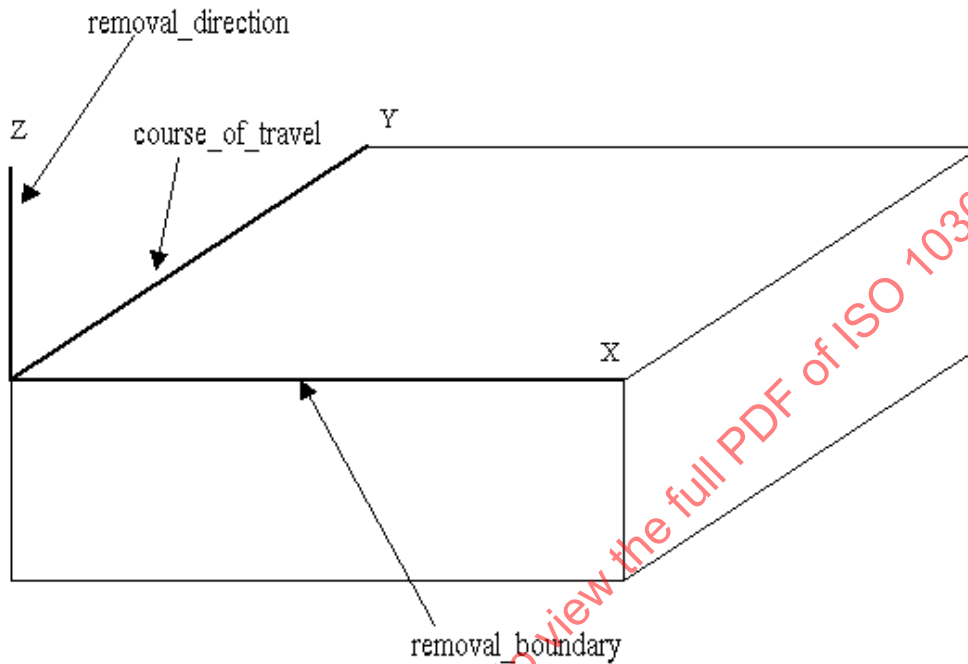


Figure 72 - Planar_face

4.2.135 Planar_pocket_bottom_condition

A `Planar_pocket_bottom_condition` is a type of `Pocket_bottom_condition` (see 4.2.139) that characterizes the bottom of a pocket which is flat.

NOTE - Figure 77 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`.

4.2.135.1 floor_location

The floor_location specifies the position of the bottom of a pocket feature for a planar floor. See 4.3.139 for the application assertion.

4.2.135.2 floor_normal

The floor_normal specifies the vector which indicates the normal of a plane being defined for the bottom of a pocket. See 4.3.138 for the application assertion.

4.2.135.3 floor_radius

The floor_radius specifies the amount of curvature for an arc between the bottom and the sides of a Pocket feature. See 4.3.140 for the application assertion.

4.2.136 Planar_top_condition

A Planar_top_condition is a type of Boss_top_condition (see 4.2.11) that is the top of a Boss which is flat.

NOTE - Figure 13 illustrates a Circular_boss with a Planar_top_condition.

The data associated with a Planar_top_condition are the following:

- top_location;
- top_normal.

4.2.136.1 top_location

The top_location specifies the position of the top of a boss feature for a planar top. See 4.3.142 for the application assertion.

4.2.136.2 top_normal

The top_normal specifies the vector which indicates the normal of a plane being defined for the top of a boss. See 4.3.141 for the application assertion.

4.2.137 Plus_minus_value

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

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.

4.2.137.1 lower_limit

The lower_limit specifies the low limit value.

4.2.137.2 significant_digits

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

4.2.137.3 upper_limit

The upper_limit specifies the high limit value.

4.2.138 Pocket

A Pocket is a type of Machining_feature (see 4.2.101) 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 General_pocket (see 4.2.81), Rectangular_closed_pocket (see 4.2.150), or a Rectangular_open_pocket (see 4.2.154). The data associated with a Pocket are the following:

- bottom_condition;
- change_in_boundary;
- pocket_depth.

4.2.138.1 bottom_condition

The bottom_condition specifies the shape of the bottom of a Pocket feature. See 4.3.145 and 4.3.146 for the application assertions.

4.2.138.2 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 4.3.144 for the application assertion.

4.2.138.3 pocket_depth

The pocket_depth specifies an amount of distance from the bottom of a pocket to a point that is outside of the pocket feature. Pocket_depth places a limitation on the Pocket definition so not to interfere with other features that might be nearby. The placement and orientation of the Linear_path that defines pocket_depth shall be the same as the Pocket feature. See 4.3.143 for the application assertion.

4.2.139 Pocket_bottom_condition

A Pocket_bottom_condition is the bottom state for a pocket. The Pocket bottom may be flat, of any arbitrary shape, or the pocket may pass through the part. Each Pocket_bottom_condition is either a General_pocket_bottom_condition (see 4.2.82), or a Planar_pocket_bottom_condition (see 4.2.135).

The data associated with a Pocket_bottom_condition are the following: 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 Pocket, and a value of FALSE if it is at the end of the Pocket.

4.2.140 Position_tolerance

A Position_tolerance is a type of Geometric_tolerance (see 4.2.86) that denotes the a tolerance zone for a theoretically exact position tolerance of a surface, and is established with respect to a datum.

NOTE 1 - Figure 73 illustrates the Position_tolerance.

NOTE 2 - The Position_tolerance definition is derived from paragraph 14.10 of ISO 1101.

The data associated with a Position_tolerance are the following:

- affected_plane;
- geometric_reference;
- value_qualifier.

4.2.140.1 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 4.3.148 for the application assertion.

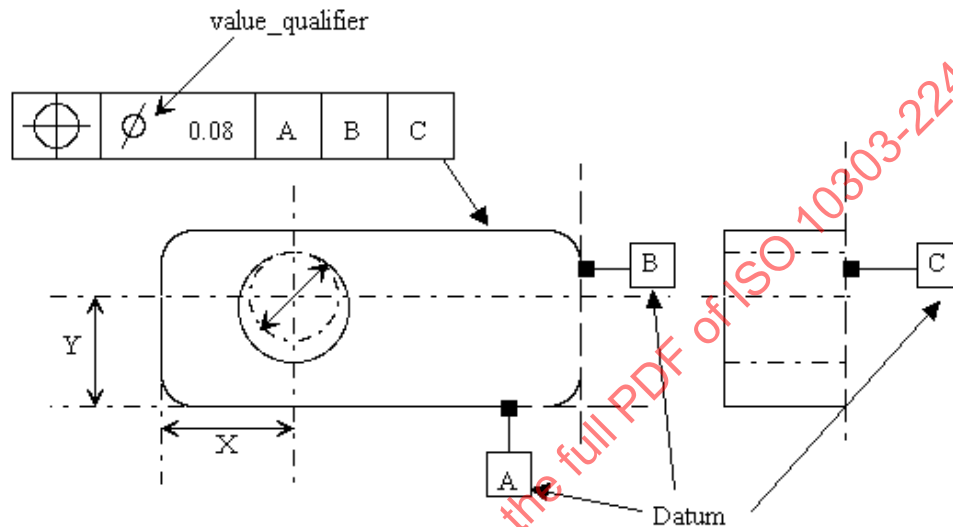


Figure 73 - Position_tolerance

4.2.140.2 geometric_reference

The `geometric_reference` specifies the datum to which the tolerance is related. See 4.3.147 for the application assertion.

4.2.140.3 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.

4.2.141 Process_property

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

EXAMPLE - Examples of `Process_property` may be paint or coat.

The data associated with a Process_property are the following:

- process_name;
- property_characteristic.

4.2.141.1 process_name

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

4.2.141.2 property_characteristic

The property_characteristic specifies the parameter to describe the Process_property. The property_characteristic need not be specified for a particular Process_property. There may be more than one property_characteristic for a Process_property. See 4.3.149 for the application assertion.

4.2.142 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 4.2.29) or an Open_profile (see 4.2.113). The data associated with a Profile are the following: placement. The placement specifies where to locate the Profile in reference to the Part orientation. See 4.3.150 for the application assertion.

4.2.143 Project_order

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

- ordered_resource;
- part_status;
- pedigree_creation_status;
- project_order_id;
- release_authorization;
- resource_acquisition_status;

- shop_work_status;
- technical_data_package_status.

4.2.143.1 ordered_resource

The ordered_resource specifies the resources required by the Project_order. There may be more than one ordered_resource for a Project_order. See 4.3.155 for the application assertion.

4.2.143.2 part_status

The part_status specifies Part that is tracked by the Project_order. There may be more than one part_status for a Project_order. See 4.3.153 for the application assertion.

4.2.143.3 pedigree_creation_status

The pedigree_creation_status specifies the Pedigree_creation_order that is tracked by the Project_order. The pedigree_creation_status need not be specified for a particular Project_order. See 4.3.154 for the application assertion.

4.2.143.4 project_order_id

The project_order_id specifies a unique identifier for the Project_order.

4.2.143.5 release_authorization

The release_authorization specifies the approval to manufacture a Part. See 4.3.151 for the application assertion.

4.2.143.6 resource_acquisition_status

The resource_acquisition_status specifies Resource_acquisition_order that is tracked by the Project_order. The resource_acquisition_status need not be specified for a particular Project_order. See 4.3.156 for the application assertion.

4.2.143.7 shop_work_status

The shop_work_status specifies the shop_work_order that is tracked by the Project_order. The shop_work_status need not be specified for a particular Project_order. See 4.3.157 for the application assertion.

4.2.143.8 technical_data_package_status

The `technical_data_package_status` specifies the `Digital_technical_data_package_work_order` that is tracked by the `Project_order`. The `technical_data_package_status` need not be specified for a particular `Project_order`. See 4.3.152 for the application assertion.

4.2.144 Projection

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

- `projection_end`;
- `projection_length`.

4.2.144.1 projection_end

The `projection_end` specifies the physical shape that is the extension of a feature. See 4.3.158 for the application assertion.

4.2.144.2 projection_length

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

4.2.145 Property

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

- `material_characteristic`;
- `property_name`;
- `part_property_characteristic`;
- `process_characteristic`;
- `property_characteristic`;
- `property_description`;
- `surface_characteristic`.

4.2.145.1 material_characteristic

The material_characteristic specifies the information that describe material for manufacturing the Part. The material_characteristic need not be specified for a particular Property. There may be more than one material_characteristic for a Property. See 4.3.159 for the application assertion.

4.2.145.2 property_name

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

4.2.145.3 part_property_characteristic

The part_property_characteristic specifies the information that describe properties of the Part. The part_property_characteristic need not be specified for a particular Property. There may be more than one part_property_characteristic for a Property. See 4.3.160 for the application assertion.

4.2.145.4 process_characteristic

The process_characteristic specifies information that describe processes for manufacturing the part. The process_characteristic need not be specified for a particular Property. There may be more than one process_characteristic for a Property. See 4.3.161 for the application assertion.

4.2.145.5 property_characteristic

The property_characteristic specifies information that describe properties of the Part. The property_characteristic need not be specified for a particular Property. There may be more than one property_characteristic for a Property. See 4.3.162 for the application assertion.

4.2.145.6 property_description

The property_description specifies the Specification that has additional information about the properties of the Part. The property_description need not be specified for a particular Property. There may be more than one property_description for a Property. See 4.3.163 for the application assertion.

4.2.145.7 surface_characteristic

The surface_characteristic specifies information that describe surface conditions of the Part. The surface_characteristic need not be specified for a particular Property. There may be more than one surface_characteristic for a Property. See 4.3.164 for the application assertion.

4.2.146 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 4.2.53) or a Numeric_parameter (see 4.2.111). The data associated with a Property_parameter are the following: parameter_name. The parameter_name specifies a word or group of words that identify a characteristic of interest for a Property_parameter.

4.2.147 Protrusion

A Protrusion is a type of Machining_feature (see 4.2.101) that is an arbitrary shape that extends out from a surrounding surface.

NOTE - Figure 74 illustrates a Protrusion.

The data associated with a Protrusion are the following: shape_volume. The shape_volume specifies the arbitrary shape that defines the shape of a Protrusion. See 4.3.165 for the application assertion.

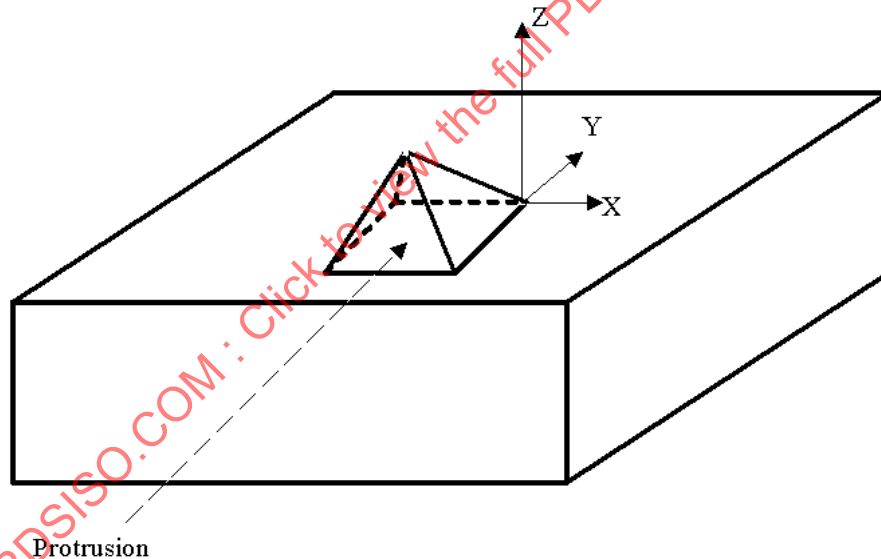


Figure 74 - Protrusion

4.2.148 Radial_dimension_tolerance

A Radial_dimension_tolerance is a type of Size_tolerance (see 4.2.172) that is the allowable variation for the radial distance from the center of a circular curve to a point on the curve.

NOTE - Figure 75 illustrates the Radial_dimension_tolerance.

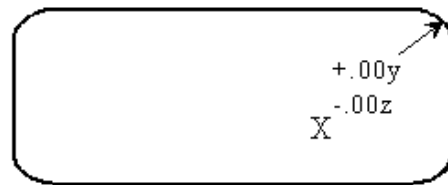


Figure 75 - Radial_dimension_tolerance

4.2.149 Radiused_slot_end_type

A Radiused_slot_end_type is a type of Slot_end_type (see 4.2.174) that is an end condition of a slot that shall be a cylindrical shape tangent to both of the adjacent Slot wall surfaces.

NOTE - Figure 76 illustrates the Radiused_slot_end_type.

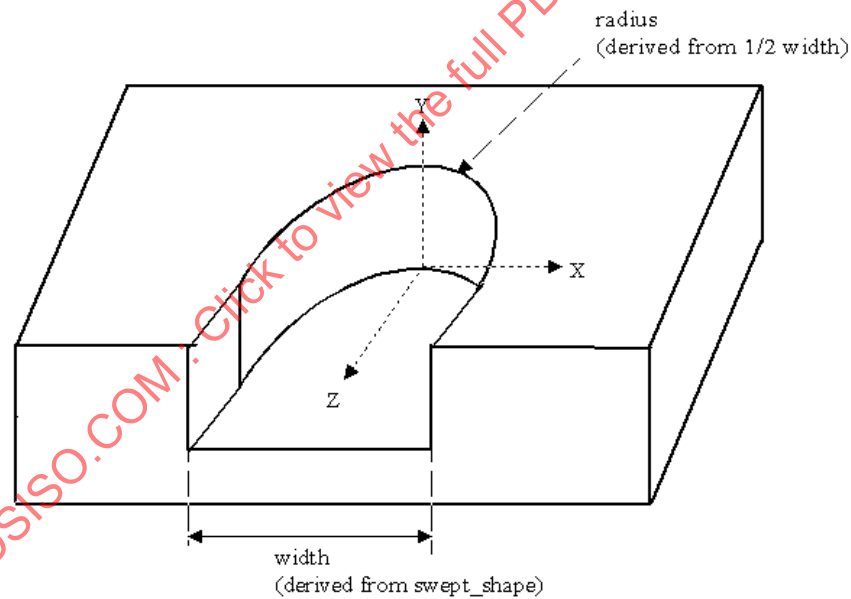


Figure 76 - Radiused_slot_end_type

4.2.150 Rectangular_closed_pocket

A Rectangular_closed_pocket is a type of Pocket (see 4.2.138) that is an enclosed volume with opposite sides that are equal in length.

NOTE - Figure 77 illustrates the Rectangular_closed_pocket.

The data associated with a Rectangular_closed_pocket are the following: 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. This placement of the profile 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 4.3.166 for the application assertion.

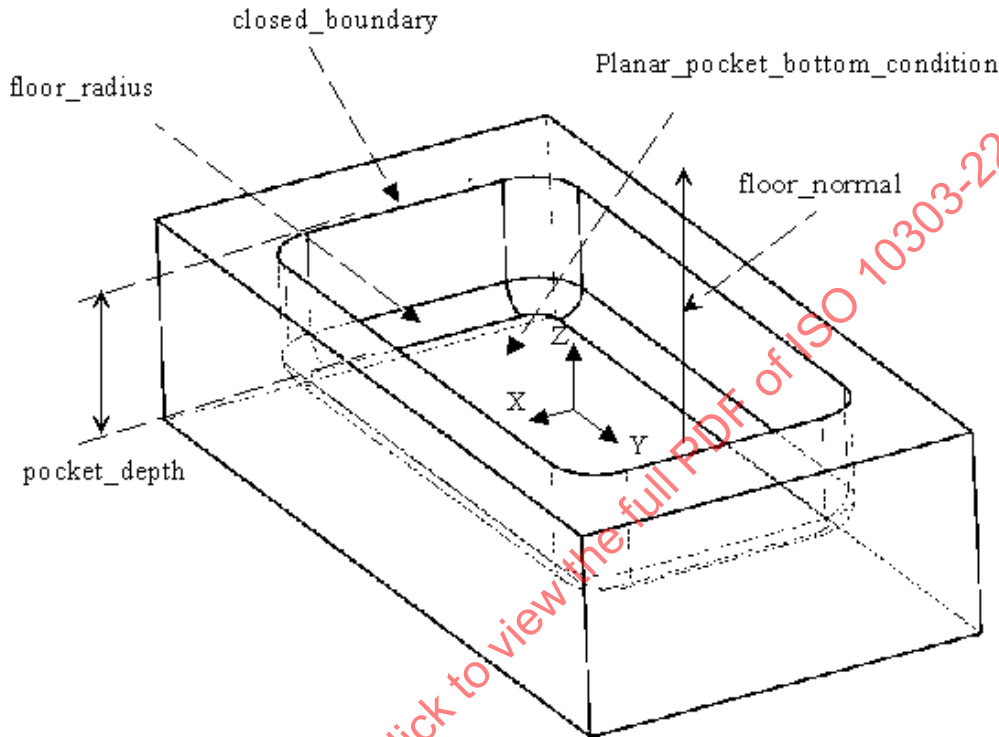


Figure 77 - Rectangular_closed_pocket

4.2.151 Rectangular_closed_profile

A Rectangular_closed_profile is a type of Closed_profile (see 4.2.29) that is an enclosed area bounded by four sides with opposite sides equal in length and corners at 90 degrees. The orientation is at the center of the rectangle, the X-axis is parallel to the length of the rectangle and the Y-axis is parallel to the width.

NOTE - Figure 78 illustrates the Rectangular_closed_profile.

The data associated with a Rectangular_closed_profile are the following:

- corner_radius;
- profile_length;
- profile_width.

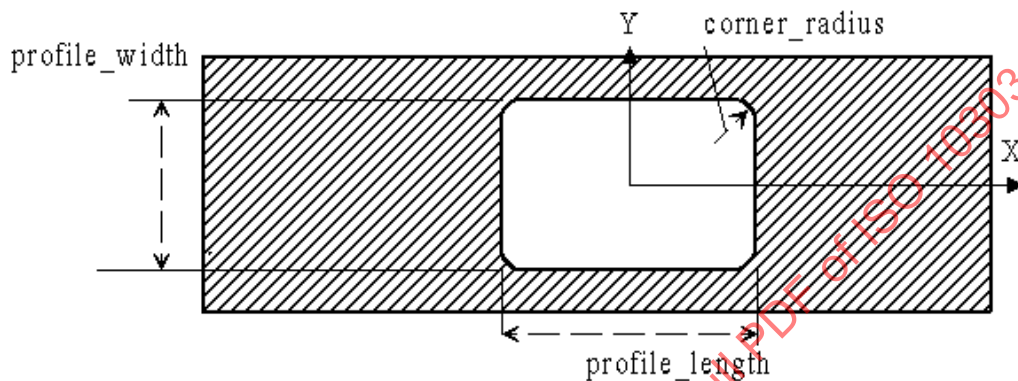


Figure 78 - Rectangular_closed_profile

4.2.151.1 corner_radius

The corner_radius specifies the size of the arc in all four corners of the rectangular profile. See 4.3.167 for the application assertion.

4.2.151.2 profile_length

The profile_length specifies the length of the side, along the X-axis, of the rectangular profile. See 4.3.167 for the application assertion.

4.2.151.3 profile_width

The profile_width specifies the length of the side, along the Y-axis, of the rectangular profile. See 4.3.167 for the application assertion.

4.2.152 Rectangular_offset_pattern

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

NOTE - Figure 79 illustrates the Rectangular_offset_pattern.

The data associated with a Rectangular_offset_pattern are the following:

- column_index;
- offset_direction;
- offset_distance;
- row_index.

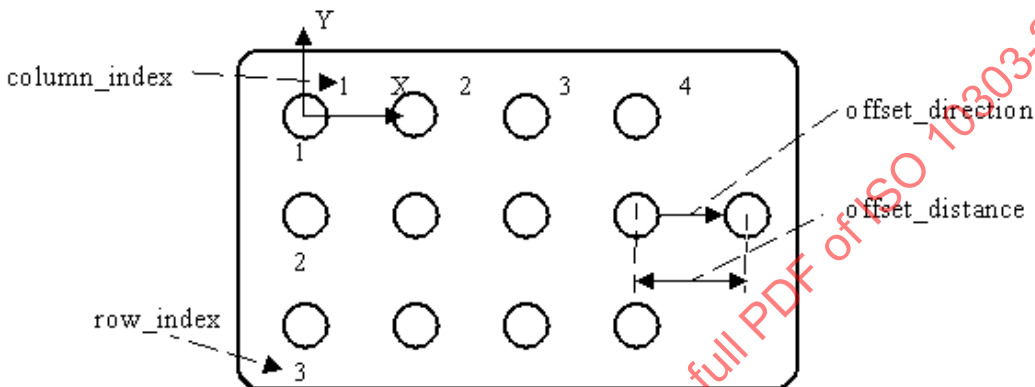


Figure 79 - Rectangular_offset_pattern

4.2.152.1 column_index

The column_index specifies the unique identification for a feature in a column of multiple features. See 4.3.169 for the application assertion.

4.2.152.2 offset_direction

The offset_direction specifies the direction to offset a base feature from its original position in the rectangular pattern. See 4.3.168 for the application assertion.

4.2.152.3 offset_distance

The offset_distance specifies the amount of offset from a feature location in a Rectangular_pattern for placing another feature. See 4.3.169 for the application assertion.

4.2.152.4 row_index

The row_index specifies the unique identification for a feature in a row of multiple features. See 4.3.169 for the application assertion.

4.2.153 Rectangular_omit_pattern

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

NOTE - Figure 80 illustrates Rectangular_omit_pattern.

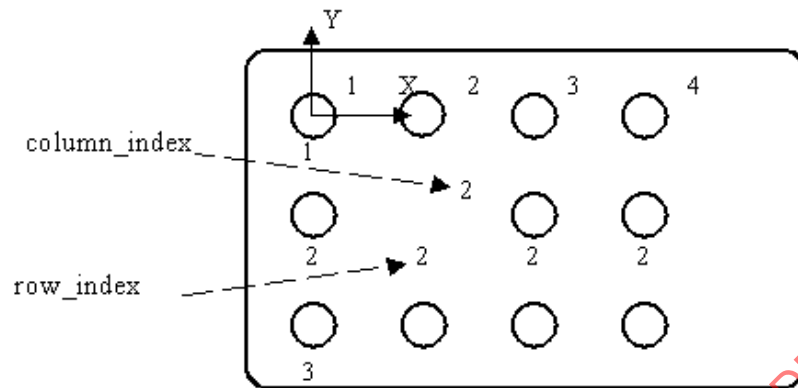


Figure 80 - Rectangular_omit_pattern

The data associated with a Rectangular_omit_pattern are the following:

- column_index;
- row_index.

4.2.153.1 column_index

The column_index specifies the unique identification for a feature in a column of multiple features. See 4.3.170 for the application assertion.

4.2.153.2 row_index

The row_index specifies the unique identification for a feature in a row of multiple features. See 4.3.170 for the application assertion.

4.2.154 Rectangular_open_pocket

A Rectangular_open_pocket is a type of Pocket (see 4.2.138) 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 - Figure 81 illustrates the Rectangular_open_pocket.

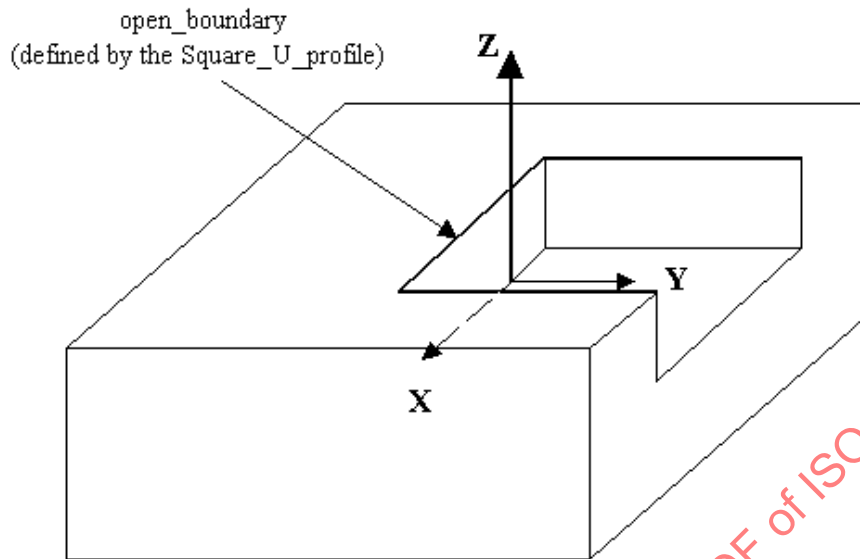


Figure 81 - Rectangular_open_pocket

The data associated with a Rectangular_open_pocket are the following: 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. This placement of the profile 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 4.3.171 for the application assertion.

4.2.155 Rectangular_pattern

A Rectangular_pattern is a type of Replicate_feature (see 4.2.157) that is a shape component arranged in a pattern of rows and columns.

NOTE - Figure 82 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.

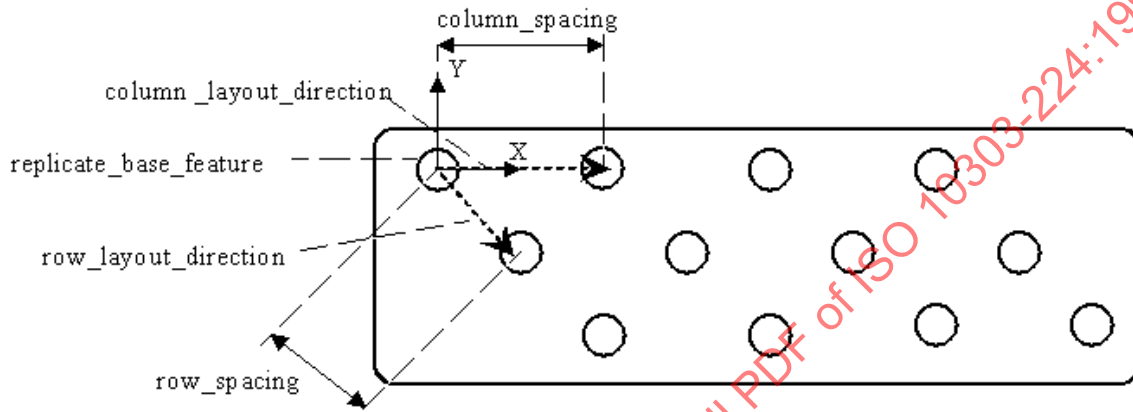


Figure 82 - Rectangular_pattern

4.2.155.1 column_layout_direction

The column_layout_direction specifies the linear direction for defining the columns of the pattern. See 4.3.172 for the application assertion.

4.2.155.2 column_spacing

The column_spacing specifies the amount of space between features in a Rectangular_pattern column. See 4.3.173 for the application assertion.

4.2.155.3 columns

The columns specifies the number of columns for placing features in the Rectangular_pattern. See 4.3.173 for the application assertion.

4.2.155.4 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 4.3.175 for the application assertion.

4.2.155.5 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 4.3.174 for the application assertion.

4.2.155.6 row_layout_direction

The row_layout_direction specifies the linear direction for defining the rows of the pattern. See 4.3.172 for the application assertion.

4.2.155.7 row_spacing

The row_spacing specifies the amount of space between features in a Rectangular_pattern row. See 4.3.173 for the application assertion.

4.2.155.8 rows

The rows specifies the number of rows for placing features in the Rectangular_pattern. See 4.3.173 for the application assertion.

4.2.156 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 4.2.101) or a Replicate_feature (see 4.2.157). The data associated with a Replicate_base are the following: 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 4.2.101) or a Replicate_feature (see 4.2.157). See 4.3.176 and 4.3.177 for the application assertion.

4.2.157 Replicate_feature

A Replicate_feature is a type of Machining_feature (see 4.2.101) that is a base 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 4.2.26), General_pattern (see 4.2.80), or a Rectangular_pattern (see 4.2.155). The data associated with a Replicate_feature are the following:

- placement;
- replicate_base_feature.

4.2.157.1 placement

The placement specifies the position and orientation of a Replicate_feature relative to the base shape for a part. See 4.3.178 for the application assertion.

4.2.157.2 replicate_base_feature

The replicate_base_feature specifies the feature that will be replicated by the Replicate_feature. See 4.3.179 for the application assertion.

4.2.158 Requisition

A Requisition is a document that contains an order for equipment or materials to perform or support the manufacturing process. Each Requisition may be one of the following: Cutting_tool_requisition (see 4.2.43), Dedicated_fixture_requisition (see 4.2.50), Machine_requisition (see 4.2.100), Material_requisition (see 4.2.107) Modular_fixture_requisition (see 4.2.108), or an Indirect_stock_requisition (see 4.2.92). The data associated with a Requisition are the following:

- quantity_ordered;
- required_delivery_date;
- requisition_date;
- requisition_description;
- requisition_number.

4.2.158.1 quantity_ordered

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

4.2.158.2 required_delivery_date

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

4.2.158.3 requisition_date

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

4.2.158.4 requisition_description

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

4.2.158.5 requisition_number

The requisition_number specifies a unique identification of the Requisition.

4.2.159 Resource_acquisition_order

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

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

The data associated with a Resource_acquisition_order are the following: order_id. The order_id specifies a unique identifier for the Resource_acquisition_order.

4.2.160 Revolved_feature

A Revolved_feature is a type of Machining_feature (see 4.2.101) 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 4.2.84), Groove (see 4.2.88), Revolved_flat (see 4.2.161), or a Revolved_round (see 4.2.162).

The data associated with a Revolved_feature are the following:

- material_side;
- radius.

4.2.160.1 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 4.3.181 for the application assertion.

4.2.160.2 radius

The radius specifies the distance from the axis of rotation to define placement of the profile that will be swept about the axis. See 4.3.180 for the application assertion.

4.2.161 Revolved_flat

A Revolved_flat is a type of Revolved_feature (see 4.2.160) that is the sweeping of a straight line about an axis.

NOTE - Figure 83 illustrates the Revolved_flat.

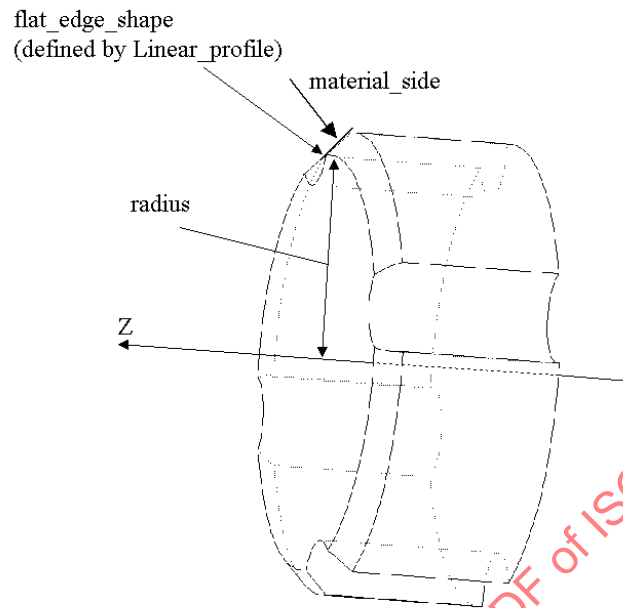
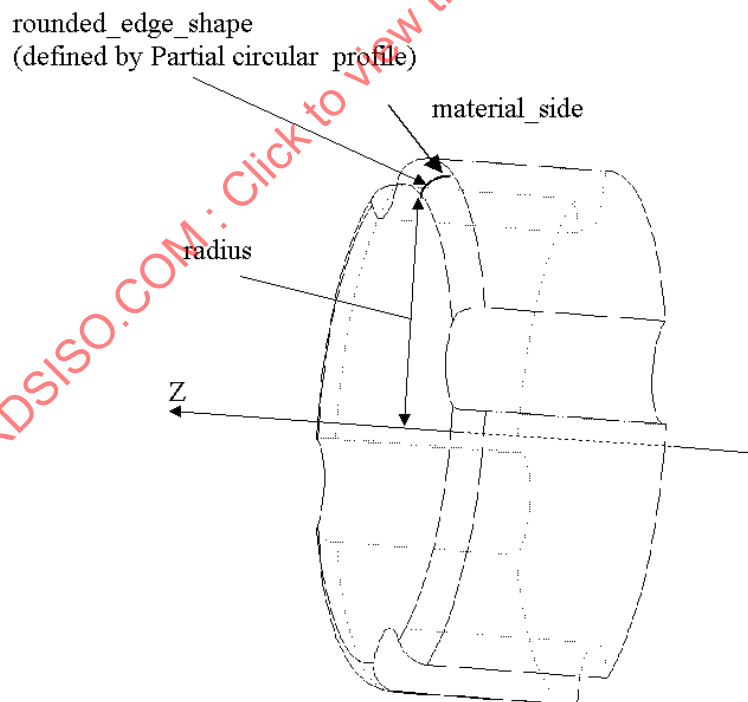
The data associated with a Revolved_flat are the following: flat_edge_shape. The flat_edge_shape specifies the line with direction and magnitude that when revolved about an axis defines the area on a part for volume removal. The placement of the profile shall be along the X-axis of the Revolved_flat 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 4.3.182 for the application assertion.

4.2.162 Revolved_round

A Revolved_round is a type of Revolved_feature (see 4.2.160) that is the sweeping of an arc about an axis.

NOTE - Figure 84 illustrates the Revolved_round.

The data associated with a Revolved_round are the following: 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 4.3.183 for the application assertion.

**Figure 83 - Revolved_flat****Figure 84 - Revolved_round**

4.2.163 Round_hole

A Round_hole is a type of Hole (see 4.2.90) that is a removal of a volume of cylindrical shape from a part. A Round_hole need not be tapered. The orientation is at a point in the bottom of the hole. The Z-axis is along the centerline with the direction out of the hole.

NOTE - Figure 85 illustrates the Round_hole with and without a taper.

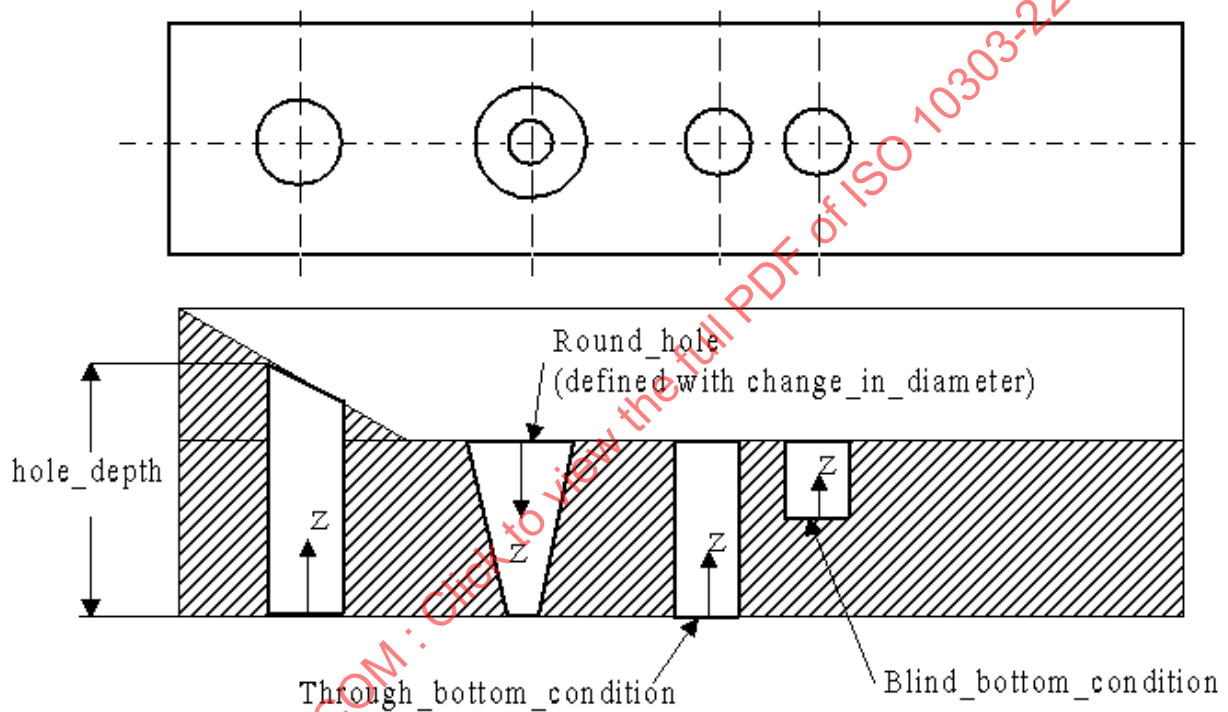


Figure 85 - Round_hole

The data associated with a Round_hole are the following:

- bottom_condition;
- change_in_diameter;
- diameter;
- hole_depth.

4.2.163.1 bottom_condition

The `bottom_condition` specifies the shape of the top of a `Round_hole` feature. See 4.3.184 and 4.3.185 for the application assertion.

4.2.163.2 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 4.3.188 and 4.3.189 for the application assertion.

4.2.163.3 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 4.3.186 for the application assertion.

4.2.163.4 hole_depth

The `hole_depth` is some amount of distance from the bottom of a `Round_hole` to a point that is outside of the `Round_hole` feature. `Hole_depth` places a limitation on the `Round_hole` depth definition so not to interfere with other features that might be nearby. The placement and orientation of the `Linear_path` shall be the same as the `Round_hole` feature. See 4.3.187 for the application assertion.

4.2.164 Rounded_end

A `Rounded_end` is a type of `Machining_feature` (see 4.2.101) that is a partially circular shape passed along a linear path.

NOTE - Figure 86 illustrates the `Rounded_end`.

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

- `course_of_travel`;
- `partial_circular_boundary`.

4.2.164.1 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 4.3.190 for the application assertion.

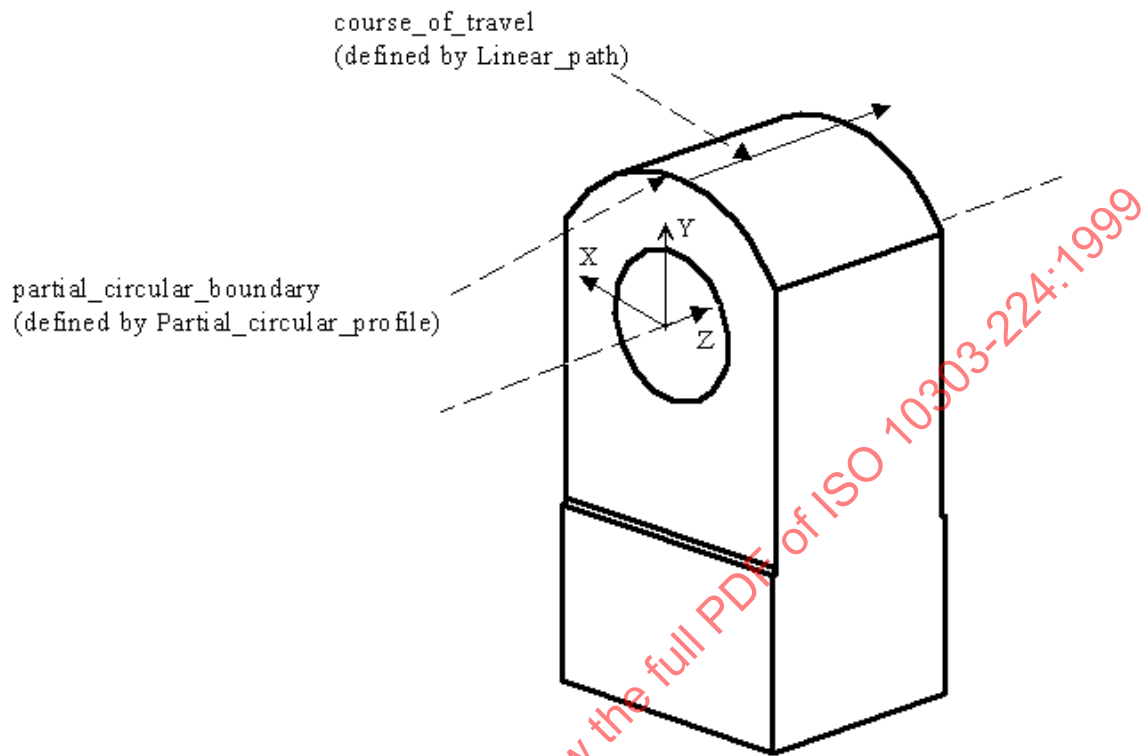


Figure 86 - Rounded_end

4.2.164.2 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 4.3.191 for the application assertion.

4.2.165 Rounded_U_profile

A `Rounded_U_profile` is a type of `Open_profile` (see 4.2.113) 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. The data associated with a `Rounded_U_profile` are the following: width. The width specifies the distance across a `Round_U_profile`. See 4.3.192 for the application assertion.

NOTE - Figure 87 illustrates the `Rounded_U_profile`.

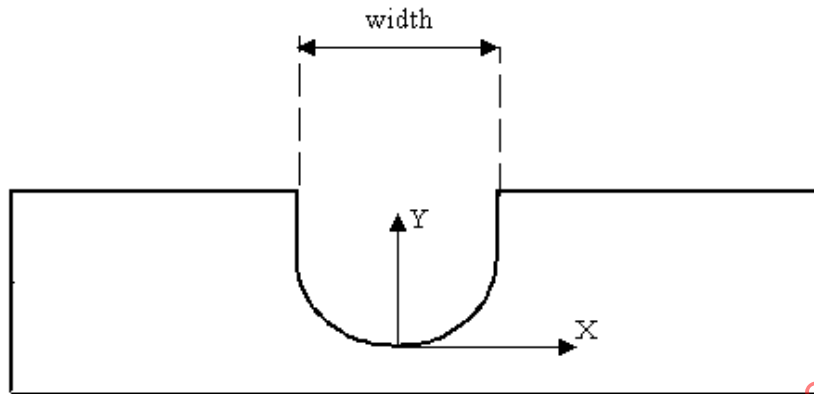


Figure 87 - Rounded_U_profile

4.2.166 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 4.2.20) or a `Second_offset` (see 4.2.167). The data associated with a `Second_chamfer_offset` are the following: `second_face`. The `second_face` is one of two faces the Chamfer feature will transition between. See 4.3.193 for the application assertion.

4.2.167 Second_offset

A `Second_offset` is a type of `Second_chamfer_offset` (see 4.2.166) that is the amount of length offset from a face for creating a Chamfer feature.

NOTE - Figure 12 illustrates a `Second_offset` for a Chamfer feature.

The data associated with a `Second_offset` are the following: `offset_amount`. The `offset_amount` specifies the offset value from the edge of a face to the Chamfer face. See 4.3.194 for the application assertion.

4.2.168 Shape

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

- base_shape_definition;
- B-rep_form;
- element.

4.2.168.1 base_shape_definition

The base_shape_definition specifies either the implicit or the explicit definition of the Part. See 4.3.195 for the application assertion.

4.2.168.2 B-rep_form

The B-rep_form specifies the boundary representation shape of the Part. The B-rep_form need not be specified for a particular Shape. There may be more than one B-rep_form for a Shape. See 4.3.196 for the application assertion.

4.2.168.3 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 4.3.197 for the application assertion.

4.2.169 Shape_aspect

A Shape_aspect is a region of interest with respect to the shape of a part. A Shape_aspect may be an element of the shape of the part or a reference shape that does not lie on the shape of the part, but is used to specify a characteristic of the shape of the part. The data associated with a Shape_aspect are the following:

- B-rep_form;
- B-rep_shape;
- element.

4.2.169.1 B-rep_form

The B-rep_form specifies aspects of the boundary representation of the Part. There may be more than one B-rep_form for a Shape_aspect. See 4.3.199 for the application assertion.

4.2.169.2 B-rep_shape

The B-rep_shape specifies the boundary representation of the shape of the Part. The B-rep_shape need not be specified for a particular Shape_aspect. There may be more than one B-rep_shape for a Shape_aspect. See 4.3.198 for the application assertion.

4.2.169.3 element

The element specifies components of the shape of the Part. The element need not be specified for a particular Shape_aspect. See 4.3.200 for the application assertion.

4.2.170 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 4.2.61), Face_shape_element (see 4.2.68), Location_element (see 4.2.98), Manufacturing_feature (see 4.2.102), or a Path_element (see 4.2.128).

4.2.171 Shop_work_order

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

4.2.172 Size_tolerance

A Size_tolerance is a type of Dimensional_tolerance (see 4.2.60) that is the size dimension tolerance characteristic for a geometric element. Each Size_tolerance is either a Angular_size_dimension_tolerance (see 4.2.4), Curved_dimension_tolerance (see 4.2.41), Diameter_dimension_tolerance (see 4.2.56), or a Radial_dimension_tolerance (see 4.2.148). The data associated with a Size_tolerance are the following: applied_shape. The applied_shape specifies the physical shape of the Part that is being tolerated. See 4.3.201 for the application assertion.

4.2.173 Slot

A Slot is a type of Machining_feature (see 4.2.101) 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.

NOTE - Figure 88 illustrates types of Slots.

The data associated with a Slot are the following:

- `course_of_travel`;
- `end_conditions`;
- `slot_depth`;
- `sweep_shape`.

4.2.173.1 `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 4.3.204 for the application assertion.

4.2.173.2 `end_conditions`

The `end_conditions` specifies the type of implicit shape at the ends of the Slot. See 4.3.205 for the application assertion.

4.2.173.3 `slot_depth`

The `slot_depth` specifies the implicit definition for the depth of the Slot in the Part. See 4.3.202 for the application assertion.

4.2.173.4 `sweep_shape`

The `sweep_shape` specifies the implicit 2D profile definition that, when combine with a Path, creates the shape of the Slot. See 4.3.203 for the application assertion.

4.2.174 `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 4.2.72), `Open_slot_end_type` (see 4.2.114), `Radiused_slot_end_type` (see 4.2.149), or a `Woodruff_slot_end_type` (see 4.2.204). The data associated with a `Slot_end_type` are the following: `first_or_second`. The `first_or_second` specifies a value of 'FIRST' if the `Slot_end_type` is closest to the positioning point of a Slot or 'SECOND' if it is the farthest away.

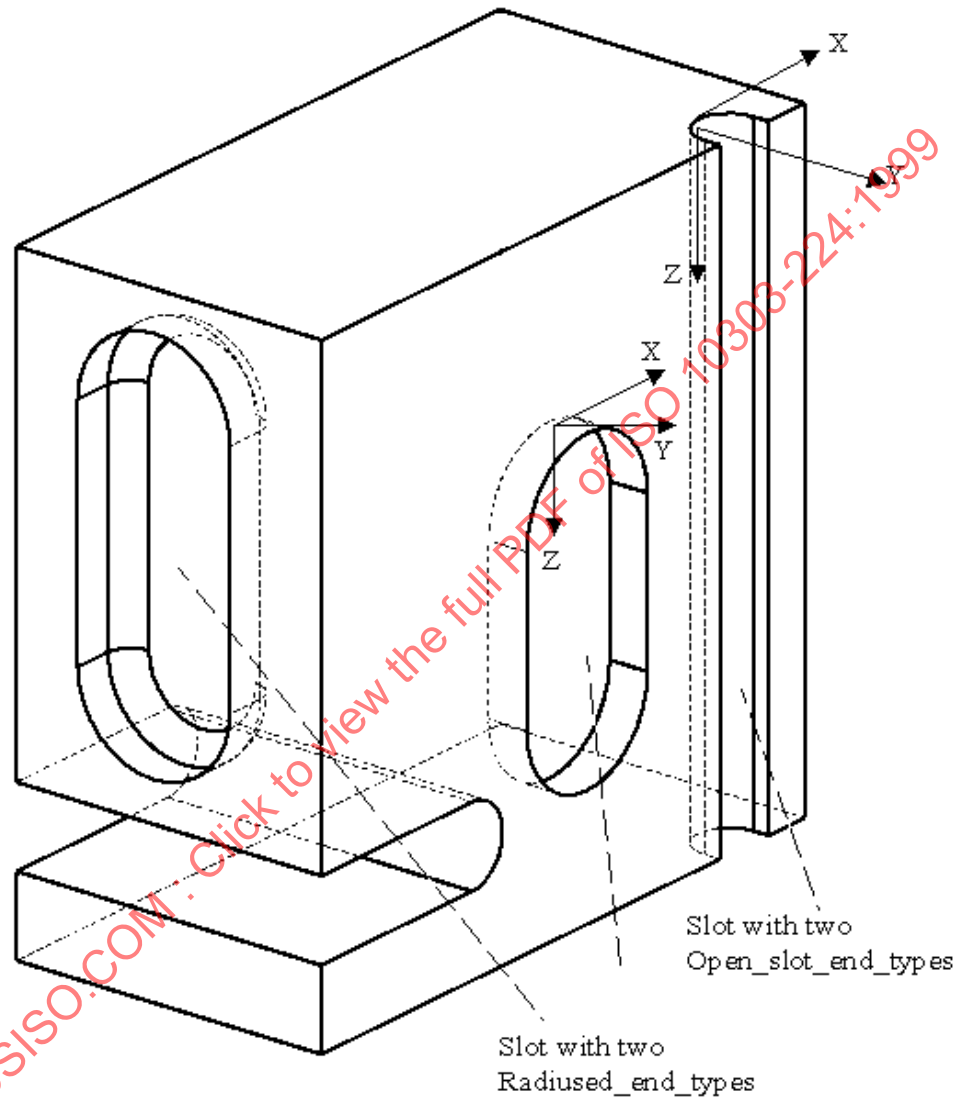


Figure 88 - Slot

4.2.175 Specification

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

- constraint;
- specification_class;
- specification_description;
- specification_id.

4.2.175.1 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 4.3.206 for the application assertion.

4.2.175.2 specification_class

The specification_class specifies a section within a Specification that is divided into classes.

4.2.175.3 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.

4.2.175.4 specification_id

The specification_id specifies a unique identifier of the document.

4.2.176 Specification_usage_constraint

A Specification_usage_constraint is a restriction on the application of information defined within a Specification. The data associated with a Specification_usage_constraint are the following:

- class_id;
- element.

4.2.176.1 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.

4.2.176.2 element

The element specifies the particular piece or area of information that is being restricted within the Specification.

4.2.177 Spherical_cap

A Spherical_cap is a type of Machining_feature (see 4.2.101) that is circular about an axis of rotation. A Spherical_cap consists of all points a given distance from a point constituting its center. The Z-axis shall be in the direction away from the material.

NOTE - Figure 89 illustrates the Spherical_cap.

The data associated with a Spherical_cap are the following:

- internal_angle;
- radius.

4.2.177.1 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. See 4.3.207 for the application assertion. The X-axis defines the start of the spherical_cap and the internal_angle is measured from this axis.

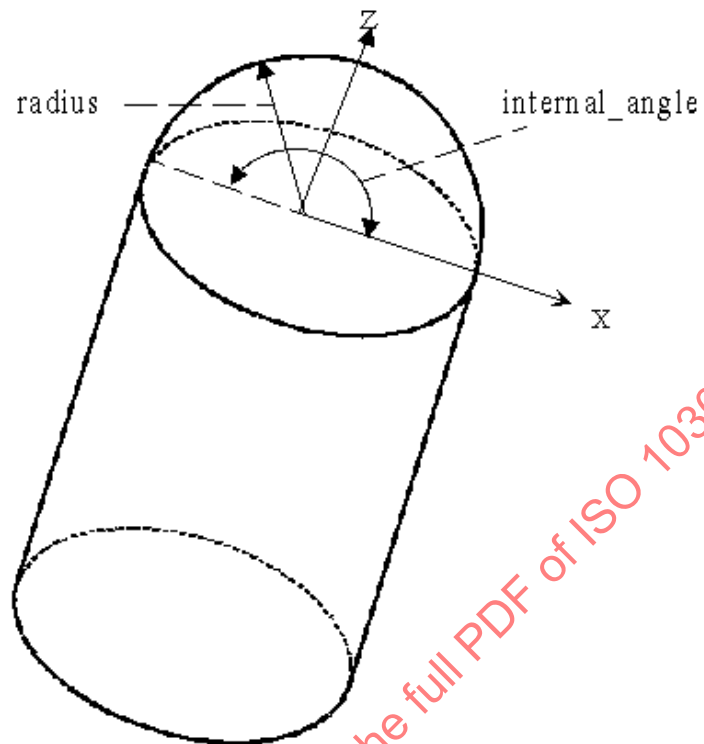
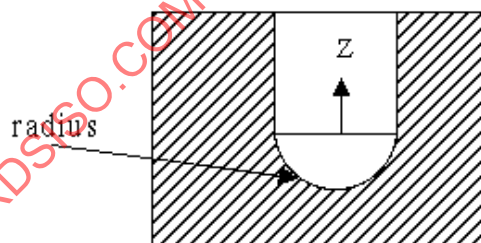
4.2.177.2 radius

The radius specifies the constant distance from a point for defining a sphere. See 4.3.207 for the application assertion.

4.2.178 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 - Figure 90 illustrates the Spherical_hole_bottom.

**Figure 89 - Spherical_cap****Figure 90 - Spherical_hole_bottom**

The data associated with `Spherical_hole_bottom` are the following: radius. The radius specifies the radius at the bottom of the `Round_hole`. See 4.3.208 for the application assertion.

4.2.179 Square_U_profile

A `Square_U_profile` is a type of `Open_profile` (see 4.2.113) 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 obtuse or acute angle that is less than or larger than a right angle. The two lines may also be at right angle to the base line. The corners of the `Square_U_profile` need not be blended by a radius.

NOTE - Figure 91 illustrates the `Square_U_profile`.

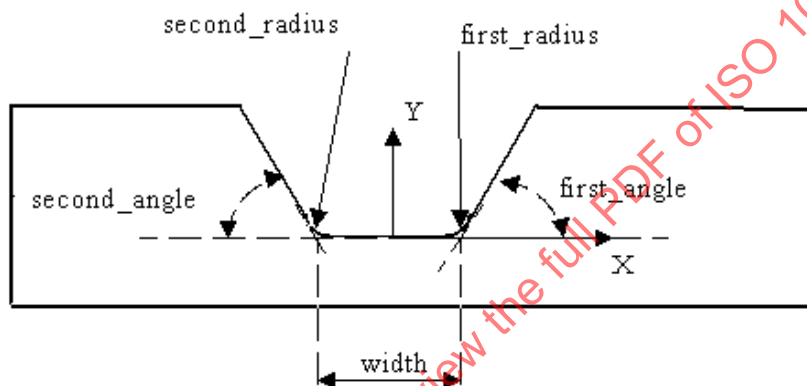


Figure 91 - Square_U_profile

The data associated with `Square_U_profile` are the following:

- `first_angle`;
- `first_radius`;
- `second_angle`;
- `second_radius`;
- `width`.

4.2.179.1 first_angle

The `first_angle` specifies the size of an angle between one side of the profile and the base. See 4.3.209 for the application assertion.

4.2.179.2 first_radius

The first_radius specifies the radius shape blend between one side of the profile and the base. See 4.3.209 for the application assertion.

4.2.179.3 second_angle

The second_angle specifies the size of an angle between the second side of the profile and the base. See 4.3.209 for the application assertion.

4.2.179.4 second_radius

The second_radius specifies the radius shape blend between the second side of the profile and the base. See 4.3.209 for the application assertion.

4.2.179.5 width

The width specifies the size of the base line for a Square_U_profile. See 4.3.209 for the application assertion.

4.2.180 Step

A Step is a type of Machining_feature (see 4.2.101) 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 a straight angle. The intersection of the two lines need not be blended with a radius.

NOTE - Figure 92 illustrates the Step.

The data associated with a Step are the following:

- course_of_travel;
- removal_boundary.

4.2.180.1 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 4.3.211 for the application assertion.

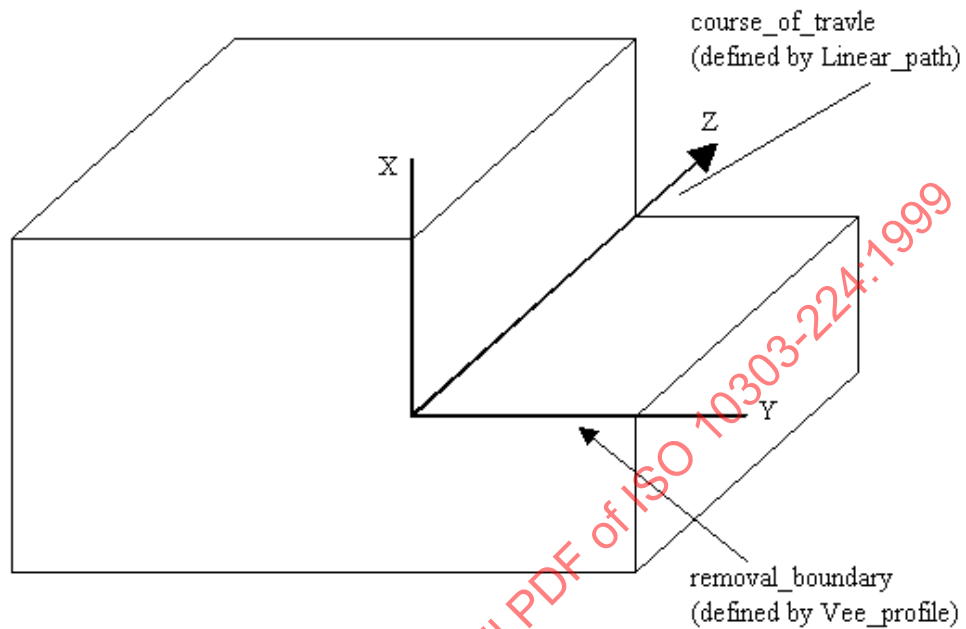


Figure 92 - Step

4.2.180.2 removal_boundary

The removal_boundary specifies the a Vee_profile (see 4.2.203) 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 4.3.210 for the application assertion.

4.2.181 Straight_knurl

A Straight_knurl is a type of Turned_knurl (see 4.2.202) that is the knurl scoring that is parallel to the axis of the scored surface.

NOTE - Figure 93 illustrates the Straight_knurl.

4.2.182 Straightness_tolerance

A Straightness_tolerance is a type of Geometric_tolerance (see 4.2.86) that is the amount of deviation a surface shall have from being straight. No element of the cylindrical surface deviates more than a specified tolerance amount from a straight line.

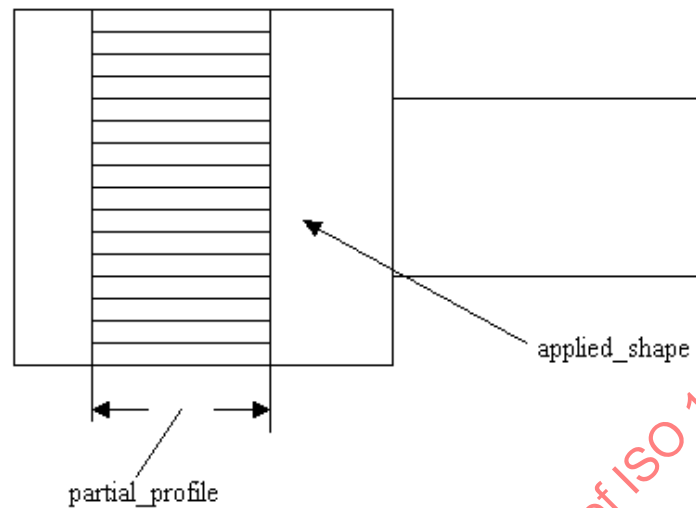


Figure 93 - Straight_knurl

NOTE 1 - Figure 94 illustrates the Straightness_tolerance.

NOTE 2 - The Straightness_tolerance definition is derived from paragraph 14.1 of ISO 1101.

The data associated with a Straightness_tolerance are the following:

- affected_plane;
- segment_size.

4.2.182.1 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 4.3.212 for the application assertion.

4.2.182.2 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.

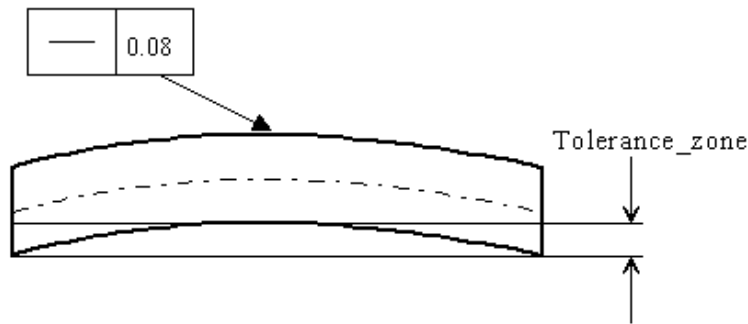


Figure 94 - Straightness_tolerance

4.2.183 Surface_profile_tolerance

A Surface_profile_tolerance is a type of Geometric_tolerance (see 4.2.86) that is a uniform boundary or zone along the true profile within which all elements of the surface shall lie.

NOTE 1 - Figure 95 illustrates Surface_profile_tolerance.

NOTE 2 - The Surface_profile_tolerance definition is derived from paragraph 14.6 of ISO 1101.

The data associated with a Surface_profile_tolerance are the following: geometric_reference. The geometric_reference specifies the datum to which the tolerance is related. See 4.3.213 for the application assertion.

4.2.184 Surface_property

A Surface_property specifies characteristics of a surface that are elements of the shape of a part. The data associated with a Surface_property are the following:

- property_characteristics;
- surface_finish.

4.2.184.1 property_characteristics

The property_characteristics specifies the parameter to describe the Surface_property. See 4.3.214 for the application assertion.

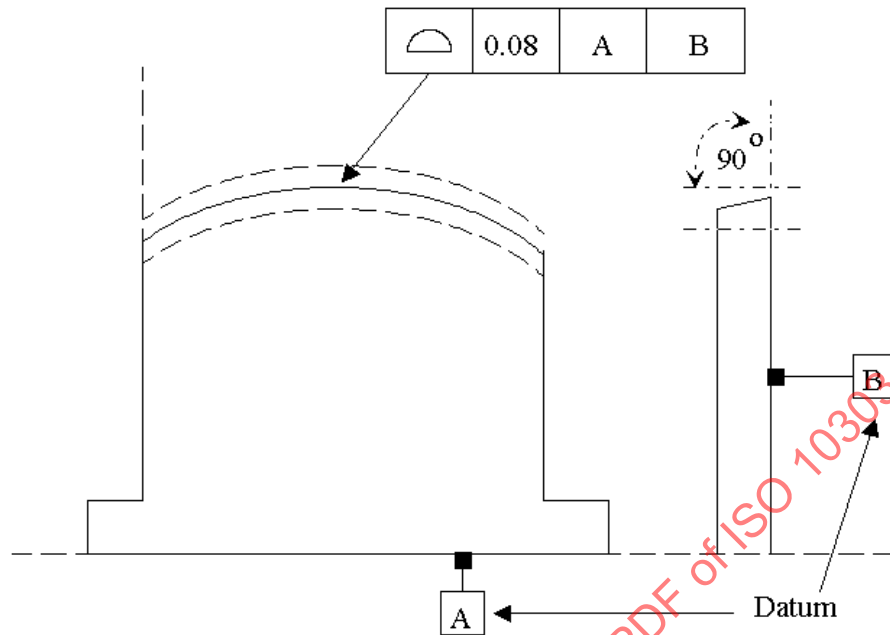


Figure 95 - Surface_profile_tolerance

4.2.184.2 surface_finish

The `surface_finish` specifies a boolean value that indicates a type of `Surface_property` is a surface finish. A value of `true` specifies the `Surface_property` is a surface finish property, a value of `false` specifies the `Surface_property` is for other surface properties.

4.2.185 Symmetry_tolerance

A `Symmetry_tolerance` is a type of `Geometric_tolerance` (see 4.2.86) that is the amount of deviation a surface shall have from being symmetrical to a datum. A plane of the part shall lie in a tolerance zone defined by two parallel planes, symmetrically located with respect to a datum plane of the center.

NOTE 1 - Figure 96 illustrates the `Symmetry_tolerance`.

NOTE 2 - The `Symmetry_tolerance` definition is derived from paragraph 14.12 of ISO 1101.

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

- `affected_plane`;
- `geometric_reference`.

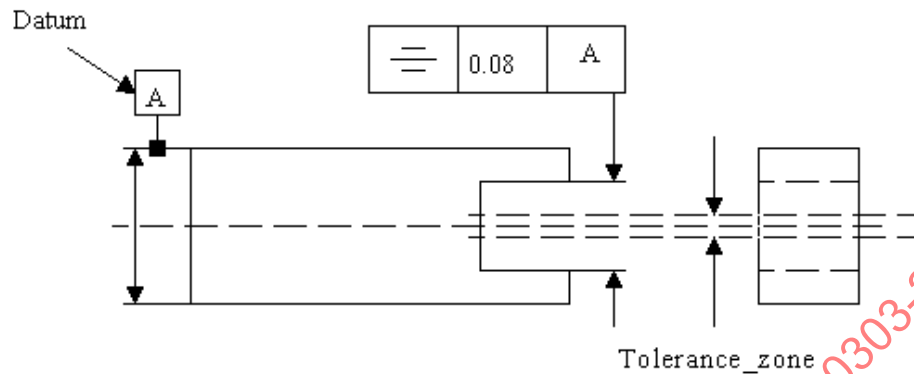


Figure 96 - Symmetry_tolerance

4.2.185.1 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 4.3.216 for the application assertion.

4.2.185.2 geometric_reference

The `geometric_reference` specifies the datum to which the tolerance is related. See 4.3.215 for the application assertion.

4.2.186 Target_area

A `Target_area` is a type of `Datum_target` (see 4.2.48) that is an enclosed area bounded by an arbitrary shape required to define a `Datum_target`. The shape of the `Target_area` is described explicitly by a set of curves. The data associated with `Target_area` are the following: `area_shape`. The `area_shape` specifies the physical form of the `Target_area` shape. See 4.3.217 for the application assertion.

4.2.187 Target_circle

A `Target_circle` is a type of `Placed_target` (see 4.2.133) that is an enclosed area bounded by a circle required to define a `Datum_target`. The origin of the `Datum_target` is the center of the circle, and the orientation is the x-y plane. The data associated with a `Target_circle` are the following: `target_diameter`. The `target_diameter` specifies the diameter value of the `Target_circle`.

