Recommended Practices Guide for Geometric Tolerances

December 15, 2004

Note: This is a draft document containing proposals for the recommended usage of the Geometric Tolerances as defined in AIC 519. This document will continue to be refined and extended. Please forward any issues, questions or suggestions to the authors below.

Contacts

David Briggs
Boeing Commercial Airplanes
P.O. Box 2707 M/S 2R-97
Seattle, WA 98124-2207 USA
david.d.briggs@boeing.com

Tom Hendrix
Boeing Commercial Airplanes
P.O. Box 2707 M/S 2R-97
Seattle, WA 98124-2207 USA
thomas.e.hendrix@boeing.com

Table of Contents

1 Introduction

Tolerances treat the uncertainty with which the realized shape or measurements of a real manufactured object compare to their design ideals. If all parts could be manufactured perfectly as designed, there would be no need for tolerancing practices. However, it is certain that this cannot be done for finite cost in any but the most trivial cases. In the drawing world, tolerances are noted on the drawing per standard notations such as ANSI Y14.5 or ISO 1101.

There are two main classes of tolerances, dimensional and geometric. Dimensional tolerances are addressed in “Recommended Practices for Dimensions and Dimensional Tolerances” written by Markus Hauser, Mike Strub and Tom Hendrix, dated April 18, 2000.

Geometric tolerances are the more complex of these two types. Geometric tolerances provide more flexible means for controlling shape than do dimensional tolerances. They achieve this by enabling tolerances to be defined independently of explicit dimensions. This enables tolerances to be specified that are more closely related to the functional requirements of the design, such as strength and fit. These tolerances are the subject of this recommended practices guide.

2 Scope

This document covers the recommended usage and implementation of geometric tolerances defined in Application Integrated Construct (AIC) 519. This AIC is the result of a harmonization effort between AP214 and AP224. This AIC is also the basis for the tolerance modules to be included in AP203 Edition 2.

This document is not intended as a primer on geometric tolerancing. The explanations included are only provided to relate common tolerancing techniques to the STEP entity structures. This is not a comprehensive coverage of any existing draughting standard but does provide a capability to exchange a variety of typical models. Future versions of this document will address additional capabilities.

This document covers the application of tolerances to boundary representation solid models. The application of tolerances to wireframe or other geometric models is not covered here.

The tolerances addressed in this document are:

Angularity
Circular runout
Circularity
Coaxiality
Concentricity
Cylindricity
Flatness

Parallelism
Perpendicularity
Position
Profile of a line
Profile of a surface
Roundness
Straightness
Symmetry
Total runout

Tolerance modifiers (Maximum and minimum material condition, regardless of feature size and projected tolerance zone) are also addressed.

3 Fundamental Concepts

A geometric tolerance describes a constraint on the acceptable deviation of a manufactured object from the ideal design. Tolerances are applied to the geometric features of a part, such as faces and holes.

The fundamental principles of geometric tolerances can be found in national and international standards such as ANSI Y14.5M-1994 or ISO 5459-1981.

There are several subtypes of the geometric_tolerance entity, which are not mutually exclusive. For example, tolerances that reference datums are of type geometric_tolerance_with_datum_reference. Tolerances that include a modifier such as maximum material condition are of type modified_geometric_tolerance. Many typical engineering tolerances combine these. In these cases, complex entities instances will occur in the Part 21 file.

3.1 Associating tolerances with Features

In STEP, the tolerance entities are associated with a shape_aspect that identifies the toleranced feature. The feature is identified by a shape_aspect which has a representation. In the case of a solid boundary representation model, the feature of the part is represented by one or more topological_representation_items such as advanced_face entities. For example, a through hole in a solid model might be represented by two semi-circular surfaces, each an advanced_face entity. These topological_representation_items are collected together by a shape_representation which is representation of the shape_aspect for the feature. This shape_representation is typically a connected_face_set and shall share the same geometric_representation_context as the solid. See Figure 1 for an example of how the tolerance entities are related to the shape elements of the toleranced feature.