4.2.188 Target_line

A Target_line is a type of Placed_target (see 4.2.133) that is a straight curve. The origin shall be the first end point of the Target_line, the second end point shall be located on the Z-axis at a specified length. The data associated with a Target_line are the following: target_length. The target_length specifies the length value of the Target_line.

4.2.189 Target_point

A Target_point is a type of Placed_target (see 4.2.133) that is a single point. The origin shall be at the Target_point.

4.2.190 Target_rectangle

A Target_rectangle is a type of Placed_target (see 4.2.133) that is an area bounded by four sides with opposite sides equal in length. The center of the rectangle is at the origin. The orientation of the rectangle is with the length along the X-axis and the width along the Y-axis. The data associated with Target_rectangle are the following:

- target_length;
- target_width.

4.2.190.1 target_length

The target_length specifies the length value of the Target_rectangle.

4.2.190.2 target_width

The target_width specifies the width value of the Target_rectangle.

4.2.191 Tee_profile

A Tee_profile is a type of Open_profile (see 4.2.113) the cross-section of which has the shape of the twentieth letter of the English alphabet in capital form. The first line has a defined length. The second line begins at the midpoint of the first line and is perpendicular to it. The second line extends infinitely. The corners of the Tee_profile need not be blended by a radius. The profile is positioned with the opening in the direction of the Y-axis. The orientation is at a point on the profile the farthest distance from the opening measured along the Y-axis. The X-axis is tangent to the bottom of the profile.

NOTE - Figure 97 illustrates the Tee_profile.

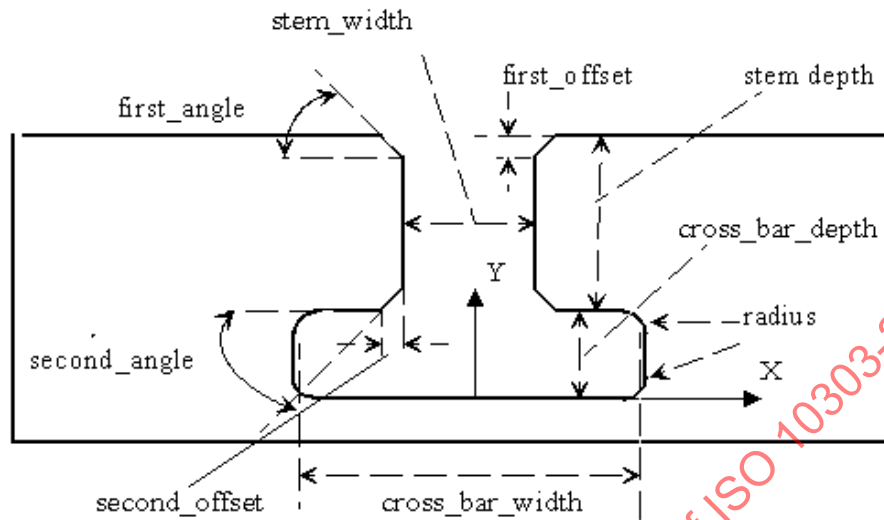


Figure 97 - Tee_profile

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.

4.2.191.1 cross_bar_depth

The cross_bar_depth specifies the depth dimension of the Tee cross bar size. See 4.3.218 for the application assertion.

4.2.191.2 cross_bar_width

The cross_bar_width specifies the width dimension of the Tee cross bar size. See 4.3.218 for the application assertion.

4.2.191.3 depth

The depth specifies the depth dimension of the Tee stem. See 4.3.218 for the application assertion.

4.2.191.4 first_angle

The first_angle specifies the angular measurement for creating a chamfer on the open end of a Tee_profile. See 4.3.218 for the application assertion.

4.2.191.5 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 4.3.218 for the application assertion.

4.2.191.6 radius

The radius specifies the arc size for blending the sides of a Tee_profile cross bar. See 4.3.218 for the application assertion.

4.2.191.7 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 4.3.218 for the application assertion.

4.2.191.8 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 4.3.218 for the application assertion.

4.2.191.9 width

The width specifies the width dimension of the Tee stem. See 4.3.218 for the application assertion.

4.2.192 Thread

A Thread is a type of Machining_feature (see 4.2.101) 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 4.2.18) or a Defined_thread (see 4.2.52).

NOTE 1 - Figure 98 illustrates the Thread and Figure 34 illustrates Thread attributes.

NOTE 2 - Threads may be used to screw parts together.

NOTE 3 - An outside thread might be on a Outer_round, an inside thread might be in a Round_hole.

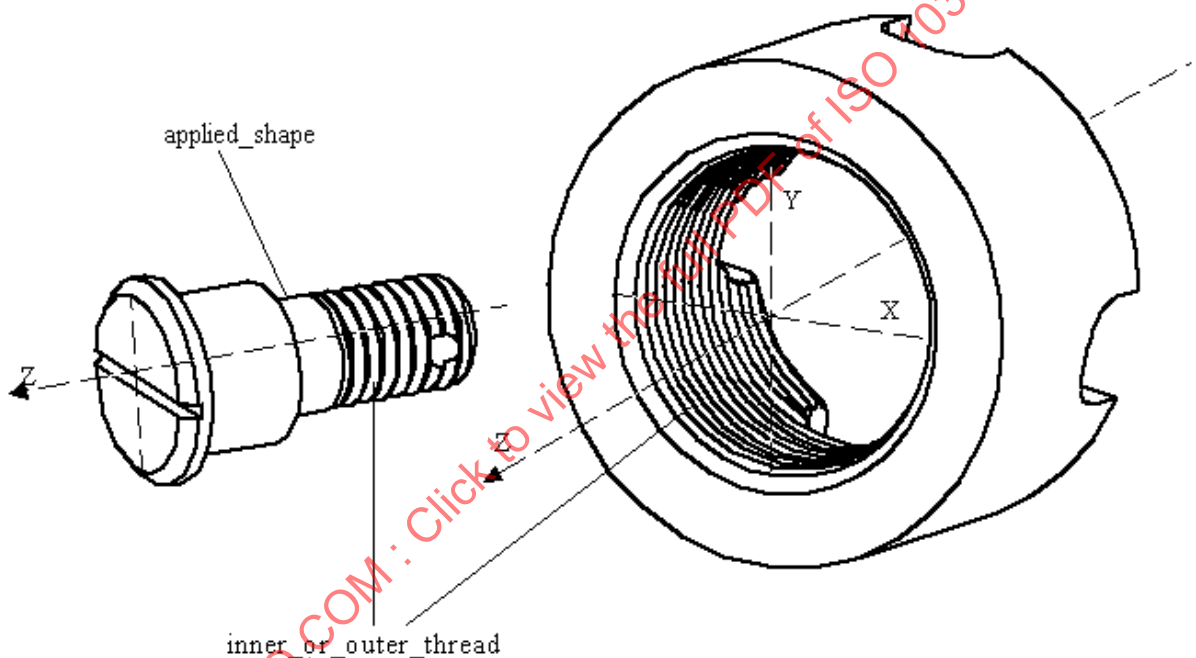


Figure 98 - Thread

The data associated with a Thread are the following:

- applied_shape;
- english_or_metric;
- fit_class;
- form;

- inner_or_outer_thread;
- major_diameter;
- number_of_threads;
- partial_profile;
- qualifier;
- thread_hand.

4.2.192.1 applied_shape

The `applied_shape` specifies the physical shape of the Part that will define where the Thread feature will be applied. See 4.3.222 for the application assertion.

4.2.192.2 english_or_metric

An `english_metric` specifies whether or not the thread type is either English or metric.

4.2.192.3 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 4.3.219 for the application assertion.

EXAMPLE - Examples of ANSI fit class are: 1A, 2A, and 3A which apply to external threads only, and 1B, 2B, and 3B which apply to internal threads only.

4.2.192.4 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 4.3.219 for the application assertion.

EXAMPLE - Examples of form are: metric, square, unified, sharp V, buttress, standard worm, and knuckle.

4.2.192.5 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.

4.2.192.6 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 4.3.220 for the application assertion.

4.2.192.7 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 4.3.220 for the application assertion.

4.2.192.8 partial_profile

The `partial_profile` specifies the limitations to be applied on the Thread feature. The `partial_profile` need not be specified for a particular Thread. See 4.3.221 for the application assertion.

4.2.192.9 qualifier

The `qualifier` specifies additional text information that describes a Thread. See 4.3.219 for the application assertion.

4.2.192.10 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 ? for the application assertion.

4.2.193 Through_bottom_condition

A `Through_bottom_condition` is a selection type of `Hole_bottom_condition_select` that shall pass through two faces of a part; the depth is specified by the feature. The `Through_bottom_condition` length is specified by the size of the Hole feature.

NOTE - Figure 99 illustrates the `Through_bottom_condition`.

4.2.194 Through_pocket_bottom_condition

A `Through_pocket_bottom_condition` is a pocket which passes through two faces of a part; the depth is specified by the size of the feature.

NOTE - Figure 51 illustrates `Through_pocket_bottom_condition`.

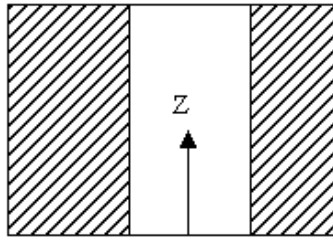


Figure 99 -Through_bottom_condition

4.2.195 Tolerance_limit

A **Tolerance_limit** is a upper or lower tolerance value applied directly to a dimension. When applied to a **Dimensional_tolerance**, the **dimensional_value** (see 4.2.60.2) shall be a tolerance value. When applied to a **Numeric_parameter_with_tolerance**, the **parameter_value** (see 4.2.111.2) shall be a tolerance value. There shall be a qualifier that describes the tolerance context. The data associated with a **Tolerance_limit** are the following: **limit_qualifier**. 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.

4.2.196 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 4.2.60.2) may be a nominal tolerance value. When applied to a **Numeric_parameter_with_tolerance**, the **parameter_value** (see 4.2.111.2) may be a nominal tolerance value. The data associated with a **Tolerance_range** are the following:

- **lower_range**;
- **significant_digits**;
- **upper_range**.

4.2.196.1 lower_range

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

4.2.196.2 significant_digits

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

4.2.196.3 upper_range

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

4.2.197 Tolerance_value

A Tolerance_value is the representation of the magnitude of the allowable deviation required for dimensions. These tolerance values may be explicitly defined or may require a specification for definition. The data associated with a Tolerance_value are the following:

- defined_value;
- envelope.

4.2.197.1 defined_value

The defined_value specifies the tolerance deviation value. See 4.3.223, 4.3.224, 4.3.225, and 4.3.226 for the application assertion.

4.2.197.2 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.

4.2.198 Tolerance_zone

A Tolerance_zone is an area where all points of the geometric element that have tolerances shall be contained.

EXAMPLE - A point, line, surface, or plane are examples for geometric elements that have tolerances zones.

The data associated with a Tolerance_zone are the following:

- common_zone;
- extended_shape;
- form_type;
- zone_definition.

4.2.198.1 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.

4.2.198.2 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 4.3.228 for the application assertion.

4.2.198.3 form_type

The `form_type` specifies the shape of the `Tolerance_zone`.

EXAMPLE - 'Cylindrical', 'parallelepiped', 'spherical' are examples for `form_type`.

4.2.198.4 zone_definition

The `zone_definition` specifies the defining boundaries for a `Tolerance_zone`. See 4.3.227 for the application assertion.

4.2.199 Tolerance_zone_definition

A `Tolerance_zone_definition` is the boundaries of a `Tolerance_zone`. Each tolerance zone shall be defined by at least one shape and may be defined with two shapes.

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

- `first_element`;
- `second_element`.

4.2.199.1 first_element

The `first_element` specifies one of two shapes for defining the boundary for the `Tolerance_zone_definition`. See 4.3.229 for the application assertion.

4.2.199.2 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 4.3.229 for the application assertion.

4.2.200 Total_runout_tolerance

A Total_runout_tolerance is a type of Geometric_tolerance (see 4.2.86) that is a compound tolerances used to control the functional relationship of one or more surfaces of a part to a datum axis. The types of surfaces controlled by Total_runout_tolerance tolerances include those surfaces constructed around a datum axis and those constructed at right angles to a datum axis. Surfaces shall be within the tolerance when the part is rotated about the datum axis.

NOTE 1 - Figure 100 illustrates the Total_runout_tolerance.

NOTE 2 - The Total_runout_tolerance definition is derived from paragraph 14.13 of ISO 1101.

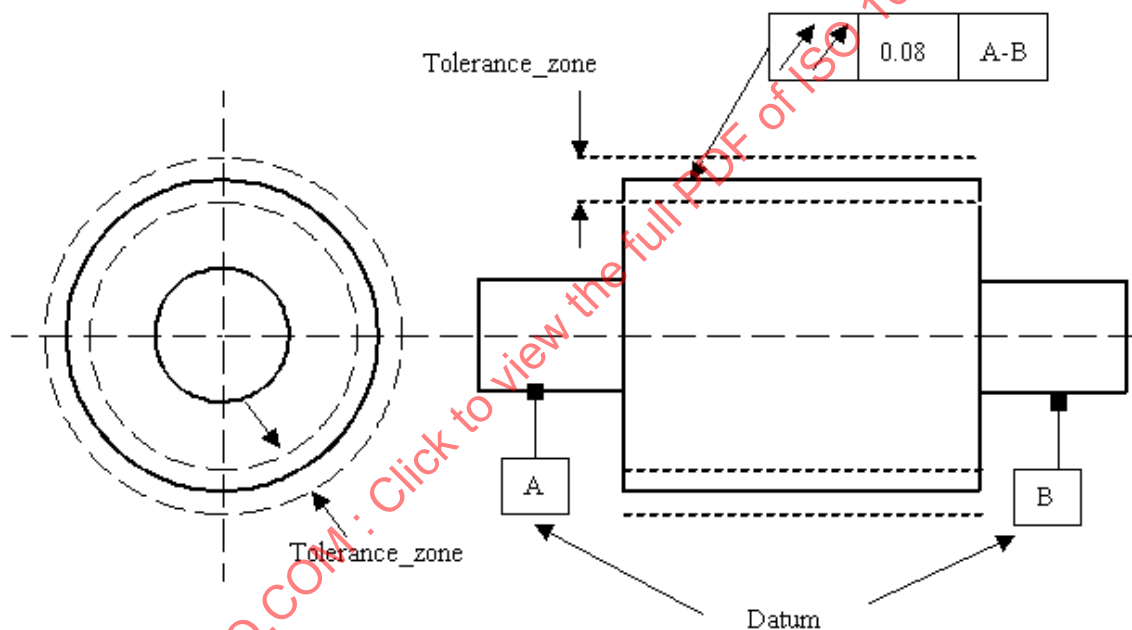


Figure 100 - Total_runout_tolerance

The data associated with a Total_runout_tolerance are the following:

- geometric_reference;
- runout.

4.2.200.1 geometric_reference

The `geometric_reference` specifies the datum to which the tolerance is related. See 4.3.230 for the application assertion.

4.2.200.2 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`.

4.2.201 Transition_feature

A `Transition_feature` is a type of `Manufacturing_feature` (see 4.2.102) 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 4.2.19), `Edge_round` (see 4.2.63), or a `Fillet` (see 4.2.69).

4.2.202 Turned_knurl

A `Turned_knurl` is a type of `Knurl` (see 4.2.93) 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 4.2.55), `Diamond_knurl` (see 4.2.58), or a `Straight_knurl` (see 4.2.181).

NOTE 1 - Figure 101 illustrates the `Turned_knurl`.

NOTE 2 - 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`.

4.2.202.1 diametral_pitch

The `diametral_pitch` specifies the ratio of the number of teeth in the circumference to the nominal diameter. See 4.3.231 for the application assertion.

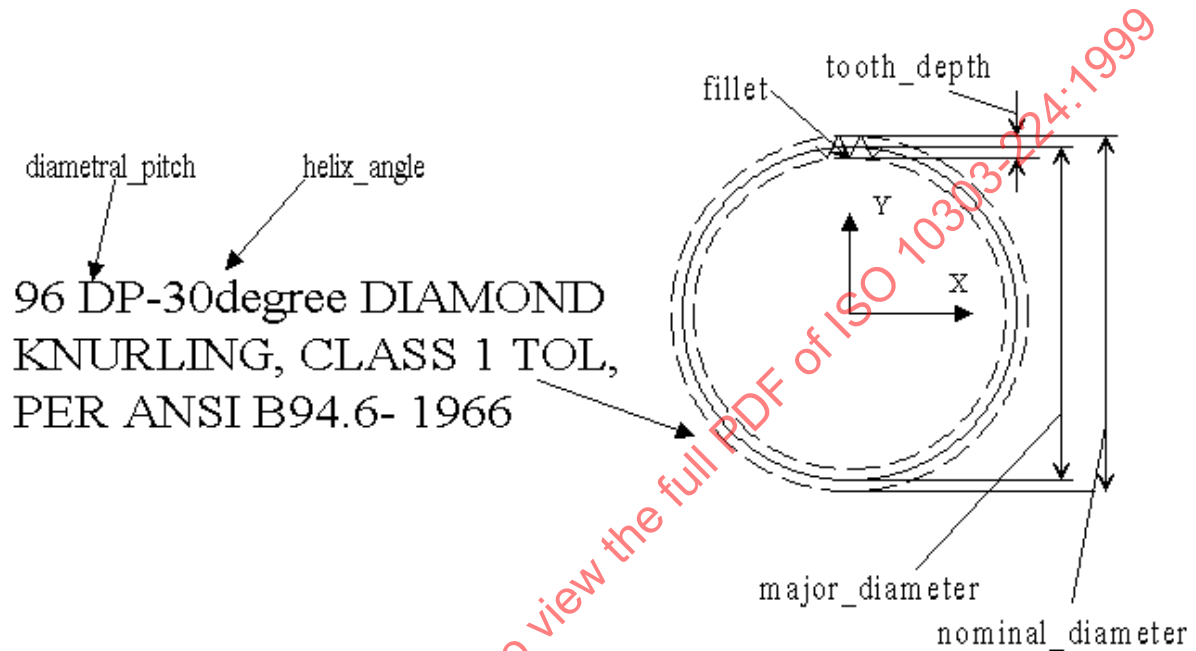


Figure 101 -Turned_knurl

4.2.202.2 major_diameter

The `major_diameter` specifies size of the part before a knurl is applied to it. See 4.3.231 for the application assertion.

4.2.202.3 nominal_diameter

The `nominal_diameter` specifies the size of the part after a knurl has been applied. See 4.3.231 for the application assertion.

4.2.202.4 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 4.3.231 for the application assertion.

4.2.202.5 root_fillet

The root_fillet specifies the dimension of a radius between teeth on a knurling tool. See 4.3.231 for the application assertion.

4.2.202.6 tooth_depth

The tooth_depth specifies the depth from the crest of a tooth to the point where two teeth intersect. See 4.3.231 for the application assertion.

4.2.203 Vee_profile

A Vee_profile is a type of Open_profile (see 4.2.113) that is a shape bounded by two lines that connect at a point and extends infinitely. The enclosed angle is less than 180 degrees. The intersection of the two lines need not be blended with a radius. The profile is positioned with the opening in the direction of the Y axis. The Y-axis intersects the angle between the two sides.

NOTE - Figure 102 illustrates the Vee_profile.

The data associated with a Vee_profile are the following:

- profile_angle;
- profile_radius;
- tilt_angle.

4.2.203.1 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 4.3.232 for the application assertion.

4.2.203.2 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. See 4.3.232 for the application assertion.

4.2.203.3 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 4.3.232 for the application assertion.

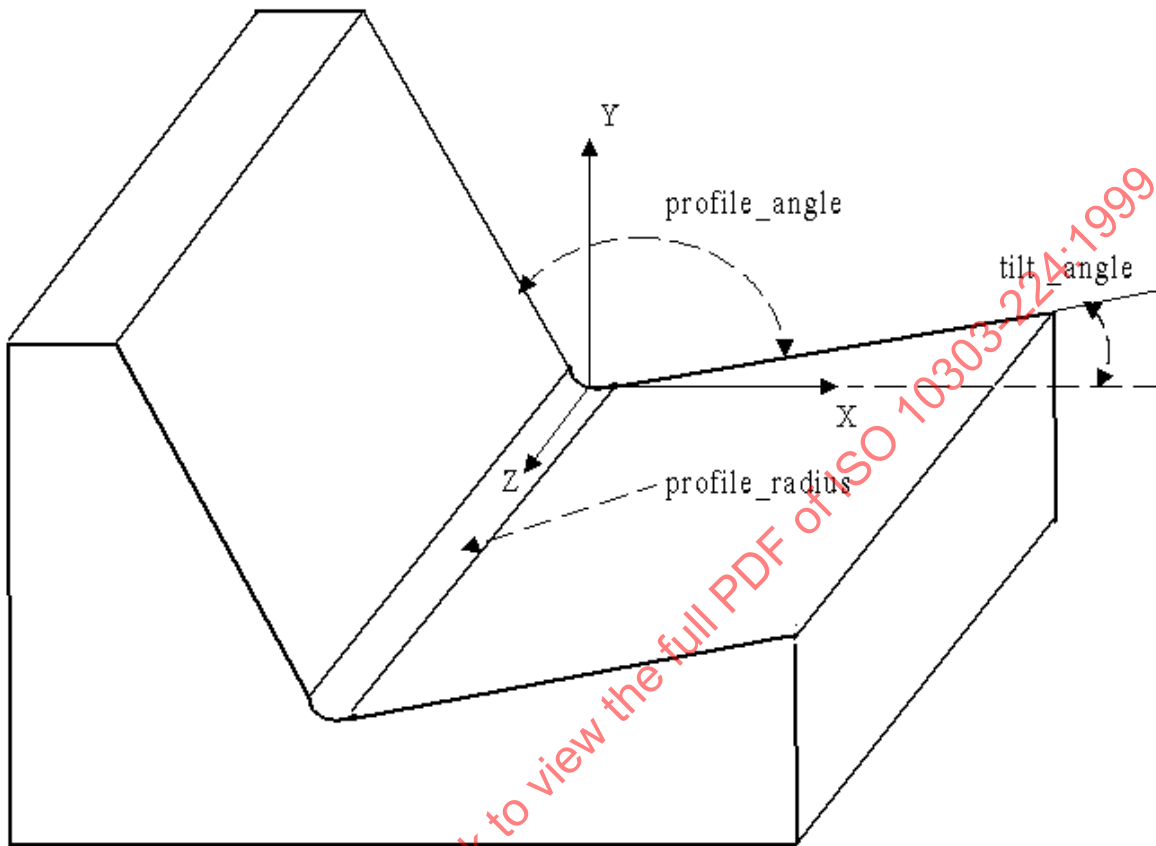


Figure 102 - Vee_profile

4.2.204 Woodruff_slot_end_type

A Woodruff_slot_end_type is a type of Slot_end_type (see 4.2.174) that is an end condition of a slot that shall be a radius tangent to the Slot bottom, and curved upward about an axis.

NOTE - Figure 103 illustrates the Woodruff_slot_end_type.

The data associated with a Woodruff_slot_end_type are the following: 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 4.3.233 for the application assertion.

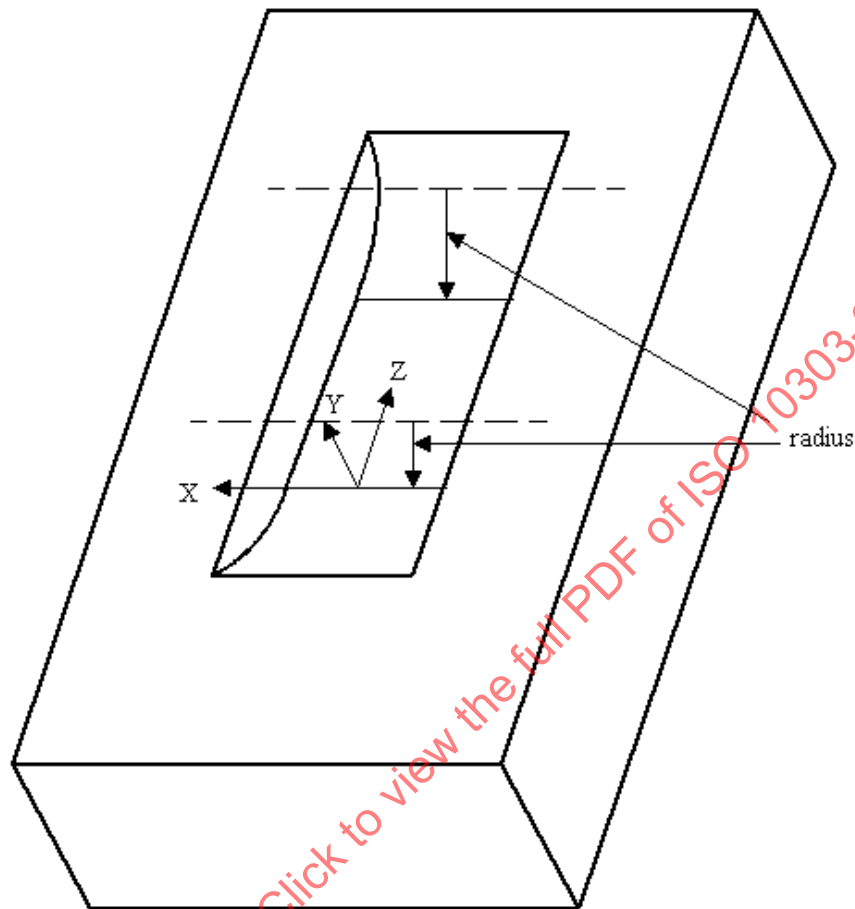


Figure 103 - Slot with Woodruff_slot_end_type at each end

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 Alternate_material to Material

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

4.3.2 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.

4.3.3 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 Approval to Person_in_organization

Each Approval has the approval_authority defined by one or many Person_in_organization objects. Each Person_in_organization is the approval_authority for zero, one, or many Approval objects.

4.3.5 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 exactly one Numeric_parameter. Each Numeric_parameter defines the fillet_radius for zero, one, or many Boss objects.

4.3.9 B-rep_model_element to B-rep_model

Each B-rep_model_element has the element defined by exactly one B-rep_model. Each B-rep_model defines the element for zero, one, or many B-rep_model_element objects.

4.3.10 B-rep_shape_representation to B-rep_model

Each B-rep_shape_aspect_representation has the shape_definition defined by exactly one B-rep_model. Each B-rep_model defines the shape_definition for zero, one, or many B-rep_shape_aspect_representation objects.

4.3.11 B-rep_shape_aspect_representation to B-rep_model

Each B-rep_shape_aspect_representation has the shape_definition defined by exactly one B-rep_model. Each B-rep_model defines the shape_definition for zero, one, or many B-rep_shape_aspect_representation objects.

4.3.12 Catalogue_knurl to Specification

Each Catalogue_knurl has documentation defined by exactly one Specification. Each Specification defines the documentation for zero, one, or many Catalogue_knurl objects.

4.3.13 Catalogue_marking to Specification

Each Catalogue_marking has documentation defined by exactly one Specification. Each Specification defines the documentation for zero, one, or many Catalogue_marking objects.

4.3.14 Catalogue_thread to Specification

Each Catalogue_thread has documentation defined by exactly one Specification. Each Specification defines the documentation for zero, one, or many Catalogue_thread objects.

4.3.15 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.

4.3.16 Chamfer to First_offset

Each Chamfer has the first_face_offset defined by exactly one First_offset. Each First_face_offset defines the first_face_offset for zero, one, or many Chamfer objects.

4.3.17 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.

4.3.18 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.

4.3.19 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.

4.3.20 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.

4.3.21 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.

4.3.22 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.

4.3.23 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.

4.3.24 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.

4.3.25 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.

4.3.26 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.

4.3.27 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.

4.3.28 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 exactly 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.

4.3.29 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.

4.3.30 Compound_datum to Datum_feature

Each Compound_datum has element defined by two or more Datum_feature objects. Each Datum_feature defines the element for zero, one, or many Compound_datum objects.

4.3.31 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.

4.3.32 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.

4.3.33 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.

4.3.34 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.

4.3.35 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.

4.3.36 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.

4.3.37 Constant_radius_edge_round to Numeric_parameter

Each Constant_radius_edge_round has the first_face_offset defined by exactly 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 exactly 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.

4.3.38 Constant_radius_fillet to Numeric_parameter

Each Constant_radius_fillet has the first_face_offset defined by exactly 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 exactly 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.

4.3.39 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.

4.3.40 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 tapered_hole for zero, one, or many Countersunk_hole objects.

4.3.41 Customer_order to Ordered_part

Each Customer_order has the quantity_ordered defined by one or many Ordered_part objects. Each Ordered_part defines the quantity_ordered by zero or one Customer_order.

4.3.42 Customer_order to Person_in_organization

Each Customer_order has the customer defined by exactly one Person_in_organization. Each Person_in_organization defines the customer for zero, one, or many Customer_order objects.

4.3.43 Customer_order to Project_order

Each Customer_order has the initiated_order defined by one or many Project_order objects. Each Project_order defines the initiated_order by zero or one Customer_order.

4.3.44 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.

4.3.45 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.

4.3.46 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.

4.3.47 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.

4.3.48 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.

4.3.49 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.

4.3.50 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.

4.3.51 Defined_thread to Numeric_parameter

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 by zero or one Numeric_parameter. Each Numeric_parameter defines the pitch_diameter for zero, one, or many Defined_thread objects.

4.3.52 Design_exception_notice to Engineering_change_proposal

Each Design_exception_notice has the result defined by zero, one, or many Engineering_change_proposal objects. Each Engineering_change_proposal defines the result for exactly one Design_exception_notice.

4.3.53 Design_exception_notice to Part

Each Design_exception_notice has the discrepant_part defined by one or many Part objects. Each Part defines the discrepant_part for zero, one, or many Design_exception_notice objects.

4.3.54 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.

4.3.55 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.

4.3.56 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.

4.3.57 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.

4.3.58 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.

4.3.59 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 path for zero, one, or many Distance_along_curve_tolerance objects.

4.3.60 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.

4.3.61 Engineering_change_order to Engineering_change_proposal

Each Engineering_change_order has the incorporated_proposal defined by one or more Engineering_change_proposal objects. Each Engineering_change_proposal defines the incorporated_proposal for zero, one, or many Engineering_change_order objects.

4.3.62 Engineering_change_order to Part

Each Engineering_change_order has the new_version defined by one or many Part objects. Each Part defines the new_version_changes by zero or one Engineering_change_order.

4.3.63 Explicit_base_shape_representation to B-rep_shape_representation

Each Explicit_base_shape_representation has the B-rep_form defined by exactly one B-rep_shape_representation. Each B-rep_shape_representation defines the B-rep_form for zero, one, or many Explicit_base_shape_representation objects.

4.3.64 Externally_defined_tolerance_value to Specification

Each Externally_defined_tolerance_value has documentation defined by exactly one Specification. Each Specification defines documentation for zero, one, or many Externally_defined_tolerance_value objects.

4.3.65 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.

4.3.66 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.

4.3.67 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.

4.3.68 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.

4.3.69 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.

4.3.70 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.

4.3.71 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.

4.3.72 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.

4.3.73 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.

4.3.74 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.

4.3.75 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.

4.3.76 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.

4.3.77 General_pocket to Profile

Each General_pocket has the boundary defined by exactly one Profile. Each Profile defines boundary for zero, one, or many General_pocket objects.

4.3.78 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.

4.3.79 General_pocket_bottom_condition to Numeric_parameter

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

4.3.80 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.

4.3.81 General_revolution to General_open_profile

Each General_revolution has the outer_edge_shape defined by exactly one Open_profile. Each General_open_profile defines outer_edge_shape for zero, one, or many General_revolution objects.

4.3.82 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.

4.3.83 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.

4.3.84 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.

4.3.85 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.

4.3.86 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.

4.3.87 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.

4.3.88 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.

4.3.89 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.

4.3.90 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.

4.3.91 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.

4.3.92 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.

4.3.93 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.

4.3.94 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.

4.3.95 Linear_profile_tolerance to Orientation

Each Linear_profile_tolerance has an affected_plane defined by exactly one Orientation. Each Orientation defines the affected_plane for zero, one, or many Linear_profile_tolerance objects.

4.3.96 Location_dimension_tolerance to Orientation

Each Location_dimension_tolerance has a plane and direction defined by zero or one Shape_aspect. Each Orientation defines a plane and direction for zero, one, or many Location_dimension_tolerance objects.

4.3.97 Location_tolerance to Shape_element

Each Location_tolerance has the termination_shape defined by exactly one Shape_element. Each Shape_element defines the termination_shape for zero, one, or many Location_dimension objects.

Each Location_dimension has a origin_shape defined by exactly one Shape_element. Each Shape_element defines the origin_shape for zero, one, or many Location_dimension objects.

4.3.98 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.

4.3.99 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.

4.3.100 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.

4.3.101 Material to Material_property

Each Material has the material_characteristics defined by zero, one, or many Material_property objects. Each Material_property defines the material_characteristics for exactly one Material.

4.3.102 Material to Specification

Each Material has the material_specification defined by zero, one, or many Specification objects. A Specification defines the material_specification for exactly one Material.

4.3.103 Material_property to Hardness

Each Material_property has the material_hardness defined by zero, one, or many Hardness objects. A Hardness defines material_hardness for exactly one Material_property.

4.3.104 Material_property to Property_parameter

Each Material_property has the property_characteristic defined by zero, one, or many Property_parameter objects. Each Property_parameters defines the property_characteristic for exactly one Material_property.

4.3.105 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 circumscribed_diameter defined by exactly one Numeric_parameter. Each Numeric_parameter defines the circumscribed_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.

4.3.106 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 circumscribed_diameter defined by exactly one Numeric_parameter. Each Numeric_parameter defines the circumscribed_diameter for zero, one, or many Ngon_profile objects.

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

4.3.107 Numeric_parameter_with_tolerance to Plus_minus_value

Each Numeric_parameter_with_tolerance has the implicit_tolerance defined by exactly one Plus_minus_value. Each Plus_minus_value defines the implicit_tolerance for zero, one, or many Numeric_parameter_with_tolerance objects.

NOTE - This assertion is established through the Numeric_parameter_tolerance_select.

4.3.108 Numeric_parameter_with_tolerance to Tolerance_limit

Each Numeric_parameter_with_tolerance has the implicit_tolerance defined by exactly one Tolerance_limit. Each Tolerance_limit defines the implicit_tolerance for zero, one, or many Numeric_parameter_with_tolerance objects.

NOTE - This assertion is established through the Numeric_parameter_tolerance_select.

4.3.109 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.

4.3.110 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.

4.3.111 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.

4.3.112 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.

4.3.113 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.

4.3.114 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.

4.3.115 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.

4.3.116 Part to Alternate_material

Each Part has the alternate_material_definition defined by zero, one, or many Alternate_material objects. Each Alternate_material defines the alternate_material_definition for exactly one Part.

4.3.117 Part to Approval

Each Part has the manufacture_authorization defined by zero, one, or many Approval objects. Each Approval defines the manufacture_authorization for zero, one, or many Part objects.

4.3.118 Part to Material

Each Part has the material_definition defined by one or more Material objects. Each Material defines the material_definition for one or more Part objects.

4.3.119 Part to Ordered_part

Each Part has the quantity_ordered defined by one or more Ordered_part objects. Each Ordered_part defines quantity_ordered for exactly one part.

4.3.120 Part to Organization

Each Part has the manufactured_by_organization defined by one or more organization objects. Each Organization defines the manufactured_by_organization for one or more Part objects.

Each Part has the owned_by_organization defined by exactly one Organization objects. Each Organization defines the owned_by_organization for zero, one, or many Part objects.

4.3.121 Part to Person_in_organization

Each Part has the manufactured_by_person defined by one or more Person_in_organization objects. Each Person_in_organization defines the manufactured_by_person for zero, one, or many Part objects.

Each Part has the owned_by_person defined by one or more Person_in_organization objects. Each Person_in_organization defines the owned_by_person for zero, one, or many Part objects.

4.3.122 Part to Property

Each Part has the property_characteristics defined by zero, one, or many Property objects. Each Property defines the property_characteristic for exactly one Part.

4.3.123 Part to Shape

Each Part has the physical_form defined by exactly one Shape. Each Shape defines the physical_form for zero, one, or many Part objects.

4.3.124 Part_property to Property_parameter

Each Part_property has the property_characteristic defined by zero, one, or many Property_parameter objects. Each Property_parameter defines the property_characteristic for exactly one Part_property.

4.3.125 Partial_area_definition to Numeric_parameter

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

4.3.126 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.

4.3.127 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.

4.3.128 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.

4.3.129 Path to Orientation

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

4.3.130 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.

4.3.131 Perpendicularity_tolerance to Orientation

Each Perpendicularity_tolerance has the affected_plane defined by zero or one Orientation. Each Orientation defines the affected_plane for zero, one, or many Perpendicularity_tolerance objects.

4.3.132 Person_in_organization to Organization

Each Person_in_organization has the company defined by exactly one Organization. Each Organization defines the company for zero, one, or many Person_in_organization objects.

4.3.133 Person_in_organization to Person

Each Person_in_organization has the employee defined by exactly one Person. Each Person defines the employee for zero, one, or many Person_in_organization objects.

4.3.134 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.

4.3.135 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.

4.3.136 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.

4.3.137 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.

4.3.138 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.

4.3.139 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.

4.3.140 Planar_pocket_bottom_condition to Numeric_parameter

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

4.3.141 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.

4.3.142 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.

4.3.143 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.

4.3.144 Pocket to Angle_taper

Each Pocket 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 Pocket objects.

4.3.145 Pocket to Pocket_bottom_condition

Each Pocket has the bottom_condition defined by exactly one Pocket_bottom_condition. Each Pocket_bottom_condition defines the bottom_condition for zero, one, or many Pocket objects.

4.3.146 Pocket to Through_pocket_bottom_condition

Each Pocket has the bottom_condition defined by exactly one Through_pocket_bottom_condition. Each Through_pocket_bottom_condition defines the bottom_condition for zero, one, or many Pocket objects.

4.3.147 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.

4.3.148 Position_tolerance to Orientation

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

4.3.149 Process_property to Property_parameter

Each Process_property has the property_characteristic defined by zero, one, or many Property_parameters objects. Each Property_parameter defines the property_characteristic for exactly one Process_property.

4.3.150 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.

4.3.151 Project_order to Approval

Each Project_Order has the release_authorization defined by exactly one Approval. Each Approval defines the release_authorization for zero or one Project_Order.

4.3.152 Project_order to Digital_technical_data_package_work_order

Each Project_order has the technical_data_package_status defined by zero or one Digital_technical_data_package_work_order objects. Each Digital_technical_data_package_work_order defines the technical_data_package_status for exactly one Project_order.

4.3.153 Project_order to Part

Each Project_order has the part_status defined by one or more Part objects. Each Part defines the part_status for exactly one Project_order.

4.3.154 Project_order to Pedigree_creation_order

Each Project_order has the pedigree_creation_order of zero or one Pedigree_creation_order. Each Pedigree_creation_order defines the pedigree_creation_order for exactly one Project_order.

4.3.155 Project_order to Requisition

Each Project_order has the ordered_resource defined by zero, one, or many Requisition objects. Each Requisition defines the ordered_resource for exactly one Project_order.

4.3.156 Project_order to Resource_acquisition_order

Each Project_order has the resource_acquisition_status defined by zero or one Resource_acquisition_order. Each Resource_acquisition_order defines the resource_acquisition_order for exactly one Project_order.

4.3.157 Project_order to Shop_work_order

Each Project_order has the shop_work_status defined by zero or one Shop_work_order. Each Shop_work_order defines the shop_work_status for exactly one Project_order.

4.3.158 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.

4.3.159 Property to Material_property

Each Property has the material_characteristic defined by zero, one, or many Material_property objects. Each Material_property defines the material_characteristic for one or more Property objects.

4.3.160 Property to Part_property

Each Property has the part_property_characteristic defined by zero, one, or many Part_property objects. Each Part_property defines the part_property_characteristic for one or more Property objects.

4.3.161 Property to Process_property

Each Property has the process_characteristic defined by zero, one, or many Process_property objects. Each Process_property defines the process_characteristic for one or more Property objects.

4.3.162 Property to Shape_aspect

Each Property has the property_characteristic defined by zero, one, or many Shape_aspect objects. Each Shape_aspect defines the property_characteristic for zero, one, or many Property objects.

4.3.163 Property to Specification

Each Property has the property_description defined by zero, one, or many Specification objects. Each Specification defines the property_description for exactly one Property.

4.3.164 Property to Surface_property

Each Property has the surface_characteristic defined by zero, one, or many Surface_property objects. Each Surface_property defines the surface_characteristic for one or more Property objects.

4.3.165 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.

4.3.166 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.

4.3.167 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 exactly one Numeric_parameter. Each Numeric_parameter defines the corner_radius for zero, one, or many Rectangular_closed_profile objects.

4.3.168 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.

4.3.169 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.

4.3.170 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.

4.3.171 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.

The open boundary is defined by a Square_U_profile (see 4.2.179) that when swept along a path defines the area on a part for volume removal. The orientation and placement of the Square_U_profile shall be the same as the Rectangular_open_pocket.

4.3.172 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.

4.3.173 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.

4.3.174 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.

4.3.175 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.

4.3.176 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.

4.3.177 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.

4.3.178 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.

4.3.179 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.

4.3.180 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.

4.3.181 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.

4.3.182 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.

4.3.183 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.

4.3.184 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.

4.3.185 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.

4.3.186 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.

4.3.187 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.

4.3.188 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.

4.3.189 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.

4.3.190 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.

4.3.191 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.

4.3.192 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.

4.3.193 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.

4.3.194 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.

4.3.195 Shape to Base_shape

Each Shape has the base_shape_definition defined by exactly one Base_shape. Each Base_shape defines the base_shape_definition for exactly one Shape.

4.3.196 Shape to B-rep_shape_representation

Each Shape has the B-rep_form defined by zero, one, or many B-rep_shape_representation objects. Each B-rep_shape_representation defines the B-rep_form for exactly one Shape.

4.3.197 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 exactly one Shape.

4.3.198 Shape_aspect to B-rep_model_element

Each Shape_aspect has the B-rep_shape defined by zero, one, or many B-rep_model_element objects. Each B-rep_model_element defines the B-rep_shape for exactly one Shape_aspect.

4.3.199 Shape_aspect to B-rep_shape_aspect_representation

Each Shape_aspect has the B-rep_form defined by zero, one, or many B-rep_shape_aspect_representation objects. Each B-rep_shape_aspect_representation defines the B-rep_form for exactly one Shape_aspect.

4.3.200 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.

4.3.201 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.

4.3.202 Slot to Numeric_parameter

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

4.3.203 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.

4.3.204 Slot to Path

Each Slot has the course_of_travel defined by exactly one Path. Each Path defines the course_of_travel for zero, one, or many Slot objects.

4.3.205 Slot to Slot_end_type

Each 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 Slot objects.

4.3.206 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 constrain for exactly one Specification.

4.3.207 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.

4.3.208 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.

4.3.209 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 exactly 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 exactly 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.

4.3.210 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.

4.3.211 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.

4.3.212 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.

4.3.213 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.

4.3.214 Surface_property to Property_parameter

Each Surface_property has the property_characteristic defined by zero, one, or many Property_parameter objects. A Property_parameter defines the property_characteristic for exactly one Surface_property.

4.3.215 Symmetry_tolerance to Datum

Each Symmetry has the geometric_reference defined by one, two, or three Datum objects. Each Datum defines the geometric_reference for zero, one, or many Symmetry objects.

4.3.216 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.

4.3.217 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.

4.3.218 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 exactly 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.

4.3.219 Thread to Descriptive_parameter

Each Thread has the fit_class defined by zero or one Descriptive_parameter. Each Descriptive_parameter defines the fit_class 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 exactly 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.

4.3.220 Thread to Numeric_parameter

Each Thread has the major_diameter defined by exactly one Numeric_parameter. Each Numeric_parameter defines the major_diameter 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.

4.3.221 Thread to Partial_area_definition

Each Thread 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 Thread objects.

4.3.222 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.

4.3.223 Tolerance_value to Externally_defined_tolerance_value

Each Tolerance_value has the defined_value by zero, one Externally_defined_tolerance_value. Each Externally_defined_tolerance_value defines the defined_value for zero, one, or many Tolerance_value objects.

NOTE - This assertion is established through Tolerance_definition_select.

4.3.224 Tolerance_value to Plus_minus_value

Each Tolerance_value has the defined_value by zero, one Plus_minus_value. Each Plus_minus_value defines the defined_value for zero, one, or many Tolerance_value objects.

NOTE - This assertion is established through Tolerance_definition_select.

4.3.225 Tolerance_value to Tolerance_limit

Each Tolerance_value has the defined_value by zero or one tolerance_limit. Each tolerance_limit defines the values for zero, one, or many Tolerance_value objects.

NOTE - This assertion is established through Tolerance_definition_select.

4.3.226 Tolerance_value to Tolerance_range

Each Tolerance_value has the defined_value by zero or one tolerance_ranges. Each tolerance_range defines the defined_value for zero, one, or many Tolerance_value objects.

NOTE - This assertion is established through Tolerance_definition_select.

4.3.227 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.

4.3.228 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.

4.3.229 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.

4.3.230 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.

4.3.231 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.

4.3.232 Vee_profile to Numeric_parameter

Each Vee_profile has the profile_radius defined by exactly 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.

4.3.233 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 table

This clause contains the mapping table 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). The mapping table is organized in five columns.

Column 1) Application element: Name of an application element as it appears in the application object definition in 4.2. Application object names are written in uppercase. Attribute names and assertions are listed after the application object to which they belong and are written in lower case.

Column 2) AIM element: Name of an AIM element as it appears in the AIM (see annex A), the term 'IDENTICAL MAPPING', or the term 'PATH'. AIM entities are written in lower case. Attribute names of AIM entities are referred to as <entity name>.<attribute name>. The mapping of an application element may result in several related AIM elements. Each of these AIM elements requires a line of its own in the table. The term 'IDENTICAL MAPPING' indicates that both application objects of the application assertion map to the same AIM element. The term 'PATH' indicates that the application assertion maps to the entire reference path.

Column 3) Source: For those AIM elements that are interpreted from the integrated resources, this is the number of the corresponding part of ISO 10303. For those AIM elements that are created for the purpose of this part of ISO 10303, this is the number of this part.

Column 4) Rules: One or more numbers may be given that refer to rules that apply to the current AIM element or 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. The expanded names of the rules are listed after the table.

Column 5) Reference path: To describe fully the mapping of an application object, it may be necessary to specify a reference path through several related AIM elements. The reference path column documents the role of an AIM element relative to the AIM element in the row succeeding 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 up 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:

- a) [] : multiple AIM elements or sections of the reference path are required to satisfy an information requirement;
- b) () : multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;
- c) {} : enclosed section constrains the reference path to satisfy an information requirement;
- d) -> : attribute references the entity or select type given in the following row;
- e) <- : entity or select type is referenced by the attribute in the following row;
- f) [i] : attribute is an aggregation of which a single member is given in the following row;
- g) [n] : attribute is an aggregation of which member n is given in the following row;
- h) => : entity is a supertype of the entity given in the following row;
- i) <= : entity is a subtype of the entity given in the following row;
- j) = : the string, select, or enumeration type is constrained to a choice or value;
- k) \ : the line continuation for strings that wrap.

Table 1 - Mapping table for design_exception UoF

Application element	AIM element	Source	Rules	Reference path
DESIGN_EXCEPTION_- NOTICE	versioned_action_request	41	3	
issuing_date	date	41	6	<pre> versioned_action_request feature_based_pp_dated_item = versioned_action_request feature_based_pp_dated_item <- feature_based_pp_date_assignment.items[i] feature_based_pp_date_assignment <= {date_assignment date_assignment.role -> date_role date_role.name = 'issuing date'} date_assignment date_assignment.assigned_date -> date </pre>
notice_description	versioned_action_request.purpose	41		
notice_number	versioned_action_request.id	41		
technical_- recommendation	action_method	41		<pre> versioned_action_request <- action_request_solution.request action_request_solution action_request_solution.method -> action_method </pre>
design_exception_notice to engineering_change_- proposal (as result)	PATH			<pre> versioned_action_request <- action_request_solution.request action_request_solution action_request_solution.method -> action_method feature_based_pp_document_item = action_method feature_based_pp_document_item <- feature_based_pp_document_reference.items[i] feature_based_pp_document_reference <= document_reference document_reference.assigned_document -> document document {document.kind -> document_type document_type.product_data_type = 'engineering change proposal'} </pre>

Table 1 - Mapping table for design_exception UoF (continued)

Application element	AIM element	Source	Rules	Reference path
design_exception_notice to part (as discrepant_part)	PATH			<pre> versioned_action_request <- action_request_assigned_action_request action_request_assignment => feature_based_pp_action_request_assignment feature_based_pp_action_request_items[i] -> feature_based_pp_action_request_item feature_based_pp_action_request_item product_definition_formation </pre>
ENGINEERING_- CHANGE_ORDER	action_directive	41		
change_order_number	action_directive.name	41		
engineering_change_- order to engineering_- change_proposal (as incorporated_- proposal)	PATH			<pre> action_directive <- directed_action.directive directed_action <= executed_action <= action action.chosen_method -> action_method feature_based_pp_document_item = action_method feature_based_pp_document_item <- feature_based_pp_document_reference.items[i] feature_based_pp_document_reference <= document_reference document_reference.assigned_document -> document {document document.kind -> document_type document_type document_type.product_data_type = 'engineering change proposal'} </pre>

Table 1 - Mapping table for design_exception UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
engineering_change_order to part (as new_version)	PATH			action_directive <- directed_action.directive directed_action <= executed_action <= action <- action_assignment.assigned_action action_assignment => feature_based_pp_action_assignment feature_based_pp_action_assignment.items[i] -> feature_based_pp_action_item feature_based_pp_action_item product_definition_formation
ENGINEERING_CHANGE_PROPOSAL	document	41		{document document.kind -> document_type document_type.product_data_type = 'engineering change proposal'}
change_proposal_number	document.id	41		

Table 2 - Mapping table for feature_definition_item UoF

Application element	AIM element	Source	Rules	Reference path
ANGLE_TAPER	taper	224	23	taper <= shape_aspect {shape_aspect shape_aspect.description = 'angle taper'}
angle_taper to numeric_- parameter (as angle)	PATH		9, 19, 24	taper <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> representation_item {representation_item.name = 'taper angle'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
BLIND_BOTTOM_- CONDITION	hole_bottom	224	23	hole_bottom <= shape_aspect {shape_aspect (shape_aspect.description = 'conical') (shape_aspect.description = 'flat') (shape_aspect.description = 'flat with radius') (shape_aspect.description = 'spherical')}

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
start_or_end	descriptive_representation_- item.description	45	9, 19, 24	<pre> 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 = 'blind bottom orientation'} representation_item => descriptive_representation_item descriptive_representation_item.description {descriptive_representation_item.description = 'hole start'} {descriptive_representation_item.description = 'hole end'}} </pre>
BOSS_TOP_CONDITION	boss_top	224	23	<pre> boss_top <= shape_aspect </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
start_or_end	descriptive_representation_- item.description	45	9, 19, 24	<pre> boss_top <= 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 => descriptive_representation_item descriptive_representation_item.description {(descriptive_representation_item.description = 'boss start') {(descriptive_representation_item.description = 'boss end')}}} </pre>
CHAMFER_ANGLE	chamfer_offset	224	23	<pre> chamfer_offset <= shape_aspect {shape_aspect shape_aspect.description = 'second offset'} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
chamfer_angle to numeric_parameter (as angle_amount)	PATH		9, 19, 24	<pre> chamfer_offset <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'offset angle'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>
CIRCULAR_PATH	path_feature_component	224	23	<pre> path_feature_component <= shape_aspect {shape_aspect (shape_aspect.description = 'partial circular') (shape_aspect.description = 'complete circular')}} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_path to numeric_parameter (as radius)	PATH		9, 19, 24	<pre> path_feature_component <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
COMPLETE_- CIRCULAR_PATH	path_feature_component	224	23	<pre> path_feature_component <= shape_aspect {shape_aspect shape_aspect.description = 'complete circular'} </pre>
CONICAL_HOLE_- BOTTOM	hole_bottom	224	23	<pre> hole_bottom <= shape_aspect {shape_aspect shape_aspect.description = 'conical'} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
conical_hole_bottom to numeric_parameter (as tip_angle)	PATH		9, 19, 24	<pre> hole_bottom <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'tip angle'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
conical_hole_bottom to numeric_parameter (as tip_radius)	PATH		9, 19, 24	<pre> hole_bottom <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'tip radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
DIAMETER_TAPER	taper	224	23	<pre> taper <= shape_aspect {shape_aspect {shape_aspect.description = 'diameter taper'}} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
diameter_taper to numeric_parameter (as final_diameter)	PATH		9, 19, 24	<pre> taper <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'taper diameter'} measure_representation_item => measure_representation_item <= {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
FIRST_OFFSET	chamfer_offset	224	23	<pre> chamfer_offset <= shape_aspect {shape_aspect shape_aspect.description = 'first offset'} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
first_offset to face_shape_element (as face_shape)	PATH		19	<pre> chamfer_offset <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'first face shape'} face_shape_representation </pre>
first_offset to numeric_parameter (as offset_amount)	PATH		9, 19, 24	<pre> 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} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
FLAT_HOLE_BOTTOM	hole_bottom	224	23	hole_bottom <= shape_aspect {shape_aspect shape_aspect.description = 'flat'}
FLAT_SLOT_END_TYPE	slot_end	224	23	slot_end <= shape_aspect {shape_aspect shape_aspect.description = 'flat'}
flat_slot_end_type to numeric_parameter (as first_radius)	PATH		9, 19, 24	slot_end <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'first radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
flat_slot_end_type to numeric_parameter (as second_radius)	PATH		9, 19, 24	<pre> slot_end <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'second radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
FLAT_WITH_RADIUS - HOLE_BOTTOM	hole_bottom	224	23	<pre> hole_bottom <= shape_aspect {shape_aspect shape.aspect.description = 'flat with radius'} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
flat_with_radius_hole_- bottom to numeric_- parameter (as corner_radius)	PATH		9, 19, 24	<pre> hole_bottom <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'corner radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
GENERAL_PATH	path_feature_component	224	23	<pre> path_feature_component <= shape_aspect {shape_aspect {shape_aspect.description = 'complex'}} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_path to path_- element (as sweep_path)	PATH		19	path_feature_component <= shape_aspect shape_definition = shape_aspect shape_definition <= property_definition.definition property_definition.definition property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'sweep path'} representation => path_shape_representation
GENERAL_POCKET_- BOTTOM_CONDITION	pocket_bottom	224	23	pocket_bottom <= shape_aspect {shape_aspect shape_aspect.description = 'non-planar'}
general_pocket_bottom_- condition to face_shape_- element (as floor)	PATH		19	pocket_bottom <= shape_aspect shape_definition = shape_aspect shape_definition <= property_definition.definition property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'floor'} representation => face_shape_representation

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_pocket_bottom_- condition to numeric_- parameter (as floor_radius)	PATH		9, 19, 24	pocket_bottom <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
GENERAL_TOP_- CONDITION	boss_top	224	23	boss_top <= shape_aspect {shape_aspect shape_aspect.description = 'non-planar'}

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_top_condition to face_shape_element (as top_face)	PATH		19	boss_top <= shape_aspect shape_definition = shape_aspect shape_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'top face'} representation => face_shape_representation
LINEAR_PATH	path_feature_component	224	23	path_feature_component <= shape_aspect {shape_aspect shape_aspect.description = 'linear'}

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
linear_path to numeric_- parameter (as distance)	PATH		9, 19, 24	path_feature_component <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'distance'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
OPEN_SLOT_END_TYPE	slot_end	224	23	slot_end <= shape_aspect {shape_aspect shape_aspect.description = 'open'}
PARTIAL_AREA_- DEFINITION	applied_area	224	23	applied_area <= shape_aspect

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
partial_area_definition to numeric_parameter (as applied_length)	PATH		9, 19, 24	<pre> applied_area <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'length'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
partial_area_definition to orientation (as placement)	PATH		9, 19, 24	<pre> applied_area <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement </pre>
PARTIAL_CIRCULAR_- PATH	path_feature_component	224	23	<pre> path_feature_component <= shape_aspect {shape_aspect shape_aspect.description = 'partial circular'} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
partial_circular_path to numeric_parameter (as sweep_angle)	PATH		9, 19, 24	<pre> path_feature_component <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'sweep angle'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>
PATH	path_feature_component	224	23	<pre> path_feature_component <= shape_aspect </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
path to orientation (as placement)	PATH		9, 19, 24	<pre> path_feature_component <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement </pre>
PLANAR_POCKET_- BOTTOM_CONDITION	pocket_bottom	224	23	<pre> pocket_bottom <= shape_aspect {shape_aspect shape_aspect.description = 'planar'} </pre>

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
planar_pocket_bottom_- condition to direction_- element (as floor_normal)	PATH		19	pocket_bottom <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'floor normal'} representation => direction_shape_representation
planar_pocket_bottom_- condition to location_- element (as floor_location)	PATH		19	pocket_bottom <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'floor location'} representation => location_shape_representation

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
planar_pocket_bottom_- condition to numeric_- parameter (as floor_radius)	PATH		9, 19, 24	pocket_bottom <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
PLANAR_TOP_- CONDITION	boss_top	224	23	boss_top <= shape_aspect {shape_aspect shape_aspect.description = 'planar'}

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
planar_top_condition to direction_element (as top_normal)	PATH		19	boss_top <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'top normal'} representation => direction_shape_representation
planar_top_condition to location_element (as top_location)	PATH		19	boss_top <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'top location'} representation => location_shape_representation
POCKET_BOTTOM_- CONDITION	pocket_bottom	224	23	pocket_bottom <= shape_aspect

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
start_or_end	descriptive_representation_- item.description	45	9, 19, 24	pocket_bottom <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item => descriptive_representation_item descriptive_representation_item.description {descriptive_representation_item.description = 'pocket start'} (descriptive_representation_item.description = 'pocket end'))}
RADIUSED_SLOT_- END_TYPE	slot_end	224	23	slot_end <= shape_aspect {shape_aspect shape_aspect.description = 'radiused'}
SECOND_CHAMFER_- OFFSET	chamfer_offset	224	23	chamfer_offset <= shape_aspect {shape_aspect shape_aspect.description = 'second offset'}

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
second_chamfer_offset to face_shape_element (as second_face)	PATH		19	chamfer_offset <= shape_aspect shape_definition = shape_aspect shape_definition <= property_definition.definition property_definition.definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.name = 'second face shape' face_shape_representation
SECOND_OFFSET	chamfer_offset	224	23	chamfer_offset <= shape_aspect {shape_aspect shape_aspect.description = 'second offset'}

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
second_offset to numeric_parameter (as offset_amount)	PATH		9, 19, 24	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} length_measure_with_unit} representation_item.name = 'offset amount'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} length_measure_with_unit}
SLOT_END_TYPE	slot_end	224	23	slot_end <= shape_aspect
first_or_second	shape_aspect_relationship.name	41		slot_end <= shape_aspect <= shape_aspect_relationship.relation_shape_aspect shape_aspect_relationship shape_aspect_relationship.name {(shape_aspect_relationship.name = 'first') (shape_aspect_relationship.name = 'second')}
SPHERICAL_HOLE_- BOTTOM	hole_bottom	224	23	hole_bottom <= shape_aspect {shape_aspect shape_aspect.description = 'spherical'}

Table 2 - Mapping table for feature_definition_item UoF (continued)

Application element	AIM element	Source	Rules	Reference path
spherical_hole_bottom to numeric_parameter (as radius)	PATH		9, 19, 24	<pre> hole_bottom <= shape_aspect shape_definition = shape_aspect 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} </pre>
THROUGH_BOTTOM_- CONDITION	hole_bottom	224	23	<pre> hole_bottom <= shape_aspect {shape_aspect shape_aspect.description = 'through'} </pre>
THROUGH_POCKET_- BOTTOM_CONDITION	pocket_bottom	224	23	<pre> pocket_bottom <= shape_aspect {shape_aspect shape_aspect.description = 'through'} </pre>
WOODRUFF_SLOT_- END_TYPE	slot_end	224	23	<pre> slot_end <= shape_aspect {shape_aspect shape_aspect.description = 'woodruff'} </pre>

Table 2 - Mapping table for feature_definition_item UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
woodruff_slot_end_- type to numeric_- parameter (as radius)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF

Application element	AIM element	Source	Rules	Reference path
CIRCULAR_CLOSED_- PROFILE	circular_closed_profile	224	23	circular_closed_profile <= shape_aspect
circular_closed_profile to numeric_parameter (as diameter)	PATH		9, 19, 24	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}
CLOSED_PROFILE	(circular_closed_profile) (closed_path_profile) (ngon_closed_profile) (rectangular_closed_profile)	224 224 224 224	23	(circular_closed_profile <= (closed_path_profile <= (ngon_closed_profile <= (rectangular_closed_profile <= shape_aspect
GENERAL_CLOSED_- PROFILE	closed_path_profile	224	23	closed_path_profile <= shape_aspect

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_closed_profile to path_element (as closed_profile_shape)	PATH		19	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 => path_shape_representation
GENERAL_OPEN_ - PROFILE	open_path_profile	224	23	open_path_profile <= shape_aspect
general_open_profile to path_element (as enclosed_boundary)	PATH		19	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 => path_shape_representation
LINEAR_PROFILE	linear_profile	224	23	linear_profile <= shape_aspect

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
linear_profile to numeric_parameter (as profile_length)	PATH		9, 19, 24	<pre> 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} </pre>
NGON_PROFILE	ngon_closed_profile	224	23	<pre> ngon_closed_profile <= shape_aspect </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
ngon_profile to numeric_parameter (as circumscribed_ diameter)	PATH		9, 19, 24	<pre> 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 => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} } representation_item.name = 'circumscribed diameter'} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
ngon_profile to numeric_parameter (as corner_radius)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
ngon_profile to numeric_parameter (as number_of_sides)	PATH		9, 19, 24	<pre> 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_value_component -> measure_value measure_value = count_measure count_measure} (linear_profile <=) (open_path_profile <=) (partial_circular_profile <=) (rounded_u_profile <=) (square_u_profile <=) (tee_profile <=) (vee_profile <=) shape_aspect} </pre>
OPEN_PROFILE	(linear_profile) (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)	224 224 224 224 224 224 224	23	
PARTIAL_CIRCULAR_PROFILE	partial_circular_profile	224	23	partial_circular_profile <= shape_aspect

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
partial_circular_profile to numeric_parameter (as radius)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
partial_circular_profile to numeric_parameter (as sweep_angle)	PATH		9, 19, 24	<pre> 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} </pre>
PROFILE	(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)	224 224 224 224 224 224 224 224 224 224 224	23	(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 <=

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
profile to orientation (as placement)	PATH		9, 19, 24	<pre> (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 </pre>
RECTANGULAR- CLOSED_PROFILE	rectangular_closed_profile	224	23	<pre> rectangular_closed_profile <= shape_aspect </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_closed_- profile to numeric_- parameter (as corner_radius)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_closed_profile to numeric_parameter (as profile_length)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_closed_- profile to numeric_- parameter (as profile_width)	PATH		9, 19, 24	<pre> 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} </pre>
ROUNDED_U_PROFILE	rounded_u_profile	224	23	rounded_u_profile <= shape_aspect

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rounded_u_profile to numeric_parameter (as width)	PATH		9, 19, 24	<pre> 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} </pre>
SQUARE_U_PROFILE	square_u_profile	224	23	<pre> square_u_profile <= shape_aspect </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
square_u_profile to numeric_parameter (as first_angle)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
square_u_profile to numeric_parameter (as first_radius)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
square_u_profile to numeric_parameter (as second_angle)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
square_u_profile to numeric_parameter (as second_radius)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
square_u_profile to numeric_parameter (as width)	PATH		9, 19, 24	<pre> 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} </pre>
TEE_PROFILE	tee_profile	224	23	tee_profile <= shape_aspect

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_-parameter (as cross_bar_depth)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_ parameter (as cross_bar_width)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_ parameter (as depth)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_- parameter (as first_angle)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_ parameter (as first_offset)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_-parameter (as radius)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_-parameter (as second_angle)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_ parameter (as second_offset)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to numeric_-parameter (as width)	PATH		9, 19, 24	tee_profile <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'width'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
VEE_PROFILE	vee_profile	224	23	vee_profile <= shape_aspect

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
vee_profile to numeric_ parameter (as profile_angle)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (continued)

Application element	AIM element	Source	Rules	Reference path
vee_profile to numeric_ parameter (as profile_radius)	PATH		9, 19, 24	<pre> 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} </pre>

Table 3 - Mapping table for feature_profile UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
vee_profile to numeric_ parameter (as tilt_angle)	PATH		9, 19, 24	<pre> 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} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF

Application element	AIM element	Source	Rules	Reference path
BOSS	boss	224	11, 23, 25	<pre> boss <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object {characterized_object (characterized_object.description = 'circular') (characterized_object.description = 'non-circular')}} </pre>
boss to boss_top_- condition (as top_condition)	PATH		22, 23	<pre> 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.related_shape_aspect -> shape_aspect => boss_top </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
boss to linear_path (as boss_height)	PATH		22, 23	<pre> boss <= feature_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.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'linear'} shape_aspect => path_feature_component </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
boss to numeric_ parameter (as fillet_radius)	PATH		9, 19, 24	<pre> boss <= feature_definition <= characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'fillet radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
CATALOGUE_KNURL	externally_defined_feature_definition	224	11, 23, 25	<pre> externally_defined_feature_definition <= [externally_defined_item] [{feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'knurl'}] </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
catalogue_knurl to specification (as documentation)	PATH			externally_defined_feature_definition feature_based_pp_document_item = externally_defined_feature_definition feature_based_pp_document_item <- feature_based_pp_document_reference.items[i] feature_based_pp_document_reference <= document_reference document_reference.assigned_document -> document => document_with_class
CATALOGUE_MARKING	externally_defined_feature_definition	224	11, 23, 25	externally_defined_feature_definition <= [externally_defined_item] [{feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'marking'}}]
catalogue_marking to specification (as documentation)	PATH			externally_defined_feature_definition feature_based_pp_document_item = externally_defined_feature_definition feature_based_pp_document_item <- feature_based_pp_document_reference.items[i] feature_based_pp_document_reference <= document_reference document_reference.assigned_document -> document => document_with_class
CATALOGUE_THREAD	externally_defined_feature_definition	224	11, 23, 25	externally_defined_feature_definition <= [externally_defined_item] [{feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'thread'}}]

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
catalogue_thread to specification (as documentation)	PATH			externally_defined_feature_definition feature_based_pp_document_item = externally_defined_feature_definition feature_based_pp_document_item <- feature_based_pp_document_reference.items[i] feature_based_pp_document_reference <= document_reference document_reference.assigned_document -> document => document_with_class
CHAMFER	chamfer	224	22, 26, 27	chamfer <= transition_feature <= shape_aspect
chamfer to face_shape_- element (as chamfer_face)	PATH		19	chamfer <= transition_feature <= 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 = 'chamfer face'} representation => face_shape_representation
chamfer to first_offset (as first_face_offset)	PATH		22, 23	chamfer <= transition_feature <= shape_aspect <- shape_aspect.relationship.relation_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

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
chamfer to second_- chamfer_offset (as second_face_offset)	PATH		22, 23	chamfer <= transition_feature <= shape_aspect <- shape_aspect_relationship.relation_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
CIRCULAR_BOSS	boss	224	11, 23, 25	boss <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'circular'}