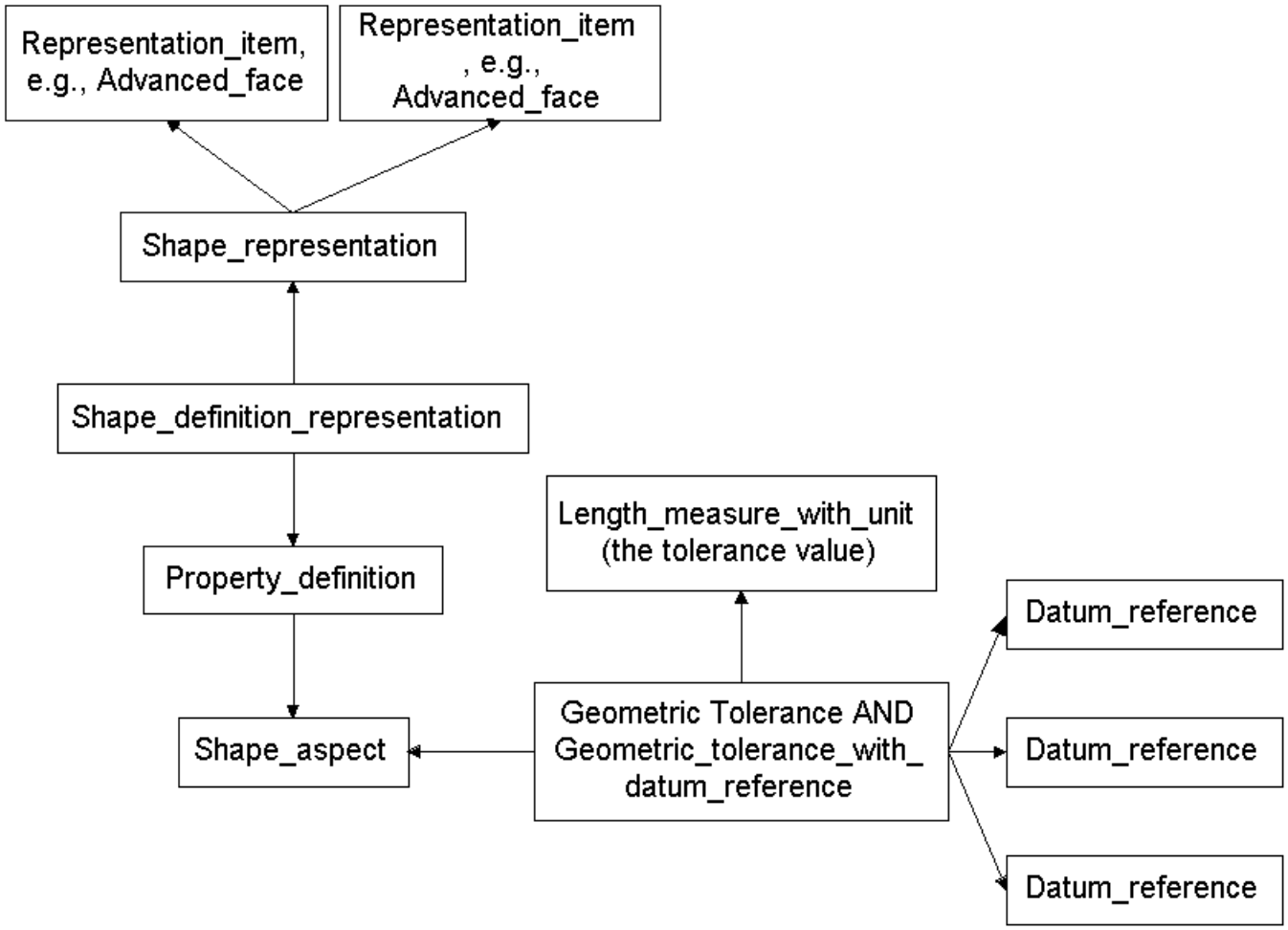


Figure 1 Relating a typical tolerance to a feature's representatation

In the case where the tolerance applies to more than one feature, such as a pattern of holes, the a composite_shape_aspect is used rather than shape_aspect. The composite_shape_aspect is related to the tolerance in the same manner as the shape_aspect.

3.2 Datum Systems

Some types of tolerances refer to one or more datums in order to represent the requirements on the shape. Datum systems are related datums that provide a reference system for describing requirements on the product shape. Datum systems are defined by datum entities and their corresponding datum_feature entities.

3.2.1 Datums

A datum is a theoretically exact geometric reference, such as an exact point, axis or plane, to which toleranced features are related. A datum is the origin from which the location or geometric characteristics of features of a part are established. A datum may be based on one or more datum features of a part (Definition from ISO 5459-1981).