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_boss to angle_-taper (as change_in_diameter)	PATH		22, 23	<pre> boss <= feature_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.relationship shape_aspect.relationship.relationship_shape_aspect -> {shape_aspect shape_aspect.description = 'angle taper'} shape_aspect => taper </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_boss to circular_closed_profile (as circular_profile)	PATH		22, 23	<pre> boss <= feature_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.relatng_shape_aspect -> shape_aspect => circular_closed_profile </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_boss to diameter_taper (as change_in_diameter)	PATH		22, 23	<pre> boss <= feature_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.related_shape_aspect -> shape_aspect.relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'diameter taper'} shape_aspect => taper </pre>
CIRCULAR_OFFSET_- PATTERN	shape_aspect	41	22	<pre> {shape_aspect <- shape_aspect.relationship.related_shape_aspect shape_aspect.relationship => feature_component_relationship => pattern_offset_membership } </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_offset_pattern to numeric_parameter (as angular_offset)	PATH		9, 19, 24	<pre> 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 = 'angular offset'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_offset_pattern to numeric_parameter (as index_number)	PATH		9, 19, 24	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'offset index'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>
CIRCULAR_OMIT_- PATTERN	shape_aspect	41	22	<pre> {shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship => feature_component_relationship => pattern_omit_membership} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_omit_pattern to numeric_parameter (as omit_index)	PATH		9, 19, 24	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'omit index'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>
CIRCULAR_PATTERN	circular_pattern	224	23	<pre> circular_pattern <= replicate_feature <= shape_aspect </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_pattern to circular_offset_pattern (as relocated_base_- feature)	PATH		22	circular_pattern <= replicate_feature <= shape_aspect <= shape_aspect.relationship.relation_shape_aspect {shape_aspect.relationship => feature_component_relationship => pattern_offset_membership} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect => replicate_feature => modified_pattern} shape_aspect
circular_pattern to circular_omit_pattern (as missing_base_feature)	PATH		22, 23	circular_pattern <= replicate_feature <= shape_aspect <= shape_aspect.relationship.relation_shape_aspect {shape_aspect.relationship => feature_component_relationship => pattern_omit_membership} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect => replicate_feature => modified_pattern} shape_aspect

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_pattern to numeric_parameter (as angular_spacing)	PATH		9, 19, 24	<pre> circular_pattern <= replicate_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'angular spacing'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_pattern to numeric_parameter (as base_feature - diameter)	PATH		9, 19, 24	<pre> circular_pattern <= replicate_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_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} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_pattern to numeric_parameter (as base_feature_rotation)	PATH		9, 19, 24	<pre> circular_pattern <= replicate_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'base feature rotation'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
circular_pattern to numeric_parameter (as number_of_features)	PATH		9, 19, 24	<pre> circular_pattern <= replicate_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition <- {property_definition_representation.definition shape_definition_representation} 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} </pre>
COMPOUND_FEATURE	composite_machining_feature	224	11, 23, 25	<pre> composite_machining_feature <= machining_feature <= instantiated_feature <= [{shape_aspect => composite_shape_aspect} shape_aspect] [feature_definition <= characterized_object] </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
feature_description	shape_aspect.description	41		composite_machining_feature <= machining_feature <= instanced_feature <= shape_aspect shape_aspect.description
feature_name	shape_aspect.name	41		composite_machining_feature <= machining_feature <= instanced_feature <= shape_aspect shape_aspect.name
compound_feature_to compound_feature_- element (as element)	PATH		11, 22, 23, 26, 27	composite_machining_feature <= machining_feature <= instanced_feature <= shape_aspect <= shape_aspect.relationship.relationship_shape_aspect {shape_aspect.relationship => feature_component_relationship} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (instanced_feature => machining_feature) (transition_feature)
COMPOUND_- FEATURE_ELEMENT	(machining_feature) (transition_feature)	224 224	11, 23, 25, 26, 27	(machining_feature <= instanced_feature <= [feature_definition <= characterized_object] [shape_aspect]) (transition_feature <= shape_aspect)
compound_feature_- element to machining_- feature (as element)	IDENTICAL MAPPING			
compound_feature_- element to transition_- feature (as element)	IDENTICAL MAPPING			

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
COMPOUND_- FEATURE_- RELATIONSHIP	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'compound feature ordering'}
compound_feature_- relationship to compound_feature_- element (as predecessor)	PATH		11, 23, 26, 27	shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => (instantiated_feature => machining_feature) (transition_feature)
compound_feature_- relationship to compound_feature_- element (as successor)	PATH		11, 23, 26, 27	shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect => (instantiated_feature => machining_feature) (transition_feature)
CONSTANT_RADIUS_- EDGE_ROUND	edge_round	224	22, 26, 27	edge_round <= transition_feature <= shape_aspect {shape_aspect shape_aspect.description = 'constant radius'}

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
constant_radius_edge_- round to numeric_- parameter (as first_face_offset)	PATH		9, 19, 24	<pre> edge_round <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'first offset'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
constant_radius_edge_- round to numeric_- parameter (as radius)	PATH		9, 19, 24	<pre> edge_round <= transition_feature <= shape_aspect shape_definition = shape_aspect 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} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
constant_radius_edge_- round to numeric_- parameter (as second_face_offset)	PATH		9, 19, 24	<pre> edge_round <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'second offset'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
CONSTANT_RADIUS_- FILLET	fillet	224	23, 26, 27	<pre> fillet <= transition_feature <= shape_aspect {shape_aspect shape_aspect.description = 'constant radius'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
constant_radius_fillet to numeric_parameter (as first_face_offset)	PATH		9, 19, 24	<pre> fillet <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition property_definition_representation.definition {property_definition_representation => shape_definition_representation} shape_definition_representation property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} 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} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
constant_radius_fillet to numeric_parameter (as radius)	PATH		9, 19, 24	<pre> fillet <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_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} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
constant_radius_fillet to numeric_parameter (as second_face_offset)	PATH		9, 19, 24	<pre> fillet <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_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 => representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
COUNTERBORE_HOLE	composite_hole	224	11, 23, 25	<pre> composite_hole <= composite_machining_feature <= machining_feature <= instantiated_feature <= {shape_aspect => composite_shape_aspect} [shape_aspect shape_aspect.description = 'counterbore']] [feature_definition = 'counterbore']] [feature_definition <= characterized_object] </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
counterbore_hole to round_hole (as larger_hole)	PATH		22, 23	<pre> composite_hole <= composite_machining_feature <= machining_feature <= instanced_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {[shape_aspect_relationship => feature_component_relationship] [shape_aspect_relationship shape_aspect_relationship.name = 'large hole']} shape_aspect_relationship.related_shape_aspect -> shape_aspect => {instanced_feature => machining_feature} instanced_feature <= feature_definition => round_hole </pre>
counterbore_hole to round_hole (as smaller_hole)	PATH		22, 23	<pre> composite_hole <= composite_machining_feature <= machining_feature <= instanced_feature <= shape_aspect <- shape_aspect_relationship.related_shape_aspect {[shape_aspect_relationship => feature_component_relationship] [shape_aspect_relationship shape_aspect_relationship.name = 'small hole']} shape_aspect_relationship.related_shape_aspect -> shape_aspect => {instanced_feature => machining_feature} instanced_feature <= feature_definition => round_hole </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
COUNTERSUNK_HOLE	composite_hole	224	11, 23, 25	<pre> composite_hole <= composite_machining_feature <= machining_feature <= instanced_feature <= [{shape_aspect => composite_shape_aspect [shape_aspect shape_aspect.description = 'countersunk']} shape_aspect [feature_definition <= characterized_object]] </pre>
countersunk_hole to round_hole (as constant_diameter_ hole)	PATH		22, 23	<pre> composite_hole <= composite_machining_feature <= machining_feature <= instanced_feature <= shape_aspect <- shape_aspect.relationship.relationing_shape_aspect {[shape_aspect_relationship => feature_component_relationship [shape_aspect_relationship shape_aspect_relationship.name = 'constant diameter hole']} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => {instanced_feature => machining_feature instanced_feature <= feature_definition => round_hole} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
countersunk_hole to round_hole (as tapered_hole)	PATH		22, 23	<pre> composite_hole <= composite_machining_feature <= machining_feature <= instanced_feature <= shape_aspect <- shape_aspect.relationship.relationship_shape_aspect {[shape_aspect.relationship => feature_component_relationship] [shape_aspect.relationship shape_aspect.relationship.name = 'tapered hole']} shape_aspect.relationship.related_shape_aspect -> {shape_aspect <- shape_aspect.relationship.relationship_shape_aspect {shape_aspect.relationship => feature_component_relationship} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> shape_aspect => taper} shape_aspect => {instanced_feature => machining_feature} instanced_feature <= feature_definition => round_hole </pre>
DEFINED_MARKING	marking	224	11, 23, 25	<pre> marking <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
defined_marking_to_descriptive_parameter (as font_name)	PATH		9, 19, 24	<pre> marking <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'font name'} representation_item => descriptive_representation_item </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
defined_marking_to_descriptive_parameter (as special_instructions)	PATH		9, 19, 24	<pre> marking <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'special_instructions'} representation_item => descriptive_representation_item </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
defined_marking_to_numeric_parameter (as character_height)	PATH		9, 19, 24	<pre> marking <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'character height'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
defined_marking_to_numeric_parameter (as character_spacing)	PATH		9, 19, 24	<pre> marking <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'character spacing'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
DEFINED_THREAD	thread	224	11, 23, 25	<pre> thread <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
defined_thread to numeric_parameter (as minor_diameter)	PATH		9, 19, 24	<pre> thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'minor diameter'} representation_item measure_representation_item <= {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
defined_thread to numeric_parameter (as pitch_diameter)	PATH		9, 19, 24	<pre> thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'pitch diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
DIAGONAL_KNURL	turned_knurl	224	11, 23, 25	<pre> turned_knurl <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'diagonal'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
diagonal_knurrl to descriptive_parameter (as helix_hand)	PATH		9, 19, 24	<pre> turned_knurrl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'helix hand'} representation_item => descriptive_representation_item </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
diagonal_knurl to numeric_parameter (as helix_angle)	PATH		9, 19, 24	<pre> turned_knurl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'helix angle'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>
DIAMOND_KNURL	turned_knurl	224	11, 23, 25	<pre> turned_knurl <= {feature_definition => instance_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'diamond'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
diamond_knurl to numeric_parameter (as helix_angle)	PATH		9, 19, 24	<pre> turned_knurl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'helix angle'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>
EDGE_ROUND	edge_round	224	23, 26, 27	<pre> edge_round <= transition_feature <= shape_aspect </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
edge_round to face_- shape_element (as edge_round_face)	PATH		19	<pre> edge_round <= transition_feature <= 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 = 'edge round face'} face_shape_representation </pre>
edge_round to face_- shape_element (as first_face_shape)	PATH		19	<pre> edge_round <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition <- property_definition.representation.definition property_definition.representation property_definition.representation.used_representation -> {representation representation.name = 'first face shape'} representation => face_shape_representation </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
edge_round to face_- shape_element (as second_face_shape)	PATH		19	<pre> edge_round <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition.definition <- property_definition.representation.definition property_definition.representation property_definition.representation.used_representation -> {representation representation.name = 'second face shape'} representation => face_shape_representation </pre>
FILLET	fillet	224, 23, 26, 27		<pre> fillet <= transition_feature <= shape_aspect </pre>
fillet to face_shape_- element (as fillet_face)	PATH		19	<pre> fillet <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition.definition <- property_definition.representation.definition property_definition.representation property_definition.representation.used_representation -> {representation representation.name = 'fillet face'} representation => face_shape_representation </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
fillet to face_shape_- element (as first_face_shape)	PATH		19	<pre> fillet <= transition_feature <= shape_aspect shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> {representation representation.name = 'first face shape'} representation => face_shape_representation </pre>
fillet to face_shape_- element (as second_face_shape)	PATH		19	<pre> fillet <= transition_feature <= 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 => face_shape_representation </pre>
GENERAL_BOSS	boss	224	11, 23, 25	<pre> boss <= {feature_definition => {instance_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'non-circular'}} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_boss to angle_-taper (as change_in_boundary)	PATH		22, 23	<pre> boss <= feature_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.relationship shape_aspect.relationship.relationship_shape_aspect -> {shape_aspect shape_aspect.description = 'angle taper'} shape_aspect => taper </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_boss to closed_- profile (as enclosed_boundary)	PATH		22, 23	<pre> boss <= feature_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.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.relationship.description = 'profile usage']} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> shape_aspect => (circular_closed_profile) (closed_path_profile) (ngon_closed_profile) (rectangular_closed_profile) </pre>
GENERAL_OUTSIDE_- PROFILE	outside_profile	224	11, 23, 25	<pre> outside_profile <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_outside_profile to profile (as boundary)	PATH		22, 23	<pre> outside_profile <= feature_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.relatng_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) </pre>
GENERAL_PATTERN	feature_pattern	224	23	<pre> feature_pattern <= replicate_feature <= shape_aspect </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_pattern to orientation (as feature_placement)	PATH		9, 19, 24	<pre> feature_pattern <= replicate_feature <= shape_aspect <- shape_aspect.relationship.relation_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect_relationship.name = 'replicate positioning']} shape_aspect_relationship shape_aspect_relationship.related_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 = 'base feature placement'} representation_item => geometric_representation_item => placement pocket <= {feature_definition => {instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'non-rectangular'}}</pre>
GENERAL_POCKET	pocket	224	11, 23, 25	

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_pocket to profile (as boundary)	PATH		22, 23	pocket <= feature_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.related_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)
GENERAL_REMOVAL_- VOLUME	removal_volume	224	11, 23, 25	removal_volume <= {feature_definition => instanced_feature => machining_feature} feature_definition <- characterized_object

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_removal_volume to shape_element (as removal_volume)	PATH		22	<pre> 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.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.relationship.description = 'volume shape usage']} shape_aspect.relationship.relationship shape_aspect.relationship.relationship_shape_aspect -> {shape_aspect shape_aspect.description = 'volume shape'} shape_aspect </pre>
GENERAL - REVOLUTION	revolved_profile	224	11, 23, 25	<pre> revolved_profile <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'open profile'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
general_revolution to general_open_profile (as outer_edge_shape)	PATH		22, 23	<pre> revolved_profile feature_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.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.relationship.description = 'profile usage']} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'outer edge shape'} shape_aspect => open_path_profile </pre>
GROOVE	revolved_profile	224	11, 23, 25	<pre> revolved_profile <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'groove'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
groove to open_profile (as sweep)	PATH		22, 23	<pre> 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.description = 'profile usage' shape_aspect.relationship.related_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) </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
HOLE	(round_hole) (composite_hole)	224 224	11, 23, 25	(round_hole <= {feature_definition => instanted_feature => machining_feature} feature_definition <= characterized_object) (composite_hole <= composite_machining_feature <= machining_feature <= instanted_feature <= {{shape_aspect => composite_shape_aspect} shape_aspect] [feature_definition <= characterized_object])
KNURL	(turned_knurl) (externally_defined_feature_definition)	224 224	11, 23	(turned_knurl <= {feature_definition => instanted_feature => machining_feature} feature_definition <= characterized_object) (externally_defined_feature_definition <= [externally_defined_item] [{{feature_definition => instanted_feature => machining_feature} feature_definition <= characterized_object {characterized_object {characterized_object.description = 'knurl'}}])

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
knurl to partial_area_definition (as partial_profile)	PATH		22, 23	<pre> (turned_knurl <=) feature_definition <= characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect shape_aspect.description = 'partial area occurrence'} shape_aspect <- shape_aspect.relationship.related_shape_aspect {{shape_aspect.relationship => feature_component_relationship} [shape_aspect.relationship shape_aspect.relationship.description = 'applied area usage']} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> shape_aspect => applied_area </pre>
knurl to shape_element (as applied_shape)	PATH		22	<pre> (turned_knurl <=) feature_definition <=) feature_definition => {instanced_feature => machining_feature} instanced_feature <= shape_aspect <- shape_aspect.relationship.related_shape_aspect {{shape_aspect.relationship => shape_defining_relationship} shape_aspect.relationship shape_aspect.relationship.name = 'applied shape'}} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> shape_aspect => applied_area </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
MACHINING_FEATURE	machining_feature	224	11, 23, 25	<pre> machining_feature <= instantiated_feature <= [shape_aspect] [feature_definition <= characterized_object] </pre>
usage_name	shape_aspect.description	41		<pre> machining_feature <= instantiated_feature <= shape_aspect shape_aspect.description </pre>
machining_feature to orientation (as placement)	PATH		9, 19, 24	<pre> machining_feature <= instantiated_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement </pre>
MANUFACTURING_FEATURE	shape_aspect	41		

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
MARKING	(marking) (externally_defined_feature_definition)	224 224	11, 23	<pre> (marking <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object (externally_defined_feature_definition <= [externally_defined_item] [{feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object {characterized_object.description = 'marking'}}]) </pre>
marking to shape_element (as applied_to_shape)	PATH		22	<pre> (marking <=) (externally_defined_feature_definition <=) feature_definition => {instantiated_feature => machining_feature} instantiated_feature <= shape_aspect <- shape_aspect_relationship.relation_shape_aspect {[shape_aspect_relationship => shape_defining_relationship] [shape_aspect_relationship shape_aspect_relationship.name = 'applied shape']} shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
marking to descriptive_- parameter (as text)	PATH			<pre> (marking <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'marking text'} representation_item => descriptive_representation_item </pre>
OUTER_DIAMETER	outer_round	224	11, 23, 25	<pre> outer_round <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'outer diameter'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
outer_diameter to angle_-taper (as reduced_size)	PATH		22, 23	<pre> outer_round <= feature_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.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'angle taper'} shape_aspect => taper </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
outer_diameter to diameter_taper (as reduced_size)	PATH		22, 23	<pre> outer_round <= feature_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.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'diameter taper'} shape_aspect => taper </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
outer_diameter to numeric_parameter (as diameter)	PATH		9, 19, 24	<pre> outer_round <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
outer_diameter to numeric_parameter (as feature_length)	PATH		9, 19, 24	<pre> outer_round <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'length'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
OUTER_DIAMETER_- TO_SHOULDER	outer_round	224	11, 23, 25	<pre> outer_round <= {feature_definition => instance_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'outer diameter to shoulder'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
outer_diameter_to_- shoulder to numeric_- parameter (as diameter)	PATH		9, 19, 24	<pre> outer_round <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
outer_diameter_to_-- shoulder to vee_profile (as v_shape_boundary)	PATH		22, 23	<pre> 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.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.relationship.description = 'profile usage']} shape_aspect.relationship.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'v-shape'} shape_aspect => vee_profile </pre>
OUTER_ROUND	outer_round	224	11, 23, 25	<pre> outer_round <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>
PLANAR_FACE	flat_face	224	11, 23, 25	<pre> flat_face <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
planar_face to direction_ element (as removal_direction)	PATH		19	<pre> flat_face <= feature_definition <= characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition.representation.definition property_definition.representation property_definition.used_representation -> {representation representation.name = 'removal direction'} representation => direction_shape_representation </pre>
planar_face to linear_path (as course_of_travel)	PATH		22, 23	<pre> 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.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.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'linear'} shape_feature_component path_feature_component </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
planar_face to linear_profile (as removal_boundary)	PATH		22, 23	<pre> flat_face <= feature_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.relatng_shape_aspect -> shape_aspect => linear_profile </pre>
POCKET	pocket	224	11, 23, 25	<pre> pocket <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
pocket to angle_taper (as change_in_boundary)	PATH		22, 23	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.relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'angle taper'} shape_aspect => taper

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
pocket to linear_path (as pocket_depth)	PATH		22, 23	<pre> pocket <= feature_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.relate_shape_aspect -> {shape_aspect shape_aspect.description = 'linear'} shape_aspect => path_feature_component </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
pocket to pocket_- bottom_condition (as bottom_condition)	PATH		22, 23	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.relationship shape_aspect_relationship.related_shape_aspect -> {shape_aspect (shape_aspect.description = 'planar') (shape_aspect.description = 'non-planar')} shape_aspect => pocket_bottom

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
pocket to through_- pocket_bottom_condition (as bottom_condition)	PATH		22, 23	pocket <= feature_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.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.relationship_shape_aspect -> {shape_aspect (shape_aspect.description = 'through')} shape_aspect => pocket_bottom
PROTRUSION	protrusion	224	11, 23, 25	protrusion <= {feature_definition => instance.feature => machining_feature} feature_definition <= characterized_object

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
protrusion to shape_- element (as shape_volume)	PATH		22	<pre> protrusion <= feature_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.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.description = 'volume shape usage']} shape_aspect.relationship shape_aspect.relationship.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'volume shape'} shape_aspect </pre>
RECTANGULAR_- CLOSED_POCKET	pocket	224	11, 23, 25	<pre> pocket <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'closed rectangular'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_closed_- pocket to rectangular_- closed_profile (as closed_boundary)	PATH		22, 23	<pre> pocket <= feature_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.related_shape_aspect -> shape_aspect.description = 'boundary'} rectangular_closed_profile </pre>
RECTANGULAR_- OFFSET_PATTERN	shape_aspect	41	22	<pre> {shape_aspect <- shape_aspect.relationship.related_shape_aspect shape_aspect.relationship => feature_component_relationship => pattern_offset_membership} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_offset_pattern to direction_element (as offset_direction)	PATH		19	<pre> shape_aspect <- shape_definition = shape_aspect shape_definition <- property_definition.definition property_definition.definition <- property_definition_representation.definition property_definition_representation.used_representation -> {representation representation.name = 'offset direction'} direction_shape_representation </pre>
rectangular_offset_pattern to numeric_parameter (as column_index)	PATH		9, 19, 24	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'column index'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value = count_measure count_measure} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_offset_pattern to numeric_parameter (as offset_distance)	PATH		9, 19, 24	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'offset distance'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_offset_pattern to numeric_parameter (as row_index)	PATH		9, 19, 24	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'row index'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure} </pre>
RECTANGULAR_OMIT_- PATTERN	shape_aspect	41	22	<pre> {shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship => feature_component_relationship => pattern_omit_membership} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_omit_pattern to numeric_parameter (as column_index)	PATH		9, 19, 24	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'column index'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_omit_pattern to numeric_parameter (as row_index)	PATH		9, 19, 24	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'row index'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>
RECTANGULAR_OPEN_- POCKET	pocket	224	11, 23, 25	<pre> pocket <= {feature_definition => instance_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'open rectangular'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_open_pocket to square_u_profile (as open_boundary)	PATH		22, 23	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.relationship shape_aspect.relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'boundary'} shape_aspect => square_u_profile
RECTANGULAR_- PATTERN	rectangular_pattern	224	23	rectangular_pattern <= replicate_feature <= shape_aspect

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_pattern to direction_element (as column_layout_- direction)	PATH		19	<pre> rectangular_pattern <= replicate_feature <= 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 = 'column layout direction'} direction_shape_representation </pre>
rectangular_pattern to direction_element (as row_layout_direction)	PATH		19	<pre> rectangular_pattern <= replicate_feature <= 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 = 'row layout direction'} representation => direction_shape_representation </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_pattern to numeric_parameter (as column_spacing)	PATH		9, 19, 24	<pre> rectangular_pattern <= replicate_feature <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'column spacing'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_pattern to numeric_parameter (as columns)	PATH		9, 19, 24	<pre> rectangular_pattern <= replicate_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> representation_item {representation_item.name = 'number of columns'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_pattern to numeric_parameter (as row_spacing)	PATH		9, 19, 24	<pre> rectangular_pattern <= replicate_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'row spacing'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_pattern to numeric_parameter (as rows)	PATH		9, 19, 24	<pre> rectangular_pattern <= replicate_feature <= shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'number of rows'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_pattern to rectangular_offset_pattern (as relocated_base_- feature)	PATH		22	<pre> rectangular_pattern <= replicate_feature <= shape_aspect <- shape_aspect.relationship.related_shape_aspect {shape_aspect.relationship => feature_component_relationship => pattern_offset_memberships shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> {shape_aspect => replicate_feature => modified_pattern } shape_aspect </pre>
rectangular_pattern to rectangular_omit_pattern (as missing_base_feature)	PATH		22	<pre> rectangular_pattern <= replicate_feature <= shape_aspect <- shape_aspect.relationship.related_shape_aspect {shape_aspect.relationship => feature_component_relationship => pattern_omit_memberships shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> {shape_aspect => replicate_feature => modified_pattern } shape_aspect </pre>
REPLICATE_BASE	(machining_feature) (replicate_feature)	224 224	11, 22, 23	<pre> (machining_feature <= instantiated_feature <=) (replicate_feature <=) shape_aspect <- {shape_aspect <- shape_aspect.relationship.related_shape_aspect {shape_aspect.relationship => feature_component_relationship shape_aspect.relationship [shape_aspect.relationship.name = 'pattern basis'] [shape_aspect.relationship.related_shape_aspect -> shape_aspect => replicate_feature]} } </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
replicate_base to machining_feature (as base_feature)	IDENTICAL MAPPING			
replicate_base to replicate_feature (as base_feature)	IDENTICAL MAPPING			
REPLICATE_FEATURE	replicate_feature	224	23	replicate_feature <= shape_aspect
replicate_feature to orientation (as placement)	PATH		9, 19, 24	replicate_feature <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
replicate_feature to replicate_base - (as replicate_base - feature)	PATH		11, 22, 23	<pre> replicate_feature <= shape_aspect <- shape_aspect.relationship.relatng_shape_aspect {shape_aspect.relationship => feature_component_relationship} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> shape_aspect => (instanced_feature => machining_feature) (replicate_feature) </pre>
REVOLVED_FEATURE	revolved_profile	224	11, 23, 25	<pre> revolved_profile <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>
revolved_feature to direction_element (as material_side)	PATH		19	<pre> revolved_profile <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition.representation.definition property_definition.representation property_definition.representation.used_representation -> {representation representation.name = 'removal direction'} representation => direction_shape_representation </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
revolved_feature_to_numeric_parameter (as radius)	PATH		9, 19, 24	<pre> revolved_profile <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
REVOLVED_FLAT	revolved_profile	224	11, 23, 25	<pre> revolved_profile <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object.description = 'flat'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
revolved_flat to linear_- profile (as flat_edge_shape)	PATH		22, 23	<pre> revolved_profile <= feature_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.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.relationship.description = 'profile usage']} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'flat edge shape'} linear_profile </pre>
REVOLVED_ROUND	revolved_profile	224	11, 23, 25	<pre> revolved_profile <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'round'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
revolved_round to partial_circular_profile (as rounded_edge_shape)	PATH		22, 23	<pre> 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.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'rounded edge shape'} shape_aspect => partial_circular_profile </pre>
ROUND_HOLE	round_hole	224	11, 23, 25	<pre> round_hole <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
round_hole to angle_taper (as change_in_diameter)	PATH		22, 23	<pre> 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.relationship shape_aspect.relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'angle taper'} shape_aspect => taper </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
round_hole to blind_- bottom_condition (as bottom_condition)	PATH		22, 23	<pre> round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object property_definition <- property_definition.definition product_definition_shape <- shape_aspect.of_shape {shape_aspect shape_aspect.description = 'bottom condition occurrence'} shape_aspect <- shape_aspect.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.relatng_shape_aspect -> {shape_aspect (shape_aspect.description = 'conical') (shape_aspect.description = 'flat') (shape_aspect.description = 'flat with radius') (shape_aspect.description = 'spherical')}} shape_aspect => hole_bottom </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
round_hole to circular_-closed_profile (as diameter)	PATH		22, 23	<pre> 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.relatng_shape_aspect -> shape_aspect => circular_closed_profile </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
round_hole to diameter_-taper (as change_in_diameter)	PATH		22, 23	<pre> 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.relationship shape_aspect.relationship.related_shape_aspect -> {shape_aspect shape_aspect.description = 'diameter taper'} shape_aspect => taper </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
round_hole to linear_path (as hole_depth)	PATH		22, 23	<pre> round_hole <= feature_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.relate_shape_aspect -> {shape_aspect shape_aspect.description = 'linear'} shape_aspect => path_feature_component </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
round_hole to through_- bottom_condition (as bottom_condition)	PATH		22, 23	<pre> round_hole <= feature_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.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'through'} shape_aspect => hole_bottom </pre>
ROUNDED_END	rounded_end	224	11, 23, 25	<pre> rounded_end <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rounded_end to linear_ path (as course_of_travel)	PATH		22, 23	<pre> 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.name = 'course of travel'] [shape_aspect.relationship.description = 'path feature component usage']]} shape_aspect.relationship.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'linear'} shape_aspect => path_feature_component </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
rounded_end to partial_circular_profile (as partial_circular_boundary)	PATH		22, 23	<pre> 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.name = 'partial circular boundary'] [shape_aspect_relationship.description = 'profile usage']]} shape_aspect.relationship shape_aspect.relationship.relatng_shape_aspect -> shape_aspect => partial_circular_profile </pre>
SLOT	slot	224	11, 23, 25	<pre> slot <= {feature_definition => instanced_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
slot to numeric_parameter (as slot_depth)	PATH		9, 19, 24	<pre> slot <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'depth'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
slot to open_profile (as sweep_shape)	PATH		22, 23	<pre> slot <= feature_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.relatng_shape_aspect -> {shape_aspect shape_aspect.description = 'swept shape'} shape_aspect => (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile) </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
slot to path (as course_of_travel)	PATH		22, 23	<pre> slot <= feature_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.relatng_shape_aspect -> shape_aspect => path_feature_component </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
slot to slot_end_type (as end_conditions)	PATH		22, 23	<pre> slot <= feature_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.related_shape_aspect -> shape_aspect => slot_end </pre>
SPHERICAL_CAP	spherical_cap	224	11, 23, 25	<pre> spherical_cap <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
spherical_cap to numeric_parameter (as internal_angle)	PATH		9, 19, 24	<pre> spherical_cap <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'internal angle'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
spherical_cap to numeric_parameter (as radius)	PATH		9, 19, 24	<pre> spherical_cap <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'radius'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
STEP	step	224	11, 23, 25	<pre> step <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
step to linear_path (as course_of_travel)	PATH		22, 23	<pre> step <= feature_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.relateing_shape_aspect -> {shape_aspect shape_aspect.description = 'linear'} shape_aspect => path_feature_component </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
step to vee_profile (as removal_boundary)	PATH		22, 23	<pre> step <= feature_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.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.relationship.description = 'profile usage']} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> shape_aspect => vee_profile </pre>
STRAIGHT_KNURL	turned_knurl	224	11, 23, 25	<pre> turned_knurl <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'straight'} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
THREAD	(thread) (externally_defined_feature_definition)	224 224	11, 23	<pre> (thread <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object) (externally_defined_feature_definition <= [externally_defined_item] [{{feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'thread'}}]) </pre>
english_or_metric	descriptive_representation_- item.description	45	9, 19, 24	<pre> (thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'thread system'} representation_item => descriptive_representation_item descriptive_representation_item.description {(descriptive_representation_item.description = 'english')} (descriptive_representation_item.description = 'metric')} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
thread to descriptive_ - parameter (as fit_class)	PATH		9,19, 24	<pre>(thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'fit class'} representation_item => descriptive_representation_item</pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
thread to descriptive_ - parameter (as form)	PATH		9,19, 24	<pre> (thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'form'} representation_item => descriptive_representation_item </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
thread to descriptive_ - parameter (as qualifier)	PATH		9, 19, 24	(thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'qualifier'} representation_item => descriptive_representation_item

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
inner_or_outer_thread	descriptive_representation_- item.description	45	9, 19, 24	<pre> (thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'thread side'} representation_item => descriptive_representation_item descriptive_representation_item.description {(descriptive_representation_item.description = 'internal')} {(descriptive_representation_item.description = 'external')}} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
thread to numeric_parameter (as major_diameter)	PATH		9,19, 24	<pre> (thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'major_diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
thread to numeric_parameter (as number_of_threads)	PATH		9,19, 24	<pre> (thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'number of threads'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
thread to partial_area_-- definition (as partial_profile)	PATH		22, 23	<pre> (thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect shape_aspect.description = 'partial area occurrence'} shape_aspect <- shape_aspect.relationship.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.description = 'applied area usage']} shape_aspect.relationship shape_aspect.relationship.relatng_shape_aspect -> shape_aspect => applied_area </pre>
thread to shape_element (as applied_shape)	PATH		22	<pre> (thread <=) (externally_defined_feature_definition <=) feature_definition => {instanced_feature => machining_feature} instanced_feature <= shape_aspect <- shape_aspect.relationship.related_shape_aspect {[shape_aspect.relationship => shape_defining_relationship] [shape_aspect.relationship shape_aspect.relationship.name = 'applied shape']} shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> shape_aspect </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
thread to descriptive_ - parameter (as thread_hand)	PATH		9, 19, 24	<pre> (thread <=) (externally_defined_feature_definition <=) feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'hand'} representation_item => descriptive_representation_item </pre>
TRANSITION_FEATURE	transition_feature	224	23, 26, 27	<pre> transition_feature <= shape_aspect </pre>
TURNUED_KNURL	turned_knurl	224	11, 23, 25	<pre> turned_knurl <= {feature_definition => instantiated_feature => machining_feature} feature_definition <= characterized_object </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
turned_knurl to numeric_parameter (as diametral_pitch)	PATH		9, 19, 24	<pre> turned_knurl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'diametral_pitch'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
turned_knurl to numeric_parameter (as number_of_teeth)	PATH		9, 19, 24	<pre> turned_knurl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'number of teeth'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
turned_knurl to numeric_parameter (as major_diameter)	PATH		9, 19, 24	<pre>turned_knurl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'major diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}</pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
turned_knurl to numeric_parameter (as nominal_diameter)	PATH		9, 19, 24	<pre> turned_knurl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'nominal diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
turned_knurl to numeric_parameter (as root_fillet)	PATH		9, 19, 24	<pre> turned_knurl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'root fillet'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 4 - Mapping table for manufacturing_part_properties UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
turned_knurl to numeric_parameter (as tooth_depth)	PATH		9, 19, 24	<pre> turned_knurl <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'tooth depth'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>

Table 5 - Mapping table for manufacturing_part_properties UoF

Application element	AIM element	Source	Rules	Reference path
ALTERNATE_- MATERIAL	product_definition	41	12	
alternate_ranking	make_from_usage_option.ranking	44		product_definition <- product_definition.relationship.related_product_definition product_definition.usage => make_from_usage_option make_from_usage_option.ranking
alternate_material to material (as material_substitute)	PATH			product_definition <- product_definition.relationship.related_product_definition {product_definition.relationship => product_definition.usage => make_from_usage_option} product_definition.relationship product_definition.relationship.related_product_definition -> product_definition
DESCRIPTIVE_- PARAMETER	descriptive_representation_item	45		
descriptive_string	descriptive_representation_- item.description	45		
HARDNESS	representation	43		{representation representation.name = 'hardness'}
high_value	measure_representation_item	45		representation representation.items[1] -> {representation_item representation_item.name = 'high value'} representation_item => measure_representation_item
low_value	measure_representation_item	45		representation representation.items[1] -> {representation_item representation_item.name = 'low value'} representation_item => measure_representation_item

Table 5 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
nominal	measure_representation_item	45		representation representation.items[1] -> {representation_item representation_item.name = 'nominal'} representation_item => measure_representation_item
scale	unit	41		representation representation.items[1] -> representation_item => measure_representation_item <= measure_with_unit measure_with_unit.component -> unit
MATERIAL	product_definition	41		
material_description	product_definition.description	41		
material_id	material_designation.name	45		product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- material_designation.definitions[i] material_designation material_designation.name
stock_size	product_definition.id	41		
material to material_- property (as material_- characteristics)	PATH			product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition => material_property
material to specification (as material_- specification)	PATH			product_definition => product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[i] -> document => document_with_class

Table 5 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
MATERIAL_PROPERTY	material_property	45		
material_property to hardness (as material_hardness)	PATH			<pre> material_property <= property_definition <- property_definition_representation property_definition_representation.used_representation -> representation {representation representation.name = 'hardness'} </pre>
material_property to property_parameter (as property_- characteristic)	PATH			<pre> material_property <= property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item </pre>
NUMERIC_PARAMETER	measure_representation_item	45		
parameter_units	named_unit	41	7	<pre> measure_representation_item <= measure_with_unit measure_with_unit.component -> unit unit = named_unit named_unit </pre>
parameter_value	measure_value	41		<pre> measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value </pre>
NUMERIC_- PARAMETER_WITH_- TOLERANCE	[measure_representation_item] [qualified_representation_item]	45 45		
numeric_parameter_- with_tolerance to plus_- minus_value (as implicit_tolerance)	IDENTICAL MAPPING			

Table 5 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
numeric_parameter_- with_tolerance_to tolerance_limit (as implicit_tolerance)	PATH			qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier value_qualifier = type_qualifier type_qualifier
PART_PROPERTY	property_definition	41		{property_definition property_definition.name = 'part property'}
part_property_to property_parameter (as property_- characteristic)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item
PROCESS_PROPERTY	property_definition	41		{property_definition property_definition.name = 'process property'}
process_name	representation.name	43		property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.name
process_property_to property_parameter (as property_- characteristic)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item
PROPERTY	property_definition	41		
property_name	property_definition.name	41		
property_to_material_ property (as material_ characteristic)	PATH			property_definition <- property_definition_relationship.relation_property_definition property_definition_relationship property_definition_relationship.related_property_definition -> property_definition => material_property

Table 5 - Mapping table for manufacturing_part_properties UoF (continued)

Application element	AIM element	Source	Rules	Reference path
property to part_property (as part_property_- characteristic)	PATH			<pre> property_definition <- property_definition.relatng_property_definition property_definition.relationship property_definition.related_property_definition -> property_definition {property_definition property_definition.name = 'part property'}</pre>
property to process_- property (as process_- characteristic)	PATH			<pre> property_definition <- property_definition.relatng_property_definition property_definition.relationship property_definition.related_property_definition -> property_definition {property_definition property_definition.name = 'process property'}</pre>
property to shape_aspect (as property_- characteristic)	PATH			<pre> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect shape_aspect</pre>
property to specification (as property_description)	PATH			<pre> property_definition feature_based_pp_document_item = property_definition feature_based_pp_document_item <- feature_based_pp_document_reference.items[i] feature_based_pp_document_reference <= document_reference document_reference.assigned_document -> document => document_with_class</pre>
property to surface_- property (as surface_characteristic)	PATH			<pre> property_definition <- property_definition.relatng_property_definition property_definition.relationship property_definition.related_property_definition -> property_definition {property_definition property_definition.name = 'surface property'}</pre>

Table 5 - Mapping table for manufacturing_part_properties UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
PROPERTY_- PARAMETER	representation_item	43		
parameter_name	representation_item.name	43		
SURFACE_PROPERTY	property_definition	41		{property_definition property_definition.name = 'surface property'}
surface_finish	property_definition.description	41		{property_definition property_definition.description = 'surface finish'}
surface_property to property_parameter (as property_- characteristic)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item

Table 6 - Mapping table for manufacturing_process_control_documentation UoF

Application element	AIM element	Source	Rules	Reference path
CUSTOMER_ORDER	directed_action	41	18	{directed_action <= executed_action <= action action.name = 'customer order'}
delivery_date	date	41	6	directed_action feature_based_pp_dated_item = directed_action feature_based_pp_dated_item <- feature_based_pp_date_assignment.items[i] feature_based_pp_date_assignment <= {date_assignment date_assignment.role -> date_role date_role.name = 'delivery date'} date_assignment date_assignment.assigned_date -> date
material_disposition	action.description	41		directed_action <= executed_action <= action action.description
order_number	action_directive.name	41		directed_action directed_action.directive -> action_directive action_directive.name
order_status	action_status	41	4	directed_action <= executed_action <= action <- action.status.assigned_action action_status
special_instructions	action_directive.comment	41		directed_action directed_action.directive -> action_directive action_directive.comment

Table 6 - Mapping table for manufacturing_process_control_documentation UoF (continued)

Application element	AIM element	Source	Rules	Reference path
customer_order to ordered_part (as quantity_ordered)	PATH			directed_action <= executed_action <= action <- action_assignment.assigned_action ordered_part
customer_order to person_in_organization (as customer)	PATH			directed_action directed_action.directive -> action_directive feature_based_pp_person_and_organization_item = action_directive feature_based_pp_person_and_organization_item <- feature_based_pp_person_and_organization_assignment.items[i] feature_based_pp_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment person_and_organization
customer_order to project_order (as initiated_order)	PATH		18	directed_action <= executed_action <= action <- action_relationship.relationship_action action_relationship.related_action -> {action action.name = 'project order'} action => executed_action => directed_action
DIGITAL_TECHNICAL_- DATA_PACKAGE_- WORK_ORDER	directed_action	41	18	{directed_action <= executed_action <= action action.name = 'digital technical data package work order'}
order_id	action_directive.name	41		directed_action directed_action.directive -> action_directive action_directive.name
ORDERED_PART	ordered_part	224	13, 25	ordered_part <= [action_assignment] [characterized_object]

Table 6 - Mapping table for manufacturing_process_control_documentation UoF (continued)

Application element	AIM element	Source	Rules	Reference path
quantity_required	measure_with_unit.value_component	41		<pre> ordered_part <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit measure_with_unit.value_component {measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure} </pre>
quantity_unit_of_- measure	measure_with_unit.unit_component	41		<pre> ordered_part <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit measure_with_unit.unit_component </pre>
PEDIGREE_CREATION_- ORDER	directed_action	41	18	<pre> {directed_action <= executed_action <= action action.name = 'pedigree creation_order'} </pre>

Table 6 - Mapping table for manufacturing_process_control_documentation UoF (continued)

Application element	AIM element	Source	Rules	Reference path
order_id	action_directive.name	41		directed_action directed_action.directive-> action_directive action_directive.name
PROJECT_ORDER	directed_action	41	13, 18	{directed_action <= executed_action <= action action.name = 'project order'}
project_order_id	action_directive.name	41		directed_action directed_action.directive-> action_directive action_directive.name
project_order to approval (as release_authorization)	PATH		17	directed_action feature_based_pp_approved_item = directed_action feature_based_pp_approved_item <- feature_based_pp_approval_assignment.items[i] feature_based_pp_approval_assignment <= approval_assignment approval_assignment.assigned_approval-> approval
project_order to digital_- technical_data_package- work_order (as technical_data_- package_status)	PATH		18	directed_action <= executed_action <= action <- action_relationship.relation_action action_relationship action_relationship.related_action-> {action action => executed_action => directed_action action.name = 'digital technical data package work order'}

Table 6 - Mapping table for manufacturing_process_control_documentation UoF (continued)

Application element	AIM element	Source	Rules	Reference path
project_order to pedigree_creation_order (as pedigree_creation_ - status)	PATH		18	directed_action <= executed_action <= action <- action_relationship.relationing_action action_relationship.related_action -> {action action.name = 'pedigree creation order'} action => executed_action => directed_action
project_order to part (as part_status)	PATH		13	directed_action <= executed_action <= action <- action_assignment.assigned_action action_assignment => feature_based_pp_action_assignment feature_based_pp_action_assignment.items[i] -> feature_based_pp_action_item feature_based_pp_action_item = product_definition_formation product_definition_formation
project_order to requisition (as ordered_resource)	PATH			directed_action feature_based_pp_document_item = directed_action feature_based_pp_document_item <- feature_based_pp_document_reference.items[i] feature_based_pp_document_reference <= document_reference document_reference.assigned_document -> document

Table 6 - Mapping table for manufacturing_process_control_documentation UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
project_order to resource_acquisition_order (as resource_acquisition_status)	PATH		18	directed_action <= executed_action <= action <- action_relationship.relationing_action action_relationship.related_action -> {action action.name = 'resource acquisition order'} action => executed_action => directed_action
project_order to shop_work_order (as shop_work_status)	PATH		18	directed_action <= executed_action <= action <- action_relationship.relationing_action action_relationship.related_action -> {action action.name = 'shop work order'} action => executed_action => directed_action
RESOURCE_-ACQUISITION_ORDER	directed_action	41	18	{directed_action <= executed_action <= action action.name = 'resource acquisition order'}
order_id	action_directive.name	41		directed_action directed_action.directive -> action_directive action_directive.name
SHOP_WORK_ORDER	directed_action	41	18	{directed_action <= executed_action <= action action.name = 'shop work order'}
order_id	action_directive.name	41		directed_action directed_action.directive -> action_directive action_directive.name

Table 7 - Mapping table for manufacturing_process_requirement_documents UoF

Application element	AIM element	Source	Rules	Reference path
SPECIFICATION	document_with_class	41		
specification_class	document_with_class.class	41		
specification_description	document.description	41		document_with_class <= document document.description
specification_id	document.id	41		document_with_class <= document document.id
specification to specification_usage_- constraint (as constraint)	PATH			document_with_class <= document <= document_usage_constraint.source document_usage_constraint
SPECIFICATION_- USAGE_CONSTRAINT	document_usage_constraint	41		
class_id	document_usage_constraint.- subject_element_value	41		
element	document_usage_constraint.- subject_element	41		

Table 8 - Mapping table for measurement_limitations UoF

Application element	AIM element	Source	Rules	Reference path
ANGULAR_- DIMENSION_- TOLERANCE	angular_location	47	22	
major_angle	angular_location.angle_selection	47		
mirror_origin #1: The origin element is not mirrored. #2: The origin element is mirrored.	#1: (shape_aspect) #2: (shape_aspect.relationship)	41 41		angular_location <= dimensional_location <= shape_aspect.relationship shape_aspect.relationship.relatiing_shape_aspect -> #1: (shape_aspect) #2: (shape_aspect <- shape_aspect.relationship.relatiing_shape_aspect shape_aspect.relationship {shape_aspect.relationship {shape_aspect.relationship.name = 'mirroring relationship'}})
mirror_termination #1: The termination element is not mirrored. #2: The termination element is mirrored.	#1: (shape_aspect) #2: (shape_aspect.relationship)	41 41		angular_location <= dimensional_location <= shape_aspect.relationship shape_aspect.relationship.related_shape_aspect -> #1: (shape_aspect) #2: (shape_aspect <- shape_aspect.relationship.relatiing_shape_aspect shape_aspect.relationship {shape_aspect.relationship {shape_aspect.relationship.name = 'mirroring relationship'}})
ANGULAR_SIZE_- DIMENSION_- TOLERANCE	angular_size	47		

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
full_or_half	representation_item.name	43	9, 19, 24	angular_size <= dimensional_size dimensional_characteristic = dimensional_size dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item representation_item.name {(representation_item.name = 'full angle') (representation_item.name = 'half angle')}
major_angle	angular_size.angle_selection	47		
ANGULARITY_- TOLERANCE	angularity_tolerance	224	10	angularity_tolerance <= geometric_tolerance_with_datum_reference {geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name = 'angularity'}
segment_size	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41		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]
angularity_tolerance to datum (as geometric_reference)	PATH			angularity_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
CIRCULAR_RUNOUT_- TOLERANCE	circular_runout_tolerance	224	10	circular_runout_tolerance <= geometric_tolerance_with_datum_reference {geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name = 'circular runout'}

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
runout	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41	23	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]
circular_runout_tolerance to datum (as geometric_reference)	PATH			circular_runout_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
CIRCULARITY_- TOLERANCE	geometric_tolerance	47		{geometric_tolerance geometric_tolerance.name = 'circularity'}
COMPOUND_DATUM	datum (see NOTE)	47	23	datum <= shape_aspect <= shape_aspect_relationship.relation_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect => datum_feature
compound_datum to datum_feature (as element)	PATH		23	
CONCENTRICITY_- TOLERANCE	concentricity_tolerance	224	10	concentricity_tolerance <= geometric_tolerance_with_datum_reference {geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name = 'concentricity'}

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
value_qualifier	tolerance_zone_form.name	47		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
concentricity_tolerance to datum (as geometric_reference)	PATH			concentricity_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
CURVED_DIMENSION_-TOLERANCE	dimensional_size	47		{dimensional_size dimensional_size.name = 'curve length'}
CYLINDRICITY_-TOLERANCE	geometric_tolerance	47		{geometric_tolerance geometric_tolerance.name = 'cylindricity'}
DATUM	datum_reference	47		
name	shape_aspect.name	41	23	datum_reference datum_reference.referenced_datum -> datum <= shape_aspect shape_aspect.name
precedence	datum_reference.precedence	47		
DATUM_FEATURE	(datum) ([datum] [datum_feature])	47 47 47	23	{(datum) (datum <= shape_aspect <- shape_aspect.relationship.related_shape_aspect shape_aspect.relationship shape_aspect.relationship.relativing_shape_aspect -> shape_aspect => datum_feature)}
datum_feature to datum_target_set (as datum_representation)	PATH			datum datum.established_by.relationships

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
datum_feature to material_condition_modifier (as modifier)	PATH			datum <- datum_reference.referenced_datum datum_reference => referenced_modified_datum referenced_modified_datum.modifier -> limit_condition
datum_feature to shape_element (as datum_representation)	IDENTICAL MAPPING			
DATUM_TARGET	datum_target	47	23	
identifier	datum_target.target_id	47		
DATUM_TARGET_SET	datum_established_by_relationships	47	23	{datum_established_by_relationships[i] -> shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect => datum_target}
rule_description	shape_aspect_relationship.description	41		datum_established_by_relationships[i] -> shape_aspect_relationship shape_aspect_relationship.description
datum_target_set to datum_target (as target_shape)	PATH			datum_established_by_relationships datum <= shape_aspect <- shape_aspect_relationship.relatng_shape_aspect shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect => datum_target
DIAMETER - DIMENSION - TOLERANCE	dimensional_size	47		{dimensional_size dimensional_size.name = 'diameter'}
DIMENSIONAL - TOLERANCE	shape_dimension_representation	47		

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
dimension_description	(dimensional_size.name) (shape_aspect_relationship.description)	47 41	22	shape_dimension_representation <- dimensional_characteristic_representation.representation dimensional_characteristic_representation.dimension -> dimensional_characteristic (dimensional_characteristic = dimensional_size dimensional_size dimensional_size.name) (dimensional_characteristic = dimensional_location dimensional_location shape_aspect_relationship shape_aspect_relationship.description) shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item
dimension_value	measure_representation_item	45	9, 19, 24	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
significant_digits	precision_qualifier.precision_value	45	9, 19, 24	

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
unit_of_measure	unit		9, 19, 24	shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit measure_with_unit.component -> unit
dimensional_tolerance_to tolerance_value (as limit)	PATH			shape_dimension_representation <= dimensional_characteristic_representation dimensional_characteristic_representation
DISTANCE_ALONG_- CURVE_TOLERANCE	dimensional_location_with_path	47	22	
with_curve_direction	dimensional_location_with_path.path	47		
distance_along_curve_- tolerance_to_shape_aspect (as path)	PATH			dimensional_location_with_path dimensional_location_with_path.path -> shape_aspect
EXTERNALLY_- DEFINED_- TOLERANCE_VALUE	document_usage_constraint			
tolerance_value	document_usage_constraint.subject_- element_value			
externally_defined_- tolerance_value_to specification (as documentation)	PATH			document_usage_constraint document_usage_constraint.source -> document => document_with_class
FLATNESS_TOLERANCE	#1: (geometric_tolerance_with_defined_- unit) #2: (geometric_tolerance)	47		{#1: (geometric_tolerance_with_defined_unit <= geometric_tolerance) #2: (geometric_tolerance) geometric_tolerance.name = 'flatness'}
#1: if segment size is specified #2: if segment size is not specified		47		
segment_size	[measure_with_unit.value_component] [measure_with_unit.component]	41 41		#1: geometric_tolerance_with_defined_unit geometric_tolerance_with_defined_unit.unit_size -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.component]

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
GEOMETRIC_- TOLERANCE	geometric_tolerance	47	22	
geometric_tolerance_- value	measure_with_unit.value_component	41		geometric_tolerance geometric_tolerance.magnitude -> measure_with_unit measure_with_unit.value_component
significant_digits	precision_qualifier.precision_value	45		geometric_tolerance geometric_tolerance.magnitude -> measure_with_unit <- measure_qualification.qualifiers[1] -> value_qualifier value_qualifier = precision_qualifier precision_qualifier precision_qualifier.precision_value
unit_of_measure	unit	41		geometric_tolerance geometric_tolerance.magnitude -> measure_with_unit measure_with_unit.unit_component -> unit
geometric_tolerance to material_condition_- modifier (as modifier_control)	PATH			geometric_tolerance => modified_geometric_tolerance modified_geometric_tolerance.modifier -> limit_condition
geometric_tolerance to shape_aspect (as applied_shape)	PATH			geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> shape_aspect
geometric_tolerance to tolerance_zone (as zone_definition)	PATH			geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone
GEOMETRIC_- TOLERANCE_- PRECEDENCE_- RELATIONSHIP	geometric_tolerance_relationship	47		{geometric_tolerance_relationship geometric_tolerance_relationship.name = 'precedence'}

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_tolerance_- precedence_relationship to geometric_tolerance (as base_shape_tolerance)	PATH			geometric_tolerance_precedence_relationship <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect <= geometric_tolerance.toleranced_shape_aspect geometric_tolerance
geometric_tolerance_- precedence_relationship to geometric_tolerance (as pattern_shape_- tolerance)	PATH			geometric_tolerance_precedence_relationship <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect <= geometric_tolerance.toleranced_shape_aspect geometric_tolerance
LINEAR_PROFILE_- TOLERANCE	linear_profile_tolerance	224	10	linear_profile_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'linear profile'}
linear_profile_tolerance to datum (as geometric_reference)	PATH			linear_profile_tolerance <= geometric_tolerance => geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
linear_profile_tolerance to orientation (as affected_plane)	PATH			<pre> linear_profile_tolerance <= geometric_tolerance <- tolerance_zone.defined_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect.relationship.relation_shape_aspect shape_aspect.relationship.relation_shape_aspect-> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.representation.definition property_definition.representation property_definition.representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item => geometric_representation_item => placement </pre>
LOCATION_- DIMENSION_- TOLERANCE #1: if directed = true #2: if directed = false	#1: (directed_dimensional_location) #2: (dimensional_location)	224 47	22	
directed	IDENTICAL MAPPING			

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
location_dimension_- tolerance to orientation (as plane_and_direction)	PATH		22	<pre>#1: (directed_dimensional_location <= dimensional_location <=) #2: (dimensional_location <=) shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item {representation_item.name = 'orientation'}} representation_item => geometric_representation_item => placement</pre>
LOCATION_- TOLERANCE	dimensional_location	47	22	
location_tolerance to shape_element (as origin_shape)	PATH			<pre>dimensional_location <= shape_aspect_relationship shape_aspect_relationship.relatng_shape_aspect -> shape_aspect</pre>
location_tolerance to shape_element (as termination_shape)	PATH			<pre>dimensional_location <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect</pre>
MATERIAL_- CONDITION_MODIFIER	limit_condition	47		
material_type	IDENTICAL_MAPPING			
PARALLELISM_- TOLERANCE	parallelism_tolerance	224	10	<pre>parallelism_tolerance <= geometric_tolerance_with_datum_reference {geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name = 'parallelism'}</pre>

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
segment_size	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41		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]
parallelism_tolerance to datum (as geometric_reference)	PATH			parallelism_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
parallelism_tolerance to orientation (as affected_plane)	PATH			parallelism_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <= tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <= shape_aspect.relationship.relativing_shape_aspect shape_aspect.relationship shape_aspect.relationship.related_shape_aspect-> shape_aspect shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item {representation_item representation_item.name = 'affected plane'} representation_item => geometric_representation_item => placement

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
PERPENDICULARITY_- TOLERANCE	perpendicularity_tolerance	224	10	perpendicularity_tolerance <= geometric_tolerance_with_datum_reference {geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name = 'perpendicularity'}
segment_size	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41		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]
perpendicularity_- tolerance to datum (as geometric_reference)	PATH			perpendicularity_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
perpendicularity_- tolerance to orientation (as affected_plane)	PATH			<pre> perpendicularity_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect_relationship.relation_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect-> shape_aspect {shape_aspect {shape_aspect {shape_aspect.description = 'affected plane'}} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item {representation_item.name = 'affected plane'}} representation_item => geometric_representation_item => placement </pre>
PLACED_TARGET	placed_datum_target_feature	224	23	<pre> 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')}} </pre>

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
placed_target to orientation (as placement)	PATH		9, 19, 24	<pre> placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement </pre>
PLUS_MINUS_VALUE #1: if the plus_minus_value is selected for tolerance_value #2: if the plus_minus_value is selected for numeric_parameter	#1: (tolerance_value) #2: (qualified_representation_item)	47 45		

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
lower_limit	#1: (tolerance_value.lower_bound) #2: (standard_uncertainty.- uncertainty_value)	47 45		#2: {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
significant_digits	precision_qualifier.precision_value	45		#1: (tolerance_value [tolerance_value.upper_bound ->] [tolerance_value.lower_bound ->] measure_with_unit <- measure_qualification.qualifiers[1] ->) #2: (qualified_representation_item qualified_representation_item.qualifiers[i] ->) value_qualifier value_qualifier = precision_qualifier precision_qualifier precision_qualifier.precision_value
upper_limit	#1: (tolerance_value.upper_bound) #2: (standard_uncertainty.- uncertainty_value)	47 45		#2: {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
POSITION_TOLERANCE	position_tolerance	224	10	position_tolerance <= geometric_tolerance_with_datum_reference {geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name = position'}

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
value_qualifier	tolerance_zone_form.name	47		<pre> 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 </pre>
position_tolerance to datum (as geometric_reference)	PATH			<pre> position_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference </pre>
position_tolerance to orientation (as affected_plane)	PATH			<pre> position_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect_relationship.relation_shape_aspect shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item -> geometric_representation_item => placement </pre>

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
PROJECTION	projected_zone_definition	47		
projection_length	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41		projected_zone_definition projected_zone_definition.projected_length -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
projection to shape_- element (as projection_end)	PATH			projected_zone_definition projected_zone_definition.projection_end -> shape_aspect
RADIAL_DIMENSION_- TOLERANCE	dimensional_size	47		{dimensional_size dimensional_size.name = 'radius'}
SIZE_TOLERANCE	dimensional_size	47		
size_tolerance to shape_- element (as applied_shape)	PATH			dimensional_size dimensional_size.applies_to -> shape_aspect
STRAIGHTNESS_- TOLERANCE #1: if segment size is specified #2: if segment size is not specified	#1: (geometric_tolerance_with_defined_- unit) #2: (geometric_tolerance)	47 47		{#1: (geometric_tolerance_with_defined_unit <= geometric_tolerance) #2: (geometric_tolerance) geometric_tolerance.name = 'straightness'}
segment_size	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41		#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]

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
straightness_tolerance to orientation (as affected_plane)	PATH			<pre> #1: (geometric_tolerance_with_defined_unit <= geometric_tolerance <-) #2: (geometric_tolerance <-) tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect.relationship.relation_shape_aspect shape_aspect.relationship shape_aspect.relationship.related_shape_aspect-> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.representation.definition property_definition.representation property_definition.representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item => geometric_representation_item => placement </pre>
SURFACE_PROFILE_- TOLERANCE	surface_profile_tolerance	224	10	<pre> surface_profile_tolerance <= geometric_tolerance {geometric_tolerance geometric_tolerance.name = 'surface profile'} </pre>
surface_profile_tolerance to datum (as geometric_reference)	PATH			<pre> surface_profile_tolerance <= geometric_tolerance => geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference </pre>

Table 8 - Mapping table for measurement limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
SYMMETRY_- TOLERANCE	symmetry_tolerance	224	10	<pre> symmetry_tolerance <= geometric_tolerance_with_datum_reference { geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name = 'symmetry' } </pre>
symmetry_tolerance to datum (as geometric_reference)	PATH			<pre> symmetry_tolerance <= geometric_tolerance_with_datum_reference { geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference } </pre>
symmetry_tolerance to orientation (as affected_plane)	PATH			<pre> symmetry_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defined_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect_relationship.relation_shape_aspect shape_aspect_relationship.related_shape_aspect -> shape_aspect { shape_aspect shape_aspect.description = 'affected plane' } shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition.representation.definition property_definition.representation property_definition.used_representation -> representation representation.items[i] -> { representation_item representation_item.name = 'affected plane' } representation_item => geometric_representation_item => placement </pre>
TARGET_AREA	datum_target	47	23	

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
target_area to shape_- element (as area_shape)	IDENTICAL MAPPING			
TARGET_CIRCLE	placed_datum_target_feature	224	23	placed_datum_target_feature <= datum_target {datum_target <= shape_aspect shape_aspect.description = 'circle'}
target_diameter	measure_representation_item	45	9, 19, 24	placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <= property_definition.definition property_definition <= property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'target diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
TARGET_LINE	placed_datum_target_feature	224	23	placed_datum_target_feature <= datum_target {datum_target <= shape_aspect shape_aspect.description = 'line'}

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
target_length	measure_representation_item	45	9, 19, 24	<pre> placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'target length'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
TARGET_POINT	placed_datum_target_feature	224	23	<pre> placed_datum_target_feature <= datum_target {datum_target <= shape_aspect shape_aspect.description = 'point'} </pre>
TARGET_RECTANGLE	placed_datum_target_feature	224	23	<pre> placed_datum_target_feature <= datum_target {datum_target <= shape_aspect shape_aspect.description = 'rectangle'} </pre>

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
target_length	measure_representation_item	45	9, 19, 24	<pre> placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} } } </pre>

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
target_width	measure_representation_item	45	9, 19, 24	<pre> placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'target width'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit} </pre>
TOLERANCE_LIMIT	type_qualifier	45		
limit_qualifier	type_qualifier.name	45		
TOLERANCE_RANGE	shape_dimension_representation	47		
lower_range	measure_representation_item	45		<pre> shape_dimension_representation <= shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'lower range'} representation_item => measure_representation_item </pre>

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
significant_digits	precision_qualifier.precision_value	45		<pre> shape_dimension_representation <= representation representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit <- measure_qualification.qualifiers[1] -> value_qualifier = precision_qualifier precision_qualifier precision_qualifier.precision_value </pre>
upper_range	measure_representation_item	45		<pre> shape_dimension_representation <= representation representation.items[i] -> representation_item {representation_item representation_item.name = 'upper range'} representation_item => measure_representation_item </pre>
TOLERANCE_VALUE	dimensional_characteristic_representation	47		
envelope	representation.name	43	9, 19, 24	<pre> dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.name {representation.name = 'envelope tolerance'} </pre>
tolerance_value to externally_defined_tolerance_value (as defined_value)	PATH			<pre> feature_based_pp_document_item = dimensional_characteristic_representation feature_based_pp_document_item <- feature_based_pp_document_reference.items[i] feature_based_pp_document_reference <= document_reference document_reference.assigned_document -> document <- document_usage_constraint.source document_usage_constraint </pre>

Table 8 - Mapping table for measurement_limitations UoF (continued)

Application element	AIM element	Source	Rules	Reference path
tolerance_value to plus_minus_value (as defined_value)	PATH			dimensional_characteristic_representation dimensional_characteristic_representation.dimension -> dimensional_characteristic -> plus_minus_tolerance.tolerance_dimension plus_minus_tolerance plus_minus_tolerance.range -> tolerance_method_definition tolerance_method_definition = tolerance_value tolerance_value
tolerance_value to tolerance_limit (as defined_value)	PATH		9, 19, 24	dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_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
tolerance_value to tolerance_range (as defined_value)	PATH			dimensional_characteristic_representation dimensional_characteristic_representation.representation -> dimensional_shape_dimension_representation
TOLERANCE_ZONE	tolerance_zone	47	23	
common_zone	tolerance_zone.defining_tolerance	47		
form_type	tolerance_zone.form.name	47		tolerance_zone tolerance_zone.form -> tolerance_zone_form tolerance_zone_form.name
tolerance_zone to projection (as extended_shape)	PATH			tolerance_zone <- tolerance_zone_definition.zone tolerance_zone_definition => projected_zone_definition

Table 8 - Mapping table for measurement limitations UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
tolerance_zone to tolerance_zone_definition (as zone_definition)	PATH			tolerance_zone <- tolerance_zone_definition.zone tolerance_zone_definition
TOLERANCE_ZONE_- DEFINITION	tolerance_zone_definition	47		
tolerance_zone_definition to shape_element (as first_element)	PATH			tolerance_zone_definition tolerance_zone_definition.boundaries[i] -> shape_aspect
tolerance_zone_definition to shape_element (as second_element)	PATH			tolerance_zone_definition tolerance_zone_definition.boundaries[i] -> shape_aspect
TOTAL_RUNOUT_- TOLERANCE	total_runout_tolerance	224	10	total_runout_tolerance <= geometric_tolerance_with_datum_reference {geometric_tolerance_with_datum_reference <= geometric_tolerance geometric_tolerance.name = 'total runout'}
runout	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41	23	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]
total_runout_tolerance to datum (as geometric_reference)	PATH			total_runout_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
Application element	AIM element	Source	Rules	Reference path
APPROVAL	approval	41	1, 2, 5, 14, 17, 20	

Table 9 - Mapping table for part_administration_data UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
approval_date	date	41	1, 6	approval <- approval_date.dated_approval approval_date_time approval_date_time_date_time -> date_time_select date_time_select = date date
status	approval_status.name	41	5, 20	approval approval.status -> approval_status approval_status.name {(approval_status.name = 'approved') (approval_status.name = 'not yet approved') (approval_status.name = 'disapproved') (approval_status.name = 'withdrawn')}}
approval to person_in_ - organization (as approval_authority)	PATH		2	approval <- approval_person_organization.authorized_approval approval_person_organization approval_person_organization.person_organization -> person_organization_select person_organization_select = person_and_organization person_and_organization
ORGANIZATION	organization	41		
organization_address	address	41		organization <- organizational_address.organizations[i] organizational_address <= address
organization_id	organization.id	41		
organization_name	organization.name	41		
PERSON	person	41		
person_address	address	41		person <- personal_address.people[i] personal_address <= address
person_id	person.id	41		

Table 9 - Mapping table for part_administration_data UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
person_name	[person.first_name] [person.last_name]	41		
person_phone_number	address.telephone_number	41		person <- personal_address.people[i] personal_address <= address address.telephone_number
PERSON_IN_- ORGANIZATION	person_and_organization	41		
role	person_and_organization_role.name	41		person_and_organization <- person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name
person_in_organization to organization (as company)	PATH			person_and_organization person_and_organization.the_organization -> organization
person_in_organization to person (as employee)	PATH			person_and_organization person_and_organization.the_person -> person

Table 10 - Mapping table for part_model UoF

Application element	AIM element	Source	Rules	Reference path
PART	product_definition_formation	41	13, 14, 16	
part_description	product.description	41	15	product_definition_formation product_definition_formation.of_product -> product product.description
part_id	product.id	41	15	product_definition_formation product_definition_formation.of_product -> product product.id
part_name	product.name	41	15	product_definition_formation product_definition_formation.of_product -> product product.name
part_revision_id	product_definition_formation.id	41		
security_classification	security_classification	41	8, 16, 21	product_definition_formation feature_based_pp_classified_item = product_definition_formation feature_based_pp_classified_item <- feature_based_pp_security_classification_assignment.items[i] feature_based_pp_security_classification_assignment <= security_classification_assignment security_classification_assignment.assigned_security_classification -> security_classification

Table 10 - Mapping table for part model UoF (continued)

Application element	AIM element	Source	Rules	Reference path
part to alternate_material (as alternate_material_definition)	PATH		12	<pre> product_definition_formation <- product_definition.formation product_definition <- product_definition.relationship.relatng_product_definition {product_definition.relationship => product_definition_usage => make_from_usage_option} product_definition_relationship product_definition_relationship.related_product_definition -> product_definition {product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- material_designation.definitions[i] material_designation} [product_definition <- product_definition_relationship.related_product_definition {product_definition_relationship => product_definition_usage => make_from_usage_option} product_definition_relationship product_definition_relationship.relatng_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]} </pre>
part to approval (as manufacture_authorization)	PATH		14	<pre> product_definition_formation feature_based_pp_approved_item = product_definition_formation feature_based_pp_approved_item <- feature_based_pp_approval_assignment.items[i] feature_based_pp_approval_assignment <= approval_assignment approval_assignment.assigned_approval -> approval </pre>

Table 10 - Mapping table for part_model UoF (continued)

Application element	AIM element	Source	Rules	Reference path
part to material (as material_definition)	PATH		12	<pre> product_definition_formation <- product_definition.formation product_definition <- product_definition.relationship.relation_product_definition {product_definition.relationship => product_definition_usage => make_from_usage_option} product_definition_relationship product_definition.related_product_definition -> product_definition {product_definition characterized_product_definition = product_definition characterized_product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition => material_designation} </pre>
part to ordered_part (as quantity_ordered)	PATH		13	<pre> product_definition_formation feature_based_pp_ordered_item = product_definition_formation feature_based_pp_ordered_item <- ordered_part.items[i] ordered_part </pre>
part to organization (as manufactured_by_ - organization)	PATH			<pre> product_definition_formation feature_based_pp_organization_item = product_definition_formation feature_based_pp_organization_item <- feature_based_pp_organization_assignment.items[i] feature_based_pp_organization_assignment <= {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'manufacturer'} organization_assignment organization_assignment.organization -> organization </pre>

Table 10 - Mapping table for part model UoF (continued)

Application element	AIM element	Source	Rules	Reference path
part to organization (as owned_by_- organization)	PATH			<pre> product_definition_formation feature_based_pp_organization_item = product_definition_formation feature_based_pp_organization_item <- feature_based_pp_organization_assignment.items[i] feature_based_pp_organization_assignment <= {organization_assignment organization_assignment.role -> organization_role organization_role.name = 'owner'} organization_assignment organization_assignment.organization -> organization </pre>
part to person_in_- organization (as manufactured_by_- person)	PATH			<pre> product_definition_formation feature_based_pp_person_and_organization_item = product_definition_formation feature_based_pp_person_and_organization_item <- feature_based_pp_person_and_organization_assignment.items[i] feature_based_pp_person_and_organization_assignment <= {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'manufacturer'} person_and_organization_assignment person_and_organization_assignment.person_and_organization -> person_and_organization </pre>
part to person_in_- organization (as owned_by_person)	PATH			<pre> product_definition_formation feature_based_pp_person_and_organization_item = product_definition_formation feature_based_pp_person_and_organization_item <- feature_based_pp_person_and_organization_assignment.items[i] feature_based_pp_person_and_organization_assignment <= {person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'owner'} person_and_organization_assignment person_and_organization_assignment.person_and_organization -> person_and_organization </pre>

Table 10 - Mapping table for part_model UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
part to property (as property - characteristics)	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
part to shape (as physical_form)	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

Table 11 - Mapping table for requisitions UoF

Application element	AIM element	Source	Rules	Reference path
CUTTING_TOOL_- REQUISITION	document	41		{document document.kind -> document_type document_type.product_data_type = 'cutting tool requisition'}
DEDICATED_FIXTURE_- REQUISITION	document	41		{document document.kind -> document_type document_type.product_data_type = 'dedicated fixture requisition'}
INDIRECT_STOCK_- REQUISITION	document	41		{document document.kind -> document_type document_type.product_data_type = 'indirect stock requisition'}
MACHINE_- REQUISITION	document	41		{document document.kind -> document_type document_type.product_data_type = 'machine requisition'}
MATERIAL_- REQUISITION	document	41		{document document.kind -> document_type document_type.product_data_type = 'material requisition'}
MODULAR_FIXTURE_- REQUISITION	document	41		{document document.kind -> document_type document_type.product_data_type = 'modular fixture requisition'}
REQUISITION	document	41		
quantity_ordered	document.name	41		

Table 11 - Mapping table for requisitions UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
required_delivery_date	date	41	6	<pre> document=> document_with_class feature_based_pp_dated_item = document_with_class feature_based_pp_dated_item <- feature_based_pp_date_assignment.items[i] feature_based_pp_date_assignment <= {date_assignment {date_assignment.role -> date_role date_role.name = 'required delivery date'}} date_assignment.assigned_date -> date </pre>
requisition_date	date	41	6	<pre> document=> document_with_class feature_based_pp_dated_item = document_with_class feature_based_pp_dated_item <- feature_based_pp_date_assignment.items[i] feature_based_pp_date_assignment <= {date_assignment {date_assignment.role -> date_role date_role.name = 'requisition date'}} date_assignment date_assignment.assigned_date -> date </pre>
requisition_description	document.description	41		
requisition_number	document.id	41		

Table 12 - Mapping table for shape_representation_for_machining_UoF

Application element	AIM element	Source	Rules	Reference path
BASE_SHAPE	product_definition_shape	41		{product_definition_shape <= property_definition property_definition.definition -> characterized_definition <- material_designation_definitions[i]}
BLOCK_BASE_SHAPE	block_shape_representation	224	9, 19, 24	block_shape_representation <= shape_representation_with_parameters <= shape_representation
block_base_shape to numeric_parameter (as height)	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}}
block_base_shape to numeric_parameter (as width)	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}}
B-REP_MODEL	manifold_solid_brep	42		
B-REP_MODEL_-ELEMENT	(geometric_representation_item) (topological_representation_item)	224 224		

Table 12 - Mapping table for shape_representation_for_machining UoF (continued)

Application element	AIM element	Source	Rules	Reference path
B-rep_model_element to B-rep_model (as element)	PATH			<pre> (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 </pre>
B-REP_SHAPE_- ASPECT_- REPRESENTATION	advanced_brep_shape_representation	224	9, 19, 24	advanced_brep_shape_representation <= shape_representation
B-rep_shape_aspect_- representation to B-rep_- model (as shape_definition)	PATH			<pre> advanced_brep_shape_representation <= shape_representation <= representation.items[i] -> representation_item => geometric_representation_item => solid_model => manifold_solid_brep </pre>
B-REP_SHAPE_- REPRESENTATION	advanced_brep_shape_representation	224	9, 19, 24	advanced_brep_shape_representation <= shape_representation
B-rep_shape_- representation to B-rep_- model (as shape_definition)	PATH			<pre> advanced_brep_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => solid_model => manifold_solid_brep </pre>
CYLINDRICAL_BASE_- SHAPE	cylindrical_shape_representation	224	9, 19, 24	cylindrical_shape_representation <= shape_representation_with_parameters <= shape_representation

Table 12 - Mapping table for shape_representation_for_machining UoF (continued)

Application element	AIM element	Source	Rules	Reference path
cylindrical_base_shape to numeric_parameter (as diameter)	PATH			cylindrical_shape_representation <= shape_representation_with_parameters <= shape_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'diameter'}
DIRECTION_ELEMENT	direction_shape_representation	224	19	representation_item => measure_representation_item <= {measure_representation_item <= measure_with_unit => length_measure_with_unit}
EXPLICIT_BASE_- SHAPE_- REPRESENTATION	shape_representation	411	9	direction_shape_representation <= representation
explicit_base_shape_- representation to B-rep_- shape_representation (as B-rep_form)	IDENTICAL MAPPING			
FACE_SHAPE_- ELEMENT	face_shape_representation	224	19	face_shape_representation <= representation
IMPLICIT_BASE_- SHAPE_- REPRESENTATION	shape_representation_with_parameters	224	9, 19, 24	shape_representation_with_parameters <= shape_representation
implicit_base_shape_- representation to numeric_parameter (as base_shape_length)	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}

Table 12 - Mapping table for shape_representation_for_machining UoF (continued)

Application element	AIM element	Source	Rules	Reference path
implicit_base_shape_- representation to orientation (as placement)	PATH			<pre> shape_representation_with_parameters <= shape_representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement </pre>
LOCATION_ELEMENT	location_shape_representation	224	19	location_shape_representation <= representation
NGON_BASE_SHAPE	ngon_shape_representation	224	9, 19, 24	<pre> ngon_shape_representation <= shape_representation_with_parameters <= shape_representation representation.items[i] -> {representation_item representation_item.name = 'circumscribed diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit} </pre>
ngon_base_shape to numeric_parameter (as circumscribed_- diameter)	PATH			
ngon_base_shape to numeric_parameter (as corner_radius)	PATH			<pre> ngon_shape_representation <= shape_representation_with_parameters <= shape_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} </pre>

Table 12 - Mapping table for shape_representation_for_machining UoF (continued)

Application element	AIM element	Source	Rules	Reference path
ngon_base_shape to numeric_parameter (as number_of_sides)	PATH			<pre> 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} </pre>
ORIENTATION		42		
axis	<pre> placement (axis1_placement.axis) ((axis2_placement_3d.axis) (axis2_placement_3d.ref_direction)) </pre>	42 42 42		<pre> placement => (axis1_placement axis1_placement.axis) (axis2_placement_3d (axis2_placement_3d.axis) (axis2_placement_3d.ref_direction)) </pre>
location	placement.location	42		
PATH_ELEMENT	path_shape_representation	224	19	path_shape_representation <= representation
SHAPE	product_definition_shape	41		

Table 12 - Mapping table for shape_representation_for_machining UoF (continued)

Application element	AIM element	Source	Rules	Reference path
shape to base_shape_- (as base_shape_- definition)	PATH		12	<pre> 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.relatng_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 product_definition_shape <= property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> representation => shape_representation => advanced_brep_shape_representation product_definition_shape <- shape_aspect.of_shape shape_aspect </pre>
shape to B-rep_shape_- representation (as B-rep_form)	PATH		9, 19, 24	<pre> product_definition_shape <= property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> representation => shape_representation => advanced_brep_shape_representation </pre>
shape to shape_aspect (as element)	PATH			<pre> product_definition_shape <- shape_aspect.of_shape shape_aspect </pre>
SHAPE_ASPECT	shape_aspect	41	23	

Table 12 - Mapping table for shape_representation_for_machining UoF (concluded)

Application element	AIM element	Source	Rules	Reference path
shape_aspect to B-rep_- model_element (as B-rep_shape)	PATH			<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => (geometric_representation_item) (topological_representation_item) </pre>
shape_aspect to B-rep_- shape_aspect_- representation (as B-rep_form)	PATH		9, 19, 24	<pre> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => shape_representation => advanced_brep_shape_representation </pre>
shape_aspect to shape_- element (as element)	IDENTICAL MAPPING			
SHAPE_ELEMENT	shape_aspect	41	23	<pre> shape_aspect {shape_aspect shape_aspect.product.definitional = TRUE} </pre>

The following rules are referenced in the preceding tables:

- 1) approval_requires_approval_date_time
- 2) approval_requires_approval_person_organization
- 3) dependent_instantiable_action_request_status
- 4) dependent_instantiable_action_status
- 5) dependent_instantiable_approval_status
- 6) dependent_instantiable_date
- 7) dependent_instantiable_named_unit
- 8) dependent_instantiable_security_classification_level
- 9) dependent_instantiable_shape_representation
- 10) geometric_tolerance_subtype_exclusiveness
- 11) machining_feature_life_cycle
- 12) material_is_specified_for_part
- 13) part_requires_project_order
- 14) part_to_approval
- 15) product_requires_version
- 16) product_definition_formation_requires_security_classification
- 17) project_order_requires_approval
- 18) project_order_tracking_relationships
- 19) representation_subtype_exclusiveness
- 20) restrict_approval_status
- 21) restrict_security_classification_level
- 22) shape_aspect_relationship_subtype_exclusiveness
- 23) shape_aspect_subtype_exclusiveness
- 24) shape_representation_subtype_exclusiveness
- 25) subtype_mandatory_characterized_object
- 26) transition_feature_life_cycle
- 27) transition_feature_on_part_boundary

Note - The datum that represents a compound_datum uses at least four shape_aspect_relationships. At least two shape_aspect_relationship reference the datum associated to the compound_datum. Each datum referenced by the compound_datum has a shape_aspect_relationship to a datum_feature. The compound_datum has a shape_aspect_relationship that references the same datum_features.

5.2 AIM EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources and the AICs and contains the types, entity specializations, rules, and functions that are specific to this part of ISO 10303. This clause also specifies modifications to the textual material for constructs that are imported from the integrated resources and the AICs. The definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. Requirements stated in the integrated resources which refer to such items and subtypes apply exclusively to those items which are imported into the AIM.

*)

SCHEMA feature_based_process_planning;

```
USE FROM action_schema -- ISO 10303-41
    (action_directive,
     action_method,
     action_relationship,
     action_request_solution,
     action_request_status,
     action_status,
     directed_action,
     versioned_action_request);

USE FROM application_context_schema -- ISO 10303-41
    (application_context,
     application_protocol_definition);

USE FROM approval_schema -- ISO 10303-41
    (approval,
     approval_date_time,
     approval_person_organization,
     approval_relationship,
     approval_status);

USE FROM date_time_schema -- ISO 10303-41
    (date,
     calendar_date,
     ordinal_date,
     week_of_year_and_day_date);

USE FROM document_schema -- ISO 10303-41
    (document_with_class,
     document_usage_constraint,
     document_relationship);

USE FROM external_reference_schema -- ISO 10303-41
    (externally_defined_item);

USE FROM geometric_model_schema -- ISO 10303-42
    (brep_with_voids,
     manifold_solid_brep);

USE FROM geometry_schema -- ISO 10303-42
    (axis2_placement_2d,
     axis2_placement_3d,
     bezier_curve,
     bezier_surface,
     b_spline_curve_with_knots,
     b_spline_surface_with_knots,
     cartesian_point,
     cartesian_transformation_operator_3d,
     circle,
     conical_surface,
     cylindrical_surface,
     degenerate_toroidal_surface,
```

direction,
ellipse,
geometric_representation_context,
hyperbola,
line,
parabola,
pcurve,
plane,
point,
polyline,
quasi_uniform_curve,
quasi_uniform_surface,
rational_b_spline_curve,
rational_b_spline_surface,
spherical_surface,
surface_curve,
surface_of_linear_extrusion,
surface_of_revolution,
swept_surface,
toroidal_surface,
uniform_curve,
uniform_surface,
vector);

USE FROM management_resources_schema -- ISO 10303-41
(action_assignment,
action_request_assignment,
approval_assignment,
date_assignment,
document_reference,
organization_assignment,
person_and_organization_assignment,
security_classification_assignment);

USE FROM material_property_representation_schema
(material_property_representation);

USE FROM material_property_definition_schema -- ISO 10303-45
(material_designation,
material_property,
property_definition_relationship);

USE FROM measure_schema -- ISO 10303-41
(context_dependent_measure,
context_dependent_unit,
conversion_based_unit,
count_measure,
derived_unit,
descriptive_measure,
global_unit_assigned_context,
length_measure_with_unit,
length_unit,
measure_with_unit,
named_unit,
parameter_value,


```

    plane_angle_measure_with_unit,
    plane_angle_unit,
    positive_plane_angle_measure,
    ratio_measure_with_unit,
    ratio_unit,
    si_unit,
    solid_angle_measure_with_unit,
    solid_angle_unit);

USE FROM person_organization_schema                                -- ISO 10303-41
(organization_relationship,
 organizational_address,
 organizational_project,
 person_and_organization,
 personal_address);

USE FROM product_definition_schema                                -- ISO 10303-41
(product,
 product_definition,
 product_definition_formation,
 product_definition_with_associated_documents);

USE FROM product_property_definition_schema                        -- ISO 10303-41
(characterized_object,
 product_definition_shape,
 property_definition,
 shape_aspect,
 shape_aspect_relationship);

USE FROM product_property_representation_schema                    -- ISO 10303-41
(property_definition_representation,
 shape_definition_representation,
 shape_representation);

USE FROM product_structure_schema                                  -- ISO 10303-44
(make_from_usage_option);

USE FROM qualified_measure_schema                                  -- ISO 10303-45
(descriptive_representation_item,
 expanded_uncertainty,
 measure_representation_item,
 measure_qualification,
 precision_qualifier,
 qualified_representation_item,
 qualitative_uncertainty,
 standard_uncertainty,
 type_qualifier,
 uncertainty_qualifier);

USE FROM representation_schema                                    -- ISO 10303-43
(definitional_representation,
 global_uncertainty_assigned_context,
 mapped_item,
 parametric_representation_context,

```

```
        representation,  
        representation_item);  
  
USE FROM security_classification_schema -- ISO 10303-41  
    (security_classification,  
     security_classification_level);  
  
USE FROM shape_aspect_definition_schema -- ISO 10303-47  
    (composite_shape_aspect,  
     datum_feature,  
     datum_reference,  
     datum_target,  
     referenced_modified_datum,  
     datum);  
  
USE FROM shape_dimension_schema -- ISO 10303-47  
    (angular_location,  
     angular_size,  
     dimensional_characteristic_representation,  
     dimensional_location,  
     dimensional_location_with_path,  
     dimensional_size,  
     dimensional_size_with_path,  
     shape_dimension_representation);  
  
USE FROM shape_tolerance_schema -- ISO 10303-47  
    (geometric_tolerance,  
     geometric_tolerance_relationship,  
     geometric_tolerance_with_datum_reference,  
     geometric_tolerance_with_defined_unit,  
     limits_and_fits,  
     modified_geometric_tolerance,  
     plus_minus_tolerance,  
     projected_zone_definition,  
     runout_zone_definition,  
     runout_zone_orientation,  
     tolerance_value,  
     tolerance_zone,  
     tolerance_zone_definition);  
  
USE FROM support_resource_schema -- ISO 10303-41  
    (identifier,  
     label,  
     text);  
  
USE FROM topology_schema -- ISO 10303-42  
    (closed_shell,  
     connected_face_set,  
     oriented_closed_shell,  
     edge_curve,  
     edge_loop,  
     face_bound,  
     face_outer_bound,  
     face_surface,  
     path,
```

```

vertex_loop,
vertex_point);
(*

```

NOTE - The schemas referenced above can be found in the following parts of ISO 10303:

action_schema	ISO 10303-41
application_context_schema	ISO 10303-41
approval_schema	ISO 10303-41
date_time_schema	ISO 10303-41
document_schema	ISO 10303-41
external_reference_schema	ISO 10303-41
geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
management_resources_schema	ISO 10303-41
material_property_definition_schema	ISO 10303-45
measure_schema	ISO 10303-41
person_organization_schema	ISO 10303-41
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
product_structure_schema	ISO 10303-44
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
security_classification_schema	ISO 10303-41
shape_aspect_definition_schema	ISO 10303-47
shape_dimension_schema	ISO 10303-47
shape_tolerance_schema	ISO 10303-47
support_resource_schema	ISO 10303-41
topology_schema	ISO 10303-42

5.2.1 Fundamental concepts and assumptions

ISO 10303-224 is designed to be used in the representation and exchange of data to a process planning function for the manufacture of mechanical parts. This schema embodies several fundamental concepts that are manifested in the information model. The process planning function in an organization can be assisted a great deal by identifying machining oriented part shape features so that the process planner can more readily identify machining equipment, tools, and processes to manufacture a part. The fundamental concept of this schema is to enable the representation and exchange of this information about a part.

5.2.1.1 Hybrid approach to feature representation

Two ways to represent the shape of part features are specified in ISO 10303-224, explicit shape representation and implicit shape representation. The explicit representation of shape is accomplished by specifying a boundary representation solid model that defines the shape of the part feature. The implicit representation of shape is accomplished by specifying the shape defining parameters that are specified in the model for each type of part feature. The implicit representation is always dependent upon a base shape representation to which the features are applied. The schema provides the capability to use a hybrid approach to represent the shape of a part feature. This capability allows for the specification of an explicit

representation, an implicit representation, or both an explicit and implicit representation for any of the part features defined for a part. The base shape representation may be specified either implicitly or explicitly as well. The explicit representation may be used in conjunction with the implicit representation to verify the interpretation of the implicit representation or to share data with another application that does not rely on the implicit representation of part features.

5.2.1.2 Use of the integrated resources for feature definition

A part's machining features are elements or aspects of the shape of a part. As such, the **shape_aspect** entity from ISO 10303-41 is specialized in this schema to specify the concepts of the different types of features. Three basic types of features are specified as **machining_feature**, **replicate_feature**, and **transition_feature**. These basic types are again specialized into the individual types of features. Two types of parameters are used to specify the implicit shape representation of a feature, a numeric and a descriptive parameter. The **measure_representation_item** is used to specify the numeric parameter providing for both a real value of length and angle or an integer value of count and the units by which each value is specified. The **descriptive_representation_item** is used to specify the descriptive parameter. The element of the implicit shape that is being defined for each feature is specified by the name of the parameter. Constraints are defined in each specialization to specify the required parameters and the value types for the implicit representation of each feature. The **shape_representation_with_parameters** entity is used to specify the implicit representation of the shape of the feature.

Additionally, there are components of the part's features defined that are also elements of the shape of the part. As such, they are also specified using specializations of the **shape_aspect** entity. Unlike the feature representations, the feature components shall always have an implicit representation specified.

EXAMPLE - A profile to be swept or a taper definition are types of feature components.

The feature components define parts of the individual features and are required to be used in certain feature definitions. The feature components are related to the individual part's features through a specialization of the **shape_aspect_relationship** called the **feature_component_relationship**. Every **feature_component_relationship** specifies a component that has been defined for a particular feature.

There is also a specialization of the **shape_aspect_relationship** defined in the schema called **shape_defining_relationship**. This entity is used to specify other elements of the shape of the part (**shape_aspects**) that participate in the implicit representation of the shape of the feature. These elements all have a geometric or topological representation that consists of a single **representation_item**.

EXAMPLE - A location or a path are types of shape elements that are shape defining.

5.2.2 Feature based process planning types

5.2.2.1 feature_based_pp_action_item

A **feature_based_pp_action_item** identifies a **product_definition_formation** that is being tracked by an **action_directive** for a project.

EXPRESS specification:

```
*)  
TYPE feature_based_pp_action_item = SELECT  
  (product_definition_formation);  
END_TYPE;  
(*
```

5.2.2.2 feature_based_pp_action_request_item

A **feature_based_pp_action_request_item** identifies a **product_definition_formation** for which a design exception has been identified.

EXPRESS specification:

```
*)  
TYPE feature_based_pp_action_request_item = SELECT  
  (product_definition_formation);  
END_TYPE;  
(*
```

5.2.2.3 feature_based_pp_approved_item

A **feature_based_pp_approved_item** identifies a **directed_action** or **product_definition_formation** to which an **approval** may be assigned.

EXPRESS specification:

```
*)  
TYPE feature_based_pp_approved_item = SELECT  
  (directed_action,  
   product_definition_formation);  
END_TYPE;  
(*
```

5.2.2.4 feature_based_pp_classified_item

A **feature_based_pp_classified_item** identifies a **product_definition_formation** to which a **security_classification** may be assigned.

EXPRESS specification:

```
*)  
TYPE feature_based_pp_classified_item = SELECT  
  (product_definition_formation);  
END_TYPE;  
(*
```

5.2.2.5 feature_based_pp_dated_item

A **feature_based_pp_dated_item** identifies a **directed_action**, **document_with_class**, or **versioned_action_request** to which a **date** may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_dated_item = SELECT
  (directed_action,
   document_with_class,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.6 feature_based_pp_document_item

A **feature_based_pp_document_item** identifies an **action_method**, **externally_defined_feature_definition**, **directed_action**, **dimensional_characteristic_representation**, or **property_definition** to which a referenced document may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_document_item = SELECT
  (action_method,
   externally_defined_feature_definition,
   directed_action,
   dimensional_characteristic_representation,
   property_definition);
END_TYPE;
(*
```

5.2.2.7 feature_based_pp_ordered_item

A **feature_based_pp_ordered_item** identifies the **product_definition_formation** that is being ordered by a customer.

EXPRESS specification:

```
*)
TYPE feature_based_pp_ordered_item = SELECT
  (product_definition_formation);
END_TYPE;
(*
```

5.2.2.8 feature_based_pp_organization_item

A **feature_based_pp_organization_item** identifies an **action_directive** or **product_definition_formation** to which an **organization** may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_organization_item = SELECT
  (action_directive,
   product_definition_formation);
END_TYPE;
(*
```

5.2.2.9 feature_based_pp_person_and_organization_item

A **feature_based_pp_person_and_organization_item** identifies an **action_directive** or **product_definition_formation** to which a **person_and_organization** may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_person_and_organization_item = SELECT
  (action_directive,
   product_definition_formation);
END_TYPE;
(*
```

5.2.3 Feature based process planning entities

5.2.3.1 Feature based process planning entity definitions

5.2.3.1.1 advanced_brep_shape_representation

The **advanced_brep_shape_representation** is a subtype of **shape_representation** in which the **representation_items** are specialisations of **manifold_solid_brep** entities. These differ from the more general B-rep in having only explicit geometric forms for their faces and edges. The face geometry is restricted to elementary surfaces, swept surfaces or B-spline surfaces.

EXPRESS specification:

```
*)
ENTITY advanced_brep_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1: SIZEOF (QUERY (it <* SELF.items |
    NOT (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP',
```

```

        'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.AXIS2_PLACEMENT_3D'] *
        TYPEOF(it)) = 1))) = 0;
WR2: SIZEOF (QUERY (it <* SELF.items |
        SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP',
        'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM'] * TYPEOF(it)) = 1)) >= 1;

WR3: SIZEOF (QUERY (msb <* QUERY (it <* SELF.items |
        'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) |
        NOT (SIZEOF (QUERY (csh <* msb.shells(msb) |
        NOT (SIZEOF (QUERY (fcs <* csh.cfs_faces |
        NOT ('FEATURE_BASED_PROCESS_PLANNING.ADVANCED_FACE' IN
        TYPEOF(fcs)))) = 0))) = 0))) = 0;
WR4: SIZEOF (QUERY (msb <* QUERY (it <* items |
        'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) |
        'FEATURE_BASED_PROCESS_PLANNING.ORIENTED_CLOSED_SHELL' IN
        TYPEOF (msb.outer))) = 0;
WR5: SIZEOF (QUERY (brv <* QUERY (it <* items |
        'FEATURE_BASED_PROCESS_PLANNING.BREP_WITH_VOIDS' IN TYPEOF(it)) |
        NOT (SIZEOF (QUERY (csh <* brv.voids |
        csh.orientation)) = 0))) = 0;
WR6: SIZEOF (QUERY (mi <* QUERY (it <* items |
        'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM' IN TYPEOF(it)) |
        NOT ('FEATURE_BASED_PROCESS_PLANNING.' +
        'ADVANCED_BREP_SHAPE_REPRESENTATION' IN
        TYPEOF (mi.mapping_source.mapped_representation)))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **items** attribute of the **representation** supertype may contain **manifold_solid_breps**, **mapped_items** and **axis2_placement_3ds** only. A **faceted_brep** is excluded from the **items** SET since this would be of type **faceted_brep** and of type **manifold_solid_brep**.

WR2: At least one **representation_item** in the **items** set shall be a **manifold_solid_brep** entity or a **mapped_item** (see also WR6).

WR3: For each **manifold_solid_brep** in the **items** set each **face** shall be an **advanced_face**. This entity ensures that the following conditions are met:

- a) Each **face** is a **face_surface**;
- b) Each **face_surface** has its geometry defined by an **elementary_surface**, **swept_surface** or a **b_spline_surface**;
- c) The **edges** used to define the boundaries of the **face** shall all reference an **edge_curve**;
- d) Each **curve** used to define the geometry of the **faces** and face bounds shall be either a **conic**, or a **line** or a **polyline** or a **b_spline_curve**;

e) The **edges** used to define the face boundaries shall all be trimmed by vertices of type **vertex-point**;

f) No **loop** used to define a **face_bound** shall be of the oriented subtype.

WR4: For each **manifold_solid_brep** in the items set the **outer_shell** attribute shall not be of the oriented subtype.

WR5: If a **brep_with_voids** is included in the **items** set then each **shell** in the **voids** set shall be an **oriented_closed_shell** with orientation value FALSE.

WR6: If a **mapped_item** is included in the **items** set then the **mapped_representation** of the **mapping_source** attribute shall be an **advanced_brep_shape_representation**.

NOTE - If a **cartesian_transformation_operator_3d** is included as **mapped_item.mapping_target** with an **axis2_placement_3d** corresponding to the original coordinate system as **mapped_representation.mapping_origin** then the resulting **mapped_item** is a transformed copy of the **advanced_brep_shape_representation**. The precise definition of the transformation, including translation, rotation, scaling and, if appropriate, mirroring, is given by the transformation operator.

5.2.3.1.2 advanced_face

An **advanced_face** is a special subtype of **face_surface** which has additional constraints to ensure that the geometry is directly and completely defined. The **advanced_face** is the top level entity which is used to formulate the precise meaning of a topologically bounded surface corresponding to the scope of this part of ISO 10303.

EXPRESS specification:

*)

```
ENTITY advanced_face
  SUBTYPE OF (face_surface);
WHERE
  WR1 : SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.ELEMENTARY_SURFACE',
    'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_SURFACE',
    'FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE' ] *
    TYPEOF (face_geometry)) = 1;
  WR2 : SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* SELF.bounds |
    'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN
    TYPEOF (bnds.bound)) |
    NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
    NOT('FEATURE_BASED_PROCESS_PLANNING.EDGE_CURVE' IN
    TYPEOF(oe.edge_element)))) = 0))) = 0;
  WR3 : SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* SELF.bounds |
    'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN
    TYPEOF (bnds.bound)) |
    NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
    NOT (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.LINE',
```

```

'FEATURE_BASED_PROCESS_PLANNING.CONIC',
'FEATURE_BASED_PROCESS_PLANNING.POLYLINE',
'FEATURE_BASED_PROCESS_PLANNING.SURFACE_CURVE',
'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_CURVE'] *
TYPEOF (oe.edge_element\edge_curve.edge_geometry)) = 1))) = 0))) = 0;
WR4 : SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* SELF.bounds |
'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN
TYPEOF (bnds.bound)) |
NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
NOT (((('FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN
TYPEOF (oe.edge_start)) AND
('FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN
TYPEOF (oe.edge_start\vertex_point.vertex_geometry)))) AND
(('FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN
TYPEOF (oe.edge_end)) AND
('FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN
TYPEOF (oe.edge_end\vertex_point.vertex_geometry)))))) = 0))) = 0;
WR5 : SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* SELF.bounds |
'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN
TYPEOF (bnds.bound)) |
'FEATURE_BASED_PROCESS_PLANNING.ORIENTED_PATH' IN
TYPEOF (elp_fbnds.bound))) = 0;
WR6 : (NOT ('FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE' IN
TYPEOF (face_geometry))) OR
(SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.LINE',
'FEATURE_BASED_PROCESS_PLANNING.CONIC',
'FEATURE_BASED_PROCESS_PLANNING.POLYLINE',
'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_CURVE'] *
TYPEOF (face_geometry\swept_surface.swept_curve)) = 1);
WR7 : SIZEOF (QUERY (vlp_fbnds <* QUERY (bnds <* SELF.bounds |
'FEATURE_BASED_PROCESS_PLANNING.VERTEX_LOOP' IN
TYPEOF (bnds.bound)) |
NOT (('FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN
TYPEOF (vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex)) AND
('FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN
TYPEOF (vlp_fbnds\face_bound.bound\vertex_loop.
loop_vertex\vertex_point.vertex_geometry)))))) = 0;
WR8 : SIZEOF (QUERY (bnd <* SELF.bounds |
NOT (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP',
'FEATURE_BASED_PROCESS_PLANNING.VERTEX_LOOP'] *
TYPEOF (bnd.bound)) = 1))) = 0;
WR9 : SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* SELF.bounds |
'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN
TYPEOF (bnds.bound)) |
NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
('FEATURE_BASED_PROCESS_PLANNING.SURFACE_CURVE' IN
TYPEOF (oe.edge_element\edge_curve.edge_geometry)) AND
(NOT (SIZEOF (QUERY (sc_ag <*
oe.edge_element\edge_curve.edge_geometry\
surface_curve.associated_geometry |
NOT ('FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN
TYPEOF (sc_ag)))) = 0)))) = 0))) = 0;
WR10 : ((NOT ('FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE' IN
TYPEOF (face_geometry))) OR
((NOT ('FEATURE_BASED_PROCESS_PLANNING.POLYLINE' IN

```

```

        TYPEOF(face_geometry\swept_surface.swept_curve))) OR
        (SIZEOF(face_geometry\swept_surface.swept_curve\polyline.points)
        < 3))) AND
        (SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* SELF.bounds |
        'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN
        TYPEOF (bnds.bound))) |
        NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
        ('FEATURE_BASED_PROCESS_PLANNING.POLYLINE' IN
        TYPEOF(oe.edge_element\edge_curve.edge_geometry)) AND
        (NOT (SIZEOF (oe.edge_element\edge_curve.edge_geometry\
        polyline.points) < 3)))) = 0))) = 0);
    END_ENTITY;
    (*

```

Formal propositions:

WR1: The geometry used in the definition of the face shall be restricted. The face geometry shall be an **elementary_surface**, **swept_surface**, or **b_spline_surface**.

WR2: The geometry of all bounding edges of the face shall be fully defined as **edge_curves**.

WR3: The types of curve used to define the geometry of edges shall be restricted to **lines**, **conics**, **polylines**, **surface_curves**, or **b_spline_curves**.

WR4: All vertices used in the face definition shall be of type **vertex_point** with geometry defined by a **cartesian_point**.

WR5: The use of oriented paths in the definition of the **edge_loops** of the **advanced_face** is prohibited.

WR6: If the face geometry is of type **swept_surface** then the **swept_curve** used in the definition shall be of type **line**, **conic**, **polyline**, or **b_spline_curve**.

WR7: For any **vertex_loop** used to bound the face the **loop_vertex** shall be of type **vertex_point** and the geometry shall be defined by a **cartesian_point**.

WR8: The face bounds shall be defined by either **edge_loops** or **vertex_loops**.

WR9: If a **surface_curve** is used as part of a face bound then the **associated_geometry** attribute shall reference **pcurves** not **surfaces**.

WR10: If a **polyline** is used either to define a **swept_surface** or as part of a face bound, it shall contain at least three points.

5.2.3.1.3 angularity_tolerance

An **angularity_tolerance** is a type of **geometric_tolerance_with_datum_reference** which represents the maximum variation of a considered **shape_aspect**. See ARM definition for **Angularity_tolerance** in paragraph 4.2.5 for more information.

EXPRESS specification:

```

*)
ENTITY angularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'angularity';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system)
        = 1;
END_ENTITY;
(*)

```

Formal propositions:

WR1: The **name** for the **angularity_tolerance** shall be 'angularity'.

WR2: There shall be exactly one **datum_reference** in the **datum_system** set for the **angularity_tolerance**.

5.2.3.1.4 applied_area

An **applied_area** is a type of **shape_aspect** that is the representation of the bounded enclosed area used in the definition of the **thread** and **turned_knurl feature_definitions**. See ARM definition for Partial_area_-definition in paragraph 4.2.124 for more information.

EXPRESS specification:

```

*)
ENTITY applied_area
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,

```

```

'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (
['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'length')) = 1))) = 0))) = 0;
WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN
TYPEOF (it)) AND (it.name = 'orientation')) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **applied_area** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **applied_area** shall have its implicit representation specified by exactly one **shape-representation_with_parameters**.

WR3: The **applied_area** shall have an implicit representation that contains exactly two **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of the **applied_area** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

WR5: Exactly one **representation_item** used for the implicit representation of the **applied_area** shall be of type **placement** with a **name** of 'orientation'.

5.2.3.1.5 block_shape_representation

A **block_shape_representation** specifies the representation of a shape that is a rectangular volume defined as a rectangular area of a defined length. The enclosed area is defined by four straight sides with opposite sides equal in length. See ARM definition for **Block_base_shape** in paragraph 4.2.9 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 |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')))) = 1;
  WR3: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'length')))) = 1;
  WR4: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'width')))) = 1;
  WR5: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'height')))) = 1;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **block_shape_representation** shall contain exactly four **representation_items** in its set of **items**.

WR2: One of the **representation_items** 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_items** 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_items** 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_items** 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.

5.2.3.1.6 boss

A **boss** is a type of **feature_definition** that is an enclosed shape protrusion with a reference location and position. A **boss** shall be circular or non-circular. See ARM definition for Boss in paragraph 4.2.10 for more information.

EXPRESS specification:

```

*)
ENTITY boss
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SELF.description IN ['circular', 'non-circular'];
  WR2: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (pd))) |
    NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
    (sa_occ.description = 'boss height occurrence') AND
    (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    (sar.description = 'path feature component usage') AND
    ('FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
    ('FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
    TYPEOF (sdr.relateing_shape_aspect)) AND
    (sdr.relateing_shape_aspect.description = 'linear') AND
    (sdr.name = 'boss height')) = 1))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <*
    impl_rep.used_representation.items |
    (SIZEOF (
    ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND (it.name = 'fillet radius')) = 1))) = 0))) = 0;
  WR4: (NOT (SELF.description = 'circular')) OR

```

```

(SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'circular profile occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE' IN
TYPEOF (sdr.relate_shape_aspect))) = 1))) = 1))) = 0);
WR5: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'top condition occurrence') AND
(SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'boss top usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar)))) |
'FEATURE_BASED_PROCESS_PLANNING.BOSS_TOP' IN
TYPEOF (fcr.relate_shape_aspect))) = 1))) = 1))) = 0;
WR6: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'change in diameter occurrence') AND
(SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'taper usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar)))) |
'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN
TYPEOF (fcr.relate_shape_aspect))) = 1))) <= 1))) = 0;
WR7: (NOT (SELF.description = 'non-circular')) OR
(SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'enclosed boundary occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +

```



```

'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE'] *
TYPEOF (sdr.relatng_shape_aspect)) = 1)) = 1))) = 1))) = 0);
WR8: (NOT (SELF.description = 'non-circular')) OR
(SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'change in boundary occurrence') AND
(SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') |
(sar.description = 'taper usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.TAPER' IN
TYPEOF (fcr.related_shape_aspect)) AND
(fcr.related_shape_aspect.description = 'angle taper'))
= 1))) = 1))) = 0);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **description** for the **boss** shall be either 'circular' or 'non-circular'.

WR2: The **boss** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'boss height occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'boss height' and a **description** of 'path feature component usage' in which the **relating_shape_aspect** is a **path_feature_component** with a **description** of 'linear'.

WR3: The implicit representation of the **boss** shall have exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'fillet radius'.

WR4: If the **boss** has a **description** of 'circular' the **boss** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'circular profile occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' in which the **relating_shape_aspect** is a **circular_closed_profile**.

WR5: The **boss** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'top condition occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'boss top usage' in which the **relating_shape_aspect** is a **boss_top**.

WR6: The **boss** shall be the basis shape for at most one **shape_aspect** with a **description** of 'change in diameter occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** in with a **description** of 'taper usage' in which the **relating_shape_aspect** is a **taper**.

WR7: If the **boss** has a **description** of 'non-circular', the **boss** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'enclosed boundary occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'profile usage' in which the **relating_shape_aspect** is either a **circular_closed_profile**, **rectangular_closed_profile**, **ngon_closed_profile**, or **closed_path_profile**.

WR8: If the **boss** has a **description** of 'non-circular', the **boss** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'change in boundary occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'taper usage' in which the **relating_shape_aspect** is a **taper** with a **description** of 'angle taper'.

5.2.3.1.7 boss_top

A **boss_top** is a type of **shape_aspect** that is the end condition for a **boss feature_definition**. See ARM definition for **Boss_top_condition** in paragraph 4.2.11 for more information.

EXPRESS specification:

```
*)
ENTITY boss_top
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape_definition);
  WR2: SELF.description IN ['planar', 'non-planar'];
  WR3: (NOT (SELF.description = 'planar')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') |
      NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'DIRECTION_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation))) = 1))) = 0));
  WR4: (NOT (SELF.description = 'planar')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') |
      NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'LOCATION_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation))) = 1))) = 0));
  WR5: (NOT (SELF.description = 'non-planar')) OR
```

```

(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'FACE_SHAPE_REPRESENTATION'
IN TYPEOF (pdr.used_representation))) = 1))) = 0);
WR6: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') |
(sar.description = 'boss top usage') AND
('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF (sar))) |
(fcr.related_shape_aspect.description = 'top condition occurrence')
AND ('FEATURE_BASED_PROCESS_PLANNING.BOSS' IN TYPEOF
(fcr.related_shape_aspect.of_shape.definition)))) >= 1;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) = 1))) = 0;
WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0;
WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'boss top orientation') AND
(it.description IN ['boss start', 'boss end']))) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **boss_top** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **description** of the **boss_top** shall be either 'planar' or 'non-planar'.

WR3: If the **description** of the **boss_top** is 'planar', the **boss_top** shall have exactly one **direction_shape_-representation**.

WR4: If the **description** of the **boss_top** is 'planar', the **boss_top** shall have exactly one **location_shape_-representation**.

WR5: If the **description** of the **boss_top** is 'non-planar', the **boss_top** shall have exactly one **face_shape_-representation**.

WR6: The **boss_top** shall be the **relating_shape_aspect** in at least one **feature_component_relationship** with a **name** of 'boss top usage' in which the **related_shape_aspect** is a **shape_aspect** of a **boss** with a **name** of 'top condition occurrence'.

WR7: The **boss_top** shall have exactly one **shape_representation_with_parameters** to specify its implicit representation.

WR8: The **boss_top** shall be represented implicitly by exactly one **representation_item**.

WR9: The implicit representation of the **boss_top** shall have exactly one **representation_item** in its **items** set which is a **descriptive_representation_item** with a **name** of 'boss top orientation' and a **description** of either 'boss start' or 'boss end'.

Informal propositions:

IP1: The location of the **boss_top** shall be at the center of the face that is the mating face to a **boss** feature.

IP2: The **boss_top** shall be defined on the mating face in the X-Y plane with the Z direction coincident to that of the mating **boss** feature.

5.2.3.1.8 chamfer

A **chamfer** is a type of **transition_feature** that is the representation of a linear transition between two **shape_aspects**. A **chamfer** defines the outside profile of a part. See ARM definition for Fillet in paragraph 4.2.19 for more information.

EXPRESS specification:

```
*)
ENTITY chamfer
  SUBTYPE OF (transition_feature);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
```

```

NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
IN TYPEOF (pdr.used_representation)) AND
(pdr.used_representation.name = 'chamfer face')))) <= 1))) = 0;
WR2: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') |
'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF (sar)) |
('FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET' IN
TYPEOF (fcr.related_shape_aspect)) AND
(fcr.related_shape_aspect.description = 'first offset')))) = 1;
WR3: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') |
'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF (sar)) |
('FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET' IN
TYPEOF (fcr.related_shape_aspect)) AND
(fcr.related_shape_aspect.description = 'second offset')))) = 1;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **chamfer** shall have zero or one **face_shape_representations** in the role of the chamfer face.

WR2: The **chamfer** shall be related to exactly one **chamfer_offset** that specifies the first offset for the chamfer.

WR3: The **chamfer** shall be related to exactly one **chamfer_offset** that specifies the second offset for the chamfer.

5.2.3.1.9 chamfer_offset

A **chamfer_offset** is a type of **shape_aspect** that is the representation of the linear setbacks from the edge of two surfaces to create a **chamfer feature_definition**. See ARM definition for First_offset in paragraph 4.2.70, Second_chamfer_offset in paragraph 4.2.166, and Second_offset in paragraph 4.2.167 for more information.

EXPRESS specification:

```

*)
ENTITY chamfer_offset
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: SELF.description IN ['first offset', 'second offset'];
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +

```

```

'DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) = 1))) = 0;
WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0;
WR4: (NOT (SELF.description = 'first offset')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'offset amount')) = 1))) = 0))) = 0);
WR5: (NOT (SELF.description = 'first offset')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
IN TYPEOF (pdr.used_representation)) AND
(pdr.used_representation.name = 'first face shape')) = 1))) = 0);
WR6: (NOT (SELF.description = 'second offset')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
((SIZEOF
(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'offset amount')) OR
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *

```

```

        TYPEOF (it)) = 2) AND (it.name = 'offset angle')))) = 1))) = 0))) = 0));
WR7: (NOT (SELF.description = 'second offset')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'second face shape')))) = 1))) = 0));
WR8: SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar)) |
        'FEATURE_BASED_PROCESS_PLANNING.CHAMFER' IN
        TYPEOF (sdr.relatng_shape_aspect))) >= 1;

END_ENTITY;
(*

```

Formal propositions:

WR1: The **description** for the **chamfer_offset** shall be either 'first offset' or 'second offset'.

WR2: The **chamfer_offset** shall have its implicit representation specified by exactly one **shape_representation_with_parameters**.

WR3: The **chamfer_offset** shall have an implicit representation that contains exactly one **representation_item** in its set of **items**.

WR4: If the **description** of the **chamfer_offset** is 'first offset', exactly one **representation_item** used for the implicit representation of the **chamfer_offset** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'offset amount'.

WR5: If the **description** of the **chamfer_offset** is 'first offset', the **chamfer_offset** shall have exactly one **face_shape_representation** with a **name** of 'first face shape'.

WR6: If the **description** of the **chamfer_offset** is 'second offset', exactly one **representation_item** used for the implicit representation of the **chamfer_offset** shall be either of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'offset amount', or of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'offset angle'.

WR7: If the **description** of the **chamfer_offset** is 'second offset', the **chamfer_offset** shall have exactly one **face_shape_representation** with a **name** of 'second face shape'.

WR8: The **chamfer_offset** shall be the **related_shape_aspect** in at least one **feature_component_relationship** in which the **relating_shape_aspect** is a **chamfer**.

5.2.3.1.10 circular_closed_profile

A **circular_closed_profile** is a type of **shape_aspect** that is an enclosed 2D area defined by a radius, with orientation and location. The location is defined to be at the center of the circular arc, and the orientation of the **circular_closed_profile** is the X-Y plane. See ARM definition for Circular_closed_profile in paragraph 4.2.22 for more information.

EXPRESS specification:

```

*)
ENTITY circular_closed_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')) = 1)) = 0))) = 0;
  WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |

```



```

        (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND
        (it.name = 'diameter')) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **circular_closed_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **circular_closed_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **circular_closed_profile** shall contain two **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **circular_closed_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used the implicit representation of a **circular_closed_profile** shall be of type **measure_representation_item** and **length_angle_measure_with_unit** with a **name** of 'diameter'.

Informal propositions:

IP1: The location of the **circular_closed_profile** shall be defined at the center of the circle.

IP2: The **circular_closed_profile** shall be defined in the X-Y plane.

5.2.3.1.11 circular_pattern

A **circular_pattern** is a type of **replicate_feature** that relates a base feature and one or more **shape_aspects** which are the placement of the base feature at a specified circular location on the base part. A diameter is defined with a placement on the part and a number of base features are equally spaced around this diameter. See ARM definition for Circular_pattern in paragraph 4.2.26 for more information.

EXPRESS specification:

```

*)
ENTITY circular_pattern
  SUBTYPE OF (replicate_feature);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')) <= 3;
  WR2: SIZEOF (QUERY (sar <* USEDIN (SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    NOT ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF (sar)))) = 0;

```

WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
 'DEFINITION') |
 NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF (pdr.used_representation))) = 1))) = 0;
 WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF (pdr.used_representation)) |
 NOT (SIZEOF (impl_rep.used_representation.items) = 5))) = 0))) = 0;
 WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF (pdr.used_representation)) |
 NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
 (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
 'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
 TYPEOF (it))) = 2) AND (it.name = 'diameter')))) = 1))) = 0))) = 0;
 WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF (pdr.used_representation)) |
 NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
 (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
 'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
 TYPEOF (it))) = 2) AND (it.name = 'base feature rotation')))) = 1))) = 0))) = 0;
 WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF (pdr.used_representation)) |
 NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
 ('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
 TYPEOF (it)) AND
 ('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
 TYPEOF (it\measure_with_unit.value_component)) AND
 (it.name = 'number of features')))) = 1))) = 0))) = 0;
 WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |

```

NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'angular spacing')) = 1))) = 0)))
= 0;
WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN
TYPEOF (it)) AND (it.name = 'orientation')) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **circular_pattern** shall be the **relating_shape_aspect** in no more than three **shape_aspect_-relationships**.

WR2: The **circular_pattern** shall be the **relating_shape_aspect** in only **feature_component_-relationships**.

WR3: The **circular_pattern** shall have exactly one implicit representation.

WR4: The implicit representation of the **circular_pattern** shall contain five **representation_items** in its set of **items**.

WR5: The implicit representation of the **circular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'diameter'.

WR6: The implicit representation of the **circular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'base feature rotation'.

WR7: The implicit representation of the **circular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of features'.

WR8: The implicit representation of the **circular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'angular spacing'.

WR9: The implicit representation of the **circular_pattern** shall contain exactly one **representation_item** of type **placement** with a **name** of 'orientation'.

Informal propositions:

IP1: The location of the **circular_pattern** shall be defined at the center of the circle.

IP2: The **circular_pattern** shall be defined in the X-Y plane, with the X direction intersecting the placement position of the first base feature.

5.2.3.1.12 circular_runout_tolerance

A **circular_runout_tolerance** is a type of **geometric_tolerance_with_datum_reference** which represents the maximum variation of a considered **shape_aspect** with respect to a fixed position during one complete revolution about a **datum_reference**. See ARM definition for Angularity_tolerance in paragraph 4.2.27 for more information.

EXPRESS specification:

```
*)
ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'circular runout';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system)
    <= 2;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **name** for the **circular_runout_tolerance** shall be 'circular runout'.

WR2: There shall be at most two **datum_references** in the **datum_system** set for the **circular_runout_tolerance**.

5.2.3.1.13 closed_path_profile

A **closed_path_profile** is a type of **shape_aspect** that is a closed two dimensional area defined by a connected set of curves with a location and orientation. The start vertex and end vertex for the set of curves are the same point creating an enclosed area. The **closed_path_profile** is located at the approximate center

of the enclosed area, and the orientation is in the X-Y plane. See ARM definition for General_closed_profile in paragraph 4.2.76 for more information.

EXPRESS specification:

```

*)
ENTITY closed_path_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')) = 1))) = 0))) = 0;
  WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PATH_SHAPE_REPRESENTATION'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
END_ENTITY;
(

```

Formal propositions:

WR1: The **closed_path_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **closed_path_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **closed_path_profile** shall contain one **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **closed_path_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: The **closed_path_profile** shall have exactly one **path_shape_representation**.

Informal propositions:

IP1: The location of the **closed_path_profile** shall be defined at the approximate center of the enclosed area.

IP2: The **closed_path_profile** shall be defined in the X-Y plane.

5.2.3.1.14 composite_hole

A **composite_hole** is a feature that is the representation of two **round_holes** for the purpose of creating two commonly used **round_hole** combinations, the countersunk hole and the counterbore hole. The counterbore hole is the representation of two **round_holes** with one having a larger diameter than the second. The countersunk hole is the representation of two **round_holes**, the first having a larger diameter than the second with a linear decrease in diameter until it is the same as the second. This linear decrease shall be represented by a **taper**.

EXPRESS specification:

```

*)
ENTITY composite_hole
  SUBTYPE OF (composite_machining_feature);
WHERE
  WR1: SELF.description IN ['counterbore', 'countersunk'];
  WR2: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') |
    'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF (sar)) |
    'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF
    (fcr.related_shape_aspect))) = 2;
  WR3: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') |

```

```

'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF (sar)) |
(NOT ('FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF
(fcr.related_shape_aspect))) OR
(SIZEOF (QUERY (fcr2 <* QUERY (sar2 <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') - fcr |
'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF (sar2)) |
(NOT ('FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF
(fcr2.related_shape_aspect))) OR
(SIZEOF (QUERY (pd <* USEDIN (fcr.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(NOT (
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(it.name = 'diameter') AND
(SIZEOF (QUERY (pd2 <* USEDIN (fcr2.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (pdr2 <* USEDIN (pd2,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(NOT (
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr2.used_representation))) OR
(SIZEOF (QUERY (pd2 <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep2 <* QUERY (pdr2 <* USEDIN (pd2,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr2.used_representation)) |
NOT (SIZEOF (QUERY (it2 <* impl_rep2.used_representation.items |
(it2.name = 'diameter') AND
(it\measure_with_unit.value_component =
it2\measure_with_unit.value_component))) = 0))) = 0))) = 0))) = 0)))
= 0))) = 0))) = 0))) = 0))) = 0))) = 0))) = 0;

```

```

WR4: (NOT (SELF.description = 'countersunk')) OR
      (SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') |
        'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF (sar)) |
        ('FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF
        (fcr.related_shape_aspect)) AND
        (SIZEOF (QUERY (rh_sdr <* QUERY (rh_sar <*
        USEDIN (fcr.related_shape_aspect,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') |
        'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF (rh_sar)) |
        'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN
        TYPEOF (rh_sdr.related_shape_aspect))) = 1))) = 1);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **description** of the **composite_hole** shall be either 'counterbore' or 'countersunk'.

WR2: The **composite_hole** shall be the **relating_shape_aspect** in exactly two instances of **shape_aspect_-relationship** with a **related_shape_aspect** of type **round_hole**.

WR3: Each **round_holes** that comprise the **composite_hole** shall not have diameters that are equal in length.

WR4: If the **description** of the **composite_hole** is 'countersunk', exactly one **round_hole** that comprises it shall be a tapered hole.

5.2.3.1.15 composite_machining_feature

A **composite_machining_feature** is a type of **machining_feature** that is the representation of several **machining_features** and their relationships to one another for the purpose of creating unique user defined features.

EXPRESS specification:

```

*)
ENTITY composite_machining_feature
  SUBTYPE OF (machining_feature);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
        TYPEOF (SELF);
  WR2: SIZEOF (QUERY (sar <* SELF\composite_shape_aspect.
        component_relationships |
        NOT ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF (sar)))) = 0;

```



```

WR3: SIZEOF (QUERY (sar <* SELF\composite_shape_aspect.
    component_relationships |
    NOT ('FEATURE_BASED_PROCESS_PLANNING.MACHINING_FEATURE' IN
    TYPEOF (sar.related_shape_aspect)))) = 0;
END_ENTITY;
(*)

```

Formal propositions:

WR1: The **composite_machining_feature** shall be a **composite_shape_aspect**.

WR2: The **composite_machining_feature** shall be the **relating_shape_aspect** in only **shape_aspect_relationships** of type **feature_component_relationship**.

WR3: The **composite_machining_feature** shall be the **relating_shape_aspect** in only **shape_aspect_relationships** with a **related_shape_aspect** of type **machining_feature**.

5.2.3.1.16 concentricity_tolerance

A **concentricity_tolerance** is type of **geometric_tolerance_with_datum_reference** that represents a cylindrical **shape_aspect** which shall be concentric when rotated about a **datum_reference**. See ARM definition for Concentricity_tolerance in paragraph 4.2.35 for more information.

EXPRESS specification:

```

*)
ENTITY concentricity_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
    WR1: SELF\geometric_tolerance.name = 'concentricity';
    WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) =
        1;
END_ENTITY;
(*)

```

Formal propositions:

WR1: The **name** for the **concentricity_tolerance** shall be 'concentricity'.

WR2: There shall be exactly one **datum_reference** in the **datum_system** set for the **concentricity_tolerance**.

5.2.3.1.17 cylindrical_shape_representation

A **cylindrical_shape_representation** specifies representation of a shape that is a cylindrical volume defined as a circular area of a defined length. The enclosed area is defined by a circle with a specified radius. See ARM definition for Cylindrical_base_shape in paragraph 4.2.44 for more information.

EXPRESS specification:

```

*)
ENTITY cylindrical_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
WHERE
  WR1: SIZEOF (SELF.items) = 3;
  WR2: SIZEOF (QUERY (it <* SELF.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')))) = 1;
  WR3: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'length')))) = 1;
  WR4: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'diameter')))) = 1;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **cylindrical_shape_representation** shall contain exactly three **representation_items** in its set of **items**.

WR2: One of the **representation_items** 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_items** 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_items** used for the implicit representation of a **cylindrical_shape_representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'diameter'.

Informal propositions:

IP1: The location of the **cylindrical_shape_representation** shall be defined to be at the center of the circle that defines the cylinder.

IP2: The **cylindrical_shape_representation** shall be defined by forming a circular profile in the X-Y plane, and the length along the z direction.

5.2.3.1.18 directed_dimensional_location

A **directed_dimensional_location** specifies is a type of **dimension_location** that identifies the direction to measure the location dimension.

EXPRESS specification:

```
*)
ENTITY directed_dimensional_location
  SUBTYPE OF (dimensional_location);
END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_aspect_relationship.relying_shape_aspect: the origin of the directed dimension.

SELF\shape_aspect_relationship.related_shape_aspect: the target of the directed dimension.

5.2.3.1.19 direction_shape_representation

A **direction_shape_representation** is a type of **representation** that represents a direction vector in two or three dimensional space.

EXPRESS specification:

```
*)
ENTITY direction_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF (SELF.items) = 1;
  WR2: SIZEOF (QUERY (it < * SELF.items |
    NOT ('FEATURE_BASED_PROCESS_PLANNING.DIRECTION' IN TYPEOF
      (it)))) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **direction_shape_representation** shall have exactly one **representation_item** in its set of **items**.

WR2: The geometric element that is used to represent the **direction_shape_representation** shall be a **direction**.

5.2.3.1.20 edge_round

An **edge_round** is a type of **transition_feature** that is the representation of a circular convex transition between two **shape_aspects**. An **edge_round** defines the outside profile of a part. See ARM definition for **Edge_round** in paragraph 4.2.63 for more information.

EXPRESS specification:

```

*)
ENTITY edge_round
  SUBTYPE OF (transition_feature);
WHERE
  WR1: (NOT (SELF\shape_aspect.description = 'constant radius')) XOR
        (NOT (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation))) = 1))) = 0));
  WR2: (NOT (SELF\shape_aspect.description = 'constant radius')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 3))) = 0))) = 0);
  WR3: (NOT (SELF.description = 'constant radius')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'radius')) = 1))) = 0))) = 0);
  WR4: (NOT (SELF.description = 'constant radius')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'first offset')) = 1))) = 0))) = 0);

```

```

WR5: (NOT (SELF.description = 'constant radius')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'second offset')) = 1))) = 0))) = 0);
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'edge round face')) = 1))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'first face shape')) = 1))) = 0;
WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'second face shape')) = 1))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: An **edge_round** has an implicit representation if and only if it has a **description** of 'constant radius'.

WR2: If the description of the **edge_round** is 'constant radius', the implicit representation shall contain three **representation_items** in its set of **items**.

WR3: If the description of the **edge_round** is 'constant radius', exactly one **representation_item** used for the implicit representation of the **edge_round** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR4: If the description of the **edge_round** is 'constant radius', exactly one **representation_item** used for the implicit representation of the **edge_round** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'first offset'.

WR5: If the description of the **edge_round** is 'constant radius', exactly one **representation_item** used for the implicit representation of the **edge_round** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'second offset'.

WR6: The **edge_round** shall have zero or one **face_shape_representation** with a **name** of 'edge round face'.

WR7: The **edge_round** shall have exactly one **face_shape_representation** with a **name** of 'first face shape'.

WR8: The **edge_round** shall have exactly one **face_shape_representation** with a **name** of 'second face shape'.

5.2.3.1.21 externally_defined_feature_definition

An **externally_defined_feature_definition** relates the document containing a feature specification to a **feature_definition**. See ARM definition for Catalogue_thread in paragraph 4.2.18, Catalogue_marking in paragraph 4.2.17, and Catalogue_knurl in paragraph 4.2.16 for more information.

EXPRESS specification:

```
*)
ENTITY externally_defined_feature_definition
  SUBTYPE OF (feature_definition, externally_defined_item);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT ({1 <= SIZEOF (impl_rep.items) <= 8}))) = 0))) = 0;
  WR2: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (pd)) |
    NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
    (sa_occ.description = 'partial area occurrence') AND
    (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    (sar.description = 'applied area usage') AND
    ('FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
```

```

'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN
  TYPEOF (sdr.relying_shape_aspect))) = 1))) <= 1))) = 0;
WR3: SIZEOF (USEDIN (SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
  'FEATURE_BASED_PP_DOCUMENT_REFERENCE.ITEMS')) = 1;
WR4: (NOT (SELF.description = 'marking')) OR
  (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND (it.name = 'marking text')) = 1))) = 0))) = 0);
WR5: (NOT (SELF.description = 'thread')) OR
  (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND (it.name = 'thread system') AND
      ((it.description = 'english') OR
      (it.description = 'metric')) = 1))) = 0))) = 0);
WR6: (NOT (SELF.description = 'thread')) OR
  (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND (it.name = 'removal direction') AND
      ((it.description = 'internal') OR
      (it.description = 'external')) = 1))) = 0))) = 0);
WR7: (NOT (SELF.description = 'thread')) OR
  (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN

```

```

        TYPEOF (it)) AND (it.name = 'qualifier')))) = 1))) = 0))) = 0));
WR8: (NOT (SELF.description = 'thread')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        ('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF (it)) AND (it.name = 'hand')))) <= 1))) = 0))) = 0));
WR9: (NOT (SELF.description = 'thread')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        ('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF (it)) AND (it.name = 'fit class')))) <= 1))) = 0))) = 0));
WR10: (NOT (SELF.description = 'thread')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        ('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF (it)) AND (it.name = 'form')))) <= 1))) = 0))) = 0));
WR11: (NOT (SELF.description = 'thread')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it))=2) AND (it.name = 'major diameter')))) <= 1))) = 0))) = 0));
WR12: (NOT (SELF.description = 'thread')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +

```



```

'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.RATIO_MEASURE_WITH_UNIT'] *
TYPEOF (it))=2) AND (it.name = 'number of threads'))<= 1)))= 0)))= 0));
END_ENTITY;
(*

```

Formal propositions:

WR1: The implicit representation of the **externally_defined_feature_definition** shall contain between one and eight **representation_items** in its set of **items**.

WR2: The **externally_defined_feature_definition** shall be the basis shape for at most one **shape_aspect** with a **description** of 'partial area occurrence' that is the **related_shape_aspect** in exactly one **shape_-defining_relationship** with a description of 'applied area usage' in which the **relating_shape_aspect** is an **applied_area**.

WR3: The **externally_defined_feature_definition** shall be in the set of **items** of exactly one **feature_-based_pp_document_assignment** that defines the document containing the feature specification.

WR4: If the **description** of the **externally_defined_feature_definition** is 'marking', exactly one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **descriptive_representation_item** with a **name** of 'marking text'.

WR5: If the **description** of the **externally_defined_feature_definition** is 'thread', exactly one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **descriptive_representation_item** with a **name** of 'thread system' and a **description** of either 'english' or 'metric'.

WR6: If the **description** of the **externally_defined_feature_definition** is 'thread', then exactly one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **descriptive_representation_item** with a **name** of 'removal direction' and a **description** of either 'internal' or 'external'.

WR7: If the **description** of the **externally_defined_feature_definition** is 'thread', exactly one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **descriptive_representation_item** with a **name** of 'qualifier'.

WR8: If the **description** of the **externally_defined_feature_definition** is 'thread', at most one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **descriptive_representation_item** with a **name** of 'hand'.

WR9: If the **description** of the **externally_defined_feature_definition** is 'thread', at most one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **descriptive_representation_item** with a **name** of 'fit class'.

WR10: If the **description** of the **externally_defined_feature_definition** is 'thread', at most one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **descriptive_representation_item** with a **name** of 'form'.

WR11: If the **description** of the **externally_defined_feature_definition** is 'thread', at most one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'major diameter'.

WR12: If the **description** of the **externally_defined_feature_definition** is 'thread', at most one **representation_item** used for the implicit representation of the **externally_defined_feature_definition** shall be of type **measure_representation_item** and **ratio_measure_with_unit** with a **name** of 'number of threads'.

5.2.3.1.22 face_shape_representation

A **face_shape_representation** is a type of **representation** that represents geometry defined by a portion of an associated surface.

EXPRESS specification:

```
*)
ENTITY face_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF (SELF.items) = 1;
  WR2: SIZEOF (QUERY (it <* SELF.items |
    NOT ('FEATURE_BASED_PROCESS_PLANNING.FACE_SURFACE' IN TYPEOF
      (it)))) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **face_shape_representation** shall have exactly one **representation_item** in its set of **items**.

WR2: The geometric element that is used to represent the **face_shape_representation** shall be a **face_surface**.

5.2.3.1.23 feature_based_pp_action_assignment

A **feature_based_pp_action_assignment** specifies those **feature_based_pp_action_items** to which an **action** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_action_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF feature_based_pp_action_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_action_items** for which a particular **action** is applicable.

5.2.3.1.24 feature_based_pp_action_request_assignment

A **feature_based_pp_action_request_assignment** specifies those **feature_based_pp_action_request_items** for which a design exception has been identified.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_action_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF feature_based_pp_action_request_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_action_request_items** for which a particular **action_request** is applicable.

5.2.3.1.25 feature_based_pp_approval_assignment

A **feature_based_pp_approval_assignment** specifies those **feature_based_pp_approved_items** to which an **approval** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_approval_assignment
  SUBTYPE OF (approval_assignment);
```

```

    items : SET [1:?] OF feature_based_pp_approved_item;
END_ENTITY;
(*

```

Attribute definition:

items: the set of **feature_based_pp_approved_items** to which an **approval** is assigned.

5.2.3.1.26 feature_based_pp_date_assignment

A **feature_based_pp_date_assignment** specifies those **feature_based_pp_dated_items** to which a date is assigned.

EXPRESS specification:

```

*)
ENTITY feature_based_pp_date_assignment
    SUBTYPE OF (date_assignment);
    items : SET [1:?] OF feature_based_pp_dated_item;
END_ENTITY;
(*

```

Attribute definition:

items: the set of **feature_based_pp_dated_items** to which a date is assigned.

5.2.3.1.27 feature_based_pp_document_reference

A **feature_based_pp_document_reference** specifies those **feature_based_pp_document_items** to which a referenced document is assigned.

EXPRESS specification:

```

*)
ENTITY feature_based_pp_document_reference
    SUBTYPE OF (document_reference);
    items : SET [1:?] OF feature_based_pp_document_item;
END_ENTITY;
(*

```

Attribute definition:

items: the set of **feature_based_pp_document_items** to which a referenced document is assigned.

5.2.3.1.28 feature_based_pp_organization_assignment

A **feature_based_pp_organization_assignment** specifies those **feature_based_pp_organization_items** to which an **organization** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF feature_based_pp_organization_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_organization_items** to which an **organization** is assigned.

5.2.3.1.29 feature_based_pp_person_and_organization_assignment

A **feature_based_pp_person_and_organization_assignment** specifies those **feature_based_pp_person_and_organization_items** to which a **person_and_organization** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF feature_based_pp_person_and_organization_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_person_and_organization_items** to which a **person_and_organization** is assigned.

5.2.3.1.30 feature_based_pp_security_classification_assignment

A **feature_based_pp_security_classification_assignment** specifies those **feature_based_pp_classified_items** to which a **security_classification** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
```

```

    items : SET [1:?] OF feature_based_pp_classified_item;
END_ENTITY;
(*

```

Attribute definition:

items: the set of **feature_based_pp_classified_items** to which a **security_classification** is assigned.

5.2.3.1.31 feature_component_definition

A **feature_component_definition** is a type of **characterized_object** that is used as a constituent of a preconceived form pattern.

EXPRESS specification:

```

*)
ENTITY feature_component_definition
  SUBTYPE OF (characterized_object);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')) = 1))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **feature_component_definition** shall be the item for which exactly one **shape_aspect** is defined.

5.2.3.1.32 feature_component_relationship

A **feature_component_relationship** is a kind of **shape_aspect_relationship** in which the **related_shape_aspect** is an implicitly represented component, such as a **taper**, of the **relating_shape_aspect**.

EXAMPLE - A Chamfer may have the implicit chamfer offset amount defined with a **feature_component_relationship** and the explicit geometry for the chamfer defined with a **shape_defining_relationship**.

EXPRESS specification:

```

*)
ENTITY feature_component_relationship
  SUPERTYPE OF (ONEOF (pattern_omit_membership, pattern_offset_membership))
  SUBTYPE OF (shape_aspect_relationship);
WHERE
  WR1: (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
    'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_MACHINING_FEATURE',

```

```

    'FEATURE_BASED_PROCESS_PLANNING.TRANSITION_FEATURE'] *
    TYPEOF (SELF.relying_shape_aspect)) = 1)
    OR ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_DEFINITION' IN
    TYPEOF (SELF.relying_shape_aspect.of_shape))OR(
    'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.relying_shape_aspect.of_shape));
END_ENTITY;
(*)

```

Formal proposition:

WR1: The **shape_aspect** referenced as the **relying_shape_aspect** shall be of type **replicate_feature**, **composite_machining_feature**, or **transition_feature** or shall be an aspect of the shape of a **feature_definition** or a **feature_component_definition**.