Since the datum is intended to be the idealized geometry, unbounded geometric entities are used as the representation_item. For a boundary representation solid model, these entities are planes, lines, etc.

3.2.2 Datums features

Datum features are tangible features of a part, for example a face that provides a reference system for measurements of the actual part. Datum features must lie on the physical boundary of the shape. Consequentially, datum_feature entities are related to topological entities that represent those boundaries in the solid model such as an advanced_face.

Where tolerances contain references to datums, the tolerance entity is considered a subtype of geometric_tolerance_with_datum_reference. It thus inherits the datum_system attribute, which is the mechanism for pointing to the datum_reference entities.

In the figure below, the tolerance entity, e.g., a positional_tolerance uses three datum_reference entities to describe the datum system for the tolerance. The datum_reference points to a datum entity, which in turns points to the geometric elements that represents the datum. The datum_feature, i.e., the feature on the part corresponding to the datum, references the topological elements of the solid model representing that feature, i.e., the advanced_face entities. See Figure 2 for details.

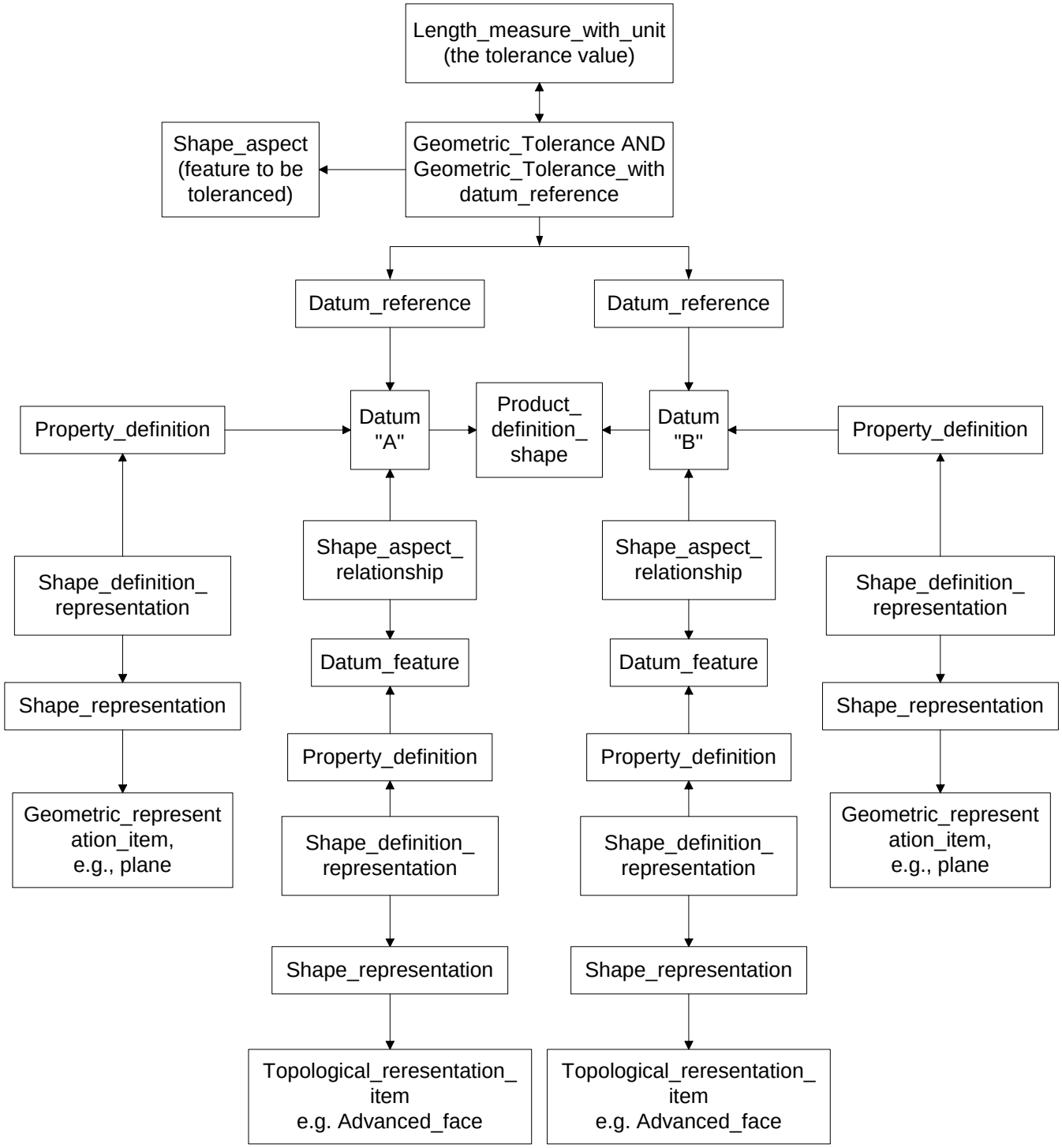


Figure 2 Datums and datum_features

3.2.3 Datum targets

A datum_target designates a specific point, line or area of contact on a part that is used in establishing a data reference frame (definition from ANSI Y14.5). It differs from a datum_feature in that it identifies a

restricted region of a feature, i.e. a point, line or area of a surface rather than a topological feature. Typically, two or more datum_target elements are used to define a datum

A datum_target is represented by a placed_datum_target_feature. The placed_datum_target_feature is related to datum that it helps establish via the mechanism depicted in Figure 3 below. The point, line or area of the datum_target is represented by a shape_representation_with_parameters containing a measure_representation. See Figure 3.