5.2.3.1.33 feature_definition

A **feature_definition** is a type of **characterized_object** that is a preconceived form pattern.

EXPRESS specification:

```

*)
ENTITY feature_definition
  SUBTYPE OF (characterized_object);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN
    TYPEOF (it))) = 1))) = 0))) = 0;
END_ENTITY;
(*)

```

Formal propositions:

WR1: The **feature_definition** shall have exactly one implicit **representation**.

WR2: Exactly one **representation_item** in the implicit **representation** of the **feature_definition** shall be a **placement**. The **placement** shall define a reference location and orientation for the origin of the **feature_definition**.

5.2.3.1.34 feature_pattern

A **feature_pattern** is a type of **replicate_feature** that relates a base feature and one or more **shape_aspects** which are the placement of the base feature at a specified location on the base part.

EXPRESS specification:

```
*)
ENTITY feature_pattern
  SUBTYPE OF (replicate_feature);
WHERE
  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    (('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP' IN
    TYPEOF (sar)) AND (sar.name = 'replicate_positioning')) AND
    (NOT (SIZEOF (QUERY (pd <* USEDIN (sar.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN
    TYPEOF (it)) AND (it.name = 'base feature placement')))) >= 1)))) = 0)))) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **feature_pattern** shall be the **relating_shape_aspect** in exactly one **shape_aspect_relationship** that is a **shape_defining_relationship** with the name of 'replicate_positioning' in which the **related_shape_aspect** has an implicit representation defined by a **shape_representation_with_parameters** which contains one or more **representation_items** in its set of **items** of type **placement** with the name of 'base feature placement'.

5.2.3.1.35 fillet

A **fillet** is a type of **transition_feature** that is the representation of a circular concave transition between two **shape_aspects**. A **fillet** defines the outside profile of a part. See ARM definition for Fillet in paragraph 4.2.69 for more information.

EXPRESS specification:

```

*)
ENTITY fillet
  SUBTYPE OF (transition_feature);
WHERE
  WR1: (NOT (SELF\shape_aspect.description = 'constant radius')) XOR
        (NOT (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation))) = 1))) = 0))) = 0));
  WR2: (NOT (SELF\shape_aspect.description = 'constant radius')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 3))) = 0))) = 0));
  WR3: (NOT (SELF.description = 'constant radius')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'radius')) = 1))) = 0))) = 0));
  WR4: (NOT (SELF.description = 'constant radius')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'first offset')) = 1))) = 0))) = 0));
  WR5: (NOT (SELF.description = 'constant radius')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'second offset')))) = 1))) = 0))) = 0);
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'fillet face')))) <= 1))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'first face shape')))) = 1))) = 0;
WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'second face shape')))) = 1))) = 0;
WR9: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') |
        'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF (sar))
        'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN
        TYPEOF (fcr.related_shape_aspect))) <= 1;
END_ENTITY;
(*

```

Formal propositions:

WR1: A **fillet** has an implicit representation if and only if it has a **description** of 'constant radius'.

WR2: If the description of the **fillet** is 'constant radius', the implicit representation shall contain three **representation_items** in its set of **items**.

WR3: If the description of the **fillet** is 'constant radius', exactly one **representation_item** used for the implicit representation of the **fillet** shall be of type **measure_representation_item** and **length_measure_-with_unit** with a **name** of 'radius'.

WR4: If the description of the **fillet** is 'constant radius', exactly one **representation_item** used for the implicit representation of the **fillet** shall be of type **measure_representation_item** and **length_measure_-with_unit** with a **name** of 'first offset'.

WR5: If the description of the **fillet** is 'constant radius', exactly one **representation_item** used for the implicit representation of the **fillet** shall be of type **measure_representation_item** and **length_measure_-with_unit** with a **name** of 'second offset'.

WR6: The **fillet** shall have zero or one **face_shape_representation** with a **name** of 'fillet face'.

WR7: The **fillet** shall have exactly one **face_shape_representation** with a **name** of 'first face shape'.

WR8: The **fillet** shall have exactly one **face_shape_representation** with a **name** of 'second face shape'.

WR9: The **fillet** shall be the **relating_shape_aspect** in zero or one instance of **shape_aspect_relationship** of type **feature_component_relationship** with a **related_shape_aspect** of type **taper**.

5.2.3.1.36 flat_face

A **flat_face** is a type of **feature_definition** that is the representation of a volume that is removed from the base shape. This removal shall be planar and representation shall be a **linear_profile** moving along a linear **path_feature_component**. See ARM definition for Planar_face in paragraph 4.2.134 for more information.

EXPRESS specification:

```

*)
ENTITY flat_face
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION'))
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
    IN TYPEOF (pdr.used_representation)) AND
    (pdr.used_representation.name = 'removal direction'))))
    = 1))) = 0;
  WR2: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (pd)) |
    NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
    (sa_occ.description = 'course of travel occurrence') AND
    (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |

```

```

(sar.description = 'path feature component usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
TYPEOF (sdr.relating_shape_aspect)) AND
(sdr.relating_shape_aspect.description = 'linear') AND
(sdr.name = 'course of travel')) = 1))) = 1))) = 0;
WR3: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'removal boundary occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE' IN
TYPEOF (sdr.relating_shape_aspect)) AND
(sdr.name = 'removal boundary')) = 1))) = 1))) = 0;
END_ENTITY;
(*

```

Formal proposition:

WR1: The **flat_face** shall have exactly one **direction_shape_representation** with a **name** of 'removal direction'.

WR2: The **flat_face** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'course of travel occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'path feature component usage' and a **name** of 'course of travel' and a **relating_shape_aspect** that references a **path_feature_component** with a **description** of 'linear'.

WR3: The **flat_face** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'removal boundary occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **name** of 'removal boundary' and a **relating_shape_aspect** that references a **linear_profile**.

5.2.3.1.37 hole_bottom

A **hole_bottom** is a type of **shape_aspect** that is the representation of the end condition for a hole **feature_definition**. See ARM definition for **Blind_bottom_condition** in paragraph 4.2.8 and **Through_bottom_condition** in paragraph 4.2.193 for more information.

EXPRESS specification:

```

*)
ENTITY hole_bottom
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SELF.description IN ['through', 'flat', 'flat with radius',
    'spherical', 'conical'];
  WR3: (NOT (SELF.description = 'through')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (USEDIN (pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION.REPRESENTATION.DEFINITION')) = 0))) = 0);
  WR4: (NOT (SELF.description IN ['flat', 'flat with radius', 'spherical',
    'conical'])) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION.REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0);
  WR5: (NOT (SELF.description = 'flat')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION.REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
  WR6: (NOT (SELF.description IN ['flat with radius', 'spherical'])) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION.REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
  WR7: (NOT (SELF.description = 'conical')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION.REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) |

```

```

NOT (SIZEOF (impl_rep.used_representation.items) = 3))) = 0))) = 0);
WR8: (NOT (SELF.description IN ['flat', 'flat with radius', 'spherical',
'conical'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND (it.name = 'blind bottom orientation') AND
(it.description IN ['hole start', 'hole end']))) = 1))) = 0))) = 0);
WR9: (NOT (SELF.description = 'flat with radius')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'corner radius')))) = 1))) = 0))) = 0);
WR10: (NOT (SELF.description = 'spherical')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'radius')))) = 1))) = 0))) = 0);
WR11: (NOT (SELF.description = 'conical')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF
(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'tip radius')))) = 1))) = 0))) = 0);

```

```

WR12: (NOT (SELF.description = 'conical')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT ( SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation) ) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF
        (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'tip angle')) = 1))) = 0))) = 0));
WR13: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') |
        (sar.description = 'hole bottom usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF (sar))) |
        (fcr.related_shape_aspect.description = 'bottom condition occurrence')
        AND
        ('FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF
        (fcr.related_shape_aspect.of_shape.definition)))) >= 1;
END_ENTITY;
(*)

```

Formal propositions:

WR1: The **hole_bottom** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **description** of the **hole_bottom** shall be either 'through', 'flat', 'flat with radius', 'spherical', or 'conical'.

WR3: If the **description** of the **hole_bottom** is 'through', the **hole_bottom** shall not have any representation.

WR4: If the **description** of the **hole_bottom** is 'flat', 'flat with radius', 'spherical', or 'conical', the **hole_bottom** shall have exactly one **shape_representation_with_parameters** to specify its implicit representation.

WR5: If the **description** of the **hole_bottom** is 'flat', the **hole_bottom** shall be represented implicitly by exactly one **representation_item**.

WR6: If the **description** of the **hole_bottom** is 'flat with radius' or 'spherical', the **hole_bottom** shall be represented implicitly by exactly two **representation_items**.

WR7: If the **description** of the **hole_bottom** is 'conical', the **hole_bottom** shall be represented implicitly by exactly three **representation_items**.

WR8: If the **description** of the **hole_bottom** is 'flat', 'flat with radius', 'spherical', or 'conical', the implicit representation of the **hole_bottom** shall have exactly one **representation_item** in its **items** set which is a **descriptive_representation_item** with a **name** of 'blind bottom orientation' and a **description** of either 'hole start' or 'hole end'.

WR9: If the **description** of the **hole_bottom** is 'flat with radius', the implicit representation of the **hole_bottom** shall have exactly one **representation_item** in its **items** set which is a **measure_representation_item** and **length_measure_with_unit** with a **name** of 'corner radius'.

WR10: If the **description** of the **hole_bottom** is 'spherical', the implicit representation of the **hole_bottom** shall have exactly one **representation_item** in its **items** set which is a **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR11: If the **description** of the **hole_bottom** is 'conical', the implicit representation of the **hole_bottom** shall have exactly one **representation_item** in its **items** set which is a **measure_representation_item** and **length_measure_with_unit** with a **name** of 'tip radius'.

WR12: If the **description** of the **hole_bottom** is 'conical', the implicit representation of the **hole_bottom** shall have exactly one **representation_item** in its **items** set which is a **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'tip angle'.

WR13: The **hole_bottom** shall be the **relating_shape_aspect** in at least one **feature_component_relationship** with a **description** of 'hole bottom usage' in which the **related_shape_aspect** is an aspect of the shape of a **round_hole** with a **description** of 'bottom condition occurrence'.

Informal propositions:

IP1: The location of the **hole_bottom** shall be at the center of the circular face that is the mating face to a **hole** feature for spherical, conical, and flat hole bottom types. For flat_with_radius, the location shall be at the absolute bottom.

IP2: The **hole_bottom** shall be defined on the mating face in the X-Y plane with the Z direction coincident to that of the mating **hole** feature.

5.2.3.1.38 instanced_feature

An **instanced_feature** is a type of **shape_aspect** and of **feature_definition** that is the representation of a preconceived pattern for the machining application purpose.

EXPRESS specification:

```
*)
ENTITY instanced_feature
  SUBTYPE OF (feature_definition, shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION' IN
```



```

        TYPEOF (SELF.of_shape.definition);
    WR2: SELF.product_definitional;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **instanced_feature** shall be an aspect of the shape of a **product_definition**.

WR2: The **instanced_feature** shall lie on the boundary of the part.

5.2.3.1.39 linear_profile

A **linear_profile** is a type of **shape_aspect** that is the representation of a straight line with orientation and location. See ARM definition for Linear_profile in paragraph 4.2.95 for more information.

EXPRESS specification:

```

*)
ENTITY linear_profile
    SUBTYPE OF (shape_aspect);
WHERE
    WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);
    WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation))) = 1))) = 0;
    WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0;
    WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND

```

```

        (it.name = 'orientation')))) = 1))) = 0))) = 0;
WR5: sizeof (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (sizeof (QUERY (it <* impl_rep.used_representation.items |
    (sizeof (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'profile length')))) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **linear_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **linear_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **linear_profile** shall contain two **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **linear_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **linear_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'profile length'.

Informal propositions:

IP1: The location of the **linear_profile** shall be one end of the profile.

IP2: The X direction of the **linear_profile** shall be defined coincident to the X direction of the mating feature.

5.2.3.1.40 linear_profile_tolerance

A **linear_profile_tolerance** is a type of **geometric_tolerance** that defines the allowable deviation of a **curve** that may require **datum_reference** to control the definition. See ARM definition for **linear_profile_-tolerance** in paragraph 4.2.96 for more information.

EXPRESS specification:

```

*)
ENTITY linear_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: SELF\geometric_tolerance.name = 'linear profile';
  WR2: (NOT ('FEATURE_BASED_PROCESS_PLANNING.' +
    'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF))) OR
(SIZEOF
  (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **name** for the **linear_profile_tolerance** shall be 'linear profile'.

WR2: There shall be at three or less **datum_references** in the **datum_system** set for the **linear_profile_tolerance**.

5.2.3.1.41 location_shape_representation

A **location_shape_representation** is a type of **representation** that represents a geometric position or a reference point.

EXPRESS specification:

```

*)
ENTITY location_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF (SELF.items) = 1;
  WR2: SIZEOF (QUERY it <* SELF.items |
    NOT ('FEATURE_BASED_PROCESS_PLANNING.POINT' IN
    TYPEOF(it))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **location_shape_representation** shall have exactly one **representation_item** in its set of **items**.

WR2: The geometric element that is used to represent the **location_shape_representation** shall be a **point**.

5.2.3.1.42 machining_feature

A **machining_feature** is a type of **instanced_feature** that is the representation of a preconceived pattern for the machining application purpose.

EXPRESS specification:

```

*)
ENTITY machining_feature
  SUBTYPE OF (instanced_feature);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) >= 1))) = 0;
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.BOSS',
    'FEATURE_BASED_PROCESS_PLANNING.TURNED_KNURL',
    'FEATURE_BASED_PROCESS_PLANNING.THREAD',
    'FEATURE_BASED_PROCESS_PLANNING.MARKING',
    'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE',
    'FEATURE_BASED_PROCESS_PLANNING.OUTSIDE_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.POCKET',
    'FEATURE_BASED_PROCESS_PLANNING.REMOVAL_VOLUME',
    'FEATURE_BASED_PROCESS_PLANNING.REVOLVED_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.OUTER_ROUND',
    'FEATURE_BASED_PROCESS_PLANNING.FLAT_FACE',
    'FEATURE_BASED_PROCESS_PLANNING.PROTRUSION',
    'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_END',
    'FEATURE_BASED_PROCESS_PLANNING.SLOT',
    'FEATURE_BASED_PROCESS_PLANNING.SPHERICAL_CAP',
    'FEATURE_BASED_PROCESS_PLANNING.STEP',
    'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_MACHINING_FEATURE',
    'FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_FEATURE_DEFINITION']
    * TYPEOF (SELF)) = 1;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN
    TYPEOF (it)) AND (it.name = 'orientation')))) = 1))) = 0))) = 0;
  WR5: (NOT (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.TURNED_KNURL',
    'FEATURE_BASED_PROCESS_PLANNING.THREAD',
    'FEATURE_BASED_PROCESS_PLANNING.MARKING',
    'FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_FEATURE_DEFINITION']

```

```

    * TYPEOF (SELF)) = 1)) OR
    (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') |
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP' IN
    TYPEOF (sar)) |
    sdr.name = 'applied shape')) = 1);
    WR6: number_of_representation_items (SELF);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **machining_feature** shall have at least one representation.

WR2: The **machining_feature** shall have exactly one implicit representation.

WR3: The **machining_feature** shall be either a **boss**, **turned_knurl**, **thread**, **marking**, **round_hole**, **outside_profile**, **pocket**, **removal_volume**, **revolved_profile**, **outer_round**, **flat_face**, **protrusion**, **rounded_end**, **slot**, **spherical_cap**, **step**, **composite_machining_feature**, or **externally_defined_machining_feature**.

WR4: Exactly one **representation_item** used for the implicit representation of a **machining_feature** shall be of type **placement** with a **name** of 'orientation'.

WR5: If the **machining_feature** is a **turned_knurl**, **thread**, **marking**, or **externally_defined_feature_definition** it shall be the **relating_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'applied shape'.

WR6: The **machining_feature** shall have the proper number of **representation_items** used to specify its implicit representation depending on its **feature_definition** type. The **feature_definition** types and their corresponding number of **representation_items** are as follows:

- A **boss** shall have exactly two **representation_items** in its implicit representation;
- A **flat_face** shall have exactly one **representation_item** in its implicit representation;
- A **marking** shall have exactly six **representation_items** in its implicit representation;
- An **outer_round** shall have exactly three **representation_items** in its implicit representation if it has a description of 'outer diameter' or exactly two **representation_items** if it has a description of 'outer diameter to shoulder';
- An **outside_profile** shall have exactly one **representation_item** in its implicit representation;
- A **pocket** shall have exactly one **representation_item** in its implicit representation;
- A **protrusion** shall have exactly one **representation_item** in its implicit representation;
- A **removal_volume** shall have exactly one **representation_item** in its implicit representation;
- A **revolved_profile** shall have exactly two **representation_items** in its implicit representation;
- A **round_hole** shall have exactly one **representation_item** in its implicit representation;
- A **rounded_end** shall have exactly one **representation_item** in its implicit representation;
- A **slot** shall have exactly two **representation_items** in its implicit representation;

- A **spherical_cap** shall have exactly three **representation_items** in its implicit representation;
- A **step** shall have exactly one **representation_item** in its implicit representation;
- A **thread** shall have between eight and eleven **representation_items** in its implicit representation;
- A **turned_knurl** shall have between six and nine **representation_items** in its implicit representation.

Informal propositions:

IP1: If the **machining_feature** is a **boss** the location shall be defined at the top of the **boss** at the approximate center of the feature.

IP2: If the **machining_feature** is a **boss** then it 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 of the part face.

IP3: If the **machining_feature** is a **boss** then the location, X direction, and Y direction of the **path_feature_component** with a **description** of 'boss height' shall be the same as the **boss**.

IP4: If the **machining_feature** is a **boss** then the placement of the **circular_closed_profile** that defines the diameter of a **boss** with a **description** of 'circular' shall be with the origin, X direction, and Y direction equal to that of the **boss**.

IP5: If the **machining_feature** is a **boss** then the placement of the **rectangular_closed_profile**, **ngon_closed_profile**, or **closed_path_profile** that defines the shape of a **boss** with a **description** of 'non-circular' shall be with the origin, X direction, and Y direction equal to that of the **boss**.

IP6: If the **machining_feature** is a **flat_face** then the location of the **flat_face** shall be defined at one of the corners of the removal volume.

IP7: If the **machining_feature** is a **flat_face** then the direction of volume removal shall be in the Z direction.

IP8: If the **machining_feature** is a **flat_face** then the origin, X direction, and Y direction of the **path_feature_component** shall be equal to that of the **flat_face**.

IP9: If the **machining_feature** is a **flat_face** then the origin, X direction, Y direction of the **flat_face** shall be equal to that of the **linear_profile**.

IP10: If the **machining_feature** is an **outer_round** then the location of the **outer_round** shall be defined at a position in the Z direction.

IP11: If the **machining_feature** is an **outer_round** then the Z direction is the axis of revolution.

IP12: If the **machining_feature** is an **outer_round** and if the description of the **outer_round** is 'outer diameter to shoulder', the placement of the **vee_profile** shall be along the X direction of the **outer_round**

at a specified distance from the origin. The orientation of the **vee_profile** shall be in the same direction in the Y axis. The Y axis of the **vee_profile** shall be in the same direction as the X axis of the **outer_round**.

IP13: If the **machining_feature** is an **outside_profile** then the origin, X direction, Y direction of the **profile** that defines the shape of the **outside_profile** shall be with the X and Y axes equal to those specified by the placement of the **outside_profile**.

IP14: If the **machining_feature** is a **pocket** then the volume removal of the **pocket** shall be in the Z direction.

IP15: If the **machining_feature** is a **pocket** then the placement of the **circular_closed_profile**, **ngon_closed_profile**, **rectangular_closed_profile**, **closed_path_profile**, **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** that defines the shape of the **pocket** shall be with the origin, X direction, and Y direction equal to that of the **pocket**.

IP16: If the **machining_feature** is a **pocket** and if the **description** of the **pocket** is 'open rectangular', the origin, X direction, Y direction of the **square_u_profile** shall be equal to that of the **pocket**.

IP17: If the **machining_feature** is a **protrusion** then the representation of the **shape_aspect** that is the **relating_shape_aspect** in the **shape_defining_relationship** shall define a volume.

IP18: If the **machining_feature** is a **protrusion** then the location of the **protrusion** shall be at a position on the base of the feature where it mates with the part.

IP19: If the **machining_feature** is a **protrusion** then the Z direction of the **protrusion** shall be with in a direction away from the part.

IP20: If the **machining_feature** is a **protrusion** then the location of the **removal_volume** shall be defined at the bottom and approximate center of the **removal_volume**.

IP21: If the **machining_feature** is a **protrusion** then the Z direction of the **removal_volume** shall be in the direction of volume removal.

IP22: If the **machining_feature** is a **revolved_profile** and if the description of the **revolved_profile** is 'open profile', then the placement of the **open_path_profile** that defines the shape of the **revolved_profile** shall be with the origin, x direction, and y direction equal to that of the **revolved_profile**.

IP23: If the **machining_feature** is a **revolved_profile** and if the description of the **revolved_profile** is 'groove', then the placement of the **linear_profile**, **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** that defines the shape of the **revolved_profile** shall be placed a specified distance away from the origin of the **revolved_profile** along the X direction.

NOTE - The orientation of the **linear_profile**, **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** is independent of the origin of the **revolved_profile**. The **revolved_profile** may be defined on different faces of a part dependent upon the orientation of the profile.

IP24: If the **machining_feature** is a **revolved_profile** and if the description of the **revolved_profile** is 'flat', then the placement of the **linear_profile** that defines the shape of the **revolved_profile** shall be placed a specified distance away from the origin of the **revolved_profile** along the X direction. The Y direction of the **linear_profile** shall be equal to that of the **revolved_profile**. The X direction and Z direction of the **linear_profile** shall be independent from that of the **revolved_profile**.

IP25: If the **machining_feature** is a **revolved_profile** and if the description of the **revolved_profile** is 'round', then the placement of the **partial_circular_profile** that defines the shape of the **revolved_profile** shall be placed a specified distance away from the origin of the **revolved_profile** along the X direction. The Z direction of the **partial_circular_profile** shall be equal to the Y direction of the **revolved_profile**. The X direction and Y direction of the **partial_circular_profile** shall be independent from that of the **revolved_profile**.

IP26: If the **machining_feature** is a **round_hole** then the location of the **round_hole** shall be defined at either the bottom or the top center of the hole.

IP27: If the **machining_feature** is a **round_hole** then the Z direction of the **round_hole** shall be in the direction of volume removal.

IP28: If the **machining_feature** is a **round_hole** then the origin, X direction, Y direction of the **circular_closed_profile** shall be equal to that of the **round_hole**.

IP29: If the **machining_feature** is a **round_hole** then the origin, X direction, Y direction of the **path_feature_component** shall be equal to that of the **round_hole**.

IP30: If the **machining_feature** is a **rounded_end** then the origin, X direction, Y direction of the **partial_circular_profile** shall be equal to that of the **rounded_end**.

IP31: If the **machining_feature** is a **rounded_end** then the origin, X direction, Y direction of the **path_feature_component** shall be equal to that of the **rounded_end**.

IP32: If the **machining_feature** is a **slot** then the location of the **slot** shall be defined at one end of the **slot**.

IP33: If the **machining_feature** is a **slot** and if the **description** of the **path_feature_component** relating to the **slot** is 'linear' or 'complex', then the Z direction of the **slot** shall be in the direction of **slot** end and the volume removal in the Y direction.

IP34: If the **machining_feature** is a **slot** and if the **description** of the **path_feature_component** relating to the **slot** is 'circular' or 'partial circular', then the Z direction of the **slot** shall be the axis of revolution.

IP35: If the **machining_feature** is a **slot** and if the **description** of the **path_feature_component** relating to the **slot** is 'linear' or 'complex', then the origin, X direction, Y direction of the **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** shall be equal to that of the **slot**.

IP36: If the **machining_feature** is a **slot** and if the **description** of the **path_feature_component** relating to the **slot** is 'circular' or 'partial circular', then the origin, X direction, Y direction of the **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** shall be in the X direction of the **slot** at a distance equal to the radius of the **path_feature_component** away from the origin of the slot.

IP37: If the **machining_feature** is a **slot** then the origin, X direction, Y direction of the **path_feature_component** shall be equal to that of the **slot**.

IP38: If the **machining_feature** is a **spherical_cap** then the location of the **spherical_cap** shall be at a position on the base of the feature where it mates with the part.

IP39: If the **machining_feature** is a **spherical_cap** then the Z direction of the **spherical_cap** shall be in the direction of away from the part.

IP40: If the **machining_feature** is a **step** then the origin, X direction, Y direction of the **vee_profile** shall be equal to that of the **step**.

IP41: If the **machining_feature** is a **step** then the origin, X direction, Y direction of the **path_feature_component** shall be equal to that of the **step**.

IP42: If the **machining_feature** is a **thread** then the location of the **thread** shall be at a distance from the end of a cylindrical face, and on the axis of the cylindrical face.

IP43: If the **machining_feature** is a **thread** then the Z direction of the **thread** shall be in the direction of the center axis of the cylindrical face.

IP44: If the **machining_feature** is a **turned_knurl** then the location of the **turned_knurl** shall be at a distance from the end of a cylindrical face, and on the axis of the cylindrical face.

IP45: If the **machining_feature** is a **turned_knurl** then the Z direction of the **turned_knurl** shall be in the direction of the center axis of the cylindrical face.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **machining_feature** entity:

- machining_feature_life_cycle (See 5.2.4.15).

5.2.3.1.43 marking

A **marking** is a type of **feature_definition** that is the representation of a text or symbol shapes that are applied to all or a portion of a **shape_aspect**. See ARM definition for **Defined_marking** in paragraph 4.2.51 for more information.

EXPRESS specification:

```

*)
ENTITY marking
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION')) |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF (it)) AND (it.name = 'marking text')))) = 1))) = 0))) = 0;
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION')) |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF (it)) AND (it.name = 'special instructions')))) <= 1))) = 0)))
    = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION')) |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF (it)) AND (it.name = 'font name')))) <= 1))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION')) |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']) *

```

```

        TYPEOF (it)) = 2) AND (it.name = 'character height')))) <= 1))) = 0)))
    = 0;
WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']) <
    TYPEOF (it)) = 2) AND (it.name = 'character spacing')))) <= 1))) = 0)))
    = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: Exactly one **representation_item** used for the implicit representation of the **marking** shall be of type **descriptive_representation_item** with a **name** of 'marking text'.

WR2: At most one **representation_item** used for the implicit representation of the **marking** shall be of type **descriptive_representation_item** with a **name** of 'special instructions'.

WR3: At most one **representation_item** used for the implicit representation of the **marking** shall be of type **descriptive_representation_item** with a **name** of 'font name'.

WR4: At most one **representation_item** used for the implicit representation of the **marking** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'character height'.

WR5: At most one **representation_item** used for the implicit representation of the **marking** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'character spacing'.

5.2.3.1.44 modified_pattern

A **modified_pattern** is a type of **replicate_feature** that represents the result of a collection of omissions and offsets of a base feature in either a **rectangular_pattern** or a **circular_pattern**.

EXPRESS specification:

```

*)
ENTITY modified_pattern
  SUBTYPE OF (replicate_feature);
WHERE
  WR1: SIZEOF (QUERY (sar <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +

```

```

        'RELATING_SHAPE_ASPECT') |
        (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
        'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
        TYPEOF (sar.related_shape_aspect)) = 1) AND
        (sar.related_shape_aspect.description = 'base pattern')) = 1;
    WR2: SIZEOF (QUERY (sar <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') |
        NOT (SIZEOF (QUERY (msar <* USEDIN (sar.related_shape_aspect,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') |
        (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.PATTERN_OFFSET_MEMBERSHIP',
        'FEATURE_BASED_PROCESS_PLANNING.PATTERN OMIT_MEMBERSHIP'] *
        TYPEOF (sar)) = 1) AND (sar :<: msar))) >= 1))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **modified_pattern** shall be the **relating_shape_aspect** in exactly one **shape_aspect-relationship** with a **related_shape_aspect** that references either a **circular_pattern** or a **rectangular_pattern** and has the **description** of 'base pattern'.

WR2: The **modified_pattern** shall be the **related_shape_aspect** in at least one **shape_aspect-relationship** in which the **related_shape_aspect** is the **related_shape_aspect** in a different instance of **shape_aspect-relationship** that is either a **pattern_offset_membership** or **pattern_omit_membership**.

5.2.3.1.45 ngon_closed_profile

A **ngon_closed_profile** is a type of **shape_aspect** that represents an enclosed 2D area with location and orientation. The enclosed area is defined by three or more straight sides. The location is defined to be at the center of the enclosed area, and the orientation of the **ngon_closed_profile** is the X-Y plane with the X direction parallel to one of the sides of the ngon. See ARM definition for Ngon_profile in paragraph 4.2.110 for more information.

EXPRESS specification:

```

*)
ENTITY ngon_closed_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) = 1))) = 0;
WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 4))) = 0))) = 0;
WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
(it.name = 'orientation')))) = 1))) = 0))) = 0;
WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'number of sides')))) = 1))) = 0))) = 0;
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it.name = 'circumscribed diameter')))) = 1))) = 0))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it.name = 'corner radius')))) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **ngon_closed_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **ngon_closed_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **ngon_closed_profile** shall contain four **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **ngon_closed_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **ngon_closed_profile** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of sides'.

WR6: Exactly one **representation_item** used for the implicit representation of a **ngon_closed_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'circumscribed diameter'.

WR7: Exactly one **representation_item** used for the implicit representation of a **ngon_closed_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'corner radius'.

Informal propositions:

IP1: The location of the **ngon_closed_profile** shall be defined at the center of the enclosed area.

IP2: The **ngon_closed_profile** shall be defined in the X-Y plane with one of the sides of the ngon parallel to the X direction intersecting the negative Y direction.

5.2.3.1.46 **ngon_shape_representation**

An **ngon_shape_representation** specifies representation of a shape that is a volume defined as a ngon area of a defined length. The enclosed area is defined by three or more straight sides. See ARM definition for Ngon_base_shape in paragraph 4.2.109 for more information.

EXPRESS specification:

```

*)
ENTITY ngon_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
WHERE
  WR1: SIZEOF (SELF.items) = 5;
  WR2: SIZEOF (QUERY (it <* SELF.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')))) = 1;
  WR3: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'length')))) = 1;
  WR4: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'corner radius')))) = 1;
  WR5: SIZEOF (QUERY (it <* SELF.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'circumscribed diameter')))) = 1;
  WR6: SIZEOF (QUERY (it <* SELF.items |
    ('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND ('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
    TYPEOF (it\measure_with_unit.value_component)) AND
    (it.name = 'number of sides')))) = 1;
END_ENTITY;
(

```

Formal propositions:

WR1: The **ngon_shape_representation** shall contain exactly five **representation_items** in its set of **items**.

WR2: One of the **representation_items** 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_items** 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_items** 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_items** 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'.

WR6: One of the **representation_items** 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.

5.2.3.1.47 open_path_profile

An **open_path_profile** is a type of **shape_aspect** that represents a connected set of curves with a location and orientation. The start vertex and end vertex for the set of curves do not connect creating an area open on one side. The **open_path_profile** orientation is in the X-Y plane. See ARM definition for General_open_profile in paragraph 4.2.77 for more information.

EXPRESS specification:

```
*)
ENTITY open_path_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```



```

        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
        (it.name = 'orientation')))) = 1))) = 0))) = 0;
WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
END_ENTITY;
(*)

```

Formal propositions:

WR1: The **open_path_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **open_path_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **open_path_profile** shall contain one **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of an **open_path_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: The **open_path_profile** shall have exactly one **path_shape_representation**.

Informal propositions:

IP1: The **open_path_profile** shall be defined in the X-Y plane.

5.2.3.1.48 ordered_part

An **ordered_part** represents the measured amount of product that is to be manufactured.

EXPRESS specification:

```

*)
ENTITY ordered_part
    SUBTYPE OF (action_assignment, characterized_object);
    items : SET [1:?] OF feature_based_pp_ordered_item;
WHERE
    WR1: SIZEOF (USEDIN (SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION.DEFINITION')) = 1;

```

```

WR2: sizeof (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (sizeof (USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1))) = 0;
WR3: sizeof (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    NOT (sizeof (items) = 1))) = 0))) = 0;
WR4: sizeof (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (sizeof (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    NOT (sizeof (QUERY (it <* pdr.used_representation_items |
    ('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND (it.name = 'quantity')) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Attribute definition:

items: the set of **product_definition_formation**s for which an **action** is being assigned.

Formal propositions:

WR1: The **ordered_part** shall have exactly one property defined for it.

WR2: The property defined for the **ordered_part** shall have exactly one representation.

WR3: The representation of the property defined for the **ordered_part** shall contain exactly one **representation_item** in its set of **items**.

WR4: The **representation_item** in the set of **items** for the representation of the property for the **ordered_part** shall be of type **measure_representation_item** with a **name** of 'quantity'.

5.2.3.1.49 **outer_round**

An **outer_round** is a type of **feature_definition** that is the representation of a volume that is removed from the base shape. The **outer_round** may be represented in one of two ways. This first shall be a removal of a cylindrical volume represented by a diameter about the Z direction of the **feature_definition** placement axis. It may have a change in diameter which is represented by a **taper**. The second shall be a removal of a cylindrical volume represented by sweeping a **vee_profile** about the Z direction of the **feature_definition** placement axis. See ARM definition for Outer_diameter in paragraph 4.2.118 and Outer_diameter_to_shoulder in paragraph 4.2.119 for more information.

EXPRESS specification:

```

*)
ENTITY outer_round
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SELF.description IN ['outer diameter', 'outer diameter to shoulder'];
  WR2: (NOT (SELF.description = 'outer diameter')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') |
      NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
          (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
            'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF (it)) = 2) AND (it.name = 'length')) = 1))) = 0))) = 0);
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF (pdr.used_representation)) |
      NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF (it)) = 2) AND (it.name = 'diameter')) = 1))) = 0))) = 0;
  WR4: (NOT (SELF.description = 'outer diameter to shoulder')) OR
    (SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF (pd))) |
      NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
        (sa_occ.description = 'v-shape boundary occurrence') AND
        (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
          'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
          'RELATED_SHAPE_ASPECT') |
          (sar.description = 'profile usage') AND
          ('FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
          ('FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE' IN
          TYPEOF (sdr.relatng_shape_aspect)) AND
          (sdr.relatng_shape_aspect.description = 'v-shape')) =
          1))) = 1))) = 0);
  WR5: (NOT (SELF.description = 'outer diameter')) OR
    (SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |

```

```

'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
  'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
  (sa_occ.description = 'reduced size occurrence') AND
  (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
  'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
  'RELATED_SHAPE_ASPECT') |
  (sar.description = 'taper usage') AND
  ('FEATURE_BASED_PROCESS_PLANNING.' +
  'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar)))) |
  ('FEATURE_BASED_PROCESS_PLANNING.TAPER' IN
  TYPEOF (sdr.relate_shape_aspect)) AND
  (sdr.name = 'reduced size')))) = 1))) <= 1))) = 0);
END_ENTITY;
(*

```

Formal proposition:

WR1: The **outer_round** shall have a description of either 'outer diameter' or 'outer diameter to shoulder'.

WR2: If the description of the **outer_round** is 'outer diameter', exactly one **representation_item** used for the implicit representation of the **outer_round** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

WR3: Exactly one **representation_item** used for the implicit representation of the **outer_round** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'diameter'.

WR4: If the description of the **outer_round** is 'outer diameter to shoulder', the **outer_round** shall be the basis shape for at most one **shape_aspect** with a **description** of 'v-shape boundary occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **vee_profile** with a **description** of 'v-shape'.

WR5: If the description of the **outer_round** is 'outer diameter', the **outer_round** shall be the basis shape for at most one **shape_aspect** with a description of 'reduced size occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'taper usage' and a **relating_shape_aspect** that references a **taper** with a **description** of 'reduced size'.

5.2.3.1.50 outside_profile

An **outside_profile** is a type of **feature_definition** that is the representation of the outside boundary of the base shape. This boundary may be enclosed and have either a rectangular base shape that is represented by a **rectangular_closed_profile** or it may have a non-rectangular base shape that is represented by either a **ngon_closed_profile**, **rectangular_closed_profile**, **closed_path_profile**, or a **circular_closed_profile**. If the boundary is not enclosed the base shape would be defined by a **linear_profile**, **square_u_profile**, **partial_circular_profile**, **round_u_profile**, **vee_profile**, **tee_profile**, or an **open_path_profile**. See ARM definition for General_outside_profile in paragraph 4.2.78 for more information.

EXPRESS specification:

```

*)
ENTITY outside_profile
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (pd)) |
    NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
    (sa_occ.description = 'boundary occurrence') AND
    (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    (sar.description = 'profile usage') AND
    ('FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
    (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
    TYPEOF (sdr.relatng_shape_aspect)) = 1) AND
    (sdr.relatng_shape_aspect.description = 'outside boundary'))
    = 1))) = 1))) = 0;
END_ENTITY;
(*

```

Formal proposition:

WR1: The **outside_profile** the basis shape for exactly one **shape_aspect** with a **description** of 'boundary occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **circular_closed_profile**, **ngon_closed_profile**, **rectangular_closed_profile**, **closed_path_profile**, **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **linear_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** with a **description** of 'outside boundary'.

Informal propositions:

IP1: The **outside_profile** shall be linear along the Z-axis of the feature.

5.2.3.1.51 parallelism_tolerance

A **parallelism_tolerance** is a type of **geometric_tolerance_with_datum_reference** that defines the allowable deviation of a **shape_aspect** that may require **datum_reference** to control the definition. See ARM definition for Parallelism_tolerance in paragraph 4.2.121 for more information.

EXPRESS specification:

```
*)
ENTITY parallelism_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'parallelism';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) =
    1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **name** for the **parallelism_tolerance** shall be 'parallelism'.

WR2: There shall be exactly one **datum_reference** in the **datum_system** set for the **parallelism_tolerance**.

5.2.3.1.52 partial_circular_profile

A **partial_circular_profile** is a type of **shape_aspect** that is the representation of arc of constant radius, with a location and a position. The **partial_circular_profile** is located at the arc origin, the X direction starts at the arc origin in the direction of one of the arc end points and the Y direction orthogonal in the direction of volume removal. See ARM definition for Partial_circular_profile in paragraph 4.2.126 for more information.

EXPRESS specification:

```
*)
ENTITY partial_circular_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
```

```

IN TYPEOF (pdr.used_representation))) = 1))) = 0;
WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 3))) = 0))) = 0;
WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
(it.name = 'orientation')))) = 1))) = 0))) = 0;
WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it.name = 'radius')))) = 1))) = 0))) = 0;
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it.name = 'sweep angle')))) = 1)))
= 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **partial_circular_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **partial_circular_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **partial_circular_profile** shall contain three **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **partial_circular_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **partial_circular_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR6: Exactly one **representation_item** used for the implicit representation of a **partial_circular_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'sweep angle'.

Informal propositions:

IP1: The location of the **partial_circular_profile** shall be at the arc origin.

IP2: The **partial_circular_profile** shall be defined with the X direction starting at the arc origin in the direction of one of the arc end points and the Y direction orthogonal.

5.2.3.1.53 path_feature_component

A **path_feature_component** is a type of **shape_aspect** that is the representation 2D or 3D curve or set of curves. See ARM definition for Path in paragraph 4.2.127 for more information.

EXPRESS specification:

```
*)
ENTITY path_feature_component
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SELF.description IN ['partial circular', 'complete circular', 'linear',
    'complex'];
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR4: (NOT (SELF.description = 'partial circular')) OR
```



```

(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 3))) = 0))) = 0));
WR5: (NOT (SELF.description = 'partial circular')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'radius')) = 1))) = 0))) = 0);
WR6: (NOT (SELF.description = 'partial circular')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'sweep angle')) = 1))) = 0))) = 0);
WR7: (NOT (SELF.description = 'complete circular')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
WR8: (NOT (SELF.description = 'complete circular')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |

```

```

        (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'radius')) = 1))) = 0))) = 0);
WR9: (NOT (SELF.description = 'linear')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
WR10: (NOT (SELF.description = 'linear')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'distance')) = 1))) = 0))) = 0);
WR11: (NOT (SELF.description = 'linear')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation))) = 1))) = 0);
WR12: (NOT (SELF.description = 'complex')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
WR13: (NOT (SELF.description = 'complex')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        (pdr.used_representation.name = 'sweep path')) = 1))) = 0);

```

```

WR14: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
  'DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
  'FEATURE_BASED_PROCESS_PLANNING.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  'FEATURE_BASED_PROCESS_PLANNING.' +
  'SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF (pdr.used_representation)) |
  NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
  ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
  (it.name = 'orientation')))) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **path_feature_component** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **description** for the **path_feature_component** shall be either 'partial circular', 'complete circular', 'linear', or 'complex'.

WR3: The **path_feature_component** shall have its implicit representation specified by exactly one **shape_representation_with_parameters**.

WR4: If the **description** of the **path_feature_component** is 'partial circular', the **path_feature_component** shall have an implicit representation that contains exactly three **representation_items** in its set of **items**.

WR5: If the **description** of the **path_feature_component** is 'partial circular', exactly one **representation_item** used for the implicit representation of the **path_feature_component** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR6: If the **description** of the **path_feature_component** is 'partial circular', exactly one **representation_item** used for the implicit representation of the **path_feature_component** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'sweep angle'.

WR7: If the **description** of the **path_feature_component** is 'complete circular', the **path_feature_component** shall have an implicit representation that contains exactly two **representation_items** in its set of **items**.

WR8: If the **description** of the **path_feature_component** is 'complete circular', exactly one **representation_item** used for the implicit representation of the **path_feature_component** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR9: If the **description** of the **path_feature_component** is 'linear', the **path_feature_component** shall have an implicit representation that contains exactly two **representation_items** in its set of **items**.

WR10: If the **description** of the **path_feature_component** is 'linear', exactly one **representation_item** used for the implicit representation of the **path_feature_component** shall be of type **measure_-representation_item** and **length_measure_with_unit** with a **name** of 'distance'.

WR11: If the **description** of the **path_feature_component** is 'linear', the **path_feature_component** shall have exactly one **direction_shape_representation** as its **related_shape_aspect**.

WR12: If the **description** of the **path_feature_component** is 'complex', the **path_feature_component** shall have an implicit representation that contains exactly one **representation_item** in its set of **items**.

WR13: If the **description** of the **path_feature_component** is 'complex', the **path_feature_component** shall have exactly one **path_shape_representation** with a **name** of 'sweep path'.

WR14: Exactly one **representation_item** used for the implicit representation of the **path_feature_-component** shall be of type **placement** with a **name** of 'orientation'.

Informal propositions:

IP1: The location of the **path_feature_component** may be the same location used by the feature that references **path_feature_component**.

IP2: The origin, X direction, and Y direction of the **path_feature_component** may be the same as that used by the feature that references **path_feature_component**.

5.2.3.1.54 path_shape_representation

A **path_shape_representation** is a type of **representation** which represents a curved path using an ordered collection of geometric curves such that the collection of curves are continuous and directed.

EXPRESS specification:

```

*)
ENTITY path_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF (SELF.items) = 1;
  WR2: SIZEOF (QUERY (it <* SELF.items |
    NOT ('FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF
      (it)))) = 0;
  WR3: SIZEOF (QUERY (it <* SELF.items |
    NOT (SIZEOF (QUERY (el <* it\path.edge_list |
      NOT ('FEATURE_BASED_PROCESS_PLANNING.EDGE_CURVE' IN
        TYPEOF (el.edge_element)))) = 0))) = 0;
END_ENTITY;
(

```

Formal propositions:

WR1: The **path_shape_representation** shall have exactly one **representation_item** in its set of **items**.

WR2: The geometric element that is used to represent the **path_shape_representation** shall be an **edge_loop**.

WR3: The **edges** that comprise the **edge_loop** shall contain the curves that define their shape.

5.2.3.1.55 pattern_offset_membership

A **pattern_offset_membership** specifies the relationships necessary to offset a base feature from either a **rectangular_pattern** or a **circular_pattern**.

EXPRESS specification:

```

*)
ENTITY pattern_offset_membership
  SUBTYPE OF (feature_component_relationship);
WHERE
  WR1: SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
    'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
    TYPEOF (SELF.relate_shape_aspect)) = 1;
  WR2: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF (sar)) AND (sar :<=: SELF)) |
    'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN
    TYPEOF (fcr.relate_shape_aspect))) >= 1;
  WR3: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF (sar)) AND (sar :<=: SELF)) |
    ('FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN
    TYPEOF (fcr.relate_shape_aspect)) AND
    (NOT (SIZEOF (QUERY (modfcr <* QUERY (modsar <*
    USEDIN (fcr.relate_shape_aspect, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
    'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
    TYPEOF (modsar.related_shape_aspect)) = 1) AND
    (modsar :<=: fcr)) |
    NOT (modfcr.related_shape_aspect == SELF.relate_shape_aspect))))
    = 0)))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)))) = 0;

```

WR5: (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
 TYPEOF (SELF.relate_shape_aspect))) OR
 (SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 NOT (SIZEOF (pdr.used_representation.items) = 2))) = 0))) = 0);
 WR6: (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
 TYPEOF (SELF.relate_shape_aspect))) OR
 (SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF (pdr.used_representation)) |
 NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
 ('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
 TYPEOF (it)) AND
 ('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
 TYPEOF (it.measure_with_unit.value_component)) AND
 (it.name = 'index number')))) = 1))) = 0))) = 0);
 WR7: (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
 TYPEOF (SELF.relate_shape_aspect))) OR
 (SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF (pdr.used_representation)) |
 NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
 ('FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT' IN
 TYPEOF (it)) AND (it.name = 'offset')))) = 1))) = 0))) = 0);
 WR8: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
 TYPEOF (SELF.relate_shape_aspect))) OR
 (SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 NOT (SIZEOF (pdr.used_representation.items) = 3))) = 0))) = 0);
 WR9: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
 TYPEOF (SELF.relate_shape_aspect))) OR
 (SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
 NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 'FEATURE_BASED_PROCESS_PLANNING.' +
 'SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF (pdr.used_representation)) |

```

NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'row index')) = 1))) = 0))) = 0);
WR10: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF (SELF.relatng_shape_aspect))) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'column index')) = 1))) = 0))) = 0);
WR11: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF (SELF.relatng_shape_aspect))) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT' IN
TYPEOF (it)) AND (it.name = 'offset distance')) = 1))) = 0))) = 0);
WR12: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF (SELF.relatng_shape_aspect))) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
IN TYPEOF (pdr.used_representation)) AND
(pdr.used_representation.name = 'offset direction')) = 1))) = 0);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **pattern_offset_membership** shall define an offset relationship for a member of either a **circular_pattern** or a **rectangular_pattern**

WR2: The **shape_aspect** that defines the offset in the **pattern_offset_membership** shall be incorporated in at least one **modified_pattern**.

WR3: The **pattern_offset_membership** shall define a set of entity instances such that the **related_shape_aspect** defines the offset which is used as the **related_shape_aspect** in an instance of a **feature_component_relationship** that relates the offset to the **modified_pattern** that incorporates all of the offsets for exactly one **circular_pattern** or **rectangular_pattern**.

NOTE - Figure 104 depicts the instances defined by this constraint.

WR4: The **shape_aspect** that is referenced by **related_shape_aspect** of the **pattern_offset_membership** shall have exactly one **representation**.

WR5: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_offset_membership** is a **circular_pattern**, the **representation** of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_offset_membership** shall contain two **representation_items** in its set of **items**.

WR6: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_offset_membership** is a **circular_pattern**, exactly one **representation_item** used for the representation of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_offset_membership** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'index number'.

WR7: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_offset_membership** is a **circular_pattern**, exactly one **representation_item** used for the representation of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_offset_membership** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'offset'.

WR8: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_offset_membership** is a **rectangular_pattern**, the **representation** of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_offset_membership** shall contain three **representation_items** in its set of **items**.

WR9: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_offset_membership** is a **rectangular_pattern**, exactly one **representation_item** used for the representation of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_offset_membership** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'row index'.

WR10: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_offset_membership** is a **rectangular_pattern**, exactly one **representation_item** used for the representation of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_offset_membership** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'column index'.

WR11: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_offset_membership** is a **rectangular_pattern**, exactly one **representation_item** used for the representation of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_offset_membership** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'offset distance'.

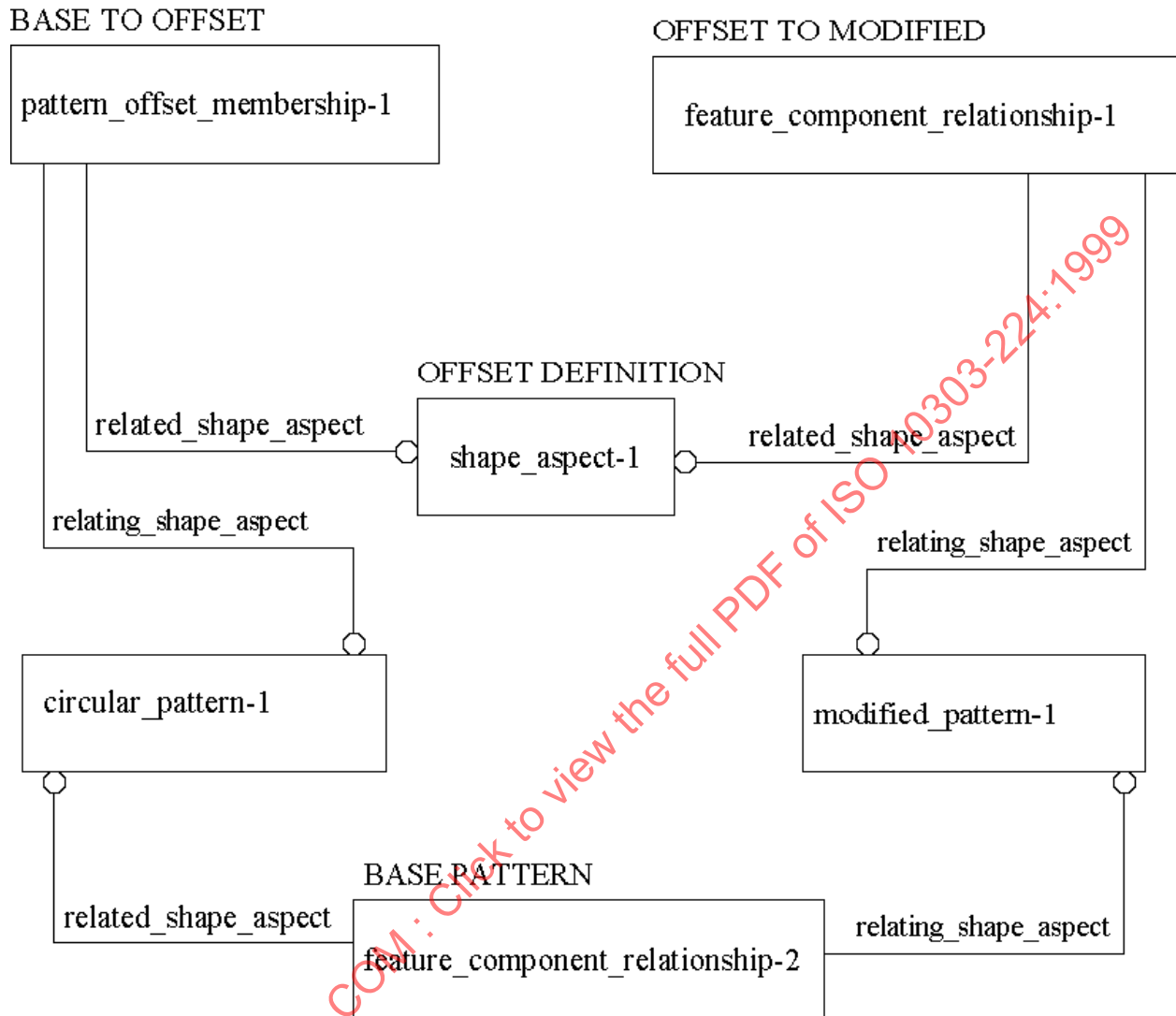


Figure 104 - Pattern offset required instances

WR12: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_offset_membership** is a **rectangular_pattern**, the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_offset_membership** shall have exactly one **direction_shape_representation** with a **name** of 'offset direction'.

5.2.3.1.56 pattern_omit_membership

A **pattern_omit_membership** specifies the relationships necessary to omit a base feature from either a **rectangular_pattern** or a **circular_pattern**.

EXPRESS specification:

```

*)
ENTITY pattern_omit_membership
  SUBTYPE OF (feature_component_relationship);
WHERE
  WR1: SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
    'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
    TYPEOF (SELF.relate_shape_aspect)) = 1;
  WR2: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF (sar)) AND (sar :<=: SELF)) |
    'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN
    TYPEOF (fcr.relate_shape_aspect))) >= 1;
  WR3: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF (sar)) AND (sar :<=: SELF)) |
    ('FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN
    TYPEOF (fcr.relate_shape_aspect)) AND
    (NOT (SIZEOF (QUERY (modfcr <* QUERY (modsar <*
    USEDIN (fcr.relate_shape_aspect, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
    'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
    TYPEOF (modsar.related_shape_aspect)) = 1) AND
    (modsar :<=: fcr)) |
    NOT (modfcr.related_shape_aspect :=: SELF.relate_shape_aspect))))
    = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1))) = 0;
  WR5: (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
    TYPEOF (SELF.relate_shape_aspect))) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    NOT (SIZEOF (pdr.used_representation.items) = 1))) = 0))) = 0);
  WR6: (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
    TYPEOF (SELF.relate_shape_aspect))) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |

```

```

NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'index number')) = 1))) = 0))) = 0);
WR7: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF (SELF.relate_shape_aspect))) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
NOT (SIZEOF (pdr.used_representation.items) = 2))) = 0))) = 0);
WR8: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF (SELF.relate_shape_aspect))) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'row index')) = 1))) = 0))) = 0);
WR9: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF (SELF.relate_shape_aspect))) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'column index')) = 1))) = 0))) = 0);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **pattern_omit_membership** shall define an omit relationship for a member of either a **circular_-pattern** or a **rectangular_pattern**

WR2: The **shape_aspect** that defines the offset in the **pattern_omit_membership** shall be incorporated in at least one **modified_pattern**.

WR3: The **pattern_omit_membership** shall define a set of entity instances such that the **related_shape_aspect** defines the offset which is used as the **related_shape_aspect** in an instance of a **feature_component_relationship** that relates the offset to the **modified_pattern** that incorporates all of the offsets for exactly one **circular_pattern** or **rectangular_pattern**.

NOTE - Figure 105 depicts the instances defined by this constraint.

WR4: The **shape_aspect** that is referenced by **related_shape_aspect** of the **pattern_omit_membership** shall have exactly one **representation**.

WR5: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_omit_membership** is a **circular_pattern**, the **representation** of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_omit_membership** shall contain one **representation_item** in its set of **items**.

WR6: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_omit_membership** is a **circular_pattern**, exactly one **representation_item** used for the representation of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_omit_membership** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'index number'.

WR7: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_omit_membership** is a **rectangular_pattern**, the **representation** of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_omit_membership** shall contain two **representation_items** in its set of **items**.

WR8: If the **shape_aspect** referenced by the **relating_shape_aspect** of the **pattern_omit_membership** is a **rectangular_pattern**, exactly one **representation_item** used for the representation of the **shape_aspect** referenced by the **related_shape_aspect** of the **pattern_omit_membership** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'row index'.

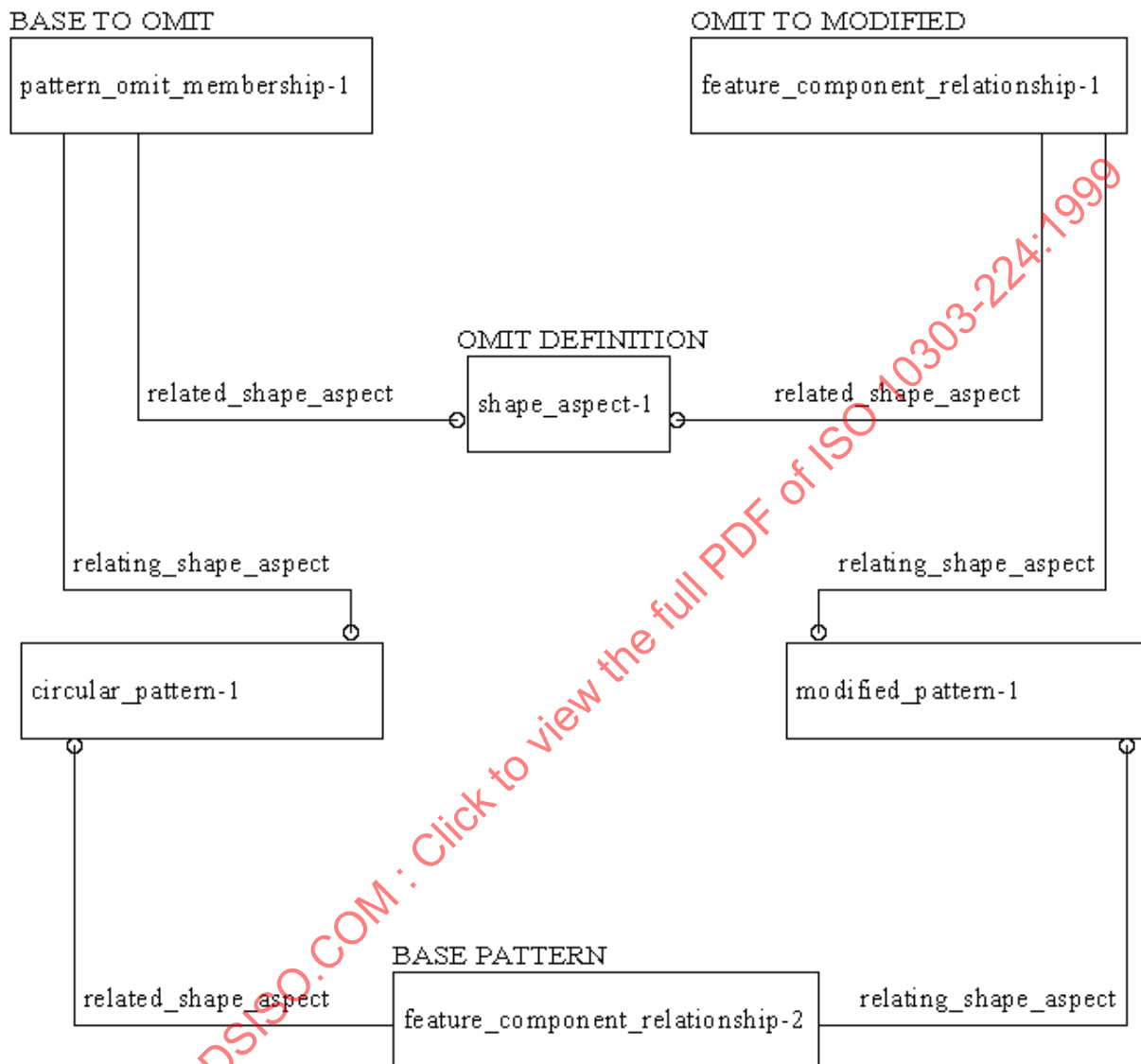


Figure 105 - Pattern omit required instances

WR9: If the `shape_aspect` referenced by the `relating_shape_aspect` of the `pattern_omit_membership` is a **rectangular_pattern**, exactly one **representation_item** used for the representation of the `shape_aspect` referenced by the `related_shape_aspect` of the `pattern_omit_membership` shall be of type **measure_-representation_item** with a `value_component` of type **count_measure** and a `name` of 'column index'.

5.2.3.1.57 perpendicularity_tolerance

A **perpendicularity_tolerance** is a type of **geometric_tolerance_with_datum_reference** that defines the allowable deviation of a **shape_aspect** that may require **datum_reference** to control the definition. See ARM definition for Perpendicularity_tolerance in paragraph 4.2.130 for more information.

EXPRESS specification:

```
*)
ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'perpendicularity';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system)
    = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **name** for the **perpendicularity_tolerance** shall be 'perpendicularity'.

WR2: There shall be exactly one **datum_reference** in the **datum_system** set for the **perpendicularity_tolerance**.

5.2.3.1.58 placed_datum_target_feature

A **placed_datum_target_feature** is a type of **datum_target** that represents the implicit definition. See ARM definition for Placed_target in paragraph 4.2.133 for more information.

EXPRESS specification:

```
*)
ENTITY placed_datum_target_feature
  SUBTYPE OF (datum_target);
WHERE
  WR1: SELF.description IN ['point', 'line', 'rectangle', 'circle'];
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (dtm_rep <* QUERY (pdr <* USEDIN (pd,
```

```

'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* dtm_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
(it.name = 'orientation')))) = 1))) = 0))) = 0;
WR4: (NOT (SELF.description = 'point')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (dtm_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (dtm_rep.used_representation.items) = 1))) = 0))) = 0))) = 0);
WR5: (NOT (SELF.description IN ['line', 'circle'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (dtm_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (dtm_rep.used_representation.items) = 2))) = 0))) = 0))) = 0);
WR6: (NOT (SELF.description = 'rectangle')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (dtm_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (dtm_rep.used_representation.items) = 3))) = 0))) = 0))) = 0);
WR7: (NOT (SELF.description = 'circle')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (dtm_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* dtm_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it.name = 'target diameter')))) = 1))) = 0))) = 0);

```

```

WR8: (NOT (SELF.description = 'line')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (dtm_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* dtm_rep.used_representation.items |
        (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it))) = 2) AND (it.name = 'target length')) = 1))) = 0))) = 0);
WR9: (NOT (SELF.description = 'rectangle')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (dtm_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* dtm_rep.used_representation.items |
        (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it))) = 2) AND (it.name = 'target length')) = 1))) = 0))) = 0);
WR10: (NOT (SELF.description = 'rectangle')) OR
      (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (dtm_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* dtm_rep.used_representation.items |
        (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it))) = 2) AND (it.name = 'target width')) = 1))) = 0))) = 0);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **description** for the **placed_datum_target_feature** shall be either 'point', 'line', 'rectangle' or 'circle'.

WR2: A **placed_datum_target_feature** shall have exactly one implicit representation.

WR3: Exactly one **representation_item** used for the representation of the **placed_datum_target_feature** shall be of type **placement** with a **name** of 'orientation'.

WR4: If the **placed_datum_target_feature** is a point, the **representation** shall contain exactly one **representation_items** in its set of **items**.

WR5: If the **placed_datum_target_feature** is a line or circle, the **representation** shall contain exactly two **representation_items** in its set of **items**.

WR6: If the **placed_datum_target_feature** is a rectangle, the **representation** shall contain exactly three **representation_items** in its set of **items**.

WR7: If the **description** of the **placed_datum_target_feature** is 'circle', exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_-representation_item** and **length_measure_with_unit** with a **name** of 'target diameter'.

WR8: If the **description** of the **placed_datum_target_feature** is 'line', exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_-representation_item** and **length_measure_with_unit** with a **name** of 'target length'.

WR9: If the **description** of the **placed_datum_target_feature** is 'rectangle', exactly one **representation_-item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_-representation_item** and **length_measure_with_unit** with a **name** of 'target length'.

WR10: If the **description** of the **placed_datum_target_feature** is 'rectangle', exactly one **representation_-item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_-representation_item** and **length_measure_with_unit** with a **name** of 'target width'.

5.2.3.1.59 pocket

A **pocket** is a type of **feature_definition** that is the representation of a volume that is removed from the base shape. This volume may be enclosed and have either a rectangular base shape that is represented by a **rectangular_closed_profile** or it may have a non_rectangular base shape that is represented by either a **ngon_closed_profile**, **rectangular_closed_profile**, **closed_path_profile**, or a **circular_closed_profile**. If the volume is not enclosed the base shape would be defined by a **square_u_profile**, **partial_circular_-profile**, **round_u_profile**, **vee_profile**, **tee_profile**, or an **open_path_profile**. The bottom of the pocket is represented by a **pocket_bottom**. See ARM definition for General_pocket in paragraph 4.2.81 and Rectangular_pocket in paragraph 4.2.154 for more information.

EXPRESS specification:

```
*)
ENTITY pocket
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SELF.description IN ['closed rectangular', 'open rectangular',
    'non-rectangular'];
  WR2: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') ) |
```

```

'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
  'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
  (sa_occ.description = 'pocket depth occurrence') AND
  (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
  'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
  'RELATED_SHAPE_ASPECT') |
  (sar.description = 'path feature component usage') AND
  ('FEATURE_BASED_PROCESS_PLANNING.' +
  'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
  ('FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
  TYPEOF (sdr.relate_shape_aspect))) AND
  (sdr.name = 'pocket depth') AND
  (sdr.relate_shape_aspect.description = 'linear')))) = 1))) = 1))) = 0;
WR3: (NOT (SELF.description = 'non-rectangular')) OR
  (SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION_DEFINITION') |
  'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF (pd)) |
  NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
  'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
  (sa_occ.description = 'boundary occurrence') AND
  (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
  'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
  'RELATED_SHAPE_ASPECT') |
  (sar.description = 'profile usage') AND
  ('FEATURE_BASED_PROCESS_PLANNING.' +
  'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
  (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
  'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
  TYPEOF (sdr.relate_shape_aspect)) = 1) AND
  (sdr.relate_shape_aspect.description = 'boundary'))))
  = 1))) = 1))) = 0);
WR4: (NOT (SELF.description = 'closed rectangular')) OR
  (SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION_DEFINITION') |
  'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF (pd)) |
  NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
  'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
  (sa_occ.description = 'closed boundary occurrence') AND
  (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
  'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
  'RELATED_SHAPE_ASPECT') |
  (sar.description = 'profile usage') AND
  ('FEATURE_BASED_PROCESS_PLANNING.' +

```

```

'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE'
IN TYPEOF (sdr.relate_shape_aspect)) AND
(sdr.relate_shape_aspect.description = 'boundary'))))
= 1))) = 1))) = 0);
WR5: (NOT (SELF.description = 'open rectangular')) OR
(SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'open boundary occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE'
IN TYPEOF (sdr.relate_shape_aspect)) AND
(sdr.relate_shape_aspect.description = 'boundary'))))
= 1))) = 1))) = 0);
WR6: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'bottom condition occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'pocket bottom usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar))) |
'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM' IN
TYPEOF (sdr.relate_shape_aspect))) = 1))) = 1))) = 0;
WR7: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'change in boundary occurrence') AND
(SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') |
(sar.description = 'taper usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.TAPER' IN
TYPEOF (fcr.related_shape_aspect)) AND
(fcr.related_shape_aspect.description = 'angle taper'))))
= 1))) <= 1))) = 0;

```

END_ENTITY;
(*

Formal proposition:

WR1: The **pocket** shall have a description of 'open rectangular', 'closed rectangular', or 'non-rectangular'.

WR2: The **pocket** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'pocket depth occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'path feature component usage' and a **relating_shape_aspect** that references a **path_feature_component** with a **name** of 'pocket depth' and a **description** of 'linear'.

WR3: If the **pocket** has a description of 'non-rectangular', the **pocket** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'boundary occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **circular_closed_profile**, **ngon_closed_profile**, **rectangular_closed_profile**, **closed_path_profile**, **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** with a **description** of 'boundary'.

WR4: If the **pocket** has a description of 'closed rectangular', the **pocket** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'closed boundary occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **rectangular_closed_profile** with a **description** of 'boundary'.

WR5: If the **pocket** has a description of 'open rectangular', the **pocket** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'open boundary occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **square_u_profile** with a **description** of 'boundary'.

WR6: The **pocket** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'bottom condition occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'pocket bottom usage' and a **relating_shape_aspect** that references a **pocket_bottom**.

WR7: The **pocket** shall be the basis shape for at most one **shape_aspect** with a **description** of 'change in boundary occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'taper usage' in which the **relating_shape_aspect** is a **taper** with a **description** of 'angle taper'.

5.2.3.1.60 pocket_bottom

A **pocket_bottom** is a type of **shape_aspect** that is the representation of the end condition for a **pocket feature_definition**. See ARM definition for **Pocket_bottom_condition** in paragraph 4.2.139 for more information.