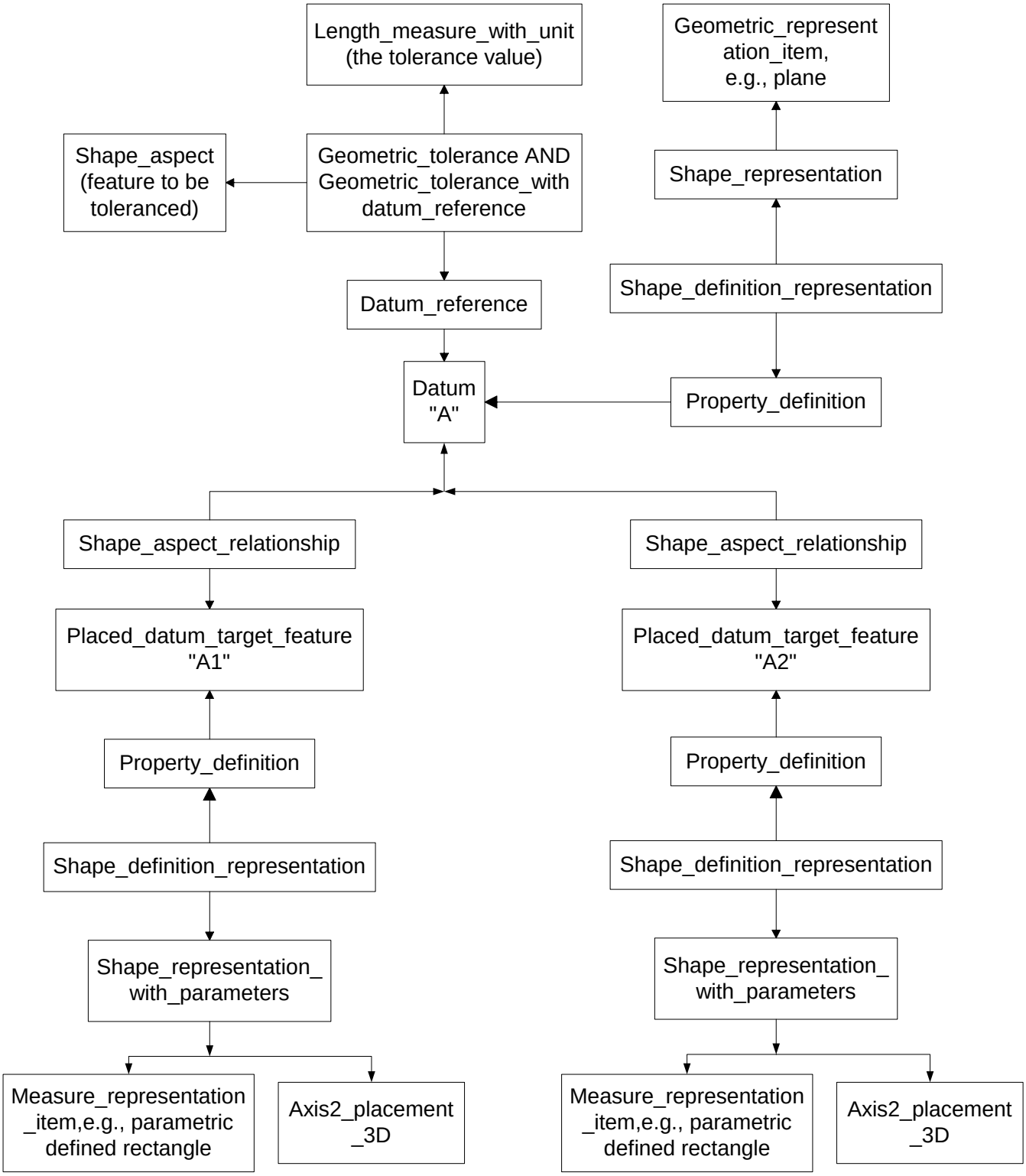


Figure 3 Datum targets

3.2.4 Modified Datum Reference

A reference to a datum can contain a modifier that specifies a condition, such as maximum material condition. For example, a control frame whose visual representation is depicted in Figure 4 is represented in a similar manner to an unmodified datum reference except that instead of a datum_reference, the subtype referenced_modified_datum is used. The modifier attribute contains the condition information as a value of type limit_condition.

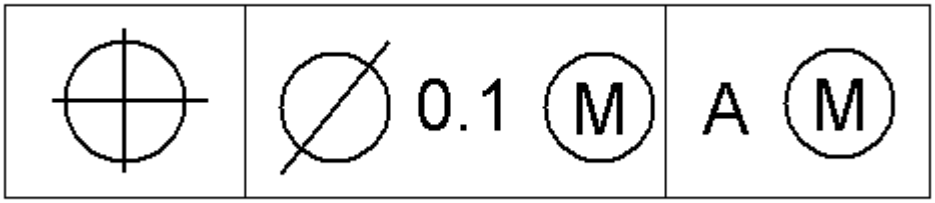


Figure 4 Modified datum reference

3.3 Modified Tolerance

Certain engineering tolerances are modified by adding conditions. The conditions identified in STEP are defined in the limit_condition enumerated type. The values defined are: maximum_material_condition, least_material_condition and regardless_of_feature_size. These are referred to in tolerance standards as MMC, LMC and RFS respectively.

When the tolerance is modified, the tolerance entity is considered a subtype of modified_geometric_tolerance and thus inherits the modifier attribute. The modifier attribute contains the enumerated value of type condition. The general structure is depicted in Figure 5.

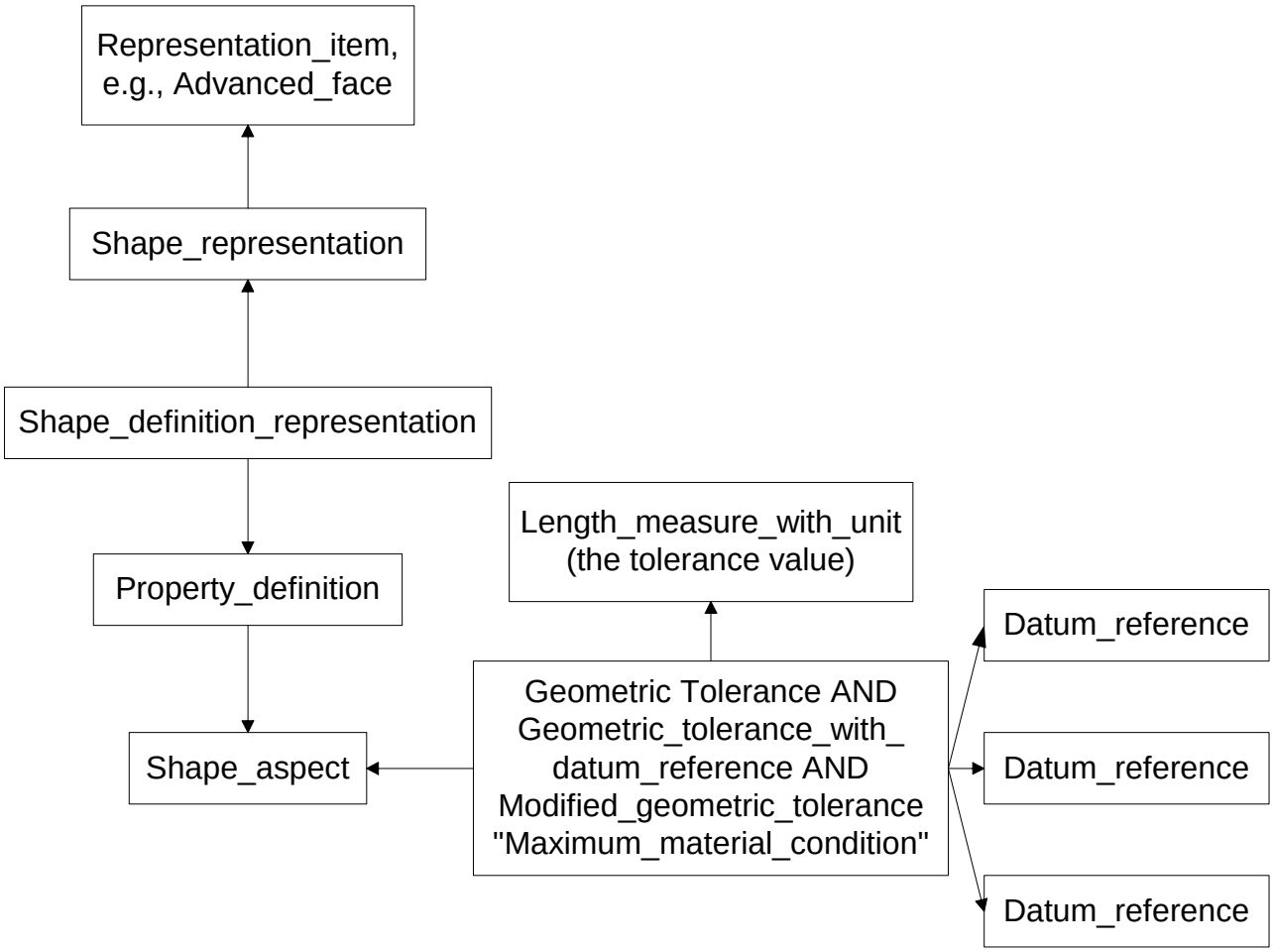


Figure 5 Modified Tolerance

3.4 Toleranced Features with multiple elements

In many cases, a tolerance is applied to a feature of the part that is comprised of multiple shape elements. For example, a tolerance applied to two disjoint, parallel faces of a part that form one of the sides. In these cases, a composite_shape_aspect is used to relate the various features together and the tolerance is applied to the composite_shape_aspect. See Figure 6 for details.