EXPRESS specification:

```

*)
ENTITY pocket_bottom
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SELF.description IN ['planar', 'non-planar', 'through'];
  WR3: (NOT (SELF.description = 'non-planar')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') |
      NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation))) = 1))) = 1);
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR5: (NOT (SELF.description IN ['planar', 'non-planar'])) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
  WR6: (NOT (SELF.description = 'through')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
  WR7: (NOT (SELF.description IN ['planar', 'non-planar'])) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'

```

```

IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'radius')) = 1))) = 0))) = 0));
WR8: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') |
(sar.description = 'pocket bottom usage') AND
('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF (sar))) |
(fcr.related_shape_aspect.description = 'bottom condition occurrence')
AND
('FEATURE_BASED_PROCESS_PLANNING.POCKET' IN TYPEOF
(fcr.related_shape_aspect.of_shape.definition)))) >= 1;
WR9: ((NOT (SELF.description IN ['planar', 'non-planar'])) OR
(SIZEOF(QUERY (pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION_DEFINITION')
| (NOT (SIZEOF(QUERY (impl_rep <* QUERY (pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY (it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'pocket bottom orientation')
AND (it.description IN ['pocket start', 'pocket end']))) )) =
1)) )) = 0)) )) = 0));
END_ENTITY;
(*

```

Formal propositions:

WR1: The **pocket_bottom** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **description** of the **pocket_bottom** shall be either 'planar', 'non-planar', or 'through'.

WR3: If the **description** of the **pocket_bottom** is 'non-planar', the **pocket_bottom** shall have exactly one **face_shape_representation**.

WR4: The **pocket_bottom** shall have its implicit representation specified by exactly one **shape_-representation_with_parameters**.

WR5: If the **description** of the **pocket_bottom** is 'planar' or 'non-planar', the **pocket_bottom** shall be represented implicitly by exactly two **representation_items**.

WR6: If the **description** of the **pocket_bottom** is 'through', the **pocket_bottom** shall be represented implicitly by exactly one **representation_item**.

WR7: If the **description** of the **pocket_bottom** is 'planar' or 'non-planar', exactly one **representation_item** used for the implicit representation of the **pocket_bottom** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR8: The **pocket_bottom** shall be the **relating_shape_aspect** in at least one **feature_component_relationship** with a **description** of 'pocket bottom usage' in which the **related_shape_aspect** is an aspect of the shape of a **pocket** with a **description** of 'bottom condition occurrence'.

WR9: If the **description** of the **pocket_bottom** is 'planar' or 'non-planar', the implicit representation of the **pocket_bottom** shall have exactly one **representation_item** in its **items** set which is a **descriptive_representation_item** with a **name** of 'pocket bottom orientation' and a **description** of either 'pocket start' or 'pocket end'.

Informal propositions:

IP1: The location of the **pocket_bottom** shall be at the approximate center of the mating **pocket** feature.

IP2: The **pocket_bottom** shall be defined with the volume removal in Z direction of the **pocket** feature.

5.2.3.1.61 position_tolerance

A **position_tolerance** is a type of **geometric_tolerance_with_datum_reference** that defines the allowable deviation of a **shape_aspect** that may require **datum_reference** to control the definition.. See ARM definition for Position_tolerance in paragraph 4.2.140 for more information.

EXPRESS specification:

```
*)
ENTITY position_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'position';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <=
    3;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **name** for the **position_tolerance** shall be 'position'.

WR2: There shall be at most three **datum_references** in the **datum_system** set for the **position_tolerance**.

5.2.3.1.62 protrusion

A **protrusion** is a type of **feature_definition** that is the representation of a volume that is extruding from the base shape. This volume shall be enclosed, except for the mating surface which may or may not be included in the definition of the volume, and have a shape defining representation. See ARM definition for Protrusion in paragraph 4.2.147 for more information.

EXPRESS specification:

```
*)
ENTITY protrusion
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (pd)) |
    NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
    (sa_occ.description = 'shape volume occurrence') AND
    (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    (sar.description = 'volume shape usage') AND
    ('FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
    sdr.relate_shape_aspect.description = 'volume shape'))
    = 1))) = 1))) = 0;
END_ENTITY;
(*
```

Formal proposition:

WR1: The **protrusion** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'shape volume occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'volume shape usage' and **relating_shape_aspect** with a **description** of 'volume shape'.

5.2.3.1.63 rectangular_closed_profile

A **rectangular_closed_profile** is a type of **shape_aspect** that represents an enclosed 2D area with location and orientation. The enclosed area is defined by four straight sides with opposite sides equal in length. The location is defined to be at the center of the rectangle, and the orientation of the **rectangular_closed_profile** is the X-Y plane with the X direction along the length of the rectangle and the Y direction orthogonal to the X direction in the direction of the width. See ARM definition for Rectangular_closed_profile in paragraph 4.2.151 for more information.

EXPRESS specification:

```

*)
ENTITY rectangular_closed_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 4))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')) = 1))) = 0))) = 0;
  WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'width')) = 1))) = 0))) = 0;
  WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |

```

```

        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'length')))) = 1))) = 0))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'corner radius')))) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **rectangular_closed_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **rectangular_closed_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **rectangular_closed_profile** shall contain four **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **rectangular_closed_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **rectangular_closed_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

WR6: Exactly one **representation_item** used for the implicit representation of a **rectangular_closed_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

WR7: Exactly one **representation_item** used for the implicit representation of a **rectangular_closed_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'corner radius'.

Informal propositions:

IP1: The location of the **rectangular_closed_profile** shall be defined at the center of the rectangle.

IP2: The **rectangular_closed_profile** shall be defined in the X-Y plane with the length in the X direction and the width in the Y direction of the rectangle. The Xdirection is orthogonal to the Y direction.

5.2.3.1.64 rectangular_pattern

A **rectangular_pattern** is a type of **replicate_feature** that relates a base feature and one or more **shape_aspects** which are the placement of the base feature at a specified row and column location on the base part. A position is defined on the part and a number of base features are equally place in rows and columns. See ARM definition for Rectangular_pattern in paragraph 4.2.155 for more information.

EXPRESS specification:

```

*)
ENTITY rectangular_pattern
  SUBTYPE OF (replicate_feature);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')) <= 5;
  WR2: SIZEOF (QUERY (sar <* USEDIN (SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
    NOT (SIZEOF (
      ['FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP',
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'] *
      TYPEOF (sar)) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
      IN TYPEOF (pdr.used_representation)) AND
      (pdr.used_representation.name = 'row layout direction')))) = 1))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
      IN TYPEOF (pdr.used_representation)) AND
      (pdr.used_representation.name = 'column layout direction'))))
    = 1))) = 0;
  WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,

```

```

'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) = 1))) = 0;
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 5))) = 0))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'number of rows')) = 1))) = 0))) = 0;
WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'number of columns')) = 1))) = 0))) = 0;
WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'row spacing')) = 1))) = 0))) = 0;
WR10: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'

```

```

    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND (it.name = 'column spacing')) = 1))) = 0)))
    = 0;
WR11: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN
    TYPEOF (it)) AND (it.name = 'orientation')) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **rectangular_pattern** shall be the **relating_shape_aspect** in no more than five **shape_aspect_-relationships**.

WR2: The **rectangular_pattern** shall be the **relating_shape_aspect** in only **feature_component_-relationships** or **shape_defining_relationships**.

WR3: The **rectangular_pattern** shall have exactly one **direction_shape_representation** with a **name** of 'row layout direction'.

WR4: The **rectangular_pattern** shall have exactly one **direction_shape_representation** with a **name** of 'column layout direction'.

WR5: The **rectangular_pattern** shall have exactly one implicit representation.

WR6: The implicit representation of the **rectangular_pattern** shall contain five **representation_items** in its set of **items**.

WR7: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_-item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of rows'.

WR8: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_-item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of columns'.

WR9: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_-item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'row spacing'.

WR10: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'column spacing'.

WR11: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_item** of type **placement** with a **name** of 'orientation'.

Informal propositions:

IP1: The location of the **rectangular_pattern** shall be defined at the position of the first base feature.

IP2: The **rectangular_pattern** shall be defined in the X-Y plane.

5.2.3.1.65 removal_volume

A **removal_volume** is a type of **feature_definition** that is the representation of a volume that is removed from the base shape. This volume may or may not be enclosed but in either case shall have a shape defining representation. See ARM definition for General_removal_volume in paragraph 4.2.83 for more information.

EXPRESS specification:

```
*)
ENTITY removal_volume
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (pd)) |
    NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
    (sa_occ.description = 'shape volume occurrence') AND
    (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') |
    (sar.description = 'volume shape usage') AND
    ('FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
    sdr.relatng_shape_aspect.description = 'volume shape'))
    = 1))) = 1))) = 0;
END_ENTITY;
(*
```

Formal proposition:

WR1: The **removal_volume** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'shape volume occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'volume shape usage' and a **relating_shape_aspect** with a **description** of 'volume shape'.

5.2.3.1.66 replicate_feature

A **replicate_feature** is a type of **shape_aspect** that is either a **feature_definition** or another **replicate_feature** for the purpose of multiple representations.

EXPRESS specification:

```
*)
ENTITY replicate_feature
  SUPERTYPE OF (ONEOF (circular_pattern, rectangular_pattern,
    modified_pattern, feature_pattern))
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') |
    'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF (sar)) |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
    'FEATURE_BASED_PROCESS_PLANNING.MACHINING_FEATURE'] *
    TYPEOF (fcr.related_shape_aspect)) = 1) AND
    (fcr.name = 'pattern basis')) = 1;
END_ENTITY;
(*
```

Formal proposition:

WR1: The **replicate_feature** shall be the **relating_shape_aspect** in exactly one **shape_aspect_relationship** that is of type **feature_component_relationship** with a name of 'pattern basis' and a **related_shape_aspect** that is either of type **replicate_feature** or **machining_feature**.

5.2.3.1.67 revolved_profile

A **revolved_profile** is a type of **feature_definition** that is the representation of a volume that is removed from the base shape. The volume of removal is defined by either sweeping a linear_profile about an axis (see ARM definition for Revolved_flat in 4.2.161) or a partial_circular_profile about an axis (see ARM definition of Revolved_round in 4.2.162) or an open_path_profile about an axis (see ARM definition of General_revolution in 4.2.84) or open_profile about an axis (see ARM definition of Groove in 4.2.88).

EXPRESS specification:

```
*)
ENTITY revolved_profile
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SELF.description IN ['groove', 'flat', 'round', 'open profile'];
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
```

```

'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it))) = 2) AND (it.name = 'radius')) = 1))) = 0))) = 0;
WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
IN TYPEOF (pdr.used_representation)) AND
(pdr.used_representation.name = 'removal direction')) = 1))) = 0;
WR4: (NOT (SELF.description = 'open profile')) OR
(SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'outer edge shape occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'
IN TYPEOF (sdr.relatng_shape_aspect)) AND
(sdr.relatng_shape_aspect.description = 'outer edge shape'))
= 1))) = 1))) = 0);
WR5: (NOT (SELF.description = 'flat')) OR
(SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'flat edge shape occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE'
IN TYPEOF (sdr.relatng_shape_aspect)) AND
(sdr.relatng_shape_aspect.description = 'flat edge shape'))
= 1))) = 1))) = 0);
WR6: (NOT (SELF.description = 'round')) OR

```



```

(SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd))) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'rounded edge shape occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
('FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE'
IN TYPEOF (sdr.relatng_shape_aspect)) AND
(sdr.relatng_shape_aspect.description = 'rounded edge shape'))
= 1))) = 1))) = 0);
WR7: (NOT (SELF.description = 'groove')) OR
(SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd))) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'sweep occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'sweep usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
(SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
TYPEOF (sdr.relatng_shape_aspect)) = 1) AND
(sdr.relatng_shape_aspect.description = 'sweep')) = 1))) = 1))) = 0);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **revolved_profile** shall have a description of 'groove', 'flat', 'round', or 'open profile'.

WR2: The **revolved_profile** shall contain exactly one **representation_item** in its set of **items** which is a **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR3: The **revolved_profile** shall have exactly one **direction_shape_representation** with a **name** of 'removal direction'.

WR4: If the **revolved_profile** has a description of 'open profile', the **revolved_profile** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'outer edge shape occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references an **open_path_profile** with a **description** of 'outer edge shape'.

WR5: If the **revolved_profile** has a description of 'flat', the **revolved_profile** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'flat edge occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **linear_profile** with a **description** of 'flat edge shape'.

WR6: If the **revolved_profile** has a description of 'round', the **revolved_profile** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'rounded edge shape occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references an **partial_circular_profile** with a **description** of 'rounded edge shape'.

WR7: If the **revolved_profile** has a description of 'groove', the **revolved_profile** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'sweep occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** with a **description** of 'sweep'.

Informal proposition:

IP1: If the orientation of the **revolved_profile** causes the feature to be self intersecting, the intersecting edges shall be trimmed at the axis of revolution.

5.2.3.1.68 round_hole

A **round_hole** is a type of **feature_definition** that is the representation of a circular enclosed volume this is removed from the base shape. The bottom of the **round_hole** is represented by a **hole_bottom**. A **round_hole** may have a change in diameter which is represented by a **taper**. See ARM definition for Round_hole in paragraph 4.2.163 for more information.

EXPRESS specification:

```
*)
ENTITY round_hole
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (pd)) |
    NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
    (sa_occ.description = 'diameter occurrence') AND
```

```

(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT')) |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE' IN
TYPEOF (sdr.relateing_shape_aspect)) AND
(sdr.name = 'diameter')))) = 1))) = 1))) = 0;
WR2: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'hole depth occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT')) |
(sar.description = 'path feature component usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
TYPEOF (sdr.relateing_shape_aspect)) AND
(sdr.name = 'hole depth') AND
(sdr.relateing_shape_aspect.description = 'linear')))) = 1))) = 1))) = 0;
WR3: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'bottom condition occurrence') AND
(SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT')) |
(sar.description = 'hole bottom usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar))) |
'FEATURE_BASED_PROCESS_PLANNING.HOLE_BOTTOM' IN
TYPEOF (fcr.relateing_shape_aspect)))) = 1))) = 1))) = 0;
WR4: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'change in diameter occurrence') AND
(SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT')) |
(sar.description = 'taper usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar))) |
'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN

```

```

        TYPEOF (fcr.relating_shape_aspect))) = 1))) <= 1))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **round_hole** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'diameter occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **name** of 'diameter' and a **relating_shape_aspect** that is a **circular_closed_profile**.

WR2: The **round_hole** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'hole depth occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'path feature component usage' and a **name** of 'hole depth' in which the **relating_shape_aspect** is a **path_feature_component** with a **description** of 'linear'.

NOTE - The specification of the hole depth places a limitation on the depth of the **round_hole** so that it will not interfere with other features of the part that are nearby.

WR3: The **round_hole** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'bottom condition occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'hole bottom usage' and the **relating_shape_aspect** is a **hole_bottom**.

WR4: The **round_hole** shall be the basis shape for at most one **shape_aspect** with a **description** of 'change in diameter occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'taper usage' and the **relating_shape_aspect** is a **taper**.

5.2.3.1.69 rounded_end

A **rounded_end** is a type of **feature_definition** that is the representation of a volume that is removed from the base shape. The removal shall be represented by **partial_circular_profile** swept along about a linear path represented by a **path_feature_component**. See ARM definition for **Rounded_end** in paragraph 4.2.164 for more information.

EXPRESS specification:

```

*)
ENTITY rounded_end
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF (pd)) |
    NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
    (sa_occ.description = 'partial circular boundary occurrence') AND
    (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,

```

```

'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE' IN
TYPEOF (sdr.relate_shape_aspect)) AND
(sdr.name = 'partial circular boundary')) = 1))) = 1))) = 0;
WR2: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'course of travel occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'path feature component usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
('FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
TYPEOF (sdr.relate_shape_aspect)) AND
(sdr.relate_shape_aspect.description = 'linear'))))
= 1))) = 1))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **rounded_end** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'partial circular boundary occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and **name** of 'partial circular boundary' in which the **relating_shape_aspect** references a **partial_circular_profile**.

WR2: The **rounded_end** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'course of travel occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'path feature component usage' and a **relating_shape_aspect** that references a **path_feature_component** with a **description** of 'linear'.

5.2.3.1.70 rounded_u_profile

A **rounded_u_profile** is a type of **shape_aspect** that is the representation of an arc with connected straight lines at each end, with a location and a position. The **rounded_u_profile** is located at the midpoint of the arc and positioned in the X-Y plane with the X direction tangent to the arc at the midpoint, and the Y direction orthogonal in the direction of volume removal. See ARM definition for Rounded_U_profile in paragraph 4.2.165 for more information.

EXPRESS specification:

```

*)
ENTITY rounded_u_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')) = 1))) = 0))) = 0;
  WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'width')) = 1))) = 0))) = 0;
END_ENTITY;
(

```

Formal propositions:

WR1: The **rounded_u_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **rounded_u_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **rounded_u_profile** shall contain two **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **rounded_u_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **rounded_u_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

Informal propositions:

IP1: The location of the **rounded_u_profile** shall be at the midpoint of the arc.

IP2: The **rounded_u_profile** shall be defined with the X direction tangent to the arc at the midpoint, and the Y direction orthogonal in the direction of the open end of the profile..

5.2.3.1.71 shape_defining_relationship

A **shape_defining_relationship** is a kind of **shape_aspect_relationship** in which the **related_shape_aspect** takes a part in the definition of the shape of the **relating_shape_aspect**.

EXAMPLE - A Chamfer may have the implicit chamfer offset amount defined with a **feature_component_relationship** and the explicit geometry for the chamfer defined with a **shape_defining_relationship**.

EXPRESS specification:

```
*)
ENTITY shape_defining_relationship
  SUBTYPE OF (shape_aspect_relationship);
END_ENTITY;
(*
```

5.2.3.1.72 shape_representation_with_parameters

A **shape_representation_with_parameters** is a kind of **shape_representation** in which the shape of a **product_definition** or **shape_aspect** is defined implicitly using measurements and descriptive parameters.

EXAMPLE - The shape of a box may be specified by a **shape_representation_with_parameters** by giving measurements for its height, length, and width.

EXPRESS specification:

```

*)
ENTITY shape_representation_with_parameters
  SUBTYPE OF (shape_representation);
WHERE
  WR1: SIZEOF (QUERY (it <* SELF.items |
    NOT (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT',
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'] *
    TYPEOF (it)) = 1))) = 0;
END_ENTITY;
(*

```

Formal proposition:

WR1: The elements that specify the parameters for a **shape_representation_with_parameters** shall be of type **placement**, **measure_representation_item**, or **descriptive_representation_item**.

5.2.3.1.73 slot

A **slot** is a type of **feature_definition** that is the representation of a volume that is removed from the base shape. This removal shall be represented by sweeping either a **square_u_profile**, **partial_circular_profile**, **round_u_profile**, **vee_profile**, **tee_profile**, or an **open_path_profile** swept along a path represented by a **path_feature_component**. Each end of the **slot** shall be represented by a **slot_end**. See ARM definition for Slot in paragraph 4.2.173 for more information.

EXPRESS specification:

```

*)
ENTITY slot
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 2))) = 0;
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'

```



```

IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'depth')) = 1))) = 0))) = 0;
WR3: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'swept shape occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
TYPEOF (sdr.relate_shape_aspect)) = 1) AND
(sdr.relate_shape_aspect.description = 'swept shape'))
= 1))) = 1))) = 0;
WR4: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'course of travel occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'path feature component usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
TYPEOF (sdr.relate_shape_aspect))) = 1))) = 1))) = 0;
WR5: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'end condition occurrence') AND
(SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'slot end usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF (sar)))) |

```

```

        ('FEATURE_BASED_PROCESS_PLANNING.SLOT_END' IN
        TYPEOF (fcr.relater_shape_aspect)) AND
        (fcr.relater_shape_aspect.description = 'end condition'))))
        = 1))) = 2))) = 0;
END_ENTITY;
(*)

```

Formal propositions:

WR1: The implicit representation of the **slot** shall contain two **representation_items** in its set of **items**.

WR2: Exactly one **representation_item** used for the implicit representation of the **slot** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'depth'.

WR3: The **slot** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'swept shape occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile** with a **description** of 'swept shape'.

WR4: The **slot** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'course of travel occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'path feature component usage' and a **relating_shape_aspect** that references a **path_feature_component**.

WR5: The **slot** shall be the basis shape for exactly two **shape_aspects** with a **description** of 'end condition occurrence' that are each the **related_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'slot end usage' and a **relating_shape_aspect** that references a **slot_end** with a **description** of 'end condition'.

5.2.3.1.74 slot_end

A **slot_end** is a type of **shape_aspect** that is the representation of the end condition for a **slot feature_definition**. See ARM definition for Slot_end_type in paragraph 4.2.174 for more information.

EXPRESS specification:

```

*)
ENTITY slot_end
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);
  WR2: SELF.description IN ['open', 'radiused', 'flat', 'woodruff'];
  WR3: (NOT (SELF.description IN ['open', 'radiused'])) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |

```

```

NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) = 0))) = 0);
WR4: (NOT (SELF.description IN ['flat', 'woodruff'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) = 1))) = 0);
WR5: (NOT (SELF.description IN ['flat'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
WR6: (NOT (SELF.description = 'flat')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'first radius')))) = 1))) = 0))) = 0);
WR7: (NOT (SELF.description = 'flat')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'second radius')))) = 1))) = 0))) = 0);
WR8: (NOT (SELF.description = 'woodruff')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |

```

```

NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT' IN
TYPEOF (it)) AND (it.name = 'radius')))) = 1))) = 0))) = 0);
WR9: (NOT (SELF.description IN ['woodruff'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
WR10: SIZEOF (QUERY (fcr <* QUERY (sar <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') |
(sar.description = 'slot end usage') AND
('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF (sar))) |
(fcr.related_shape_aspect.description = 'end condition occurrence') AND
('FEATURE_BASED_PROCESS_PLANNING.SLOT' IN TYPEOF
(fcr.related_shape_aspect.of_shape.definition)))) >= 1;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **slot_end** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **description** for the **slot_end** shall be either 'open', 'radiused', 'flat', or 'woodruff'.

WR3: If the **description** of the **slot_end** is 'open' or 'radiused', the **slot_end** shall not have any implicit representation.

WR4: If the **description** of the **slot_end** is 'flat' or 'woodruff', the **slot_end** shall have its implicit representation specified by exactly one **shape_representation_with_parameters**.

WR5: If the **description** of the **slot_end** is 'flat', the **slot_end** shall have an implicit representation that contains exactly two **representation_item** in its set of **items**.

WR6: If the **description** of the **slot_end** is 'flat', exactly one **representation_item** used for the implicit representation of the **slot_end** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'first radius'.

WR7: If the **description** of the **slot_end** is 'flat', exactly one **representation_item** used for the implicit representation of the **slot_end** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'second radius'.

WR8: If the **description** of the **slot_end** is 'woodruff', exactly one **representation_item** used for the implicit representation of the **slot_end** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR9: If the **description** of the **slot_end** is 'woodruff', the **slot_end** shall have an implicit representation that contains exactly one **representation_item** in its set of **items**.

WR10: The **slot_end** shall be the **relating_shape_aspect** in at least one **feature_component_relationship** with a **description** of 'slot end usage' in which the **related_shape_aspect** is an aspect of the shape of a **slot** with a **description** of 'end condition occurrence'.

Informal propositions:

IP1: If the **description** of the **slot_end** is 'open', then it shall have no placement or orientation.

IP2: If the **description** of the **slot_end** is 'flat', then location shall be at the midpoint of the base line.

IP3: If the **description** of the **slot_end** is 'radiused', then location shall be at the midpoint of the radius.

IP4: If the **description** of the **slot_end** is 'woodruff', then location shall be at the point where the slot floor makes contact with the radius.

IP5: If the **description** of the **slot_end** is 'open', 'radiused', or 'woodruff', then the **slot_end** is defined with the base line in the X direction and the Y direction orthogonal in the direction of volume removal. The Z direction is coincident to that of the mating **slot** feature.

5.2.3.1.75 spherical_cap

A **spherical_cap** is a type of **feature_definition** that is the representation of a volume that is extruding from the base shape. The volume shall be spherical of a defined radius. See ARM definition for Spherical_cap paragraph 4.2.177 for more information.

EXPRESS specification:

```
*)
ENTITY spherical_cap
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
```

```

'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'radius')) = 1))) = 0))) = 0;
WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'internal angle')) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: Exactly one **representation_item** used for the implicit representation of the **spherical_cap** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR2: Exactly one **representation_item** used for the implicit representation of the **spherical_cap** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'internal angle'.

Informal propositions:

IP1: The orientation of the **spherical_cap** shall be with the Z-axis away from material.

IP2: The X-axis shall define the start of the **spherical_cap** and the internal angle is measured from the X-axis.

5.2.3.1.76 square_u_profile

A **square_u_profile** is a type of **shape_aspect** that is the representation of three connected straight lines, a location and a position. The **square_u_profile** is located at the midpoint of the base line and positioned in the X-Y plane with the base line on the X direction and the Y direction orthogonal in the direction of volume removal. See ARM definition for Square_U_profile in paragraph 4.2.179 for more information.

EXPRESS specification:

```

*)
ENTITY square_u_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 6))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')) = 1))) = 0))) = 0;
  WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'width')) = 1))) = 0))) = 0;
  WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |

```

```

        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'first radius')))) = 1))) = 0))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'second radius')))) = 1))) = 0))) = 0;
WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'first angle')))) = 1))) = 0))) = 0;
WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'second angle')))) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **square_u_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **square_u_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **square_u_profile** shall contain six **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

WR6: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'first radius'.

WR7: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'second radius'.

WR8: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'first angle'.

WR9: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'second angle'.

Informal propositions:

IP1: The location of the **square_u_profile** shall be defined at the midpoint of the base line.

IP2: The **square_u_profile** shall be defined in the X-Y plane with the base line in the X direction and the Y direction orthogonal in the direction of the open end of the profile.

5.2.3.1.77 step

A **step** is a type of **feature_definition** that is the representation of a volume that is removed from the base shape. This removal shall be represented by sweeping a **vee_profile** along a path represented by a **path_feature_component**. See ARM definition for Step in paragraph 4.2.180 for more information.

EXPRESS specification:

```
*)
ENTITY step
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
```

```

'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'course of travel occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'path feature component usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
('FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT'
IN TYPEOF (sdr.relatng_shape_aspect))) AND
(sdr.relatng_shape_aspect.description = 'linear')))) = 1))) = 1))) = 0;
WR2: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'removal boundary occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |
(sar.description = 'profile usage') AND
('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar)))) |
'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE'
IN TYPEOF (sdr.relatng_shape_aspect))) = 1))) = 1))) = 0;
END_ENTITY;
(*

```

Formal proposition:

WR1: The **step** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'course of travel occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'path feature component usage' and a **relating_shape_aspect** that references a **path_feature_component** with a **description** of 'linear'.

WR2: The **step** shall be the basis shape for exactly one **shape_aspect** with a **description** of 'removal boundary occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'profile usage' and a **relating_shape_aspect** that references a **vee_profile**.

5.2.3.1.78 surface_profile_tolerance

A **surface_profile_tolerance** is a type of **geometric_tolerance** that defines the allowable deviation of a **shape_aspect** that may require **datum_reference** to control the definition. See ARM definition for **Surface_profile_tolerance** in paragraph 4.2.183 for more information.

EXPRESS specification:

```

*)
ENTITY surface_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: SELF\geometric_tolerance.name = 'surface profile';
  WR2: (NOT ('FEATURE_BASED_PROCESS_PLANNING.' +
    'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF(SELF)))OR(SIZEOF
    (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
END_ENTITY;
(*

```

Formal propositions:

WR1: The **name** for the **surface_profile_tolerance** shall be 'surface profile'.

WR2: There shall be three or less **datum_references** in the **datum_system** set for the **surface_profile_tolerance**.

5.2.3.1.79 symmetry_tolerance

A **symmetry_tolerance** is a type of **geometric_tolerance_with_datum_reference** that defines the allowable deviation of a **shape_aspect** that may require **datum_reference** to control the definition. See ARM definition for Symmetry_tolerance in paragraph 4.2.185 for more information.

EXPRESS specification:

```

*)
ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'symmetry';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <=
    3;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **name** for the **symmetry_tolerance** shall be 'symmetry'.

WR2: There shall be at most three **datum_references** in the **datum_system** set for the **symmetry_tolerance**.

5.2.3.1.80 taper

A **taper** is a type of **shape_aspect** which represents a linear change applied to a machining feature. See ARM definition for Angle_taper in paragraph 4.2.2 and Diameter_taper in paragraph 4.2.57 for more information.

EXPRESS specification:

```

*)
ENTITY taper
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SELF.description IN ['angle taper', 'diameter taper'];
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0;
  WR5: (NOT (SELF.description = 'angle taper')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'taper angle')) = 1))) = 0))) = 0);
  WR6: (NOT (SELF.description = 'diameter taper')) OR
    (SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND
(it.name = 'taper diameter')) = 1))) = 0))) = 0));
END_ENTITY;
(*

```

Formal propositions:

WR1: The **taper** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **description** of the **taper** shall be either 'angle taper' or 'diameter taper'.

WR3: The **taper** shall have exactly one implicit representation defined by a relationship to a **shape_-representation_with_parameters**.

WR4: The **shape_representation_with_parameters** that represents the **taper** shall contain one **representation_item** in its set of **items**.

WR5: The **representation_item** used for the implicit representation of an angular **taper** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'taper angle'.

WR6: The **representation_item** used for the implicit representation of a diameter **taper** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'taper diameter'.

Informal proposition:

IP1: The taper angle shall be $-90 \text{ degrees} < \text{taper} < 90 \text{ degrees}$.

5.2.3.1.81 tee_profile

A **tee_profile** is a type of **shape_aspect** which is the representation of an upside down T shape with a location and a position. The **tee_profile** consist of a stem and a perpendicular cross bar. The stem has two parallel lines connected at their ends by the cross bar. The cross bar is a rectangle. The profile is symmetric about the centerline of the stem. The **tee_profile** is located at the midpoint of the side of the cross bar opposite the stem, and positioned with the side of the cross bar opposite the stem on the X direction and the profile symmetric with respect to the Y direction. See ARM definition for Tee_profile in paragraph 4.2.191 for more information.

EXPRESS specification:

```

*)
ENTITY tee_profile
  SUBTYPE OF (shape_aspect);

```

WHERE

```

WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 10))) = 0))) = 0;
WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')))) = 1))) = 0))) = 0;
WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'width')))) = 1))) = 0))) = 0;
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',

```

```

'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF (it)) = 2) AND
  (it.name = 'depth')) = 1))) = 0))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF (pdr.used_representation)) |
  NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'cross bar width')) = 1))) = 0))) = 0;
WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF (pdr.used_representation)) |
  NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'cross bar depth')) = 1))) = 0))) = 0;
WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF (pdr.used_representation)) |
  NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'first offset')) = 1))) = 0))) = 0;
WR10: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF (pdr.used_representation)) |
  NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'second offset')) = 1))) = 0))) = 0;

```

```

WR11: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF (pdr.used_representation)) |
  NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'first angle')))) = 1))) = 0))) = 0;
WR12: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF (pdr.used_representation)) |
  NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'second angle')))) = 1))) = 0))) = 0;
WR13: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF (pdr.used_representation)) |
  NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF (it)) = 2) AND
    (it.name = 'radius')))) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **tee_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **tee_profile** shall have exactly one implicit representation defined by a relationship to a **shape_-representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **tee_profile** shall contain ten **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

WR6: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'depth'.

WR7: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'cross bar width'.

WR8: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'cross bar depth'.

WR9: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'first offset'.

WR10: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'second offset'.

WR11: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'first angle'.

WR12: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'second angle'.

WR13: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

Informal propositions:

IP1: The location of the **tee_profile** shall be at the midpoint of the side of the cross bar opposite the stem.

IP2: The **tee_profile** shall be defined with the side of the cross bar opposite the stem in the X direction and the profile symmetric with respect to the Y direction.

5.2.3.1.82 thread

A **thread** is a type of **feature_definition** that is the representation of a thread shape that is applied to all or a portion of a **shape_aspect**. See ARM definition for Thread in paragraph 4.2.192 for more information.

EXPRESS specification:

```

*)
ENTITY thread
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF (it)) = 2) AND (it.name = 'major diameter')))) = 1))) = 0))) = 0;
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF (it)) = 2) AND (it.name = 'minor diameter')))) <= 1))) = 0)))
    = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF (it)) = 2) AND (it.name = 'pitch diameter')))) <= 1))) = 0)))
    = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.RATIO_MEASURE_WITH_UNIT'] *

```

```

    TYPEOF (it)) = 2) AND (it.name = 'number of threads')))) = 1))) = 0)))
    = 0;
WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND (it.name = 'fit class')))) <= 1))) = 0))) = 0;
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND (it.name = 'form')))) = 1))) = 0))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND (it.name = 'hand')))) = 1))) = 0))) = 0;
WR8: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    ('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF (it)) AND (it.name = 'qualifier')))) = 1))) = 0))) = 0;
WR9: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |

```

```

        ('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF (it)) AND (it.name = 'thread system') AND
        ((it.description = 'english') OR
        (it.description = 'metric')))) = 1))) = 0))) = 0;
WR10: SIZEOF (QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        ('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF (it)) AND (it.name = 'thread side') AND
        ((it.description = 'internal') OR
        (it.description = 'external')))) = 1))) = 0))) = 0;
WR11: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF (pd)) |
        NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
        (sa_occ.description = 'partial area occurrence') AND
        (SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') |
        (sar.description = 'applied area usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
        'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN
        TYPEOF (sdr.relateing_shape_aspect))) = 1))) <= 1))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'major diameter'.

WR2: At most one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'minor diameter'.

WR3: At most one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'pitch diameter'.

WR4: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **ratio_measure_with_unit** with a **name** of 'number of threads'.

WR5: At most one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'fit class'.

WR6: At most one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'form'.

WR7: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'hand'.

WR8: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'qualifier'.

WR9: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'thread system' and a **description** of either 'english' or 'metric'.

WR10: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'thread side' and a **description** of either 'internal' or 'external'.

WR11: The **thread** shall be the basis shape for at most one **shape_aspect** with a **description** of 'partial area occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'applied area usage' and the **relating_shape_aspect** is an **applied_area**.

5.2.3.1.83 total_runout_tolerance

A **total_runout_tolerance** is a type of **geometric_tolerance_with_datum_reference** that defines the allowable deviation of a **shape_aspect** that may require **datum_reference** to control the definition. See ARM definition for **Total_runout_tolerance** in paragraph 4.2.200 for more information.

EXPRESS specification:

```
*)
ENTITY total_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SELF\geometric_tolerance.name = 'total runout';
  WR2: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <=
    2;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **name** for the **total_runout_tolerance** shall be 'total runout'.

WR2: There shall be at most two **datum_references** in the **datum_system** set for the **total_runout_tolerance**.

5.2.3.1.84 transition_feature

A **transition_feature** is a type of **shape_aspect** that is the representation of a transition between two **shape_aspect** entities. The placement of a **transition_feature** is dependent on the placement of the two **shape_aspect** entities.

EXPRESS specification:

```
*)
ENTITY transition_feature
  SUPERTYPE OF (ONEOF (chamfer, edge_round, fillet))
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
END_ENTITY;
(*
```

Formal propositions:

WR1: The **transition_feature** shall identify an aspect of the shape of a **product_definition**.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **named_unit** entity:

- transition_feature_life_cycle (See 5.2.4.30);
- transition_feature_on_part_boundary (See 5.2.4.31).

5.2.3.1.85 turned_knurl

A **turned_knurl** is a type of **feature_definition** that is the representation of a diamond, diagonal, or straight shape that is applied to all or a portion of a **shape_aspect**. See ARM definition for Turned_knurl in paragraph 4.2.202 for more information.

EXPRESS specification:

```
*)
ENTITY turned_knurl
  SUBTYPE OF (feature_definition);
WHERE
  WR1: SELF.description IN ['diamond', 'diagonal', 'straight'];
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
```

```

'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND
('FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN
TYPEOF (it\measure_with_unit.value_component)) AND
(it.name = 'number of teeth')) <= 1))) = 0))) = 0;
WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'major diameter')) = 1))) = 0))) = 0;
WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'nominal diameter')) = 1))) = 0)))
= 0;
WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'tooth depth')) <= 1))) = 0))) = 0;
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'root fillet')))) <= 1))) = 0))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'diametral pitch')))) = 1))) = 0))) = 0;
WR8: (NOT (SELF.description IN ['diamond', 'diagonal'])) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
(SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF (it)) = 2) AND (it.name = 'helix angle')))) = 1))) = 0))) = 0;
WR9: (NOT (SELF.description = 'diagonal')) OR
(SIZEOF (QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation)) |
NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF (it)) AND (it.name = 'helix hand')))) = 1))) = 0))) = 0;
WR10: SIZEOF (QUERY (pds <* QUERY (pd <* USEDIN (SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (pd)) |
NOT (SIZEOF (QUERY (sa_occ <* USEDIN (pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
(sa_occ.description = 'partial area occurrence') AND
(SIZEOF (QUERY (sdr <* QUERY (sar <* USEDIN (sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') |

```



```

        (sar.description = 'applied area usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF (sar))) |
        'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN
        TYPEOF (sdr.relatng_shape_aspect))) = 1))) <= 1))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **description** for the **turned_knurl** shall be either 'diamond', 'diagonal', or 'straight'.

WR2: At most one **representation_item** used for the implicit representation of the **turned_knurl** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of teeth'.

WR3: Exactly one **representation_item** used for the implicit representation of the **turned_knurl** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'major diameter'.

WR4: Exactly one **representation_item** used for the implicit representation of the **turned_knurl** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'nominal diameter'.

WR5: At most one **representation_item** used for the implicit representation of the **turned_knurl** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'tooth depth'.

WR6: At most one **representation_item** used for the implicit representation of the **turned_knurl** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'root fillet'.

WR7: Exactly one **representation_item** used for the implicit representation of the **turned_knurl** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'diametral pitch'.

WR8: If the **turned_knurl** has a **description** of either 'diagonal' or 'diamond', exactly one **representation_item** used for the implicit representation of the **turned_knurl** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'helix angle'.

WR9: If the **turned_knurl** has a **description** of 'diagonal' exactly one **representation_item** used for the implicit representation of the **turned_knurl** shall be of type **descriptive_representation_item** with a **name** of 'helix hand'.

WR10: The **turned_knurl** shall be the basis shape for at most one **shape_aspect** with a **description** of 'partial area occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **description** of 'applied area usage' and a **relating_shape_aspect** that is an **applied_area**.

5.2.3.1.86 vee_profile

A **vee_profile** is a type of **shape_aspect** that is the representation of two connected straight lines, with a location and a position. The **vee_profile** is located at the position where the two lines connect, the X direction intersects the angle between the two connected lines in the direction of volume removal, and the Y direction is orthogonal to the X direction. See ARM definition for Vee_profile in paragraph 4.2.203 for more information.

EXPRESS specification:

```

*)
ENTITY vee_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
    TYPEOF (SELF.of_shape.definition);
  WR2: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 4))) = 0))) = 0;
  WR4: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
    'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF (it)) AND
    (it.name = 'orientation')))) = 1))) = 0))) = 0;
  WR5: SIZEOF (QUERY (pd <* USEDIN (SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |

```

```

NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM' IN
  TYPEOF (it)) AND
  ('FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT' IN
    TYPEOF (it\measure_with_unit.value_component)) AND
    (it.name = 'profile radius')) = 1))) = 0))) = 0;
WR6: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND
        (it.name = 'profile angle')) = 1))) = 0))) = 0;
WR7: SIZEOF (QUERY (pd <* USEDIN (SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
  NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
      (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND
        (it.name = 'tilt angle')) = 1))) = 0))) = 0;
END_ENTITY;
(*

```

Formal propositions:

WR1: The **vee_profile** shall be an aspect of the shape of a **feature_component_definition**.

WR2: The **vee_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters**.

WR3: The **shape_representation_with_parameters** that represents the **vee_profile** shall contain four **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **vee_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **vee_profile** shall be of type **measure_representation_item** with a **value_component** of type **length_measure_with_unit** and a **name** of 'profile radius'.

WR6: Exactly one **representation_item** used for the implicit representation of a **vee_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'profile angle'.

WR7: Exactly one **representation_item** used for the implicit representation of a **vee_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'tilt angle'.

Informal propositions:

IP1: The location of the **vee_profile** shall be at the position where the two lines connect.

IP2: The **vee_profile** shall be defined with the tilt angle measured from the X-axis and the opening of the profile in the direction of the Y-axis.

5.2.3.2 Feature based process planning imported entity modifications

5.2.3.2.1 action_relationship

The base definition of the **action_relationship** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **action_relationship** entity:

— project_order_tracking_relationships (See 5.2.4.22).

5.2.3.2.2 action_request_status

The base definition of the **action_request_status** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **action_request_status** entity:

— dependent_instantiable_action_request_status (See 5.2.4.4).

5.2.3.2.3 action_status

The base definition of the **action_status** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **action_status** entity:

- dependent_instantiable_action_status (See 5.2.4.5).

5.2.3.2.4 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).

5.2.3.2.5 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).

5.2.3.2.6 approval

The base definition of the **approval** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **approval** entity:

- approval_requires_approval_date_time (See 5.2.4.2);
- approval_requires_approval_person_organization (See 5.2.4.3).

5.2.3.2.7 approval_date_time

The base definition of the **approval_date_time** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **approval_date_time** entity:

- approval_requires_approval_date_time (See 5.2.4.2).

5.2.3.2.8 approval_person_organization

The base definition of the **approval_person_organization** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **approval_person_organization** entity:

- approval_requires_approval_person_organization (See 5.2.4.3).

5.2.3.2.9 approval_status

The base definition of the **approval_status** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **approval_status** entity:

- dependent_instantiable_approval_status (See 5.2.4.6);
- restrict_approval_status (See 5.2.4.24).

5.2.3.2.10 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 5.2.4.29).

5.2.3.2.11 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 5.2.4.7).

5.2.3.2.12 dimensional_location

The base definition of the **dimensional_location** entity is given in ISO 10303-47. The following modifications apply to this part of ISO 10303.

Attribute definitions:

relating_shape_aspect: the origin of the direction of measurement of the dimension.

related_shape_aspect: the target of the direction of measurement of the dimension.

5.2.3.2.13 directed_action

The base definition of the **directed_action** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **directed_action** entity:

- project_order_tracking_relationships (See 5.2.4.22).

5.2.3.2.14 geometric_tolerance

The base definition of the **geometric_tolerance** entity is given in ISO 10303-47. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **geometric_tolerance** entity:

- **geometric_tolerance_subtype_exclusiveness** (See 5.2.4.14).

5.2.3.2.15 **make_from_usage_option**

The base definition of the **make_from_usage_option** entity is given in ISO 10303-44. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **make_from_usage_option** entity:

- **material_is_specified_for_part** (See 5.2.4.16).

5.2.3.2.16 **named_unit**

The base definition of the **named_unit** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **named_unit** entity:

- **dependent_instantiable_named_unit** (See 5.2.4.8).

5.2.3.2.17 **precision_qualifier**

The base definition of the **precision_qualifier** entity is given in ISO 10303-45. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **precision_qualifier** entity:

- **dependent_instantiable_precision_qualifier** (See 5.2.4.9).

5.2.3.2.18 **product**

The base definition of the **product** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **product** entity:

- product_requires_version (See 5.2.4.19).

5.2.3.2.19 product_definition

The base definition of the **product_definition** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **product_definition** entity:

- material_is_specified_for_part (See 5.2.4.16).

5.2.3.2.20 product_definition_formation

The base definition of the **product_definition_formation** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition_formation** entity:

- part_requires_project_order (See 5.2.4.17);
- part_to_approval (See 5.2.4.18);
- product_requires_version (See 5.2.4.19);
- product_version_requires_security_classification (See 5.2.4.20).

5.2.3.2.21 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 5.2.4.23).

5.2.3.2.22 security_classification_level

The base definition of the **security_classification_level** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 applies to the **security_classification_level** entity:

- dependent_instantiable_security_classification_level (See 5.2.4.12),
- restrict_security_classification_level (see 5.2.4.24).

5.2.3.2.23 shape_aspect

The base definition of the **shape_aspect** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **shape_aspect** entity:

- shape_aspect_subtype_exclusiveness (See 5.2.4.27).

5.2.3.2.24 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 5.2.4.26).

5.2.3.2.25 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 5.2.4.13);
- shape_representation_subtype_exclusiveness (See 5.2.4.28).

5.2.3.2.26 type_qualifier

The base definition of the **type_qualifier** entity is given in ISO 10303-45. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **type_qualifier** entity:

- dependent_instantiable_type_qualifier (See 5.2.4.10).

5.2.3.2.27 uncertainty_qualifier

The base definition of the **uncertainty_qualifier** entity is given in ISO 10303-45. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **uncertainty_qualifier** entity:

- dependent_instantiable_uncertainty_qualifier (See 5.2.4.11).

5.2.4 Feature based process planning rules**5.2.4.1 application_context_requires_ap_definition**

The **application_context_requires_ap_definition** rule specifies that each instance of **application_context** shall be referenced by exactly one **application_protocol_definition** that specifies this part of ISO 10303.

EXPRESS specification:

```

*)
RULE application_context_requires_ap_definition FOR
  (application_context, application_protocol_definition);
WHERE
  WR1: SIZEOF (QUERY (ac <* application_context |

```

```

NOT (SIZEOF (QUERY (apd <* application_protocol_definition |
(ac :=: apd.application)
AND
(apd.application_interpreted_model_schema_name =
'feature_based_process_planning')))) = 1 ))) = 0;
END_RULE;
( *

```

Argument definitions:

application_context: the set of all instances of **application_context** entities.

application_protocol_definition: the set of all instances of **application_protocol_definition** entities.

Formal propositions:

WR1: For each instance of **application_context**, there shall be exactly one instance of **application_protocol_definition** that references the instance of **application_context** as its **application** with a value of 'feature_based_process_planning' as its **application_interpreted_model_schema_name**.

5.2.4.2 approval_requires_approval_date_time

The **approval_requires_approval_date_time** rule specifies that each instance of **approval** shall be referenced by exactly one **approval_date_time**. This rule enforces the requirement for every approval to have a date on which the approval obtained its specified status.

EXPRESS specification:

```

*)
RULE approval_requires_approval_date_time FOR (approval,
approval_date_time);
WHERE
WR1: SIZEOF (QUERY (app <* approval |
NOT (SIZEOF (QUERY (adt <* approval_date_time |
app :=: adt.dated_approval )) = 1 ))) = 0;
END_RULE;
( *

```

Argument definitions:

approval: the set of all instances of **approval** entities.

approval_date_time: the set of all instances of **approval_date_time** entities.

Formal propositions:

WR1: For each instance of **approval**, there shall be exactly one instance of **approval_date_time** which contains the instance of **approval** as its **dated_approval** attribute.

5.2.4.3 approval_requires_approval_person_organization

The **approval_requires_approval_person_organization** specifies that each instance of **approval** shall have at least one **approval_person_organization** referencing it. This rule enforces the requirement for an approval to be authorized by one or more people within their organizations.

EXPRESS specification:

```

*)
RULE approval_requires_approval_person_organization FOR
  (approval, approval_person_organization);
WHERE
  WR1: SIZEOF (QUERY (app <* approval |
    NOT (SIZEOF (QUERY (apo <* approval_person_organization |
      app := apo.authorized_approval )) >= 1 ))) = 0;
END_RULE;
(

```

Argument definitions:

approval: the set of all instances of **approval** entities.

approval_person_organization: the set of all instances of **approval_person_organization** entities.

Formal propositions:

WR1: For each instance of **approval**, there shall be one or more instances of **approval_person_organization** which contains the instance of **approval** as its **authorized_approval** attribute.

5.2.4.4 dependent_instantiable_action_request_status

The **dependent_instantiable_action_request_status** rule specifies that all instances of **action_request_status** are dependent on the usage to define another entity.

EXPRESS specification:

```

*)
RULE dependent_instantiable_action_request_status FOR (action_request_status);
WHERE
  WR1: SIZEOF (QUERY (arst <* action_request_status |
    NOT (SIZEOF (USEDIN (arst, '')) >= 1))) = 0;
END_RULE;
(

```

Argument definition:

action_request_status: the set of all instances of **action_request_status**.

Formal proposition:

WR1: For each instance of **action_request_status**, there shall be a reference to the **action_request_status** instance from an attribute of another entity.

5.2.4.5 dependent_instantiable_action_status

The **dependent_instantiable_action_status** rule specifies that all instances of **action_status** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_action_status FOR (action_status);
WHERE
  WR1: SIZEOF (QUERY (ast <* action_status |
    NOT (SIZEOF (USEDIN (ast, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

action_status: the set of all instances of **action_status**.

Formal proposition:

WR1: For each instance of **action_status**, there shall be a reference to the **action_status** instance from an attribute of another entity.

5.2.4.6 dependent_instantiable_approval_status

The **dependent_instantiable_approval_status** rule specifies that all instances of **approval_status** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_approval_status FOR (approval_status);
WHERE
  WR1: SIZEOF (QUERY (ast <* approval_status |
    NOT (SIZEOF (USEDIN (ast, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

approval_status: the set of all instances of **approval_status**.

Formal proposition:

WR1: For each instance of **approval_status**, there shall be a reference to the **approval_status** instance from an attribute of another entity.

5.2.4.7 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.

5.2.4.8 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.

5.2.4.9 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.

5.2.4.10 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.

5.2.4.11 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.

5.2.4.12 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.

5.2.4.13 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.

5.2.4.14 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 **circular_runout_tolerance**, **angularity_tolerance**, **concentricity_tolerance**, **parallelism_tolerance**, **perpendicularity_tolerance**, **total_runout_tolerance**, **surface_profile_tolerance**, **symmetry_tolerance**, **position_tolerance**, or **linear_profile_tolerance**.

EXPRESS specification:

```
*)
RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);
WHERE
  WR1: SIZEOF (QUERY (gt <* geometric_tolerance |
    NOT (SIZEOF (TYPEOF (gt) *
```

```

[ 'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_RUNOUT_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.ANGULARITY_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.CONCENTRICITY_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.PARALLELISM_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.PERPENDICULARITY_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.TOTAL_RUNOUT_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.SURFACE_PROFILE_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.SYMMETRY_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.POSITION_TOLERANCE',
  'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE_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 **circular_runout_tolerance**, **angularity_tolerance**, **concentricity_tolerance**, **parallelism_tolerance**, **perpendicularity_tolerance**, **total_runout_tolerance**, **surface_profile_tolerance**, **symmetry_tolerance**, **position_tolerance**, or **linear_profile_tolerance**.

5.2.4.15 machining_feature_life_cycle

The **machining_feature_life_cycle** rule specifies that each instance of **machining_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
  (machining_feature);
WHERE
  WR1: SIZEOF (QUERY (mf <* machining_feature |
    NOT (mf.of_shape.definition.frame_of_reference.life_cycle_stage =
      'manufacturing planning')))) = 0;
END_RULE;
(*

```

Argument definitions:

machining_feature: the set of all instances of **machining_feature** entities.

Formal propositions:

WR1: For each instance of **machining_feature**, the **life_cycle_stage** of the **product_definition** for which it is defined has a value of 'manufacturing planning'.

5.2.4.16 material_is_specified_for_part

The **material_is_specified_for_part** rule specifies that every **product_definition** that is not designated a material shall be related to a material designated **product_definition** through the **make_from_usage_option**.

EXPRESS specification:

```

*)
RULE material_is_specified_for_part FOR (product_definition,
    make_from_usage_option);
WHERE
    WR1: SIZEOF (QUERY (nmpd <* QUERY (pd <* product_definition |
        SIZEOF (USEDIN (pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
            'MATERIAL_DESIGNATION.DEFINITIONS')) = 0) |
        NOT (SIZEOF (QUERY (mfuo <* make_from_usage_option |
            NOT (nmpd := mfuo.relating_product_definition))) >= 1))) = 0;
END_RULE;
(

```

Argument definitions:

product_definition: the set of all instances of **product_definition** entities.

make_from_usage_option: the set of all instances of **make_from_usage_option** entities.

Formal propositions:

WR1: For each instance of **product_definition** that is not designated as a material through a reference by the **definitions** attribute of the **material_designation**, there shall be at least one instance of **make_from_usage_option** in which the non-material **product_definition** is the **relating_product_definition**.

5.2.4.17 part_requires_project_order

The **part_requires_project_order** rule specifies that each instance of **product_definition_formation** shall be referenced by exactly one instance of **feature_based_pp_action_assignment** that assigns a project order to the part.

EXPRESS specification:

```

*)
RULE part_requires_project_order FOR (product_definition_formation,

```

```

    feature_based_pp_action_assignment);
WHERE
  WR1: SIZEOF (QUERY (pdf <* product_definition_formation |
    NOT (SIZEOF (QUERY (fbppaa <* feature_based_pp_action_assignment |
      (pdf IN fbppaa.items) AND
      (fbppaa.assigned_action.name = 'project order')))) = 1 ))) = 0;
END_RULE;
( *

```

Argument definitions:

product_definition_formation: the set of all instances of **product_definition_formation** entities.

feature_based_pp_action_assignment: the set of all instances of **feature_based_pp_action_assignment** entities.

Formal propositions:

WR1: For each instance of **product_definition_formation**, there shall be exactly one instance of **feature_based_pp_action_assignment** that contains the instance of **product_definition_formation** in its set of **items** and references an **action** with a name of 'project order'.

5.2.4.18 part_to_approval

The **part_to_approval** rule specifies that each instance of **product_definition_formation** shall be referenced by at most one instance of **feature_based_pp_approval_assignment**.

EXPRESS specification:

```

*)
RULE part_to_approval FOR (product_definition_formation,
  feature_based_pp_approval_assignment);
WHERE
  WR1: SIZEOF (QUERY (pdf <* product_definition_formation |
    NOT (SIZEOF (QUERY (fbppa <* feature_based_pp_approval_assignment |
      pdf IN fbppa.items )) <= 1 ))) = 0;
END_RULE;
( *

```

Argument definitions:

product_definition_formation: the set of all instances of **product_definition_formation** entities.

feature_based_pp_approval_assignment: the set of all instances of **feature_based_pp_approval_assignment** entities.

Formal propositions:

WR1: For each instance of **product_definition_formation**, there shall be exactly one instance of **feature_based_pp_approval_assignment** that contains the instance of **product_definition_formation** in its set of items.

5.2.4.19 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**.

5.2.4.20 product_definition_formation_requires_security_classification

The **product_definition_formation_requires_security_classification** rule specifies that each instance of **product_definition_formation** shall be referenced by exactly one instance of **feature_based_pp_security_classification_assignment**. This rule enforces the requirement for every version of a design to have a security classification.

EXPRESS specification:

```
*)
RULE product_definition_formation_requires_security_classification FOR
    (product_definition_formation,
    feature_based_pp_security_classification_assignment);
```

WHERE

```
WR1: SIZEOF (QUERY (pdf <* product_definition_formation |
  NOT (SIZEOF (QUERY (fbppsca <*
    feature_based_pp_security_classification_assignment |
    pdf IN fbppsca.items )) = 1 ))) = 0;
```

END_RULE;

(*

Argument definitions:

product_definition_formation: the set of all instances of **product_definition_formation** entities.

feature_based_pp_security_classification_assignment: the set of all instances of **feature_based_pp_security_classification_assignment** entities.

Formal propositions:

WR1: For each instance of **product_definition_formation**, there shall be exactly one instance of **feature_based_pp_security_classification_assignment** that contains the instance of **product_definition_formation** in its set of **items**.

5.2.4.21 project_order_requires_approval

The **project_order_requires_approval** rule specifies that each instance of **directed_action** that references an **action** with a **name** of 'project order' shall be referenced by exactly one instance of **feature_based_pp_approval_assignment**. This rule enforces the requirement for every project order to have an approval.

EXPRESS specification:

*)

```
RULE project_order_requires_approval FOR
  (directed_action, feature_based_pp_approval_assignment);
```

WHERE

```
WR1: SIZEOF (QUERY (po <* QUERY (da <* directed_action |
  da.name = 'project order') |
  NOT (SIZEOF (QUERY (fbppapp <* feature_based_pp_approval_assignment |
    po IN fbppapp.items )) = 1 ))) = 0;
```

END_RULE;

(*

Argument definitions:

directed_action: the set of all instances of **directed_action** entities.

feature_based_pp_approval_assignment: the set of all instances of **feature_based_pp_approval_assignment** entities.

Formal propositions:

WR1: For each instance of **directed_action** there shall be exactly one instance of **feature_based_pp_approval_assignment** that contains the instance of **directed_action** in its set of **items**.

5.2.4.22 project_order_tracking_relationships

The **project_order_tracking_relationships** rule specifies the relationship between a project order and the other types of orders within the process planning preparation. Each order shall be carried out under a project order, and an individual project order shall be defined for tracking at most one of each type of order within the project. The types of orders defined for this part of ISO 10303 are shop work order, resource acquisition order, digital technical data package work order, and pedigree creation order.

EXPRESS specification:

```

*)
RULE project_order_tracking_relationships FOR (directed_action,
  action_relationship);
WHERE
  WR1: SIZEOF (QUERY (da <* directed_action |
    (da.name IN ['shop work order', 'resource acquisition order',
      'digital technical data package work order',
      'pedigree creation order']) AND
    NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.related_action) AND
      (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
        TYPEOF (ar.relation)) AND
      (ar.relation.name = 'project order')))) = 1))) = 0;
  WR2: SIZEOF (QUERY (da <* directed_action |
    (da.name = 'project order') AND
    NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.relation) AND
      (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
        TYPEOF (ar.related_action)) AND
      (ar.related_action.name = 'shop work order')))) <= 1))) = 0;
  WR3: SIZEOF (QUERY (da <* directed_action |
    (da.name = 'project order') AND
    NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.relation) AND
      (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
        TYPEOF (ar.related_action)) AND
      (ar.related_action.name = 'resource acquisition order'))))
      <= 1))) = 0;
  WR4: SIZEOF (QUERY (da <* directed_action |
    (da.name = 'project order') AND
    NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.relation) AND
      (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
        TYPEOF (ar.related_action)) AND
      (ar.related_action.name =
        'digital technical data package work order')))) <= 1))) = 0;
  WR5: SIZEOF (QUERY (da <* directed_action |

```



```

        (da.name = 'project order') AND
        NOT (SIZEOF (QUERY (ar <* action_relationship |
        (da :=: ar.relateing_action) AND
        (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
        TYPEOF (ar.related_action)) AND
        (ar.related_action.name = 'pedigree creation order'))))
        <= 1))) = 0;
WR6: SIZEOF (QUERY (da <* directed_action |
        (da.name = 'customer order') AND
        NOT (SIZEOF (QUERY (ar <* action_relationship |
        (da :=: ar.related_action) AND
        (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
        TYPEOF (ar.relateing_action)) AND
        (ar.relateing_action.name = 'project order')))) >= 1))) = 0;
WR7: SIZEOF (QUERY (da <* directed_action |
        (da.name = 'project order' ) AND
        NOT (SIZEOF (QUERY (ar <* action_relationship |
        (da :=: ar.relateing_action) AND
        (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
        TYPEOF (ar.related_action)) AND
        (ar.related_action.name = 'customer order'))))
        <= 1))) = 0;
END_RULE;
( *

```

Argument definitions:

directed_action: the set of all instances of **directed_action** entities.

action_relationship: the set of all instances of **action_relationship** entities.

Formal propositions:

WR1: For each instance of **directed_action** with a **name** of either 'shop work order', 'resource acquisition order', 'digital technical data package work order', or 'pedigree creation order', there shall be exactly one instance of **action_relationship** that references that **directed_action** as the **related_action**, and references an instance of **directed_action** with a **name** of 'project order' as the **relating_action**.

WR2: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'shop work order'.

WR3: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'resource acquisition order'.

WR4: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'digital technical data package work order'.

WR5: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'pedigree creation order'.

WR6: For each instance of **directed_action** with a **name** of 'customer order' there shall be at least one instance of **action_relationship** that references that **directed_action** as the **related_action**, and references an instance of **directed_action** with a **name** of 'project order' as the **relating_action**.

WR7: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'customer order'.

5.2.4.23 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**, or **path_shape_representation**.

EXPRESS specification:

```
*)
RULE representation_subtype_exclusiveness FOR (representation);
WHERE
  WR1: SIZEOF (QUERY (rep <* representation |
    NOT (SIZEOF (TYPEOF (rep) *
      ['FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.DEFINITIONAL_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.LOCATION_SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'])
    <= 2))) = 0;
END_RULE;
(*
```

Argument definitions:

representation: the set of all instances of **representation** entities.

Formal propositions:

WR1: Each instance of the subtypes of **representation** shall be one of the **shape_representation**, **definitional_representation**, **direction_shape_representation**, **face_shape_representation**, **location_shape_representation**, or **path_shape_representation** subtypes.

5.2.4.24 restrict_approval_status

The **restrict_approval_status** rule specifies that the only values of **approval_status** permitted shall be 'approved', 'not_yet_approved', 'disapproved', or 'withdrawn'.

EXPRESS specification:

```
*)
RULE restrict_approval_status FOR (approval_status);
WHERE
  WR1: SIZEOF (QUERY (ast <* approval_status |
    NOT (ast.name IN
      ['approved', 'not yet approved', 'disapproved', 'withdrawn']))) = 0;
END_RULE;
(*
```

Argument definitions:

approval_status: the set of all instances of **approval** entities.

Formal propositions:

WR1: For each instance of **approval**, the value of the **approval_status** attribute shall be either 'approved', 'not yet approved', 'disapproved', or 'withdrawn'.

Attribute value definitions:

approved: specifies that the required authorizations have been obtained for a particular role of approval for a piece of product data.

not_yet_approved: specifies that the required authorizations are being awaited for a particular role of approval for a piece of product data.

disapproved: specifies that the required authorizations have been denied for a particular role of approval for a piece of product data.

withdrawn: specifies that the required authorizations have been revoked for a particular role of approval for a piece of product data.

5.2.4.25 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.

5.2.4.26 shape_aspect_relationship_subtype_exclusiveness

The **shape_aspect_relationship_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **shape_aspect_relationship** shall be only one of **dimensional_location**, **geometric_tolerance_relationship**, **feature_component_relationship**, or **shape_defining_relationship**.

EXPRESS specification:

```

*)
RULE shape_aspect_relationship_subtype_exclusiveness FOR
  (shape_aspect_relationship);
WHERE
  WR1: SIZEOF (QUERY (sr <* shape_aspect_relationship |
    NOT (SIZEOF (TYPEOF (sr) *
      ['FEATURE_BASED_PROCESS_PLANNING.DIMENSIONAL_LOCATION',
       'FEATURE_BASED_PROCESS_PLANNING.GEOMETRIC_TOLERANCE_RELATIONSHIP',
       'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP',
       'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'])
    <= 2))) = 0;
END_RULE;
( *

```

Argument definitions:

shape_aspect_relationship: the set of all instances of **shape_aspect_relationship** entities.

Formal propositions:

WR1: Each instance of the subtypes of **shape_aspect_relationship** shall be one of the **dimensional_location**, **geometric_tolerance_relationship**, **feature_component_relationship**, or **shape_defining_relationship**.

5.2.4.27 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**, **rectangular_closed_profile**, **partial_circular_profile**, **rounded_u_profile**, **open_path_profile**, **machining_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) *
      ['FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT',
       'FEATURE_BASED_PROCESS_PLANNING.SLOT_END',
       'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM',
       'FEATURE_BASED_PROCESS_PLANNING.BOSS_TOP',
       'FEATURE_BASED_PROCESS_PLANNING.HOLE_BOTTOM',
       'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA',
       'FEATURE_BASED_PROCESS_PLANNING.TAPER',
       'FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET',

```

```

'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.MACHINING_FEATURE',
'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
'FEATURE_BASED_PROCESS_PLANNING.TRANSITION_FEATURE',
'FEATURE_BASED_PROCESS_PLANNING.DATUM',
'FEATURE_BASED_PROCESS_PLANNING.DATUM_FEATURE',
'FEATURE_BASED_PROCESS_PLANNING.DATUM_TARGET']])
<= 2))) = 0;
END_RULE;
( *

```

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**, **linear_profile**, **tee_profile**, **vee_profile**, **rectangular_closed_profile**, **partial_circular_profile**, **rounded_u_profile**, **open_path_profile**, **machining_feature**, **replicate_feature**, **transition_feature**, **datum**, **datum_feature**, or **datum_target**.

5.2.4.28 shape_representation_subtype_exclusiveness

The **shape_representation_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **shape_representation** shall be only one of **advanced_brep_shape_representation**, **shape_representation_with_parameters**, or **shape_dimension_representation**.

EXPRESS specification:

```

*)
RULE shape_representation_subtype_exclusiveness FOR (shape_representation);
WHERE
  WR1: SIZEOF (QUERY (sr <* shape_representation |
    NOT (SIZEOF (TYPEOF (sr) *
      ['FEATURE_BASED_PROCESS_PLANNING.ADVANCED_BREP_SHAPE_REPRESENTATION',
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS',
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DIMENSION_REPRESENTATION'])
    <= 2))) = 0;

```

END_RULE;
(*

Argument definitions:

shape_representation: the set of all instances of **shape_representation** entities.

Formal propositions:

WR1: Each instance of the subtypes of **shape_representation** shall be one of the **advanced_brep_shape_representation**, **shape_representation_with_parameters**, or **shape_dimension_representation**.

5.2.4.29 subtype_mandatory_characterized_object

The **subtype_mandatory_characterized_object** rule specifies the permitted usage of the **characterized_object**. The **characterized_object** entity shall be limited to its use in the definition of a **feature_definition**, **feature_component_definition**, or **ordered_part**.

EXPRESS specification:

```
*)
RULE subtype_mandatory_characterized_object FOR (characterized_object);
WHERE
  WR1: SIZEOF ( QUERY (csa <* characterized_object |
    NOT (SIZEOF ([ 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_DEFINITION',
      'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION',
      'FEATURE_BASED_PROCESS_PLANNING.ORDERED_PART']
    * TYPEOF(csa)) = 1))) = 0;
END_RULE;
(*
```

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**.

5.2.4.30 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'.

5.2.4.31 transition_feature_on_part_boundary

The **transition_feature_on_part_boundary** rule specifies that each instance of **transition_feature** shall lie on the boundary of the part for which it is defined.

EXPRESS specification:

```
*)
RULE transition_feature_on_part_boundary FOR
  (transition_feature);
WHERE
  WR1: SIZEOF (QUERY (tf <* transition_feature |
    NOT (tf.product_definitional))) = 0;
END_RULE;
(*
```

Argument definitions:

transition_feature: the set of all instances of **transition_feature** entities.

Formal propositions:

WR1: For each instance of **transition_feature**, **product_definitional** shall have a value of true.

5.2.5 feature_based_process_planning function definitions

5.2.5.1 msb_shells

The **msb_shells** function builds the set of all **closed_shells** used in the definition of a **manifold_solid_brep**.

EXPRESS specification:

```
*)
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;
(*
```

Argument definitions:

brep: (input) a **manifold_solid_brep** for which a set of **closed_shell** components is required.

schema_name: (input) a STRING giving the name of the schema to which brep belongs.

5.2.5.2 number_of_representation_items

The **number_of_representation_items** function checks the number of **representation_items** in the **items** set of the implicit representation of a **machining_feature**. The **number_of_representation_items** function returns true if the number of **representation_items** are correct for each type of **feature_definition** and false if the number of **representation_items** are incorrect.

EXPRESS specification:

```
*)
FUNCTION number_of_representation_items (mf : machining_feature) : BOOLEAN;

    IF 'FEATURE_BASED_PROCESS_PLANNING.BOSS' IN TYPEOF (mf)
    THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
```

```

    NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.FLAT_FACE' IN TYPEOF (mf)
    THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.MARKING' IN TYPEOF (mf)
    THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 6))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.OUTER_ROUND' IN TYPEOF (mf)
    THEN
    IF mf.description = 'outer diameter'
        THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (impl_rep.used_representation.items) = 3))) = 0))) = 0);
        ELSE
        IF mf.description = 'outer diameter to shoulder'
            THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
            'DEFINITION') |
            NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS'
            IN TYPEOF (pdr.used_representation)) |
            NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
            END_IF;
        END_IF;
    END_IF;

```

```

END_IF;
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.OUTSIDE_PROFILE' IN TYPEOF (mf)
THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) |
NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.POCKET' IN TYPEOF (mf)
THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) |
NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.PROTRUSION' IN TYPEOF (mf)
THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) |
NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.REMOVAL_VOLUME' IN TYPEOF (mf)
THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') |
NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation))) |
NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
END_IF;

```

```

IF 'FEATURE_BASED_PROCESS_PLANNING.REVOLVED_PROFILE' IN TYPEOF (mf)
  THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF (pdr.used_representation))) |
      NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF (mf)
  THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF (pdr.used_representation))) |
      NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_END' IN TYPEOF (mf)
  THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF (pdr.used_representation))) |
      NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.SLOT' IN TYPEOF (mf)
  THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF (pdr.used_representation))) |
      NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.SPHERICAL_CAP' IN TYPEOF (mf)
  THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +

```

```

    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 3))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.STEP' IN TYPEOF (mf)
THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.THREAD' IN TYPEOF (mf)
THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT ({8 <= SIZEOF (impl_rep.used_representation.items) <= 11})))
    = 0))) = 0);
END_IF;

IF 'FEATURE_BASED_PROCESS_PLANNING.TURNED_KNURL' IN TYPEOF (mf)
THEN RETURN (SIZEOF (QUERY (pd <* USEDIN (mf,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') |
    NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF (pdr.used_representation)) |
    NOT ({6 <= SIZEOF (impl_rep.used_representation.items) <= 9})))
    = 0))) = 0);
END_IF;
RETURN (TRUE);
END_FUNCTION;
(*

```

Argument definitions:

mf: (input) a **machining_feature** for which the number of **representation_items** is being checked.

```
*)  
END_SCHEMA;                                -- feature_based_process_planning  
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-224:1999

6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation methods supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods: ISO 10303-21. Requirements with respect to implementation methods are specified in annex C.

The Protocol Information Conformance Statement (PICS) proforma lists the options or the combinations of options that may be included in the implementation. The PICS proforma is provided in annex D.

This part of ISO 10303 provides for only one option that may be supported by an implementation. This option shall all be supported by a single class of conformance which consist of all the units of functionality for this part of ISO 10303.

This conformance class is characterized as follows:

- Feature based process planning and shape represented by advanced b-rep.

NOTE - ISO 10303-324¹⁾ defines the abstract test suite to be used in the assessment of conformance. ISO 10303-32¹⁾ describes the conformance assessment process.

¹⁾ To be published.

Annex A

(normative)

AIM EXPRESS expanded listing

The following EXPRESS is the expanded form of the short form schema given in 5.2. In the event of any discrepancy between the short form and this expanded listing, the expanded listing shall be used.

SCHEMA feature_based_process_planning;

```

TYPE angle_relator = ENUMERATION OF
    (equal,
     large,
     small);
END_TYPE; -- angle_relator

```

```

TYPE axis2_placement = SELECT
    (axis2_placement_2d,
     axis2_placement_3d);
END_TYPE; -- axis2_placement

```

```

TYPE b_spline_curve_form = ENUMERATION OF
    (polyline_form,
     circular_arc,
     elliptic_arc,
     parabolic_arc,
     hyperbolic_arc,
     unspecified);
END_TYPE; -- b_spline_curve_form

```

```

TYPE b_spline_surface_form = ENUMERATION OF
    (plane_surf,
     cylindrical_surf,
     conical_surf,
     spherical_surf,
     toroidal_surf,
     surf_of_revolution,
     ruled_surf,
     generalised_cone,
     quadric_surf,
     surf_of_linear_extrusion,
     unspecified);
END_TYPE; -- b_spline_surface_form

```

```

TYPE boolean_operand = SELECT
    (solid_model);
END_TYPE; -- boolean_operand

```



```
TYPE characterized_definition = SELECT
  (characterized_object,
   characterized_product_definition,
   shape_definition);
END_TYPE; -- characterized_definition

TYPE characterized_product_definition = SELECT
  (product_definition,
   product_definition_relationship);
END_TYPE; -- characterized_product_definition

TYPE context_dependent_measure = REAL;
END_TYPE; -- context_dependent_measure

TYPE count_measure = NUMBER;
END_TYPE; -- count_measure

TYPE curve_on_surface = SELECT
  (pcurve,
   surface_curve,
   composite_curve_on_surface);
END_TYPE; -- curve_on_surface

TYPE date_time_select = SELECT
  (date);
END_TYPE; -- date_time_select

TYPE day_in_month_number = INTEGER;
END_TYPE; -- day_in_month_number

TYPE day_in_week_number = INTEGER;
WHERE
  wr1: ((1 <= SELF) AND (SELF <= 7));
END_TYPE; -- day_in_week_number

TYPE day_in_year_number = INTEGER;
END_TYPE; -- day_in_year_number

TYPE descriptive_measure = STRING;
END_TYPE; -- descriptive_measure

TYPE dimension_count = INTEGER;
WHERE
  wr1: (SELF > 0);
END_TYPE; -- dimension_count

TYPE dimensional_characteristic = SELECT
  (dimensional_location,
   dimensional_size);
END_TYPE; -- dimensional_characteristic

TYPE feature_based_pp_action_item = SELECT
  (product_definition_formation);
END_TYPE; -- feature_based_pp_action_item
```

```
TYPE feature_based_pp_action_request_item = SELECT
  (product_definition_formation);
END_TYPE; -- feature_based_pp_action_request_item

TYPE feature_based_pp_approved_item = SELECT
  (directed_action,
   product_definition_formation);
END_TYPE; -- feature_based_pp_approved_item

TYPE feature_based_pp_classified_item = SELECT
  (product_definition_formation);
END_TYPE; -- feature_based_pp_classified_item

TYPE feature_based_pp_dated_item = SELECT
  (directed_action,
   document_with_class,
   versioned_action_request);
END_TYPE; -- feature_based_pp_dated_item

TYPE feature_based_pp_document_item = SELECT
  (action_method,
   externally_defined_feature_definition,
   directed_action,
   dimensional_characteristic_representation,
   property_definition);
END_TYPE; -- feature_based_pp_document_item

TYPE feature_based_pp_ordered_item = SELECT
  (product_definition_formation);
END_TYPE; -- feature_based_pp_ordered_item

TYPE feature_based_pp_organization_item = SELECT
  (action_directive,
   product_definition_formation);
END_TYPE; -- feature_based_pp_organization_item

TYPE feature_based_pp_person_and_organization_item = SELECT
  (action_directive,
   product_definition_formation);
END_TYPE; -- feature_based_pp_person_and_organization_item

TYPE geometric_set_select = SELECT
  (point,
   curve,
   surface);
END_TYPE; -- geometric_set_select

TYPE identifier = STRING;
END_TYPE; -- identifier
```

```
TYPE knot_type = ENUMERATION OF
  (uniform_knots,
   unspecified,
   quasi_uniform_knots,
   piecewise_bezier_knots);
END_TYPE; -- knot_type

TYPE label = STRING;
END_TYPE; -- label

TYPE length_measure = REAL;
END_TYPE; -- length_measure

TYPE limit_condition = ENUMERATION OF
  (maximum_material_condition,
   least_material_condition,
   regardless_of_feature_size);
END_TYPE; -- limit_condition

TYPE list_of_reversible_topology_item = LIST [0:?] OF
  reversible_topology_item;
END_TYPE; -- list_of_reversible_topology_item

TYPE measure_value = SELECT
  (length_measure,
   plane_angle_measure,
   parameter_value,
   context_dependent_measure,
   descriptive_measure,
   positive_length_measure,
   positive_plane_angle_measure,
   count_measure);
END_TYPE; -- measure_value

TYPE month_in_year_number = INTEGER;
WHERE
  wr1: ((1 <= SELF) AND (SELF <= 12));
END_TYPE; -- month_in_year_number

TYPE parameter_value = REAL;
END_TYPE; -- parameter_value

TYPE pcurve_or_surface = SELECT
  (pcurve,
   surface);
END_TYPE; -- pcurve_or_surface

TYPE person_organization_select = SELECT
  (person,
   organization,
   person_and_organization);
END_TYPE; -- person_organization_select
```

```

TYPE plane_angle_measure = REAL;
END_TYPE; -- plane_angle_measure

TYPE positive_length_measure = length_measure;
WHERE
    wr1: (SELF > 0);
END_TYPE; -- positive_length_measure

TYPE positive_plane_angle_measure = plane_angle_measure;
WHERE
    wr1: (SELF > 0);
END_TYPE; -- positive_plane_angle_measure

TYPE preferred_surface_curve_representation = ENUMERATION OF
    (curve_3d,
     pcurve_s1,
     pcurve_s2);
END_TYPE; -- preferred_surface_curve_representation

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 set_of_reversible_topology_item = SET [0:?] OF
    reversible_topology_item;
END_TYPE; -- set_of_reversible_topology_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,
     peta,
     tera,
     giga,
     mega,

```

```
kilo,
hecto,
deca,
deci,
centi,
milli,
micro,
nano,
pico,
femto,
atto);
END_TYPE; -- si_prefix

TYPE si_unit_name = ENUMERATION OF
(metre,
gram,
second,
ampere,
kelvin,
mole,
candela,
radian,
steradian,
hertz,
newton,
pascal,
joule,
watt,
coulomb,
volt,
farad,
ohm,
siemens,
weber,
tesla,
henry,
degree_celsius,
lumen,
lux,
becquerel,
gray,
sievert);
END_TYPE; -- si_unit_name

TYPE source_item = SELECT
(identifier);
END_TYPE; -- source_item

TYPE supported_item = SELECT
(action_directive,
action,
action_method);
END_TYPE; -- supported_item
```

```
TYPE text = STRING;
END_TYPE; -- text

TYPE tolerance_method_definition = SELECT
  (tolerance_value,
   limits_and_fits);
END_TYPE; -- tolerance_method_definition

TYPE tolerance_select = SELECT
  (geometric_tolerance,
   plus_minus_tolerance);
END_TYPE; -- tolerance_select

TYPE transformation = SELECT
  (functionally_defined_transformation);
END_TYPE; -- transformation

TYPE transition_code = ENUMERATION OF
  (discontinuous,
   continuous,
   cont_same_gradient,
   cont_same_gradient_same_curvature);
END_TYPE; -- transition_code

TYPE trimming_select = SELECT
  (cartesian_point,
   parameter_value);
END_TYPE; -- trimming_select

TYPE unit = SELECT
  (named_unit);
END_TYPE; -- unit

TYPE value_qualifier = SELECT
  (precision_qualifier,
   type_qualifier,
   uncertainty_qualifier);
END_TYPE; -- value_qualifier

TYPE vector_or_direction = SELECT
  (vector,
   direction);
END_TYPE; -- vector_or_direction

TYPE week_in_year_number = INTEGER;
WHERE
  wr1: ((1 <= SELF) AND (SELF <= 53));
END_TYPE; -- week_in_year_number

TYPE year_number = INTEGER;
END_TYPE; -- year_number
```

```
ENTITY action;
    name      : label;
    description : text;
    chosen_method : action_method;
END_ENTITY; -- action

ENTITY action_assignment
    ABSTRACT SUPERTYPE;
    assigned_action : action;
END_ENTITY; -- action_assignment

ENTITY action_directive;
    name      : label;
    description : text;
    analysis   : text;
    comment    : text;
    requests   : SET [1:?] OF versioned_action_request;
END_ENTITY; -- action_directive

ENTITY action_method;
    name      : label;
    description : text;
    consequence : text;
    purpose    : text;
END_ENTITY; -- action_method

ENTITY action_relationship;
    name      : label;
    description : text;
    relating_action : action;
    related_action : action;
END_ENTITY; -- action_relationship

ENTITY action_request_assignment
    ABSTRACT SUPERTYPE;
    assigned_action_request : versioned_action_request;
END_ENTITY; -- action_request_assignment

ENTITY action_request_solution;
    method : action_method;
    request : versioned_action_request;
END_ENTITY; -- action_request_solution

ENTITY action_request_status;
    status : label;
    assigned_request : versioned_action_request;
END_ENTITY; -- action_request_status

ENTITY action_status;
    status : label;
    assigned_action : executed_action;
END_ENTITY; -- action_status
```

```

ENTITY address;
  internal_location      : OPTIONAL label;
  street_number          : OPTIONAL label;
  street                 : OPTIONAL label;
  postal_box             : OPTIONAL label;
  town                   : OPTIONAL label;
  region                 : OPTIONAL label;
  postal_code            : OPTIONAL label;
  country                : OPTIONAL label;
  facsimile_number       : OPTIONAL label;
  telephone_number       : OPTIONAL label;
  electronic_mail_address : OPTIONAL label;
  telex_number           : OPTIONAL label;
WHERE
  wr1: (EXISTS(internal_location) OR EXISTS(street_number) OR EXISTS(
    street) OR EXISTS(postal_box) OR EXISTS(town) OR EXISTS(
    region) OR EXISTS(postal_code) OR EXISTS(country) OR EXISTS(
    facsimile_number) OR EXISTS(telephone_number) OR EXISTS(
    electronic_mail_address) OR EXISTS(telex_number));
END_ENTITY; -- address

ENTITY advanced_brep_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  wr1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP',
    'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.AXIS2_PLACEMENT_3D'] *
    TYPEOF(it)) = 1)) )) = 0);
  wr2: (SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP',
    'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM'] * TYPEOF(it))=1))) >= 1);
  wr3: (SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
    'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP' IN
    TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( csh <* msb.shells(msb)
    | (NOT (SIZEOF(QUERY ( fcs <* csh\connected_face_set.
    cfs_faces | (NOT (
    'FEATURE_BASED_PROCESS_PLANNING.ADVANCED_FACE' IN TYPEOF(fcs))) ))
    = 0)) )) = 0)) )) = 0);
  wr4: (SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
    'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP' IN
    TYPEOF(it)) ) | (
    'FEATURE_BASED_PROCESS_PLANNING.ORIENTED_CLOSED_SHELL' IN
    TYPEOF(msb.outer)) )) = 0);
  wr5: (SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
    'FEATURE_BASED_PROCESS_PLANNING.BREP_WITH_VOIDS' IN TYPEOF(
    it))) | (NOT (SIZEOF(QUERY ( csh <* brv.voids | csh.orientation
    ))=0))))=0);
  wr6: (SIZEOF(QUERY ( mi <* QUERY ( it <* items | (
    'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM' IN TYPEOF(it)) )
    | (NOT (('FEATURE_BASED_PROCESS_PLANNING.' +
    'ADVANCED_BREP_SHAPE_REPRESENTATION') IN TYPEOF(mi.
    mapping_source.mapped_representation))) )) = 0);
END_ENTITY; -- advanced_brep_shape_representation

```


ENTITY advanced_face

SUBTYPE OF (face_surface);

WHERE

```

wr1 : (SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.ELEMENTARY_SURFACE',
'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_SURFACE',
'FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE']) * TYPEOF(
face_geometry)) = 1);
wr2 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
.edge_list | (NOT (
'FEATURE_BASED_PROCESS_PLANNING.EDGE_CURVE' IN TYPEOF(oe.
edge_element)))) ) = 0)) ) = 0);
wr3 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
.edge_list | (NOT (SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.LINE',
'FEATURE_BASED_PROCESS_PLANNING.CONIC',
'FEATURE_BASED_PROCESS_PLANNING.POLYLINE',
'FEATURE_BASED_PROCESS_PLANNING.SURFACE_CURVE',
'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_CURVE']) * TYPEOF(
oe.edge_element\edge_curve.edge_geometry)) = 1)) ) = 0)) ) = 0);
wr4 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
.edge_list | (NOT ((
'FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN TYPEOF(oe.
edge_start)) AND (
'FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN TYPEOF(
oe.edge_start\vertex_point.vertex_geometry)) AND (
'FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN TYPEOF(oe.
edge_end)) AND (
'FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN TYPEOF(
oe.edge_end\vertex_point.vertex_geometry)))) ) = 0)) ) = 0);
wr5 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
bound)) ) | ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_PATH'
IN TYPEOF(elp_fbnds.bound)) ) = 0);
wr6 : ((NOT ('FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE' IN
TYPEOF(face_geometry))) OR (SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.LINE',
'FEATURE_BASED_PROCESS_PLANNING.CONIC',
'FEATURE_BASED_PROCESS_PLANNING.POLYLINE',
'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_CURVE']) * TYPEOF(
face_geometry\swept_surface.swept_curve)) = 1));
wr7 : (SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* SELF.bounds | (
'FEATURE_BASED_PROCESS_PLANNING.VERTEX_LOOP' IN TYPEOF(bnds.
bound)) ) | (NOT ((
'FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN TYPEOF(
vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex)) AND (
'FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN TYPEOF(
vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex\

```

```

        vertex_point.vertex_geometry)))))) = 0);
wr8 : (SIZEOF(QUERY ( bnd <* SELF.bounds | (NOT (SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP',
    'FEATURE_BASED_PROCESS_PLANNING.VERTEX_LOOP'] * TYPEOF(bnd.
    bound))) = 1))) = 0);
wr9 : (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.bounds | (
    'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
    bound))) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
    .edge_list | ((
    'FEATURE_BASED_PROCESS_PLANNING.SURFACE_CURVE' IN TYPEOF(oe
    .edge_element\edge_curve.edge_geometry)) AND (NOT (SIZEOF(
    QUERY ( sc_ag <* oe.edge_element\edge_curve.edge_geometry\
    surface_curve.associated_geometry | (NOT (
    'FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(sc_ag)))) ) )
    = 0)))))) = 0))) = 0);
wr10: (((NOT ('FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE' IN
    TYPEOF(face_geometry))) OR (NOT (
    'FEATURE_BASED_PROCESS_PLANNING.POLYLINE' IN TYPEOF(
    face_geometry\swept_surface.swept_curve))) OR (SIZEOF(
    face_geometry\swept_surface.swept_curve\polyline.points) <
    3)) AND (SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* SELF.
    bounds | ('FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN
    TYPEOF(bnds.bound))) ) | (NOT (SIZEOF(QUERY ( oe <*
    elp_fbnds.bound\path.edge_list | ((
    'FEATURE_BASED_PROCESS_PLANNING.POLYLINE' IN TYPEOF(oe.
    edge_element\edge_curve.edge_geometry)) AND (NOT (SIZEOF(oe
    .edge_element\edge_curve.edge_geometry\polyline.points) < 3)))
    )) = 0))) = 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: (SELF\geometric_tolerance.name = 'angularity');
    wr2: (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
        datum_system) = 1);
END_ENTITY; -- angularity_tolerance

ENTITY application_context;
  application : text;
  INVERSE
    context_elements : SET [1:?] OF application_context_element FOR
        frame_of_reference;
END_ENTITY; -- application_context

```

```

ENTITY application_context_element
  SUPERTYPE OF (ONEOF (product_context, product_definition_context));
  name : label;
  frame_of_reference : application_context;
END_ENTITY; -- application_context_element

ENTITY application_protocol_definition;
  status : label;
  application_interpreted_model_schema_name : label;
  application_protocol_year : year_number;
  application : application_context;
END_ENTITY; -- application_protocol_definition

ENTITY applied_area
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 2)) )) = 0)) )) = 0);
    wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'area length')) )) = 1)) ))
      = 0)) )) = 0);
    wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

        used_representation)) ) | (NOT (SIZEOF(QUERY ( it < *
        impl_rep.used_representation.items | ((
        'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
        AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
END_ENTITY; -- applied_area

ENTITY approval;
    status : approval_status;
    level : label;
END_ENTITY; -- approval

ENTITY approval_assignment
    ABSTRACT SUPERTYPE;
    assigned_approval : approval;
END_ENTITY; -- approval_assignment

ENTITY approval_date_time;
    date_time : date_time_select;
    dated_approval : approval;
END_ENTITY; -- approval_date_time

ENTITY approval_person_organization;
    person_organization : person_organization_select;
    authorized_approval : approval;
    role : approval_role;
END_ENTITY; -- approval_person_organization

ENTITY approval_relationship;
    name : label;
    description : text;
    relating_approval : approval;
    related_approval : approval;
END_ENTITY; -- approval_relationship

ENTITY approval_role;
    role : label;
END_ENTITY; -- approval_role

ENTITY approval_status;
    name : label;
END_ENTITY; -- approval_status

ENTITY axis1_placement
    SUBTYPE OF (placement);
    axis : OPTIONAL direction;
    DERIVE
    z : direction := NVL(normalise(axis), direction([0,0,1]));
    WHERE
    wr1: (SELF\geometric_representation_item.dim = 3);
END_ENTITY; -- axis1_placement

```

```

ENTITY axis2_placement_2d
  SUBTYPE OF (placement);
  ref_direction : OPTIONAL direction;
  DERIVE
    p : LIST [2:2] OF direction := build_2axes(ref_direction);
  WHERE
    wr1: (SELF\geometric_representation_item.dim = 2);
END_ENTITY; -- axis2_placement_2d

ENTITY axis2_placement_3d
  SUBTYPE OF (placement);
  axis          : OPTIONAL direction;
  ref_direction : OPTIONAL direction;
  DERIVE
    p : LIST [3:3] OF direction := build_axes(axis,ref_direction);
  WHERE
    wr1: (SELF\placement.location.dim = 3);
    wr2: ((NOT EXISTS(axis)) OR (axis.dim = 3));
    wr3: ((NOT EXISTS(ref_direction)) OR (ref_direction.dim = 3));
    wr4: ((NOT EXISTS(axis)) OR (NOT EXISTS(ref_direction)) OR (
      cross_product(axis,ref_direction).magnitude > 0));
END_ENTITY; -- axis2_placement_3d

ENTITY b_spline_curve
  SUPERTYPE OF (ONEOF (uniform_curve,b_spline_curve_with_knots,
    quasi_uniform_curve,bezier_curve) ANDOR rational_b_spline_curve)
  SUBTYPE OF (bounded_curve);
  degree          : INTEGER;
  control_points_list : LIST [2:?] OF cartesian_point;
  curve_form      : b_spline_curve_form;
  closed_curve    : LOGICAL;
  self_intersect  : LOGICAL;
  DERIVE
    upper_index_on_control_points : INTEGER := SIZEOF(
      control_points_list) - 1;
    control_points                : ARRAY [0:
      upper_index_on_control_points] OF
      cartesian_point := list_to_array(
        control_points_list,0,
        upper_index_on_control_points);
  WHERE
    wr1: (('FEATURE_BASED_PROCESS_PLANNING.UNIFORM_CURVE' IN TYPEOF(SELF))
      OR ('FEATURE_BASED_PROCESS_PLANNING.QUASI_UNIFORM_CURVE' IN
        TYPEOF(SELF)) OR (
        'FEATURE_BASED_PROCESS_PLANNING.BEZIER_CURVE' IN TYPEOF(SELF))
      OR (
        'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_CURVE_WITH_KNOTS'
        IN TYPEOF(SELF)));
END_ENTITY; -- b_spline_curve

ENTITY b_spline_curve_with_knots
  SUBTYPE OF (b_spline_curve);
  knot_multiplicities : LIST [2:?] OF INTEGER;
  knots               : LIST [2:?] OF parameter_value;
  knot_spec           : knot_type;

```

```

DERIVE
  upper_index_on_knots : INTEGER := SIZEOF(knots);
WHERE
  wr1: constraints_param_b_spline(degree, upper_index_on_knots,
    upper_index_on_control_points, knot_multiplicities, knots);
  wr2: (SIZEOF(knot_multiplicities) = upper_index_on_knots);
END_ENTITY; -- b_spline_curve_with_knots

ENTITY b_spline_surface
  SUPERTYPE OF (ONEOF (b_spline_surface_with_knots, uniform_surface,
    quasi_uniform_surface, bezier_surface) ANDOR
    rational_b_spline_surface)
  SUBTYPE OF (bounded_surface);
  u_degree      : INTEGER;
  v_degree      : INTEGER;
  control_points_list : LIST [2:?] OF LIST [2:?] OF cartesian_point;
  surface_form   : b_spline_surface_form;
  u_closed      : LOGICAL;
  v_closed      : LOGICAL;
  self_intersect : LOGICAL;
DERIVE
  u_upper      : INTEGER := SIZEOF(control_points_list) - 1;
  v_upper      : INTEGER := SIZEOF(control_points_list[1]) - 1;
  control_points : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF
    cartesian_point := make_array_of_array(
      control_points_list, 0, u_upper, 0, v_upper);
WHERE
  wr1: (('FEATURE_BASED_PROCESS_PLANNING.UNIFORM_SURFACE' IN TYPEOF(
    SELF)) OR ( 'FEATURE_BASED_PROCESS_PLANNING.QUASI_UNIFORM_SURFACE' IN
    TYPEOF(SELF)) OR (
    'FEATURE_BASED_PROCESS_PLANNING.BEZIER_SURFACE' IN TYPEOF(
    SELF)) OR ( 'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_SURFACE_WITH_KNOTS'
    IN TYPEOF(SELF)));
END_ENTITY; -- b_spline_surface

ENTITY b_spline_surface_with_knots
  SUBTYPE OF (b_spline_surface);
  u_multiplicities : LIST [2:?] OF INTEGER;
  v_multiplicities : LIST [2:?] OF INTEGER;
  u_knots          : LIST [2:?] OF parameter_value;
  v_knots          : LIST [2:?] OF parameter_value;
  knot_spec        : knot_type;
DERIVE
  knot_u_upper : INTEGER := SIZEOF(u_knots);
  knot_v_upper : INTEGER := SIZEOF(v_knots);
WHERE
  wr1: constraints_param_b_spline(SELF\b_spline_surface.u_degree,
    knot_u_upper, SELF\b_spline_surface.u_upper, u_multiplicities,
    u_knots);
  wr2: constraints_param_b_spline(SELF\b_spline_surface.v_degree,
    knot_v_upper, SELF\b_spline_surface.v_upper, v_multiplicities,
    v_knots);
  wr3: (SIZEOF(u_multiplicities) = knot_u_upper);
  wr4: (SIZEOF(v_multiplicities) = knot_v_upper);
END_ENTITY; -- b_spline_surface_with_knots

```

```

ENTITY bezier_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- bezier_curve

ENTITY bezier_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- bezier_surface

ENTITY block_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
    wr1: (SIZEOF(SELF.items) = 4);
    wr2: (SIZEOF(QUERY ( it <* SELF.items | ((
      'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
      AND (it.name = 'orientation')) )) = 1);
    wr3: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF(
      ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1);
    wr4: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF(
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1);
    wr5: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF(
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'height')) )) = 1);
  END_ENTITY; -- block_shape_representation

ENTITY boss
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (SELF.description IN ['circular', 'non-circular']);
    wr2: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION') |
      ( 'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
      sa_occ.description = 'boss height occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATED_SHAPE_ASPECT') | ((sar.description =
      'path feature component usage') AND ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) ) | ((
      'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
      TYPEOF(sdr.relateing_shape_aspect)) AND (sdr.
      relateing_shape_aspect.description = 'linear') AND (sdr.name
      = 'boss height')) )) = 1)) )) = 1)) )) = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
      QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'fillet radius')) )) = 1)) ))
= 0)) )) = 0);
wr4: ((NOT (SELF.description = 'circular')) OR (SIZEOF(
QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'circular profile occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE' IN
TYPEOF(sdr.relying_shape_aspect)) )) = 1)) )) = 1)) )) = 0));
wr5: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'top condition occurrence') AND (
SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'boss top usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) ) | (
'FEATURE_BASED_PROCESS_PLANNING.BOSS_TOP' IN TYPEOF(fcr.
relying_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
wr6: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') | (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'change in diameter occurrence') AND (
SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'taper usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) ) | (
'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
relying_shape_aspect)) )) = 1)) )) <= 1)) )) = 0);
wr7: ((NOT (SELF.description = 'non-circular')) OR (SIZEOF(
QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') | (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'enclosed boundary occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,

```



```

'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (SIZEOF(
['FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE'] *
TYPEOF(sdr.relatng_shape_aspect)) = 1) )) = 1)) )) = 1)) ))
= 0));
wr8: ((NOT (SELF.description = 'non-circular')) OR (SIZEOF(
QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') | (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_OF_SHAPE') | ((
sa_occ.description = 'change in boundary occurrence') AND (
SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATING_SHAPE_ASPECT') | ((sar.description =
'taper usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
related_shape_aspect)) AND (fcr.related_shape_aspect.
description = 'angle taper')))) = 1)) )) = 1)) )) = 0));
END_ENTITY; -- boss

ENTITY boss_top
SUBTYPE OF (shape_aspect);
WHERE
wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
IN TYPEOF(SELF.of_shape.definition));
wr2: (SELF.description IN ['planar', 'non-planar']);
wr3: ((NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'DIRECTION_SHAPE_REPRESENTATION') IN TYPEOF(pdr.
used_representation)) )) = 1)) )) = 0));
wr4: ((NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'LOCATION_SHAPE_REPRESENTATION') IN TYPEOF(pdr.
used_representation)) )) = 1)) )) = 0));
wr5: ((NOT (SELF.description = 'non-planar')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,

```

```

        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'FACE_SHAPE_REPRESENTATION') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0));
wr6: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATING_SHAPE_ASPECT') | ((sar.description =
        'boss top usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF(sar))) ) | ((fcr.related_shape_aspect.description
        = 'top condition occurrence') AND (
        'FEATURE_BASED_PROCESS_PLANNING.BOSS' IN TYPEOF(fcr.
        related_shape_aspect.of_shape.definition))) )) >= 1);
wr7: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0);
wr8: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 1)) )) = 0)) )) = 0);
wr9: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'boss top orientation') AND (
        it.description IN ['boss start','boss end'])) )) = 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\cartesian_transformation_operator.dim = 3);
END_ENTITY; -- cartesian_transformation_operator_3d

ENTITY chamfer
  SUBTYPE OF (transition_feature);
  WHERE
    wr1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'chamfer face')) )) <= 1)) )) = 0);
    wr2: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,

```

```

        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATING_SHAPE_ASPECT') |
        ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF(sar)) ) | ((
        'FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET' IN TYPEOF(
        fcr.related_shape_aspect)) AND (fcr.related_shape_aspect.
        description = 'first offset')) )) = 1);
wr3: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATING_SHAPE_ASPECT') |
        ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF(sar)) ) | ((
        'FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET' IN TYPEOF(
        fcr.related_shape_aspect)) AND (fcr.related_shape_aspect.
        description = 'second offset')) )) = 1);
END_ENTITY; -- chamfer

ENTITY chamfer_offset
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: (SELF.description IN ['first offset','second offset']);
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 1)) )) = 0)) )) = 0);
    wr4: ((NOT (SELF.description = 'first offset')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'offset amount')) )) = 1)) ))
        = 0)) )) = 0));
    wr5: ((NOT (SELF.description = 'first offset')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +

```

```

        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'first face shape')) )) = 1)) ))
        = 0));
wr6: ((NOT (SELF.description = 'second offset')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'offset amount')) OR ((
        SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'offset angle')))) )) = 1)) ))
        = 0)) )) = 0));
wr7: ((NOT (SELF.description = 'second offset')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'second face shape')) )) = 1)) ))
        = 0));
wr8: (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) ) | (
        'FEATURE_BASED_PROCESS_PLANNING.CHAMFER' IN TYPEOF(sdr.
        relating_shape_aspect)) )) >= 1);
END_ENTITY; -- chamfer_offset

ENTITY characterized_object;
    name : label;
    description : text;
END_ENTITY; -- characterized_object

ENTITY circle
    SUBTYPE OF (conic);
    radius : positive_length_measure;
END_ENTITY; -- circle

```

```

ENTITY circular_closed_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 2)) )) = 0)) )) = 0);
    wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (SIZEOF(QUERY ( it <* impl_rep.
      used_representation.items | ((
      'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
      AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
    wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1)) )) =
      0)) )) = 0);
  END ENTITY; -- circular_closed_profile

ENTITY circular_pattern
  SUBTYPE OF (replicate_feature);
  WHERE
    wr1: (SIZEOF(USEDIN(SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')) <= 3);
    wr2: (SIZEOF(QUERY ( sar <*
      USEDIN(SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (NOT
('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
IN TYPEOF(sar))) )) = 0);
wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) = 1)) )) = 0);
wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
impl_rep.used_representation.items) = 5))) = 0))) = 0);
wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1)) )) =
0))) = 0);
wr6: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'base feature rotation')) ))
= 1)) )) = 0))) = 0);
wr7: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
QUERY ( it <* impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (
'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component)) AND (it.name =

```

```

        'number of features')) )) = 1)) )) = 0)) )) = 0);
wr8: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
    ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'angular spacing')))) = 1)) ))
    = 0)) )) = 0);
wr9: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((
    'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
    AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
END_ENTITY; -- circular_pattern

ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: (SELF\geometric_tolerance.name = 'circular runout');
    wr2: (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: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
        IN TYPEOF(SELF.of_shape.definition));
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 1)) )) = 0)) )) = 0);

```



```

wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
  | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'FEATURE_BASED_PROCESS_PLANNING.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
  'FEATURE_BASED_PROCESS_PLANNING.' +
  'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
  used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
  impl_rep.used_representation.items | ((
  'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
  AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
  'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
  'FEATURE_BASED_PROCESS_PLANNING.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
  'FEATURE_BASED_PROCESS_PLANNING.' +
  'PATH_SHAPE_REPRESENTATION') IN TYPEOF(pdr.
  used_representation)) )) = 1)) )) = 0);
END_ENTITY; -- closed_path_profile

ENTITY closed_shell
  SUBTYPE OF (connected_face_set);
END_ENTITY; -- closed_shell

ENTITY composite_curve
  SUBTYPE OF (bounded_curve);
  segments      : LIST [1:?] OF composite_curve_segment;
  self_intersect : LOGICAL;
  DERIVE
    n_segments   : INTEGER := SIZEOF(segments);
    closed_curve : LOGICAL := segments[n_segments].transition <>
                        discontinuous;
  WHERE
    wr1: (((NOT closed_curve) AND (SIZEOF(QUERY ( temp <* segments | (
      temp.transition = discontinuous) )) = 1)) OR (closed_curve
      AND (SIZEOF(QUERY ( temp <* segments | (temp.transition =
      discontinuous) )) = 0))) );
END_ENTITY; -- composite_curve

ENTITY composite_curve_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;
  transition : transition_code;
  same_sense : BOOLEAN;
  parent_curve : curve;
  INVERSE
    using_curves : BAG [1:?] OF composite_curve FOR segments;

```

```

WHERE
  wr1: ('FEATURE_BASED_PROCESS_PLANNING.BOUNDED_CURVE' IN TYPEOF(
    parent_curve));
END_ENTITY; -- composite_curve_segment

ENTITY composite_hole
  SUBTYPE OF (composite_machining_feature);
  WHERE
    wr1: (SELF.description IN ['counterbore', 'countersunk']);
    wr2: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATING_SHAPE_ASPECT') ) |
      ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
      IN TYPEOF(sar)) ) ) | (
      'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(fcr.
      related_shape_aspect)) ) ) = 2);
    wr3: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATING_SHAPE_ASPECT') ) |
      ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
      IN TYPEOF(sar)) ) ) | ((NOT (
      'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(fcr.
      related_shape_aspect))) OR (SIZEOF(QUERY ( fcr2 <*
      QUERY ( sar2 <* (USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATING_SHAPE_ASPECT') - fcr) |
      ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
      IN TYPEOF(sar2)) ) ) | ((NOT (
      'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(fcr2.
      related_shape_aspect))) OR (SIZEOF(QUERY ( pd <* USEDIN(fcr.
      related_shape_aspect,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((NOT ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation))) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((it.name = 'diameter')
      AND (SIZEOF(QUERY ( pd2 <* USEDIN(fcr2.related_shape_aspect,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr2 <* USEDIN(pd2,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((NOT ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr2.
      used_representation))) OR (SIZEOF(QUERY ( pd2 <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')

```

```

      | (NOT (SIZEOF(QUERY ( impl_rep2 <* QUERY ( pdr2 <* USEDIN(
pd2, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr2.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it2 <*
impl_rep2.used_representation.items | ((it2.name =
'diameter') AND (it\measure_with_unit.value_component = it2\
measure_with_unit.value_component)))) = 0)))) = 0)) )) = 0)) ))
= 0))))= 0))))= 0))))= 0))))= 0))))= 0))))= 0))))= 0))))= 0)) ))= 0);
wr4: ((NOT (SELF.description = 'countersunk')) OR (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATING_SHAPE_ASPECT') ) |
('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
IN TYPEOF(sar)) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(fcr.
related_shape_aspect)) AND (SIZEOF(QUERY ( rh_sdr <*
QUERY ( rh_sar <* USEDIN(fcr.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATING_SHAPE_ASPECT') ) |
('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
IN TYPEOF(rh_sar)) ) | ('FEATURE_BASED_PROCESS_PLANNING.TAPER'
IN TYPEOF(rh_sdr.related_shape_aspect)) )) = 1)) )) = 1));
END_ENTITY; -- composite_hole

ENTITY composite_machining_feature
  SUBTYPE OF (machining_feature);
  WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
      TYPEOF(SELF));
    wr2: (SIZEOF(QUERY ( sar <* SELF\composite_shape_aspect.
      component_relationships | (NOT
      ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
      IN TYPEOF(sar)) )) = 0);
    wr3: (SIZEOF(QUERY ( sar <* SELF\composite_shape_aspect.
      component_relationships | (NOT (
      'FEATURE_BASED_PROCESS_PLANNING.MACHINING_FEATURE' IN
      TYPEOF(sar.related_shape_aspect))) )) = 0);
  END_ENTITY; -- composite_machining_feature

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 concentricity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: (SELF\geometric_tolerance.name = 'concentricity');
    wr2: (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
      datum_system) = 1);
  END_ENTITY; -- concentricity_tolerance

```

```

ENTITY conic
  SUPERTYPE OF (ONEOF (circle,ellipse,hyperbola,parabola))
  SUBTYPE OF (curve);
  position : axis2_placement;
END_ENTITY; -- conic

ENTITY conical_surface
  SUBTYPE OF (elementary_surface);
  radius : length_measure;
  semi_angle : plane_angle_measure;
  WHERE
    wr1: (radius >= 0);
END_ENTITY; -- conical_surface

ENTITY connected_face_set
  SUPERTYPE OF (ONEOF (closed_shell,open_shell))
  SUBTYPE OF (topological_representation_item);
  cfs_faces : SET [1:?] OF face;
END_ENTITY; -- connected_face_set

ENTITY context_dependent_unit
  SUBTYPE OF (named_unit);
  name : label;
END_ENTITY; -- context_dependent_unit

ENTITY conversion_based_unit
  SUBTYPE OF (named_unit);
  name : label;
  conversion_factor : measure_with_unit;
END_ENTITY; -- conversion_based_unit

ENTITY curve
  SUPERTYPE OF (ONEOF (line,conic,pcurve,surface_curve))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- curve

ENTITY cylindrical_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
    wr1: (SIZEOF(SELF.items) = 3);
    wr2: (SIZEOF(QUERY ( it <* SELF.items | ((
      'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
      AND (it.name = 'orientation')) )) = 1);
    wr3: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1);
    wr4: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1);
END_ENTITY; -- cylindrical_shape_representation

```

```

ENTITY cylindrical_surface
  SUBTYPE OF (elementary_surface);
  radius : positive_length_measure;
END_ENTITY; -- cylindrical_surface

ENTITY data_environment;
  name : label;
  description : text;
  elements : SET [1:?] OF property_definition_representation;
END_ENTITY;

ENTITY date
  SUPERTYPE OF (ONEOF (calendar_date, ordinal_date,
    week_of_year_and_day_date));
  year_component : year_number;
END_ENTITY; -- date

ENTITY date_assignment
  ABSTRACT SUPERTYPE;
  assigned_date : date;
  role : date_role;
END_ENTITY; -- date_assignment

ENTITY date_role;
  name : label;
END_ENTITY; -- date_role

ENTITY datum
  SUBTYPE OF (shape_aspect);
  identification : identifier;
  INVERSE
    established_by_relationships : SET [1:?] OF
      shape_aspect_relationship FOR
      related_shape_aspect;

  WHERE
    wr1: (SIZEOF(QUERY (x <* SELF.established_by_relationships | (
      SIZEOF(TYPEOF(x.relate_shape_aspect) * [
        'FEATURE_BASED_PROCESS_PLANNING.DATUM_FEATURE',
        'FEATURE_BASED_PROCESS_PLANNING.DATUM_TARGET']) <> 1) )) = 0);
END_ENTITY; -- datum

ENTITY datum_feature
  SUBTYPE OF (shape_aspect);
  INVERSE
    feature_basis_relationship : shape_aspect_relationship FOR
    relating_shape_aspect;

  WHERE
    wr1: (SIZEOF(QUERY ( sar <* bag_to_set(USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATING_SHAPE_ASPECT')) | (NOT (
        'FEATURE_BASED_PROCESS_PLANNING.DATUM' IN TYPEOF(sar.
        related_shape_aspect))) )) = 0);
    wr2: (SELF.product_definitional = TRUE);
END_ENTITY; -- datum_feature

```

```

ENTITY datum_reference;
    precedence      : INTEGER;
    referenced_datum : datum;
    WHERE
        wr1: (precedence > 0);
END_ENTITY; -- datum_reference

ENTITY datum_target
    SUBTYPE OF (shape_aspect);
    target_id : identifier;
    INVERSE
        target_basis_relationship : shape_aspect_relationship FOR
            relating_shape_aspect;
    WHERE
        wr1: (SIZEOF(QUERY ( sar <* bag_to_set(USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATING_SHAPE_ASPECT')) | (NOT (
            'FEATURE_BASED_PROCESS_PLANNING.DATUM' IN TYPEOF(sar.
            related_shape_aspect)))) ) = 0);
        wr2: (SELF.product_definitional = TRUE);
END_ENTITY; -- datum_target

ENTITY definitional_representation
    SUBTYPE OF (representation);
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.PARAMETRIC_REPRESENTATION_CONTEXT'
            IN TYPEOF(SELF\representation.context_of_items));
END_ENTITY; -- definitional_representation

ENTITY degenerate_toroidal_surface
    SUBTYPE OF (toroidal_surface);
    select_outer : BOOLEAN;
    WHERE
        wr1: (major_radius < minor_radius);
END_ENTITY; -- degenerate_toroidal_surface

ENTITY descriptive_representation_item
    SUBTYPE OF (representation_item);
    description : text;
END_ENTITY; -- descriptive_representation_item

ENTITY dimensional_characteristic_representation;
    dimension      : dimensional_characteristic;
    representation : shape_dimension_representation;
END_ENTITY; -- dimensional_characteristic_representation

ENTITY dimensional_exponents;
    length_exponent      : REAL;
    mass_exponent         : REAL;
    time_exponent        : REAL;
    electric_current_exponent : REAL;
    thermodynamic_temperature_exponent : REAL;
    amount_of_substance_exponent : REAL;
    luminous_intensity_exponent : REAL;
END_ENTITY; -- dimensional_exponents

```

```

ENTITY dimensional_location
  SUPERTYPE OF (ONEOF (angular_location,dimensional_location_with_path))
  SUBTYPE OF (shape_aspect_relationship);
END_ENTITY; -- dimensional_location

ENTITY dimensional_location_with_path
  SUBTYPE OF (dimensional_location);
  path : shape_aspect;
END_ENTITY; -- dimensional_location_with_path

ENTITY dimensional_size
  SUPERTYPE OF (ONEOF (angular_size,dimensional_size_with_path));
  applies_to : shape_aspect;
  name : label;
  WHERE
    wr1: (applies_to.product_definitional = TRUE);
END_ENTITY; -- dimensional_size

ENTITY dimensional_size_with_path
  SUBTYPE OF (dimensional_size);
  path : shape_aspect;
END_ENTITY; -- dimensional_size_with_path

ENTITY directed_action
  SUBTYPE OF (executed_action);
  directive : action_directive;
END_ENTITY; -- directed_action

ENTITY directed_dimensional_location
  SUBTYPE OF (dimensional_location);
END_ENTITY; -- directed_dimensional_location

ENTITY direction
  SUBTYPE OF (geometric_representation_item);
  direction_ratios : LIST [2:3] OF REAL;
  WHERE
    wr1: (SIZEOF(QUERY ( tmp <* direction_ratios | (tmp <> 0) )) > 0);
END_ENTITY; -- direction

ENTITY direction_shape_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: (SIZEOF(SELF.items) = 1);
    wr2: (SIZEOF(QUERY ( it <* SELF.items | (NOT (
      'FEATURE_BASED_PROCESS_PLANNING.DIRECTION' IN TYPEOF(it))) ))
      = 0);
END_ENTITY; -- direction_shape_representation

ENTITY document;
  id : identifier;
  name : label;
  description : text;
  kind : document_type;
  UNIQUE
  ur1 : id;

```

```

END_ENTITY; -- document

ENTITY document_reference
  ABSTRACT SUPERTYPE;
  assigned_document : document;
  source             : label;
END_ENTITY; -- document_reference

ENTITY document_relationship;
  name                : label;
  description          : text;
  relating_document   : document;
  related_document     : document;
END_ENTITY; -- document_relationship

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_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')) XOR
      (NOT (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) = 1)) ) = 0)))));
wr2: ((NOT (SELF\shape_aspect.description = 'constant radius')) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        impl_rep.used_representation.items) = 3)) )) = 0)) ) = 0));
wr3: ((NOT (SELF.description = 'constant radius')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) )
      = 0));
wr4: ((NOT (SELF.description = 'constant radius')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'first offset')) )) = 1)) ))
      = 0)) ) = 0));
wr5: ((NOT (SELF.description = 'constant radius')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) = 1)) ))
      = 0)) ) = 0));

```

```

        = 0)) )) = 0));
wr6: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
    IN TYPEOF(pdr.used_representation)) AND (pdr.
    used_representation.name = 'edge round face')) )) <= 1)) ))
    = 0);
wr7: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
    IN TYPEOF(pdr.used_representation)) AND (pdr.
    used_representation.name = 'first face shape')) )) = 1)) ))
    = 0);
wr8: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
    IN TYPEOF(pdr.used_representation)) AND (pdr.
    used_representation.name = 'second face shape')) )) = 1)) ))
    = 0);
END_ENTITY; -- edge_round

ENTITY elementary_surface
    SUPERTYPE OF (ONEOF (plane,cylindrical_surface,conical_surface,
        spherical_surface,toroidal_surface))
    SUBTYPE OF (surface);
    position : axis2_placement_3d;
END_ENTITY; -- elementary_surface

ENTITY ellipse
    SUBTYPE OF (conic);
    semi_axis_1 : positive_length_measure;
    semi_axis_2 : positive_length_measure;
END_ENTITY; -- ellipse

ENTITY executed_action
    SUBTYPE OF (action);
END_ENTITY; -- executed_action

ENTITY expanded_uncertainty
    SUBTYPE OF (standard_uncertainty);
    coverage_factor : REAL;
END_ENTITY; -- expanded_uncertainty

ENTITY external_source;
    source_id : source_item;
END_ENTITY; -- external_source

```

```

ENTITY externally_defined_feature_definition
  SUBTYPE OF (feature_definition, externally_defined_item);
  WHERE
    wr1 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation))) ) | (NOT ((1 <= SIZEOF(impl_rep.items))
        AND (SIZEOF(impl_rep.items) <= 8))) ) = 0)) ) = 0));
    wr2 : (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
        IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
        pds, 'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
        | ((sa_occ.description = 'partial area occurrence') AND (
        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'applied area usage') AND ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
        'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN TYPEOF(sdr
        .relating_shape_aspect))) ) = 1)) ) <= 1)) ) = 0);
    wr3 : (SIZEOF(USEDIN(SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'FEATURE_BASED_PP_DOCUMENT_REFERENCE.ITEMS')) = 1);
    wr4 : ((NOT (SELF.description = 'marking')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'marking text')) ) = 1)) )
        = 0)) ) = 0));
    wr5 : ((NOT (SELF.description = 'thread')) OR (SIZEOF(QUERY ( pd <*
      USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'thread system') AND ((it.
        description = 'english') OR (it.description = 'metric')))) )

```

```

= 1)) )) = 0)) )) = 0));
wr6 : ((NOT (SELF.description = 'thread')) OR (SIZEOF(QUERY ( pd <*
    USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items |
    (('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'removal direction')) AND
    ((it.description = 'internal') OR
    (it.description = 'external')))) )) = 1)) )) = 0)) )) = 0));
wr7 : ((NOT (SELF.description = 'thread')) OR
    (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items |
    (('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'qualifier')) ))= 1)) ))= 0)) ))
    = 0));
wr8 : ((NOT (SELF.description = 'thread')) OR
    (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items |
    (('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'hand')) )) <= 1)) )) = 0)) ))
    = 0));
wr9 : ((NOT (SELF.description = 'thread')) OR
    (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items |
    (('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'fit class')) )) <= 1)) )) =
    0)) )) = 0));

```

```

wr10: ((NOT (SELF.description = 'thread')) OR
  (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items |
      (('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'form')) )) <= 1)) )) = 0)) ))
  = 0));
wr11: ((NOT (SELF.description = 'thread')) OR
  (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
      * TYPEOF(it)) = 2) AND (it.name = 'major diameter')) )) <=
      1)) )) = 0)) )) = 0));
wr12: ((NOT (SELF.description = 'thread')) OR
  (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.RATIO_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'number of threads')) ))
      <= 1)) )) = 0)) )) = 0));
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 | (
      'FEATURE_BASED_PROCESS_PLANNING.FACE_OUTER_BOUND' IN TYPEOF(
        temp)) )) <= 1);
END_ENTITY; -- face

ENTITY face_bound
  SUBTYPE OF (topological_representation_item);
  bound : loop;
  orientation : BOOLEAN;
END_ENTITY; -- face_bound

ENTITY face_outer_bound
  SUBTYPE OF (face_bound);
END_ENTITY; -- face_outer_bound

ENTITY face_shape_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: (SIZEOF(SELF.items) = 1);
    wr2: (SIZEOF(QUERY ( it <* SELF.items | (NOT (
      'FEATURE_BASED_PROCESS_PLANNING.FACE_SURFACE' IN TYPEOF(it))) ))
      = 0);
END_ENTITY; -- face_shape_representation

ENTITY face_surface
  SUBTYPE OF (face, geometric_representation_item);
  face_geometry : surface;
  same_sense : BOOLEAN;
END_ENTITY; -- face_surface

ENTITY feature_based_pp_action_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF feature_based_pp_action_item;
END_ENTITY; -- feature_based_pp_action_assignment

ENTITY feature_based_pp_action_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF feature_based_pp_action_request_item;
END_ENTITY; -- feature_based_pp_action_request_assignment

ENTITY feature_based_pp_approval_assignment
  SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF feature_based_pp_approved_item;
END_ENTITY; -- feature_based_pp_approval_assignment

ENTITY feature_based_pp_date_assignment
  SUBTYPE OF (date_assignment);
  items : SET [1:?] OF feature_based_pp_dated_item;
END_ENTITY; -- feature_based_pp_date_assignment

```

```

ENTITY feature_based_pp_document_reference
  SUBTYPE OF (document_reference);
  items : SET [1:?] OF feature_based_pp_document_item;
END_ENTITY; -- feature_based_pp_document_reference

ENTITY feature_based_pp_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF feature_based_pp_organization_item;
END_ENTITY; -- feature_based_pp_organization_assignment

ENTITY feature_based_pp_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF feature_based_pp_person_and_organization_item;
END_ENTITY; -- feature_based_pp_person_and_organization_assignment

ENTITY feature_based_pp_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
  items : SET [1:?] OF feature_based_pp_classified_item;
END_ENTITY; -- feature_based_pp_security_classification_assignment

ENTITY feature_component_definition
  SUBTYPE OF (characterized_object);
  WHERE
    wr1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')) = 1)) ))
      = 0);
END_ENTITY; -- feature_component_definition

ENTITY feature_component_relationship
  SUPERTYPE OF (ONEOF (pattern_omit_membership,pattern_offset_membership))
  SUBTYPE OF (shape_aspect_relationship);
  WHERE
    wr1: (SIZEOF (['FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
      'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_MACHINING_FEATURE',
      'FEATURE_BASED_PROCESS_PLANNING.TRANSITION_FEATURE'] *
      TYPEOF (SELF.relate_shape_aspect)) = 1)
    OR ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_DEFINITION' IN
      TYPEOF (SELF.relate_shape_aspect.of_shape))OR(
      'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION' IN
      TYPEOF (SELF.relate_shape_aspect.of_shape));
END_ENTITY; -- feature_component_relationship

ENTITY feature_definition
  SUBTYPE OF (characterized_object);
  WHERE
    wr1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);

```

```

wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
  | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'FEATURE_BASED_PROCESS_PLANNING.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
  'FEATURE_BASED_PROCESS_PLANNING.' +
  'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
  used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
  impl_rep.used_representation.items | (
  'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it)))
  = 1))) )) = 0))) )) = 0));
END_ENTITY; -- feature_definition

ENTITY feature_pattern
  SUBTYPE OF (replicate_feature);
  WHERE
    wr1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
      (('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
      IN TYPEOF(sar)) AND (sar.name = 'replicate_positioning') AND
      (NOT (SIZEOF(QUERY ( pd <* USEDIN(sar.related_shape_aspect,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
      AND (it.name = 'base feature placement')) )) >= 1))) )) = 0))) ))
      = 0))) )) = 1);
END_ENTITY; -- feature_pattern

ENTITY fillet
  SUBTYPE OF (transition_feature);
  WHERE
    wr1: ((NOT (SELF\shape_aspect.description = 'constant radius')) XOR
      (NOT (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) )) = 1))) )) = 0)))));
    wr2: ((NOT (SELF\shape_aspect.description = 'constant radius')) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      impl_rep.used_representation.items = 3))) )) = 0))) )) = 0));
    wr3: ((NOT (SELF.description = 'constant radius')) OR (SIZEOF(

```



```

    QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
    ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'radius')))) = 1)) )) = 0)) ))
    = 0));
wr4: ((NOT (SELF.description = 'constant radius')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'first offset')) )) = 1)) ))
    = 0)) )) = 0));
wr5: ((NOT (SELF.description = 'constant radius')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) = 1)) ))
    = 0)) )) = 0));
wr6: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
    IN TYPEOF(pdr.used_representation)) AND (pdr.
    used_representation.name = 'fillet face')) )) <= 1)) )) = 0);
wr7: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
    IN TYPEOF(pdr.used_representation)) AND (pdr.

```

```

        used_representation.name = 'first face shape')) )) = 1)) ))
        = 0);
wr8: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'second face shape')) )) = 1)) ))
        = 0);
wr9: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATING_SHAPE_ASPECT') |
        ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF(sar)) ) | ('FEATURE_BASED_PROCESS_PLANNING.TAPER'
        IN TYPEOF(fcr.related_shape_aspect)) )) <= 1),
END_ENTITY; -- fillet

ENTITY flat_face
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'removal direction')) )) = 1)) ))
        = 0);
    wr2: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
        sa_occ.description = 'course of travel occurrence') AND (
        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'path feature component usage') AND ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
        'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
        TYPEOF(sdr.relying_shape_aspect)) AND (sdr.
        relating_shape_aspect.description = 'linear') AND (sdr.name
        = 'course of travel')) )) = 1)) )) = 1)) )) = 0);
    wr3: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
        sa_occ.description = 'removal boundary occurrence') AND (

```

```

        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
        'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE' IN TYPEOF(
        sdr.relatng_shape_aspect)) AND (sdr.name =
        'removal boundary')) )) = 1)) )) = 1)) )) = 0);
END_ENTITY; -- flat_face

ENTITY functionally_defined_transformation;
    name          : label;
    description    : text;
END_ENTITY; -- functionally_defined_transformation

ENTITY geometric_representation_context
    SUBTYPE OF (representation_context);
    coordinate_space_dimension : dimension_count;
END_ENTITY; -- geometric_representation_context

ENTITY geometric_representation_item
    SUPERTYPE OF (ONEOF (point,direction,vector,placement,
    cartesian_transformation_operator,curve,surface,edge_curve,
    face_surface,vertex_point,solid_model));
    SUBTYPE OF (representation_item);
    DERIVE
        dim : dimension_count := dimension_of(SELF);
    WHERE
        wr1: (SIZEOF(QUERY ( using_rep <* using_representations(SELF) | (
        NOT
        ('FEATURE_BASED_PROCESS_PLANNING.GEOMETRIC_REPRESENTATION_CONTEXT'
        IN TYPEOF(using_rep.context_of_items))) )) = 0);
END_ENTITY; -- geometric_representation_item

ENTITY geometric_tolerance;
    name          : label;
    description    : text;
    magnitude      : measure_with_unit;
    toleranced_shape_aspect : shape_aspect;
    WHERE
        wr1: (magnitude.value_component >= 0);
END_ENTITY; -- geometric_tolerance

ENTITY geometric_tolerance_relationship;
    name          : label;
    description    : text;
    relating_geometric_tolerance : geometric_tolerance;
    related_geometric_tolerance : geometric_tolerance;
END_ENTITY; -- geometric_tolerance_relationship

ENTITY geometric_tolerance_with_datum_reference
    SUBTYPE OF (geometric_tolerance);
    datum_system : SET [1:?] OF datum_reference;
END_ENTITY; -- geometric_tolerance_with_datum_reference

```

```

ENTITY geometric_tolerance_with_defined_unit
  SUBTYPE OF (geometric_tolerance);
  unit_size : measure_with_unit;
  WHERE
    wr1: (unit_size.value_component > 0);
END_ENTITY; -- geometric_tolerance_with_defined_unit

ENTITY global_uncertainty_assigned_context
  SUBTYPE OF (representation_context);
  uncertainty : SET [1:?] OF uncertainty_measure_with_unit;
END_ENTITY; -- global_uncertainty_assigned_context

ENTITY global_unit_assigned_context
  SUBTYPE OF (representation_context);
  units : SET [1:?] OF unit;
END_ENTITY; -- global_unit_assigned_context

ENTITY hole_bottom
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
    wr2 : (SELF.description IN ['through','flat','flat with radius',
      'spherical','conical']);
    wr3 : ((NOT (SELF.description = 'through')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.'
      + 'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(USEDIN(
      pd,'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 0)) ) )
      = 0));
    wr4 : ((NOT (SELF.description IN ['flat','flat with radius',
      'spherical','conical'])) OR (SIZEOF(QUERY ( pd <* USEDIN(
      SELF,'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.'
      + 'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) = 1)) ) ) = 0));
    wr5 : ((NOT (SELF.description = 'flat')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 1)) ) ) = 0)) ) ) = 0));
    wr6 : ((NOT (SELF.description IN ['flat with radius','spherical']))
      OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((

```

```

'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 2)) )) = 0)) )) = 0));
wr7 : ((NOT (SELF.description = 'conical')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 3)) )) = 0)) )) = 0));
wr8 : ((NOT (SELF.description IN ['flat', 'flat with radius',
'spherical', 'conical'])) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (SIZEOF(QUERY ( it <* impl_rep.
used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'blind bottom orientation')
AND (it.description IN ['hole start', 'hole end'])) )) = 1) ))
= 0)) )) = 0));
wr9 : ((NOT (SELF.description = 'flat with radius')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'corner radius')))) = 1)) ))
= 0)) )) = 0));
wr10: ((NOT (SELF.description = 'spherical')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']

```

```

        * TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) =
        0)) )) = 0));
wr11: ((NOT (SELF.description = 'conical')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'tip radius')) )) = 1)) ))
    = 0)) )) = 0));
wr12: ((NOT (SELF.description = 'conical')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'tip angle')) )) = 1)) ))
    = 0)) )) = 0));
wr13: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
    + 'RELATING_SHAPE_ASPECT') | ((sar.description =
    'hole bottom usage') AND
    ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
    IN TYPEOF(sar))) ) | ((fcr.related_shape_aspect.description
    = 'bottom condition occurrence') AND (
    'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(fcr.
    related_shape_aspect.of_shape.definition))) )) >= 1);
END_ENTITY; -- hole_bottom

ENTITY hyperbola
    SUBTYPE OF (conic);
    semi_axis : positive_length_measure;
    semi_imag_axis : positive_length_measure;
END_ENTITY; -- hyperbola

ENTITY instanced_feature
    SUBTYPE OF (feature_definition, shape_aspect);
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION' IN TYPEOF(
            SELF.of_shape.definition));
        wr2: SELF.product_definitional;
END_ENTITY; -- instanced_feature

```

```

ENTITY length_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.LENGTH_UNIT' IN TYPEOF(SELF\
      measure_with_unit.unit_component));
END_ENTITY; -- length_measure_with_unit

ENTITY length_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((SELF\named_unit.dimensions.length_exponent = 1) AND (SELF\
      named_unit.dimensions.mass_exponent = 0) AND (SELF\
      named_unit.dimensions.time_exponent = 0) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
      .dimensions.amount_of_substance_exponent = 0) AND (SELF\
      named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- length_unit

ENTITY 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 linear_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 2)) )) = 0)) )) = 0);
wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
        AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'profile length')) )) = 1)) ))
        = 0)) )) = 0);
END_ENTITY; -- linear_profile

ENTITY linear_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: (SELF\geometric_tolerance.name = 'linear profile');
    wr2: ((NOT (('FEATURE_BASED_PROCESS_PLANNING.' +
        'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF)))
        OR (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
        datum_system) <= 3));
END_ENTITY; -- linear_profile_tolerance

ENTITY location_shape_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: (SIZEOF(SELF.items) = 1);
    wr2: (SIZEOF(QUERY ( it <* SELF.items | (NOT (
        'FEATURE_BASED_PROCESS_PLANNING.POINT' IN TYPEOF(it))) )) = 0);
END_ENTITY; -- location_shape_representation

ENTITY loop
  SUPERTYPE OF (ONEOF (vertex_loop, edge_loop))
  SUBTYPE OF (topological_representation_item);
END_ENTITY; -- loop

ENTITY machining_feature
  SUBTYPE OF (instanced_feature);
  WHERE

```



```

wr1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) >= 1)) ))
    = 0);
wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) = 1)) )) = 0);
wr3: (SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.BOSS',
    'FEATURE_BASED_PROCESS_PLANNING.TURNED_KNURL',
    'FEATURE_BASED_PROCESS_PLANNING.THREAD',
    'FEATURE_BASED_PROCESS_PLANNING.MARKING',
    'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE',
    'FEATURE_BASED_PROCESS_PLANNING.OUTSIDE_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.POCKET',
    'FEATURE_BASED_PROCESS_PLANNING.REMOVAL_VOLUME',
    'FEATURE_BASED_PROCESS_PLANNING.REVOLVED_PROFILE',
    'FEATURE_BASED_PROCESS_PLANNING.UTER_ROUND',
    'FEATURE_BASED_PROCESS_PLANNING.FLAT_FACE',
    'FEATURE_BASED_PROCESS_PLANNING.PROTRUSION',
    'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_END',
    'FEATURE_BASED_PROCESS_PLANNING.SLOT',
    'FEATURE_BASED_PROCESS_PLANNING.SPHERICAL_CAP',
    'FEATURE_BASED_PROCESS_PLANNING.STEP',
    'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_MACHINING_FEATURE',
    'FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_FEATURE_DEFINITION']
    * TYPEOF(SELF)) = 1);
wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((
    'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
    AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
wr5: ((NOT (SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.TURNED_KNURL',
    'FEATURE_BASED_PROCESS_PLANNING.THREAD',
    'FEATURE_BASED_PROCESS_PLANNING.MARKING',
    'FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_FEATURE_DEFINITION']
    * TYPEOF(SELF)) = 1)) OR (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
    + 'RELATING_SHAPE_ASPECT') |
    ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
    IN TYPEOF(sar)) ) | (sdr.name = 'applied shape')) = 1));
wr6: number_of_representation_items(SELF);

```

```

END_ENTITY; -- machining_feature

ENTITY make_from_usage_option
  SUBTYPE OF (product_definition_usage);
  ranking          : INTEGER;
  ranking_rationale : text;
  quantity         : measure_with_unit;
  WHERE
    wr1: (ranking > 0);
END_ENTITY; -- make_from_usage_option

ENTITY manifold_solid_brep
  SUBTYPE OF (solid_model);
  outer : closed_shell;
END_ENTITY; -- manifold_solid_brep

ENTITY mapped_item
  SUBTYPE OF (representation_item);
  mapping_source : representation_map;
  mapping_target : representation_item;
  WHERE
    wr1: acyclic_mapped_representation(using_representations(SELF),[SELF]);
END_ENTITY; -- mapped_item

ENTITY marking
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
      QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items |
      (('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'marking text')) )) = 1)) )) =
      0))) ) = 0);
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
      QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items |
      (('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'special instructions')) )) <=
      1)) )) = 0)) )) = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
      QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'font name')))) <= 1)))) = 0)) ))
        = 0);
wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
        QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'character height')) )) <=
        1)) )) = 0)) )) = 0);
wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
        QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'character spacing')) )) <=
        1)) )) = 0)) )) = 0);
END_ENTITY; -- marking

ENTITY material_designation;
    name      : label;
    definitions : SET [1:?] OF characterized_definition;
END_ENTITY; -- material_designation

ENTITY material_property
    SUBTYPE OF (property_definition);
    UNIQUE
        ur1 : name, definition;
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.CHARACTERIZED_OBJECT' IN
        TYPEOF(SELF\property_definition.definition)) OR (SIZEOF(
        bag_to_set(USEDIN(SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) -
        QUERY ( temp <* bag_to_set(USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'MATERIAL_PROPERTY_REPRESENTATION') IN TYPEOF(temp)) )) =

```

```

0);
END_ENTITY; -- material_property

ENTITY material_property_representation
  SUBTYPE OF (property_definition_representation);
  dependent_environment : data_environment;
END_ENTITY;

ENTITY measure_qualification;
  name : label;
  description : text;
  qualified_measure : measure_with_unit;
  qualifiers : SET [1:?] OF value_qualifier;
  WHERE
    wr1: (SIZEOF(QUERY ( temp <* qualifiers | (
      'FEATURE_BASED_PROCESS_PLANNING.PRECISION_QUALIFIER' IN
      TYPEOF(temp)) )) < 2);
END_ENTITY; -- measure_qualification

ENTITY measure_representation_item
  SUBTYPE OF (representation_item, measure_with_unit);
END_ENTITY; -- measure_representation_item

ENTITY measure_with_unit
  SUPERTYPE OF (ONEOF (length_measure_with_unit,
    plane_angle_measure_with_unit, solid_angle_measure_with_unit));
  value_component : measure_value;
  unit_component : unit;
  WHERE
    wr1: valid_units(SELF);
END_ENTITY; -- measure_with_unit

ENTITY modified_geometric_tolerance
  SUBTYPE OF (geometric_tolerance);
  modifier : limit_condition;
END_ENTITY; -- modified_geometric_tolerance

ENTITY modified_pattern
  SUBTYPE OF (replicate_feature);
  WHERE
    wr1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATING_SHAPE_ASPECT') | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
        'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
        TYPEOF(sar.related_shape_aspect)) = 1) AND (sar.
        related_shape_aspect.description = 'base pattern')) )) = 1);
    wr2: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATING_SHAPE_ASPECT') | (NOT (SIZEOF(QUERY ( msar <*
        USEDIN(sar.related_shape_aspect,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.PATTERN_OFFSET_MEMBERSHIP',
        'FEATURE_BASED_PROCESS_PLANNING.PATTERN_OMIT_MEMBERSHIP'] *

```

```

        TYPEOF(sar)) = 1) AND (sar :<>: msar)) )) >= 1)) )) = 0);
END_ENTITY; -- modified_pattern

ENTITY named_unit
    SUPERTYPE OF (ONEOF (si_unit, conversion_based_unit, context_dependent_unit)
    ANDOR ONEOF (length_unit, plane_angle_unit, solid_angle_unit));
    dimensions : dimensional_exponents;
END_ENTITY; -- named_unit

ENTITY ngon_closed_profile
    SUBTYPE OF (shape_aspect);
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
            IN TYPEOF(SELF.of_shape.definition));
        wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
            'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) = 1)) )) = 0);
        wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 4)) )) = 0)) )) = 0);
        wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((
            'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
            AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
        wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
            (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (
            'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
            measure_with_unit.value_component)) AND (it.name =

```

```

        'number of sides')) )) = 1)) )) = 0)) )) = 0);
wr6: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'circumscribed diameter')) ))
    = 1)) )) = 0)) )) = 0);
wr7: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) = 1)) ))
    = 0)) )) = 0);
END_ENTITY; -- ngon_closed_profile

ENTITY ngon_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
    wr1: (SIZEOF(SELF.items) = 5);
    wr2: (SIZEOF(QUERY ( it <* SELF.items | ((
        'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
        AND (it.name = 'orientation')) )) = 1);
    wr3: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1);
    wr4: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) = 1);
    wr5: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'circumscribed diameter')) ))
        = 1);
    wr6: (SIZEOF(QUERY ( it <* SELF.items |
        (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (
        'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
        measure_with_unit.value_component)) AND (it.name =
        'number of sides')) )) = 1);

```

END_ENTITY; -- ngon_shape_representation

ENTITY open_path_profile

SUBTYPE OF (shape_aspect);

WHERE

wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
IN TYPEOF(SELF.of_shape.definition));

wr2: (SIZEOF(QUERY (pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY (pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)))) = 1)))) = 0);

wr3: (SIZEOF(QUERY (pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY (impl_rep <* QUERY (pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)))) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 1)))) = 0)))) = 0);

wr4: (SIZEOF(QUERY (pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY (impl_rep <* QUERY (pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)))) | (NOT (SIZEOF(QUERY (it <*
impl_rep.used_representation.items | ((
'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
AND (it.name = 'orientation')))) = 1)))) = 0)))) = 0);

wr5: (SIZEOF(QUERY (pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY (pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
'FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'
IN TYPEOF(pdr.used_representation)))) = 1)))) = 0);

END_ENTITY; -- open_path_profile

ENTITY open_shell

SUBTYPE OF (connected_face_set);

END_ENTITY; -- open_shell

ENTITY ordered_part

SUBTYPE OF (action_assignment, characterized_object);

items : SET [1:?] OF feature_based_pp_ordered_item;

WHERE

wr1: (SIZEOF(USEDIN(SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION.DEFINITION')) = 1);

wr2: (SIZEOF(QUERY (pd <* USEDIN(SELF,

```

        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) )) =
        0);
wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
        SIZEOF(items) = 1)) )) = 0)) )) = 0);
wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'quantity')) )) = 1)) )) = 0)) ))
        = 0);
END_ENTITY; -- ordered_part

ENTITY ordinal_date
  SUBTYPE OF (date);
  day_component : day_in_year_number;
  WHERE
    wr1: (((NOT leap_year(SELF.year_component)) AND (1 <= day_component)
        AND (day_component <= 365)) OR (leap_year(SELF.
        year_component) AND (1 <= day_component) AND (day_component
        <= 366)));
END_ENTITY; -- ordinal_date

ENTITY organization;
  id : OPTIONAL identifier;
  name : label;
  description : text;
END_ENTITY; -- organization

ENTITY organization_assignment
  ABSTRACT SUPERTYPE;
  assigned_organization : organization;
  role : organization_role;
END_ENTITY; -- organization_assignment

ENTITY organization_relationship;
  name : label;
  description : text;
  relating_organization : organization;
  related_organization : organization;
END_ENTITY; -- organization_relationship

ENTITY organization_role;
  name : label;
END_ENTITY; -- organization_role

```



```

ENTITY organizational_address
  SUBTYPE OF (address);
  organizations : SET [1:?] OF organization;
  description   : text;
END_ENTITY; -- organizational_address

ENTITY organizational_project;
  name           : label;
  description     : text;
  responsible_organizations : SET [1:?] OF organization;
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 ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_CLOSED_SHELL' IN
      TYPEOF(SELF.closed_shell_element)));
END_ENTITY; -- oriented_closed_shell

ENTITY oriented_edge
  SUBTYPE OF (edge);
  edge_element : edge;
  orientation   : BOOLEAN;
  DERIVE
    SELF\edge.edge_start : vertex := boolean_choose(SELF.orientation,
      SELF.edge_element.edge_start, SELF.
      edge_element.edge_end);
    SELF\edge.edge_end   : vertex := boolean_choose(SELF.orientation,
      SELF.edge_element.edge_end, SELF.
      edge_element.edge_start);
  WHERE
    wr1: (NOT ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_EDGE' IN TYPEOF(
      SELF.edge_element)));
END_ENTITY; -- oriented_edge

ENTITY oriented_face
  SUBTYPE OF (face);
  face_element : face;
  orientation   : BOOLEAN;
  DERIVE
    SELF\face.boundaries : SET [1:?] OF face_bound := conditional_reverse(
      SELF.orientation, SELF.face_element.boundaries);
  WHERE
    wr1: (NOT ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_FACE' IN TYPEOF(
      SELF.face_element)));
END_ENTITY; -- oriented_face

ENTITY oriented_open_shell
  SUBTYPE OF (open_shell);

```

```

    open_shell_element : open_shell;
    orientation         : BOOLEAN;
DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
        conditional_reverse(SELF.orientation,
        SELF.open_shell_element.cfs_faces);
WHERE
    wr1: (NOT ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_OPEN_SHELL' IN
        TYPEOF(SELF.open_shell_element)));
END_ENTITY; -- oriented_open_shell

ENTITY oriented_path
    SUBTYPE OF (path);
    path_element : path;
    orientation   : BOOLEAN;
DERIVE
    SELF\path.edge_list : LIST [1:?] OF UNIQUE oriented_edge :=
        conditional_reverse(SELF.orientation, SELF.
        path_element.edge_list);
WHERE
    wr1: (NOT ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_PATH' IN TYPEOF(
        SELF.path_element)));
END_ENTITY; -- oriented_path

ENTITY outer_round
    SUBTYPE OF (feature_definition);
WHERE
    wr1: (SELF.description IN ['outer diameter',
        'outer diameter to shoulder']);
    wr2: ((NOT (SELF.description = 'outer diameter')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
        QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1)) )) = 0)) ))
        = 0));
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
        QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items |
        ((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1)) )) =

```

```

0)) ) = 0);
wr4: ((NOT (SELF.description = 'outer diameter to shoulder')) OR (
  SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (
      'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
      sa_occ.description = 'v-shape boundary occurrence') AND (
      SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
      'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE' IN TYPEOF(sdr.
      relating_shape_aspect)) AND (sdr.relatng_shape_aspect.
      description = 'v-shape')) ) = 1)) ) = 1)) ) = 0));
wr5: ((NOT (SELF.description = 'outer diameter')) OR (SIZEOF(
  QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (
      'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
      sa_occ.description = 'reduced size occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
      + 'RELATED_SHAPE_ASPECT') | ((sar.description =
      'taper usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
      'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
      'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(sdr.
      relating_shape_aspect)) AND (sdr.name = 'reduced size')) ) )
      = 1)) ) <= 1)) ) = 0));
END_ENTITY; -- outer_round

ENTITY outside_profile
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
        sa_occ.description = 'boundary occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
        SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
          'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
          'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
          'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',

```

```

        'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
    TYPEOF(sdr.relatng_shape_aspect)) = 1) AND (sdr.
    relating_shape_aspect.description = 'outside boundary')) ))
    = 1)) )) = 1)) )) = 0);
END_ENTITY; -- outside_profile

ENTITY parabola
    SUBTYPE OF (conic);
    focal_dist : length_measure;
    WHERE
        wr1: (focal_dist <> 0);
END_ENTITY; -- parabola

ENTITY parallelism_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
        wr1: (SELF\geometric_tolerance.name = 'parallelism');
        wr2: (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
            datum_system) = 1);
END_ENTITY; -- parallelism_tolerance

ENTITY parametric_representation_context
    SUBTYPE OF (representation_context);
END_ENTITY; -- parametric_representation_context

ENTITY partial_circular_profile
    SUBTYPE OF (shape_aspect);
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
            IN TYPEOF(SELF.of_shape.definition));
        wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
            'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) = 1)) )) = 0);
        wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) | (NOT (SIZEOF(impl_rep.
            used_representation.items) = 3)) )) = 0)) )) = 0);
        wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')

```

```

    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'radius')))) = 1)) )) = 0)) ))
= 0);
wr6: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'sweep angle')) )) = 1)) ))
= 0)) )) = 0);
END_ENTITY; -- partial_circular_profile

ENTITY path
  SUPERTYPE OF (ONEOF (edge_loop,oriented_path))
  SUBTYPE OF (topological_representation_item);
  edge_list : LIST [1:?] OF UNIQUE oriented_edge;
  WHERE
    wr1: path_head_to_tail(SELF);
END_ENTITY; -- path

ENTITY path_feature_component
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
    wr2 : (SELF.description IN ['partial circular','complete circular',
      'linear','complex']);
    wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,

```

```

        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0);
wr4 : ((NOT (SELF.description = 'partial circular')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 3)) )) = 0)) )) = 0));
wr5 : ((NOT (SELF.description = 'partial circular')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) =
        0)) )) = 0));
wr6 : ((NOT (SELF.description = 'partial circular')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'sweep angle')) )) = 1)) ))
        = 0)) )) = 0));
wr7 : ((NOT (SELF.description = 'complete circular')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 2)) )) = 0)) )) = 0));
wr8 : ((NOT (SELF.description = 'complete circular')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,

```

```

'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1))) =
0))) = 0));
wr9 : ((NOT (SELF.description = 'linear')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 2))) = 0))) = 0));
wr10: ((NOT (SELF.description = 'linear')) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'distance')) )) = 1))) =
0))) = 0));
wr11: ((NOT (SELF.description = 'linear')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
IN TYPEOF(pdr.used_representation)) )) = 1))) = 0));
wr12: ((NOT (SELF.description = 'complex')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 1))) = 0))) = 0));
wr13: ((NOT (SELF.description = 'complex')) OR (SIZEOF(

```

```

        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'sweep path')) )) = 1)) )) = 0));
wr14: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
        QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
        + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY (it <*
        impl_rep.used_representation.items | ((
        'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
        AND (it.name = 'orientation')) )) = 1))) )) = 0)) )) = 0);
END_ENTITY; -- path_feature_component

ENTITY path_shape_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: (SIZEOF(SELF.items) = 1);
    wr2: (SIZEOF(QUERY ( it <* SELF.items | (NOT (
        'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(it))) ))
        = 0);
    wr3: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF(QUERY ( el <*
        it\path.edge_list | (NOT (
        'FEATURE_BASED_PROCESS_PLANNING.EDGE_CURVE' IN
        TYPEOF(el.edge_element))) )) = 0)) )) = 0);
END_ENTITY; -- path_shape_representation

ENTITY pattern_offset_membership
  SUBTYPE OF (feature_component_relationship);
  WHERE
    wr1 : (SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
        'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
        TYPEOF(SELF.relate_shape_aspect)) = 1);
    wr2 : (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
        relate_shape_aspect,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') |
        (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF(sar)) AND (sar :<=: SELF)) ) | (
        'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN
        TYPEOF(fcr.relate_shape_aspect)) )) >= 1);
    wr3 : (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
        relate_shape_aspect,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') |
        (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF(sar)) AND (sar :<=: SELF)) ) | ((
        'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN

```



```

    TYPEOF(fcr.relater_shape_aspect)) AND (NOT (SIZEOF(
    QUERY ( modfcr <* QUERY ( modsar <* USEDIN(fcr.
    relater_shape_aspect, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATER_SHAPE_ASPECT') | ((
    SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
    'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
    TYPEOF(modsar.relater_shape_aspect)) = 1) AND (modsar <=:
    fcr)) ) | (NOT (modfcr.relater_shape_aspect :=: SELF.
    relater_shape_aspect)) )) = 0))) )) = 0);
wr4 : (SIZEOF(QUERY ( pd <* USEDIN(SELF.relater_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
    + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1))) ))
    = 0);
wr5 : ((NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
    TYPEOF(SELF.relater_shape_aspect))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.relater_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
    SIZEOF(pdr.used_representation.items) = 2))) )) = 0))) )) = 0));
wr6 : ((NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
    TYPEOF(SELF.relater_shape_aspect))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.relater_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items |
    (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (
    'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
    \measure_with_unit.value_component)) AND (it.name =
    'index number')) )) = 1))) )) = 0))) )) = 0));
wr7 : ((NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
    TYPEOF(SELF.relater_shape_aspect))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.relater_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items |
    (('FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'
    IN TYPEOF(it)) AND (it.name = 'offset')) )) = 1))) )) = 0))) ))
    = 0));
wr8 : ((NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
    TYPEOF(SELF.relater_shape_aspect))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.relater_shape_aspect,

```

```

'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
SIZEOF(pdr.used_representation.items) = 3)) )) = 0)) )) = 0));
wr9 : ((NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF(SELF.relateing_shape_aspect))) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (
'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
\measure_with_unit.value_component)) AND (it.name =
'row index')) )) = 1)) )) = 0)) )) = 0));
wr10: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF(SELF.relateing_shape_aspect))) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (
'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
\measure_with_unit.value_component)) AND (it.name =
'column index')) )) = 1)) )) = 0)) )) = 0));
wr11: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF(SELF.relateing_shape_aspect))) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'
IN TYPEOF(it)) AND (it.name = 'offset distance')) )) = 1)) ))
= 0)) )) = 0));
wr12: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF(SELF.relateing_shape_aspect))) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +

```

```

        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'offset direction')) )) = 1)) ))
        = 0));
END_ENTITY; -- pattern_offset_membership

ENTITY pattern_omit_membership
  SUBTYPE OF (feature_component_relationship);
  WHERE
    wr1: (SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
    'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
    TYPEOF(SELF.relate_shape_aspect)) = 1);
    wr2: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
    relate_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
    + 'RELATED_SHAPE_ASPECT') |
    (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
    IN TYPEOF(sar)) AND (sar :<=: SELF)) )) | (
    'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN TYPEOF(
    fcr.relate_shape_aspect)) )) >= 1);
    wr3: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
    relate_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
    + 'RELATED_SHAPE_ASPECT') |
    (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
    IN TYPEOF(sar)) AND (sar :<=: SELF)) )) | ((
    'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN TYPEOF(
    fcr.relate_shape_aspect)) AND (NOT (SIZEOF(
    QUERY ( modfcr <* QUERY ( modsar <* USEDIN(fcr.
    relate_shape_aspect, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
    SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
    'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
    TYPEOF(modsar.relate_shape_aspect)) = 1) AND (modsar :<=:
    fcr)) )) | (NOT (modfcr.relate_shape_aspect := SELF.
    relate_shape_aspect)) )) = 0))) )) = 0);
    wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF.relate_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
    + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
    = 0);
    wr5: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
    TYPEOF(SELF.relate_shape_aspect))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.relate_shape_aspect,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
    SIZEOF(pdr.used_representation.items) = 1)) )) = 0)) )) = 0));
    wr6: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
    TYPEOF(SELF.relate_shape_aspect))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.relate_shape_aspect,

```

```

'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (
'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component)) AND (it.name =
'index number')) )) = 1)) )) = 0)) )) = 0));
wr7: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF(SELF.relate_shape_aspect))) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0));
wr8: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF(SELF.relate_shape_aspect))) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (
'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component)) AND (it.name =
'row index')) )) = 1)) )) = 0)) )) = 0));
wr9: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
TYPEOF(SELF.relate_shape_aspect))) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (
'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component)) AND (it.name =
'column index')) )) = 1)) )) = 0)) )) = 0));
END_ENTITY; -- pattern_omit_membership

```

```

ENTITY pcurve
  SUBTYPE OF (curve);
  basis_surface      : surface;
  reference_to_curve : definitional_representation;
  WHERE
    wr1: (SIZEOF(reference_to_curve\representation.items) = 1);
    wr2: ('FEATURE_BASED_PROCESS_PLANNING.CURVE' IN TYPEOF(
      reference_to_curve\representation.items[1]));
    wr3: (reference_to_curve\representation.items[1]\
      geometric_representation_item.dim = 2);
END_ENTITY; -- pcurve

ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: (SELF\geometric_tolerance.name = 'perpendicularity');
    wr2: (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
      datum_system) = 1);
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;
  UNIQUE
    ur1 : id;
  WHERE
    wr1: (EXISTS(last_name) OR EXISTS(first_name));
END_ENTITY; -- person

ENTITY person_and_organization;
  the_person      : person;
  the_organization : organization;
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;
END_ENTITY; -- person_and_organization_role

ENTITY personal_address
  SUBTYPE OF (address);
  people      : SET [1:?] OF person;
  description : text;
END_ENTITY; -- personal_address

ENTITY placed_datum_target_feature

```

```

SUBTYPE OF (datum_target);
WHERE
  wr1 : (SELF.description IN ['point','line','rectangle','circle']);
  wr2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) = 1)) )) = 0));
  wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( dtm_rep <*
    QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
    + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    dtm_rep.used_representation.items | ((
    'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
    AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0));
  wr4 : ((NOT (SELF.description = 'point')) OR (SIZEOF(QUERY ( pd <*
    USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( dtm_rep <*
    QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
    + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(dtm_rep.
    used_representation.items) = 1)) )) = 0)) )) = 0));
  wr5 : ((NOT (SELF.description IN ['line','circle'])) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( dtm_rep <*
    QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
    + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(dtm_rep.
    used_representation.items) = 2)) )) = 0)) )) = 0));
  wr6 : ((NOT (SELF.description = 'rectangle')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( dtm_rep <*
    QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
    + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(dtm_rep.
    used_representation.items) = 3)) )) = 0)) )) = 0));
  wr7 : ((NOT (SELF.description = 'circle')) OR
    (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')

```

```

| (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'target diameter')))) =
1)) )) = 0)) )) = 0));
wr8 : ((NOT (SELF.description = 'line')) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
= 0)) )) = 0));
wr9 : ((NOT (SELF.description = 'rectangle')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
= 0)) )) = 0));
wr10: ((NOT (SELF.description = 'rectangle')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'target width')))) = 1)) )) = 0)) )) = 0));
END_ENTITY; -- placed_datum_target_feature

```

ENTITY placement

```

    SUPERTYPE OF (ONEOF (axis1_placement,axis2_placement_2d,
        axis2_placement_3d))
    SUBTYPE OF (geometric_representation_item);
    location : cartesian_point;
END_ENTITY; -- placement

ENTITY plane
    SUBTYPE OF (elementary_surface);
END_ENTITY; -- plane

ENTITY plane_angle_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_UNIT' IN TYPEOF(
            SELF\measure_with_unit.unit_component));
END_ENTITY; -- plane_angle_measure_with_unit

ENTITY plane_angle_unit
    SUBTYPE OF (named_unit);
    WHERE
        wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
            named_unit.dimensions.mass_exponent = 0) AND (SELF\
            named_unit.dimensions.time_exponent = 0) AND (SELF\
            named_unit.dimensions.electric_current_exponent = 0) AND (
            SELF\named_unit.dimensions.
            thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
            .dimensions.amount_of_substance_exponent = 0) AND (SELF\
            named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- plane_angle_unit

ENTITY plus_minus_tolerance;
    range : tolerance_method_definition;
    toleranced_dimension : dimensional_characteristic;
    UNIQUE
        ur1 : toleranced_dimension;
END_ENTITY; -- plus_minus_tolerance

ENTITY pocket
    SUBTYPE OF (feature_definition);
    WHERE
        wr1: (SELF.description IN ['closed rectangular','open rectangular',
            'non-rectangular']);
        wr2: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (
            'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
            sa_occ.description = 'pocket depth occurrence') AND (SIZEOF(
            QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'path feature component usage') AND ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) ) | ((

```



```

'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
  TYPEOF(sdr.relater_shape_aspect)) AND (sdr.name =
    'pocket depth') AND (sdr.relater_shape_aspect.description =
      'linear')) )) = 1)) )) = 1)) )) = 0);
wr3: ((NOT (SELF.description = 'non-rectangular')) OR (SIZEOF(
  QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (
      'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
        sa_occ.description = 'boundary occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
        SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
        TYPEOF(sdr.relater_shape_aspect)) = 1) AND (sdr.
        relater_shape_aspect.description = 'boundary')) )) = 1)) ))
        = 1)) )) = 0));
wr4: ((NOT (SELF.description = 'closed rectangular')) OR (SIZEOF(
  QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (
      'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
        sa_occ.description = 'closed boundary occurrence') AND (
        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
        'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE'
        IN TYPEOF(sdr.relater_shape_aspect)) AND (sdr.
        relater_shape_aspect.description = 'boundary')) )) = 1)) ))
        = 1)) )) = 0));
wr5: ((NOT (SELF.description = 'open rectangular')) OR (SIZEOF(
  QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (
      'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((

```

```

sa_occ.description = 'open boundary occurrence') AND (
  SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
  + 'RELATED_SHAPE_ASPECT') | ((sar.description =
    'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
    'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE' IN TYPEOF(
    sdr.relatng_shape_aspect)) AND (sdr.relatng_shape_aspect.
    description = 'boundary')) )) = 1)) )) = 1)) )) = 0));
wr6: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELf,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
  | (
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
    sa_occ.description = 'bottom condition occurrence') AND (
    SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
  + 'RELATED_SHAPE_ASPECT') | ((sar.description =
    'pocket bottom usage') AND (
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) ) | (
    'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM' IN TYPEOF(sdr
    .relatng_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
wr7: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELf,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
  | (
    'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
    sa_occ.description = 'change in boundary occurrence') AND (
    SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
  + 'RELATING_SHAPE_ASPECT') | ((sar.description =
    'taper usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
    'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
    'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
    relatng_shape_aspect)) AND (fcr.relatng_shape_aspect.
    description = 'angle taper')) )) = 1)) )) <= 1)) )) = 0);
END_ENTITY; -- pocket

ENTITY pocket_bottom
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELf.of_shape.definition));
    wr2: (SELf.description IN ['planar', 'non-planar', 'through']);
    wr3: ((NOT (SELf.description = 'non-planar')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 1));

```

```

wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
    'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) = 1)) )) = 0);
wr5: ((NOT (SELF.description IN ['planar','non-planar'])) OR (
    SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) | (NOT (SIZEOF(impl_rep.
    used_representation.items) = 2)) )) = 0)) )) = 0));
wr6: ((NOT (SELF.description = 'through')) OR
    (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) | (NOT (SIZEOF(impl_rep.
    used_representation.items) = 1)) )) = 0)) )) = 0));
wr7: ((NOT (SELF.description IN ['planar','non-planar'])) OR (
    SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
    = 0));
wr8: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
    + 'RELATING_SHAPE_ASPECT') | ((sar.description =
    'pocket bottom usage') AND
    ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
    IN TYPEOF(sar))) )) | ((fcr.related_shape_aspect.description
    = 'bottom condition occurrence') AND (
    'FEATURE_BASED_PROCESS_PLANNING.POCKET' IN TYPEOF(fcr.
    related_shape_aspect.of_shape.definition))) )) >= 1);
wr9: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it < *
        impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'pocket bottom orientation')
        AND (it.description IN ['pocket start','pocket end']))) )) =
        1)) )) = 0))) = 0);
END_ENTITY; -- pocket_bottom

ENTITY point
    SUPERTYPE OF (cartesian_point)
    SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- point

ENTITY polyline
    SUBTYPE OF (bounded_curve);
    points : LIST [2:?] OF cartesian_point;
END_ENTITY; -- polyline

ENTITY position_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
        wr1: (SELF\geometric_tolerance.name = 'position');
        wr2: (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 : text;
    frame_of_reference : SET [1:?] OF product_context;
    UNIQUE
    ur1 : id;
END_ENTITY; -- product

ENTITY product_context
    SUBTYPE OF (application_context_element);
    discipline_type : label;
END_ENTITY; -- product_context

ENTITY product_definition;
    id : identifier;
    description : text;
    formation : product_definition_formation;
    frame_of_reference : product_definition_context;
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 : text;
    of_product : product;
    UNIQUE
    ur1 : id, of_product;
END_ENTITY; -- product_definition_formation

ENTITY product_definition_relationship;
    id : identifier;
    name : label;
    description : text;
    relating_product_definition : product_definition;
    related_product_definition : product_definition;
END_ENTITY; -- product_definition_relationship

ENTITY product_definition_shape
    SUBTYPE OF (property_definition);
    UNIQUE
    ur1 : definition;
    WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.CHARACTERIZED_PRODUCT_DEFINITION'
        IN TYPEOF(SELF\property_definition.definition));
END_ENTITY; -- product_definition_shape

ENTITY product_definition_usage
    SUPERTYPE OF (make_from_usage_option)
    SUBTYPE OF (product_definition_relationship);
    UNIQUE
    ur1 : id, relating_product_definition, related_product_definition;
    WHERE
    wr1: acyclic_product_definition_relationship(SELF,[SELF\
        product_definition_relationship.related_product_definition],
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_USAGE.' +
        'RELATED_PRODUCT_DEFINITION');
END_ENTITY; -- product_definition_usage

ENTITY product_definition_with_associated_documents
    SUBTYPE OF (product_definition);
    documentation_ids : SET [1:?] OF document;
END_ENTITY; -- product_definition_with_associated_documents

ENTITY 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    : text;
        definition      : characterized_definition;
    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      : property_definition;
        used_representation : representation;
    END_ENTITY; -- property_definition_representation

    ENTITY protrusion
        SUBTYPE OF (feature_definition);
        WHERE
            wr1: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
                'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
                sa_occ.description = 'shape volume occurrence') AND (SIZEOF(
                QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'volume shape usage') AND ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (sdr.
                relating_shape_aspect.description = 'volume shape') )) = 1)) ))
                = 1)) )) = 0);
    END_ENTITY; -- protrusion

    ENTITY qualified_representation_item
        SUBTYPE OF (representation_item);
        qualifiers : SET [1:?] OF value_qualifier;
        WHERE
            wr1: (SIZEOF(QUERY ( temp <* qualifiers | (
                'FEATURE_BASED_PROCESS_PLANNING.PRECISION_QUALIFIER' IN
                TYPEOF(temp)) )) < 2);
    END_ENTITY; -- qualified_representation_item

    ENTITY qualitative_uncertainty
        SUBTYPE OF (uncertainty_qualifier);
        uncertainty_value : text;
    END_ENTITY; -- qualitative_uncertainty

    ENTITY quasi_uniform_curve
        SUBTYPE OF (b_spline_curve);
    END_ENTITY; -- quasi_uniform_curve

```

```

ENTITY quasi_uniform_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- quasi_uniform_surface

ENTITY ratio_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'FEATURE_BASED_PROCESS_PLANNING.RATIO_UNIT' IN TYPEOF
    (SELF\measure_with_unit.unit_component);
END_ENTITY;

ENTITY ratio_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
    (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
    (SELF\named_unit.dimensions.time_exponent = 0.0) AND
    (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
    (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0)
    AND (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0)
    AND (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;

ENTITY rational_b_spline_curve
  SUBTYPE OF (b_spline_curve);
  weights_data : LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:upper_index_on_control_points] OF REAL :=
      list_to_array(weights_data, 0,
        upper_index_on_control_points);
  WHERE
    wr1: (SIZEOF(weights_data) = SIZEOF(SELF\b_spline_curve.
      control_points_list));
    wr2: curve_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_curve

ENTITY rational_b_spline_surface
  SUBTYPE OF (b_spline_surface);
  weights_data : LIST [2:?] OF LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF REAL :=
      make_array_of_array(weights_data, 0, u_upper, 0, v_upper);
  WHERE
    wr1: ((SIZEOF(weights_data) = SIZEOF(SELF\b_spline_surface.
      control_points_list)) AND (SIZEOF(weights_data[1]) = SIZEOF(
      SELF\b_spline_surface.control_points_list[1])));
    wr2: surface_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_surface

ENTITY rectangular_closed_profile
  SUBTYPE OF (shape_aspect);
WHERE
  wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
    IN TYPEOF(SELF.of_shape.definition));
  wr2: (SIZEOF(QUERY (pd <* USEDIN(SELF,

```

```

'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) = 1)) )) = 0);
wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 4)) )) = 0)) )) = 0);
wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
= 0);
wr6: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'length')))) = 1)) )) = 0)) ))
= 0);
wr7: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')

```



```

    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'])*
  TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) = 1)) ))
= 0)) )) = 0);
END_ENTITY; -- rectangular_closed_profile

ENTITY rectangular_pattern
  SUBTYPE OF (replicate_feature);
  WHERE
    wr1 : (SIZEOF(USEDIN(SELF, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')) <= 5);
    wr2 : (SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (NOT
(SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP',
      'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP']
      * TYPEOF(sar)) = 1)) )) = 0);
    wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'row layout direction')))) = 1)) ))
      = 0);
    wr4 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'column layout direction')) )) =
      1)) )) = 0);
    wr5 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
    wr6 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |

```

```

('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
 impl_rep.used_representation.items) = 5)) )) = 0)) )) = 0);
wr7 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
 pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
 QUERY ( it <* impl_rep.used_representation.items |
 (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
 IN TYPEOF(it)) AND (
 'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
 \measure_with_unit.value_component)) AND (it.name =
 'number of rows')) )) = 1)) )) = 0)) )) = 0);
wr8 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
 pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
 QUERY ( it <* impl_rep.used_representation.items |
 (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
 IN TYPEOF(it)) AND (
 'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
 \measure_with_unit.value_component)) AND (it.name =
 'number of columns')) )) = 1)) )) = 0)) )) = 0);
wr9 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
 pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
 QUERY ( it <* impl_rep.used_representation.items | ((
 SIZEOF([
 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
 'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
 * TYPEOF(it)) = 2) AND (it.name = 'row spacing')) )) = 1)) ))
 = 0)) )) = 0);
wr10: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
 pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
 IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
 QUERY ( it <* impl_rep.used_representation.items | ((
 SIZEOF([
 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
 'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
 * TYPEOF(it)) = 2) AND (it.name = 'column spacing')))) = 1)) ))
 = 0)) )) = 0);

```

```

wr11: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
    | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((
    'FEATURE_BASED_PROCESS_PLANNING.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 ( pds <* QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_OF_SHAPE') | ((
                sa_occ.description = 'shape volume occurrence') AND (SIZEOF(
                QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'volume shape usage') AND ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (sdr.
                relating_shape_aspect.description = 'volume shape') )) = 1)) ))
                = 1)) )) = 0);
END_ENTITY; -- removal_volume

ENTITY replicate_feature
    SUPERTYPE OF (ONEOF (circular_pattern, rectangular_pattern,
        modified_pattern, feature_pattern))
    SUBTYPE OF (shape_aspect);
    WHERE
        wr1: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATING_SHAPE_ASPECT') |
            ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar)) ) | ((SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
            'FEATURE_BASED_PROCESS_PLANNING.MACHINING_FEATURE'] *
            TYPEOF(fcr.related_shape_aspect)) = 1) AND (fcr.name =
            'pattern basis')) )) = 1);
END_ENTITY; -- replicate_feature

ENTITY representation;
    name : label;

```

```

        items          : SET [1:?] OF representation_item;
        context_of_items : representation_context;
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 revolved_profile
    SUBTYPE OF (feature_definition);
WHERE
    wr1: (SELF.description IN ['groove', 'flat', 'round', 'open profile']);
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'radius')))) = 1)) )) = 0)) ))
        = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'removal direction')) )) = 1)) ))
        = 0);
    wr4: ((NOT (SELF.description = 'open profile')) OR (SIZEOF(

```

```

QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'outer edge shape occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE' IN
  TYPEOF(sdr.relate_shape_aspect)) AND (sdr.
relate_shape_aspect.description = 'outer edge shape')) ))
= 1)) )) = 1)) )) = 0));
wr5: ((NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pds <*
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'flat edge shape occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE' IN TYPEOF(
sdr.relate_shape_aspect)) AND (sdr.relate_shape_aspect.
description = 'flat edge shape')) )) = 1)) )) = 1)) )) = 0));
wr6: ((NOT (SELF.description = 'round')) OR (SIZEOF(QUERY ( pds <*
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'rounded edge shape occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE' IN
  TYPEOF(sdr.relate_shape_aspect)) AND (sdr.
relate_shape_aspect.description = 'rounded edge shape')) ))
= 1)) )) = 1)) )) = 0));
wr7: ((NOT (SELF.description = 'groove')) OR (SIZEOF(QUERY ( pds <*
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,

```

```

'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'sweep occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'sweep usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE']) *
TYPEOF(sdr.relatng_shape_aspect)) = 1) AND (sdr
relatng_shape_aspect.description = 'sweep')) )) = 1)) )) =
1)) )) = 0));
END_ENTITY; -- revolved_profile

ENTITY round_hole
SUBTYPE OF (feature_definition);
WHERE
  wr1: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'diameter occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE' IN
TYPEOF(sdr.relatng_shape_aspect)) AND (sdr.name =
'diameter')) )) = 1)) )) = 1)) )) = 0);
  wr2: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'hole depth occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
TYPEOF(sdr.relatng_shape_aspect)) AND (sdr.name =
'hole depth') AND (sdr.relatng_shape_aspect.description =
'linear')) )) = 1)) )) = 1)) )) = 0);
  wr3: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')

```

```

| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'bottom condition occurrence') AND (
SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'hole bottom usage') AND (('FEATURE_BASED_PROCESS_PLANNING.'
+ 'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) ) | (
'FEATURE_BASED_PROCESS_PLANNING.HOLE_BOTTOM' IN TYPEOF(fcr.
relating_shape_aspect))) )) = 1)) )) = 1)) )) = 0);
wr4: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'change in diameter occurrence') AND (
SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'taper usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) ) | (
'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
relating_shape_aspect))) )) = 1)) )) <= 1)) )) = 0);
END_ENTITY; -- round_hole

ENTITY rounded_end
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'partial circular boundary occurrence')
AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE' IN
  TYPEOF(sdr.relying_shape_aspect)) AND (sdr.name =
'partial circular boundary')) )) = 1)) )) = 1)) )) = 0);
    wr2: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'course of travel occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'

```

```

+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
TYPEOF(sdr.relying_shape_aspect)) AND (sdr.
relying_shape_aspect.description = 'linear')) )) = 1)) )) =
1)) )) = 0);
END_ENTITY; -- rounded_end

ENTITY rounded_u_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 2)) )) = 0)) )) = 0);
    wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
      AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
    wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'width')))) = 1)))) = 0)))) = 0);

```



```

END_ENTITY; -- rounded_u_profile

ENTITY runout_zone_definition
  SUBTYPE OF (tolerance_zone_definition);
  orientation : runout_zone_orientation;
END_ENTITY;

ENTITY runout_zone_orientation;
  angle : measure_with_unit;
END_ENTITY;

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;
END_ENTITY; -- security_classification_assignment

ENTITY security_classification_level;
  name : label;
END_ENTITY; -- security_classification_level

ENTITY shape_aspect;
  name : label;
  description : text;
  of_shape : product_definition_shape;
  product_definitional : LOGICAL;
END_ENTITY; -- shape_aspect

ENTITY shape_aspect_relationship;
  name : label;
  description : text;
  relating_shape_aspect : shape_aspect;
  related_shape_aspect : shape_aspect;
END_ENTITY; -- shape_aspect_relationship

ENTITY shape_defining_relationship
  SUBTYPE OF (shape_aspect_relationship);
END_ENTITY; -- shape_defining_relationship

ENTITY shape_definition_representation
  SUBTYPE OF (property_definition_representation);
  WHERE
    wr1: (('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINITION' IN TYPEOF(
      SELF.definition.definition)) OR (
      'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(SELF.definition)));
    wr2: ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION' IN
      TYPEOF(SELF.used_representation));
END_ENTITY; -- shape_definition_representation

```

```

ENTITY shape_dimension_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: (SIZEOF(QUERY ( temp <* SELF.items | (NOT
      ('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
        IN TYPEOF(temp))) )) = 0);
    wr2: (SIZEOF(SELF.items) <= 2);
    wr3: (SIZEOF(QUERY ( pos_mri <* QUERY ( real_mri <* SELF.items | (
      'REAL' IN TYPEOF(real_mri\measure_with_unit.value_component)) )
      | (NOT (pos_mri\measure_with_unit.value_component > 0))) )= 0);
END_ENTITY; -- shape_dimension_representation

ENTITY shape_representation
  SUBTYPE OF (representation);
END_ENTITY; -- shape_representation

ENTITY shape_representation_with_parameters
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: (SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT',
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_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(SELF.name);
END_ENTITY; -- si_unit

ENTITY slot
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 2)) )) = 0);
    wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
      QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([

```

```

'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF(it)) = 2) AND (it.name = 'depth')) )) = 1)) )) = 0)) ))
= 0);
wr3: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'swept shape occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) ) | ((
SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
  TYPEOF(sdr.relying_shape_aspect)) = 1) AND (sdr.
relying_shape_aspect.description = 'swept shape')) )) = 1)) ))
= 1)) )) = 0);
wr4: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'course of travel occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) ) | (
'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
  TYPEOF(sdr.relying_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
wr5: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
  TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'end condition occurrence') AND (
SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'slot end usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.SLOT_END' IN TYPEOF(fcr.
relying_shape_aspect)) AND (fcr.relying_shape_aspect.
description = 'end condition')) )) = 1)) )) = 2)) )) = 0);

```

END_ENTITY; -- slot

ENTITY slot_end

SUBTYPE OF (shape_aspect);

WHERE

```

wr1 : ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
wr2 : (SELF.description IN ['open','radiused','flat','woodruff']);
wr3 : ((NOT (SELF.description IN ['open','radiused'])) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 0)) )) = 0));
wr4 : ((NOT (SELF.description IN ['flat','woodruff'])) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0));
wr5 : ((NOT (SELF.description IN ['flat'])) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 2)) )) = 0)) )) = 0));
wr6 : ((NOT (SELF.description = 'flat')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
      * TYPEOF(it)) = 2) AND (it.name = 'first radius')) )) = 1)) ))
      = 0)) )) = 0));
wr7 : ((NOT (SELF.description = 'flat')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'second radius')))) = 1)) ))
        = 0)) )) = 0));
wr8 : ((NOT (SELF.description = 'woodruff')) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'
        IN TYPEOF(it)) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
        = 0));
wr9 : ((NOT (SELF.description IN ['woodruff'])) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 1)) )) = 0)) )) = 0));
wr10: (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATING_SHAPE_ASPECT') | ((sar.description =
        'slot end usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF(sar))) ) | ((fcr.related_shape_aspect.description
        = 'end condition occurrence') AND (
        'FEATURE_BASED_PROCESS_PLANNING.SLOT' IN TYPEOF(fcr.
        related_shape_aspect.of_shape.definition))) )) >= 1);
END_ENTITY; -- slot_end

ENTITY solid_angle_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.SOLID_ANGLE_UNIT' IN TYPEOF(
            SELF\measure_with_unit.unit_component));
END_ENTITY; -- solid_angle_measure_with_unit

ENTITY solid_angle_unit
    SUBTYPE OF (named_unit);
    WHERE
        wr1: ((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
            named_unit.dimensions.mass_exponent = 0) AND (SELF\

```

```

        named_unit.dimensions.time_exponent = 0) AND (SELF\
        named_unit.dimensions.electric_current_exponent = 0) AND (
        SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0) AND (SELF\named_unit
        .dimensions.amount_of_substance_exponent = 0) AND (SELF\
        named_unit.dimensions.luminous_intensity_exponent = 0));
END_ENTITY; -- solid_angle_unit

ENTITY solid_model
    SUPERTYPE OF (manifold_solid_brep)
    SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- solid_model

ENTITY spherical_cap
    SUBTYPE OF (feature_definition);
    WHERE
        wr1: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
        QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)))) = 0)) ))
        = 0);
        wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
        QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'internal angle'))))) = 1)) ))
        = 0)) )) = 0);
END_ENTITY; -- spherical_cap

ENTITY spherical_surface
    SUBTYPE OF (elementary_surface);
    radius : positive_length_measure;
END_ENTITY; -- spherical_surface

ENTITY square_u_profile
    SUBTYPE OF (shape_aspect);
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
        IN TYPEOF(SELF.of_shape.definition));
        wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) = 1)) )) = 0);
wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 6)) )) = 0)) )) = 0);
wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
= 0);
wr6: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'first radius')) )) = 1)) ))
= 0)) )) = 0);
wr7: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')

```

```

        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'])*
        TYPEOF(it)) = 2) AND (it.name = 'second radius')) )) = 1)) ))
        = 0)) )) = 0);
wr8: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
        = 0)) )) = 0);
wr9: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
        = 0)) )) = 0);
END_ENTITY; -- square_u_profile

ENTITY standard_uncertainty
  SUPERTYPE OF (expanded_uncertainty)
  SUBTYPE OF (uncertainty_qualifier);
  uncertainty_value : REAL;
END_ENTITY; -- standard_uncertainty

ENTITY step
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
        sa_occ.description = 'course of travel occurrence') AND (
        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,

```



```

        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
TYPEOF(sdr.relying_shape_aspect)) AND (sdr.
relying_shape_aspect.description = 'linear')) )) = 1)) )) =
1)) )) = 0);
wr2: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION' )
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION.SHAPE' IN
TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
sa_occ.description = 'removal boundary occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE' IN TYPEOF(sdr.
relying_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
END_ENTITY; -- step

ENTITY surface
  SUPERTYPE OF (ONEOF (elementary_surface,swept_surface,bounded_surface))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- surface

ENTITY surface_curve
  SUBTYPE OF (curve);
  curve_3d : curve;
  associated_geometry : LIST [1:2] OF pcurve_or_surface;
  master_representation : preferred_surface_curve_representation;
  DERIVE
    basis_surface : SET [1:2] OF surface := get_basis_surface(SELf);
  WHERE
    wr1: (curve_3d.dim = 3);
    wr2: (('FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(
      associated_geometry[1])) OR (master_representation <>
      pcurve_s1));
    wr3: (('FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(
      associated_geometry[2])) OR (master_representation <>
      pcurve_s2));
    wr4: (NOT ('FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(
      curve_3d)));
END_ENTITY; -- surface_curve

ENTITY surface_of_linear_extrusion
  SUBTYPE OF (swept_surface);
  extrusion_axis : vector;
END_ENTITY; -- surface_of_linear_extrusion

ENTITY surface_of_revolution

```

```

SUBTYPE OF (swept_surface);
  axis_position : axis1_placement;
DERIVE
  axis_line : line := line(axis_position.location,vector(axis_position
    .z,1));
END_ENTITY; -- surface_of_revolution

ENTITY surface_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: (SELF\geometric_tolerance.name = 'surface profile');
    wr2: ((NOT (('FEATURE_BASED_PROCESS_PLANNING.' +
      'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF)))
      OR (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
        datum_system) <= 3));
END_ENTITY; -- surface_profile_tolerance

ENTITY swept_surface
  SUPERTYPE OF (ONEOF (surface_of_linear_extrusion,surface_of_revolution))
  SUBTYPE OF (surface);
  swept_curve : curve;
END_ENTITY; -- swept_surface

ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: (SELF\geometric_tolerance.name = 'symmetry');
    wr2: (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
      datum_system) <= 3);
END_ENTITY; -- symmetry_tolerance

ENTITY taper
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
    wr2: (SELF.description IN ['angle taper','diameter taper']);
    wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);
    wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 1)) )) = 0)) )) = 0);
    wr5: ((NOT (SELF.description = 'angle taper')) OR (SIZEOF(

```

```

    QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'taper angle')) )) = 1)) ))
= 0)) )) = 0));
wr6: ((NOT (SELF.description = 'diameter taper')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'taper diameter')) )) = 1)) ))
= 0)) )) = 0));
END_ENTITY; -- taper

ENTITY tee_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
      IN TYPEOF(SELF.of_shape.definition));
    wr2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);
    wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 10)) )) = 0)) )) = 0);
    wr4 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd, 'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
        AND (it.name = 'orientation')) ) = 1)) ) = 0)) ) = 0);
wr5 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'width')))) = 1)))) = 0)) ) =
        0);
wr6 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'depth')))) = 1)))) = 0)) ) =
        0);
wr7 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'cross bar width')) ) =
        1)) ) = 0)) ) = 0);
wr8 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',

```

```

        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'cross bar depth')) )) =
        1)) )) = 0)) )) = 0);
wr9 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'first offset')) )) = 1)) ))
        = 0)) )) = 0);
wr10: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'second offset')) ))= 1)) ))
        = 0)) )) = 0);
wr11: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
        = 0)) )) = 0);
wr12: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
        = 0)) )) = 0);

```

```

wr13: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
  | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1))) =
    0))) )) = 0);
END_ENTITY; -- tee_profile

ENTITY thread
  SUBTYPE OF (feature_definition);
  WHERE
    wr1 : (SIZEOF(QUERY ( pd <*
USEDIN(SELF, 'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
  | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'major diameter'))))= 1))) ))
    = 0))) )) = 0);
    wr2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
  | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'minor diameter')) )) <=
    1))) )) = 0))) )) = 0);
    wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
  | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
    pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'FEATURE_BASED_PROCESS_PLANNING.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
    'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']

```

```

* TYPEOF(it)) = 2) AND (it.name = 'pitch diameter')) )) <=
1)) )) = 0)) )) = 0);
wr4 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.RATIO_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'number of threads')) )) =
1)) )) = 0)) )) = 0);
wr5 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'fit class')) )) <= 1)) )) =
0)) )) = 0);
wr6 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'form')) )) = 1)) )) = 0)) ))
= 0);
wr7 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'hand')) )) = 1)) )) = 0)) ))
= 0);
wr8 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'qualifier')) ))= 1)) ))= 0)) ))
        = 0);
wr9 : (SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'thread system') AND ((it.
        description = 'english') OR (it.description = 'metric')))) ))
        = 1)) )) = 0)) )) = 0);
wr10: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'tread side') AND ((it.
        description = 'internal') OR (it.description = 'external')))) ))
        = 1)) )) = 0)) )) = 0);
wr11: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELf,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
        | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
        IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
        pds, 'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
        | ((sa_occ.description = 'partial area occurrence') AND (
        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'applied area usage') AND ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
        'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN TYPEOF(sdr
        .relating_shape_aspect)) )) = 1)) )) <= 1)) )) = 0);
END_ENTITY; -- thread

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 (projected_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: (SELF\geometric_tolerance.name = 'total runout');
            wr2: (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: ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION' IN TYPEOF(
                SELF.of_shape.definition));
    END_ENTITY; -- transition_feature

    ENTITY turned_knurl
        SUBTYPE OF (feature_definition);
        WHERE
            wr1 : (SELF.description IN ['diamond', 'diagonal', 'straight']);
            wr2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
                'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
```

```

QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
+ 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (
'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
\measure_with_unit.value_component)) AND (it.name =
'number of teeth')) ) <= 1)) ) = 0)) ) = 0);
wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
+ 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'major diameter')))) = 1)) )
= 0)) ) = 0);
wr4 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
+ 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'nominal diameter')) ) =
1)) ) = 0)) ) = 0);
wr5 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
+ 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'tooth depth')) ) <= 1)) )
= 0)) ) = 0);
wr6 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
+ 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((

```

```

'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'root fillet')) )) <= 1)) ))
= 0)) )) = 0);
wr7 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
QUERY ( pdr <* USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.'
+ 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'diametral pitch')) )) =
1)) )) = 0)) )) = 0);
wr8 : ((NOT (SELF.description IN ['diamond', 'diagonal'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'helix angle')) )) = 1)) ))
= 0)) )) = 0);
wr9 : ((NOT (SELF.description = 'diagonal')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'helix hand')) )) = 1)) )) =
0)) )) = 0);
wr10: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
pds, 'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
| ((sa_occ.description = 'partial area occurrence') AND (

```

```

        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'applied area usage') AND ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
        'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN TYPEOF(sdr
        .relating_shape_aspect)) )) = 1)) )) <= 1)) )) = 0);
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 : text;
    WHERE
        wr1: ((NOT ('NUMBER' IN TYPEOF(SELF\measure_with_unit.
        value_component))) OR (SELF\measure_with_unit.
        value_component >= 0));
END_ENTITY; -- uncertainty_measure_with_unit

ENTITY uncertainty_qualifier
    SUPERTYPE OF (ONEOF (standard_uncertainty, qualitative_uncertainty));
    measure_name : label;
    description : text;
END_ENTITY; -- uncertainty_qualifier

ENTITY uniform_curve
    SUBTYPE OF (b_spline_curve);
END_ENTITY; -- uniform_curve

ENTITY uniform_surface
    SUBTYPE OF (b_spline_surface);
END_ENTITY; -- uniform_surface

ENTITY vector
    SUBTYPE OF (geometric_representation_item);
    orientation : direction;
    magnitude : length_measure;
    WHERE
        wr1: (magnitude >= 0);
END_ENTITY; -- vector

ENTITY vee_profile
    SUBTYPE OF (shape_aspect);
    WHERE
        wr1: ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
        IN TYPEOF(SELF.of_shape.definition));
        wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) = 1)) )) = 0);
wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 4)) )) = 0)) )) = 0);
wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0);
wr5: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items |
(('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (
'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT' IN
TYPEOF(it.measure_with_unit.value_component)) AND (it.name
= 'profile_radius')) )) = 1)) )) = 0)) )) = 0);
wr6: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND (it.name = 'profile angle')) )) = 1)) ))
= 0)) )) = 0);
wr7: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
| (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'FEATURE_BASED_PROCESS_PLANNING.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'tilt angle')) )) = 1)) ))
        = 0)) )) = 0);
END_ENTITY; -- vee_profile

ENTITY versioned_action_request;
    id          : identifier;
    version     : label;
    purpose     : text;
    description : text;
END_ENTITY; -- versioned_action_request

ENTITY vertex
    SUBTYPE OF (topological_representation_item);
END_ENTITY; -- vertex

ENTITY vertex_loop
    SUBTYPE OF (loop);
    loop_vertex : vertex;
END_ENTITY; -- vertex_loop

ENTITY vertex_point
    SUBTYPE OF (vertex, geometric_representation_item);
    vertex_geometry : point;
END_ENTITY; -- vertex_point

ENTITY week_of_year_and_day_date
    SUBTYPE OF (date);
    week_component : week_in_year_number;
    day_component  : OPTIONAL day_in_week_number;
END_ENTITY; -- week_of_year_and_day_date

RULE application_context_requires_ap_definition FOR (application_context,
    application_protocol_definition);

WHERE
    wr1: (SIZEOF(QUERY ( ac <* application_context | (NOT (SIZEOF(
        QUERY ( apd <* application_protocol_definition | ((ac :=: apd.
        application) AND (apd.application_interpreted_model_schema_name =
        'feature_based_process_planning')) )) = 1)) )) = 0);

END_RULE; -- application_context_requires_ap_definition

RULE approval_requires_approval_date_time FOR (approval,
    approval_date_time);

WHERE
    wr1: (SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( adt <*
        approval_date_time | (app :=: adt.dated_approval) )) = 1)) ))

```

```

        = 0);

END_RULE; -- approval_requires_approval_date_time

RULE approval_requires_approval_person_organization FOR (approval,
    approval_person_organization);

WHERE
    wr1: (SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( apo <*
        approval_person_organization | (app := apo.
        authorized_approval) )) >= 1)) )) = 0);

END_RULE; -- approval_requires_approval_person_organization

RULE compatible_dimension FOR (cartesian_point, direction,
    representation_context, geometric_representation_context);

WHERE
    wr1: (SIZEOF(QUERY ( x <* cartesian_point | (SIZEOF(QUERY ( y <*
        geometric_representation_context | (item_in_context(x,y) AND (
        HIINDEX(x.coordinates) <> y.coordinate_space_dimension)) )) >
        0) )) = 0);
    wr2: (SIZEOF(QUERY ( x <* direction | (SIZEOF(QUERY ( y <*
        geometric_representation_context | (item_in_context(x,y) AND (
        HIINDEX(x.direction_ratios) <> y.coordinate_space_dimension)) )) >
        0) )) = 0);

END_RULE; -- compatible_dimension

RULE dependent_instantiable_action_request_status FOR (
    action_request_status);

WHERE
    wr1: (SIZEOF(QUERY ( arst <* action_request_status | (NOT (SIZEOF(
        USEDIN(arst, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_action_request_status

RULE dependent_instantiable_action_status FOR (action_status);

WHERE
    wr1: (SIZEOF(QUERY ( ast <* action_status | (NOT (SIZEOF(USEDIN(ast, ''))
        >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_action_status

RULE dependent_instantiable_approval_status FOR (approval_status);

WHERE
    wr1: (SIZEOF(QUERY ( ast <* approval_status | (NOT (SIZEOF(USEDIN(ast,
        '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_approval_status

RULE dependent_instantiable_date FOR (date);

```

```

WHERE
  wr1: (SIZEOF(QUERY ( dt <* date | (NOT (SIZEOF(USEDIN(dt, '')) >= 1)) ))
        = 0);

END_RULE; -- dependent_instantiable_date

RULE dependent_instantiable_named_unit FOR (named_unit);

WHERE
  wr1: (SIZEOF(QUERY ( nu <* named_unit | (NOT (SIZEOF(USEDIN(nu, '')) >=
        1)) )) = 0);

END_RULE; -- dependent_instantiable_named_unit

RULE dependent_instantiable_precision_qualifier FOR (precision_qualifier);

WHERE
  wr1: (SIZEOF(QUERY ( pq <* precision_qualifier | (NOT (SIZEOF(USEDIN(
        pq, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_precision_qualifier

RULE dependent_instantiable_security_classification_level FOR (
  security_classification_level);

WHERE
  wr1: (SIZEOF(QUERY ( scl <* security_classification_level | (NOT (
        SIZEOF(USEDIN(scl, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_security_classification_level

RULE dependent_instantiable_shape_representation FOR (
  shape_representation);

WHERE
  wr1: (SIZEOF(QUERY ( sr <* shape_representation | (NOT (SIZEOF(USEDIN(
        sr, '')) >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_shape_representation

RULE dependent_instantiable_type_qualifier FOR (type_qualifier);

WHERE
  wr1: (SIZEOF(QUERY ( tq <* type_qualifier | (NOT (SIZEOF(USEDIN(tq, ''))
        >= 1)) )) = 0);

END_RULE; -- dependent_instantiable_type_qualifier

RULE dependent_instantiable_uncertainty_qualifier FOR (
  uncertainty_qualifier);

WHERE
  wr1: (SIZEOF(QUERY ( uq <* uncertainty_qualifier | (NOT (SIZEOF(
        USEDIN(uq, '')) >= 1)) )) = 0);

```



```

END_RULE; -- dependent_instantiable_uncertainty_qualifier

RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);

WHERE
  wr1: (SIZEOF(QUERY ( gt <* geometric_tolerance | (NOT (SIZEOF(TYPEOF(
    gt) * [
      'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_RUNOUT_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.ANGULARITY_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.CONCENTRICITY_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.PARALLELISM_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.PERPENDICULARITY_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.TOTAL_RUNOUT_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.SURFACE_PROFILE_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.SYMMETRY_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.POSITION_TOLERANCE',
      'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE_TOLERANCE']) <=
    2)) )) = 0);

END_RULE; -- geometric_tolerance_subtype_exclusiveness

RULE machining_feature_life_cycle FOR (machining_feature);

WHERE
  wr1: (SIZEOF(QUERY ( mf <* machining_feature | (NOT (mf.of_shape.
    definition.frame_of_reference.life_cycle_stage =
    'manufacturing_planning')) )) = 0);

END_RULE; -- machining_feature_life_cycle

RULE material_is_specified_for_part FOR (product_definition,
  make_from_usage_option);

WHERE
  wr1: (SIZEOF(QUERY ( nmpd <* QUERY ( pd <* product_definition | (
    SIZEOF(USEDIN(pd, 'FEATURE_BASED_PROCESS_PLANNING.' +
    'MATERIAL DESIGNATION.DEFINITIONS')) = 0) ) | (NOT (SIZEOF(
    QUERY ( mfuo <* make_from_usage_option | (NOT (nmpd :=: mfuo.
    relating_product_definition)) )) >= 1)) )) = 0);

END_RULE; -- material_is_specified_for_part

RULE part_requires_project_order FOR (product_definition_formation,
  feature_based_pp_action_assignment);

WHERE
  wr1: (SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
    SIZEOF(QUERY ( fbppaa <* feature_based_pp_action_assignment |
    ((pdf IN fbppaa.items) AND (fbppaa.assigned_action.name =
    'project_order')) )) = 1)) )) = 0);

END_RULE; -- part_requires_project_order

RULE part_to_approval FOR (product_definition_formation,
  feature_based_pp_approval_assignment);

```

```

WHERE
  wr1: (SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
    SIZEOF(QUERY ( fbppa <* feature_based_pp_approval_assignment
      | (pdf IN fbppa.items) )) <= 1)) )) = 0);

END_RULE; -- part_to_approval

RULE product_definition_formation_requires_security_classification FOR (
  product_definition_formation,
  feature_based_pp_security_classification_assignment);

WHERE
  wr1: (SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
    SIZEOF(QUERY ( fbppasca <*
      feature_based_pp_security_classification_assignment | (pdf IN
      fbppasca.items) )) = 1)) )) = 0);

END_RULE; -- product_definition_formation_requires_security_classification

RULE product_requires_version FOR (product, product_definition_formation);

WHERE
  wr1: (SIZEOF(QUERY ( prod <* product | (NOT (SIZEOF(QUERY ( pdf <*
    product_definition_formation | (prod ::= pdf.of_product) )) >=
    1)) )) = 0);

END_RULE; -- product_requires_version

RULE project_order_requires_approval FOR (directed_action,
  feature_based_pp_approval_assignment);

WHERE
  wr1: (SIZEOF(QUERY ( po <* QUERY ( da <* directed_action | (da.name =
    'project order')) | (NOT (SIZEOF(QUERY ( fbppapp <*
    feature_based_pp_approval_assignment | (po IN fbppapp.items) ))
    = 1)) )) = 0);

END_RULE; -- project_order_requires_approval

RULE project_order_tracking_relationships FOR (directed_action,
  action_relationship);

WHERE
  wr1: (SIZEOF(QUERY ( da <* directed_action | ((da.name IN [
    'shop work order', 'resource acquisition order',
    'digital technical data package work order',
    'pedigree creation order']) AND (NOT (SIZEOF(QUERY ( ar <*
    action_relationship | ((da ::= ar.related_action) AND (
    'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
    relating_action)) AND (ar.relatng_action.name =
    'project order')) )) = 1))) )) = 0);
  wr2: (SIZEOF(QUERY ( da <* directed_action | ((da.name =
    'project order') AND (NOT (SIZEOF(QUERY ( ar <*
    action_relationship | ((da ::= ar.relatng_action) AND (
    'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.

```

```

        related_action)) AND (ar.related_action.name =
        'shop work order')) )) <= 1))) )) = 0);
wr3: (SIZEOF(QUERY ( da <* directed_action | ((da.name =
        'project order') AND (NOT (SIZEOF(QUERY ( ar <*
        action_relationship | ((da :=: ar.relateing_action) AND (
        'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
        related_action)) AND (ar.related_action.name =
        'resource acquisition order')) )) <= 1))) )) = 0);
wr4: (SIZEOF(QUERY ( da <* directed_action | ((da.name =
        'project order') AND (NOT (SIZEOF(QUERY ( ar <*
        action_relationship | ((da :=: ar.relateing_action) AND (
        'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
        related_action)) AND (ar.related_action.name =
        'digital technical data package work order')) )) <= 1))) )) = 0);
wr5: (SIZEOF(QUERY ( da <* directed_action | ((da.name =
        'project order') AND (NOT (SIZEOF(QUERY ( ar <*
        action_relationship | ((da :=: ar.relateing_action) AND (
        'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
        related_action)) AND (ar.related_action.name =
        'pedigree creation order')) )) <= 1))) )) = 0);
wr6: (SIZEOF(QUERY ( da <* directed_action | ((da.name =
        'customer order') AND (NOT (SIZEOF(QUERY ( ar <*
        action_relationship | ((da :=: ar.related_action) AND (
        'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
        relating_action)) AND (ar.relateing_action.name =
        'project order')) )) >= 1))) )) = 0);
wr7: (SIZEOF(QUERY ( da <* directed_action | ((da.name =
        'project order') AND (NOT (SIZEOF(QUERY ( ar <*
        action_relationship | ((da :=: ar.relateing_action) AND (
        'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
        related_action)) AND (ar.related_action.name =
        'customer order')) )) <= 1))) )) = 0);

END_RULE; -- project_order_tracking_relationships

RULE representation_subtype_exclusiveness FOR (representation);

WHERE
    wr1: (SIZEOF(QUERY ( rep <* representation | (NOT (SIZEOF(TYPEOF(rep)
        * ['FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION',
        'FEATURE_BASED_PROCESS_PLANNING.DEFINITIONAL_REPRESENTATION',
        'FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION',
        'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION',
        'FEATURE_BASED_PROCESS_PLANNING.LOCATION_SHAPE_REPRESENTATION',
        'FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'])
        <= 2)) )) = 0);

END_RULE; -- representation_subtype_exclusiveness

RULE restrict_approval_status FOR (approval_status);

WHERE
    wr1: (SIZEOF(QUERY ( ast <* approval_status | (NOT (ast.name IN [
        'approved', 'not yet approved', 'disapproved', 'withdrawn'])))= 0);

```

```

END_RULE; -- restrict_approval_status

RULE restrict_security_classification_level FOR (
    security_classification_level);

WHERE
    wr1: (SIZEOF(QUERY ( scl <* security_classification_level | (NOT (scl.
        name IN ['unclassified','classified','proprietary',
        'confidential','secret','top_secret']))) )) = 0);

END_RULE; -- restrict_security_classification_level

RULE shape_aspect_relationship_subtype_exclusiveness FOR (
    shape_aspect_relationship);

WHERE
    wr1: (SIZEOF(QUERY ( sr <* shape_aspect_relationship | (NOT (SIZEOF(
        TYPEOF(sr) * [
        'FEATURE_BASED_PROCESS_PLANNING.DIMENSIONAL_LOCATION',
        'FEATURE_BASED_PROCESS_PLANNING.GEOMETRIC_TOLERANCE_RELATIONSHIP',
        'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP',
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'])
        <= 2)) )) = 0);

END_RULE; -- shape_aspect_relationship_subtype_exclusiveness

RULE shape_aspect_subtype_exclusiveness FOR (shape_aspect);

WHERE
    wr1: (SIZEOF(QUERY ( sr <* shape_aspect | (NOT (SIZEOF(TYPEOF(sr) * [
        'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT',
        'FEATURE_BASED_PROCESS_PLANNING.SLOT_END',
        'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM',
        'FEATURE_BASED_PROCESS_PLANNING.BOSS_TOP',
        'FEATURE_BASED_PROCESS_PLANNING.HOLE_BOTTOM',
        'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA',
        'FEATURE_BASED_PROCESS_PLANNING.TAPER',
        'FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET',
        'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE',
        'FEATURE_BASED_PROCESS_PLANNING.MACHINING_FEATURE',
        'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
        'FEATURE_BASED_PROCESS_PLANNING.TRANSITION_FEATURE',
        'FEATURE_BASED_PROCESS_PLANNING.DATUM',
        'FEATURE_BASED_PROCESS_PLANNING.DATUM_FEATURE',
        'FEATURE_BASED_PROCESS_PLANNING.DATUM_TARGET']) <= 2)) )) = 0);

```

```

END_RULE; -- shape_aspect_subtype_exclusiveness

RULE shape_representation_subtype_exclusiveness FOR (
    shape_representation);

WHERE
    wr1: (SIZEOF(QUERY ( sr <* shape_representation |
        (NOT (SIZEOF(TYPEOF(sr) * [
            'FEATURE_BASED_PROCESS_PLANNING.ADVANCED_BREP_SHAPE_REPRESENTATION',
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS',
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DIMENSION_REPRESENTATION'])
            <= 2)) )) = 0);

END_RULE; -- shape_representation_subtype_exclusiveness

RULE subtype_mandatory_characterized_object FOR (characterized_object);

WHERE
    wr1: (SIZEOF(QUERY ( csa <* characterized_object | (NOT (SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.FEATURE_DEFINITION',
        'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION',
        'FEATURE_BASED_PROCESS_PLANNING.ORDERED_PART']) * 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
    i : INTEGER;
    x : SET OF representation_item;
    y : SET OF representation_item;
END_LOCAL;
x := QUERY ( z <* children_set | (
    'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM' IN TYPEOF(z)) );

```

```

IF SIZEOF(x) > 0 THEN
  REPEAT i := 1 TO HIINDEX(x) BY 1;
    IF x[i]\mapped_item.mapping_source.mapped_representation IN
      parent_set THEN
      RETURN(FALSE);
    END_IF;
    IF NOT acyclic_mapped_representation(parent_set + x[i]\mapped_item
      .mapping_source.mapped_representation, x[i]\mapped_item.
      mapping_source.mapped_representation.items) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
END_IF;
x := children_set - x;
IF SIZEOF(x) > 0 THEN
  REPEAT i := 1 TO HIINDEX(x) BY 1;
    y := QUERY ( z <* bag_to_set(USEDIN(x[i], '')) | (
      'FEATURE_BASED_PROCESS_PLANNING.REPRESENTATION.ITEM' IN TYPEOF(
      z)) );
    IF NOT acyclic_mapped_representation(parent_set, y) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
END_IF;
RETURN(TRUE);

END_FUNCTION; -- acyclic_mapped_representation

FUNCTION acyclic_product_definition_relationship(
  relation: product_definition_relationship;
  relatives: SET OF product_definition;
  specific_relation: STRING
): LOGICAL;

LOCAL
  i          : INTEGER;
  x          : SET OF product_definition_relationship;
  local_relatives : SET OF product_definition;
END_LOCAL;
REPEAT i := 1 TO HIINDEX(relatives) BY 1;
  IF relation.relativing_product_definition == relatives[i] THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
x := bag_to_set(USEDIN(relation.relativing_product_definition,
  specific_relation));
local_relatives := relatives + relation.relativing_product_definition;
IF SIZEOF(x) > 0 THEN
  REPEAT i := 1 TO HIINDEX(x) BY 1;
    IF NOT acyclic_product_definition_relationship(x[i],
      local_relatives, specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
END_IF;

```

```

RETURN(TRUE);

END_FUNCTION; -- acyclic_product_definition_relationship

FUNCTION associated_surface(
    arg: pcurve_or_surface
): surface;

LOCAL
    surf : surface;
END_LOCAL;
IF 'FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(arg) THEN
    surf := arg.basis_surface;
ELSE
    surf := arg;
END_IF;
RETURN(surf);

END_FUNCTION; -- associated_surface

FUNCTION bag_to_set(
    the_bag: BAG OF GENERIC:intype
): SET OF GENERIC:intype;

LOCAL
    i : INTEGER;
    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;
    vec : direction;
    factor : REAL;
END_LOCAL;
IF dim = 3 THEN
    u[3] := NVL(normalise(axis3), direction([0,0,1]));
    u[1] := first_proj_axis(u[3], axis1);
    u[2] := second_proj_axis(u[3], u[1], axis2);
ELSE
    u[3] := ?;
    IF EXISTS(axis1) THEN
        u[1] := normalise(axis1);

```

```

    u[2] := orthogonal_complement(u[1]);
    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
        u[2] := normalise(axis2);
        u[1] := orthogonal_complement(u[2]);
        u[1].direction_ratios[1] := -u[1].direction_ratios[1];
        u[1].direction_ratios[2] := -u[1].direction_ratios[2];
    ELSE
        u[1].direction_ratios[1] := 1;
        u[1].direction_ratios[2] := 0;
        u[2].direction_ratios[1] := 0;
        u[2].direction_ratios[2] := 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
    u : LIST [2:2] OF direction;
END_LOCAL;
u[1] := NVL(normalise(ref_direction),direction([1,0]));
u[2] := orthogonal_complement(u[1]);
RETURN(u);

END_FUNCTION; -- build_2axes

FUNCTION build_axes(
    axis, ref_direction: direction
): LIST [3:3] OF direction;

```