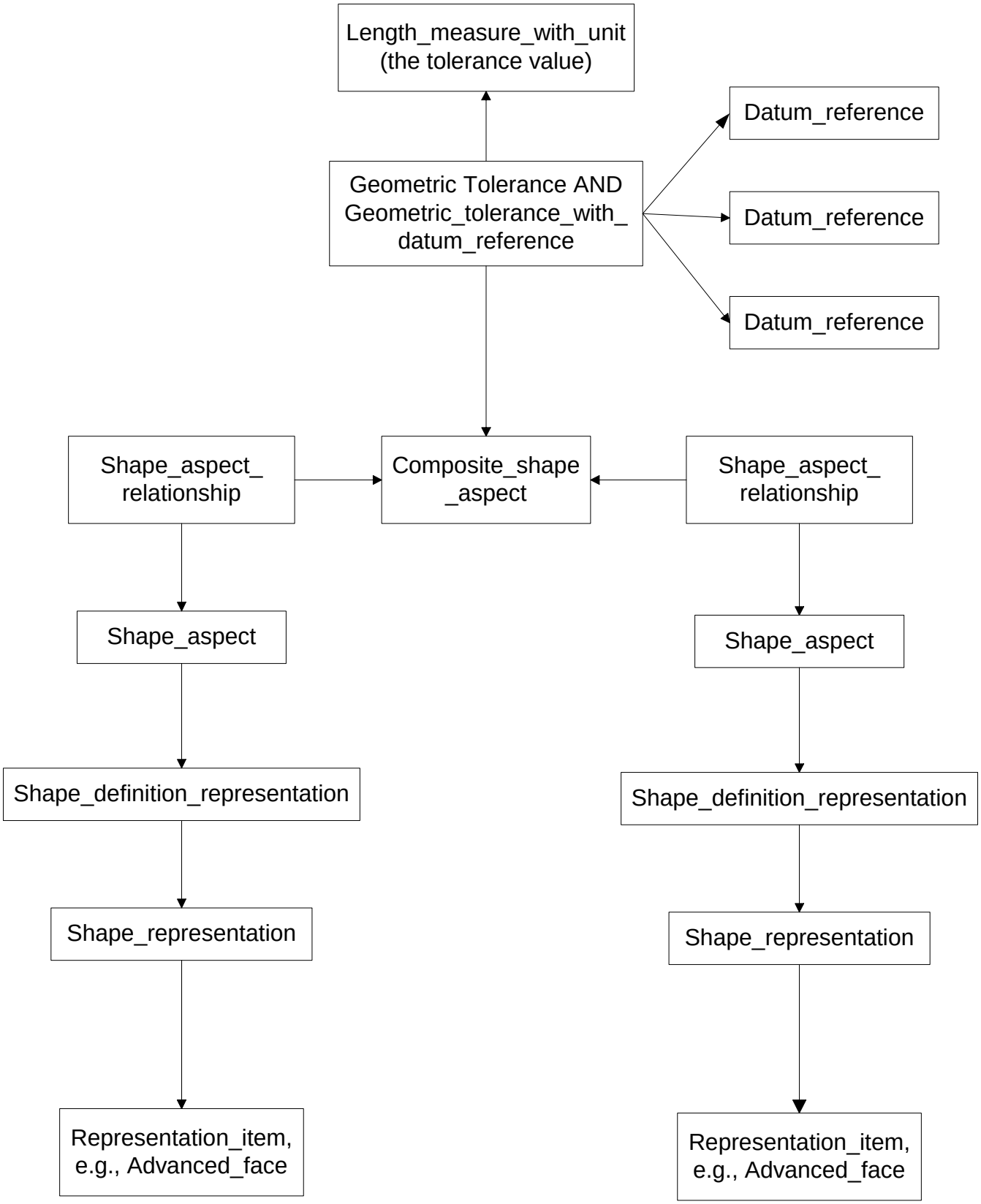


Figure 6 Tolerancing multiple features

3.5 Tolerancing patterns of Features

In addition to tolerancing an individual feature, tolerances can also be applied to a pattern of features. Typical usages include applying the same tolerance to a pattern of holes or slots. See Figure 7 for an example of a typical control frame that applies to a pattern of features .

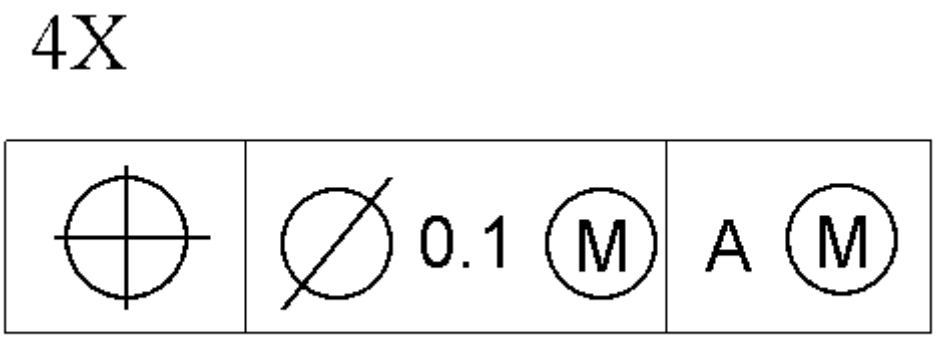


Figure 7 Pattern Control Frame

In the pattern of features, each individual feature (e.g., hole or slot) is a shape_aspect. A composite_shape_aspect is used to relate the shape_aspects together and the tolerance is applied to the composite_shape_aspect. See Figure 6 for details.

3.6 Composite Tolerances

Some tolerances have multiple requirements as represented by a multiple frame tolerance control frame whose visual representation is shown in Figure 8.

Figure 8 Composite Control Frame

This is implemented by creating the appropriate geometric_tolerance entity for each frame. The geometric_tolerance entities are then related via a geometric_tolerance_relationship. The description attribute of the geometric_tolerance_relationship entity shall contain the value “Composite Tolerance”. See Figure 9 for the structure of the tolerance depicted in Figure 8.

In the geometric_tolerance_relationship entity, the upper frame is the “relating” reference and the lower frame is the “related” reference. For tolerances with more than two frames, multiple geometric_tolerance_relationship entities are used to relate the frames together. For example, a tolerance with three frames would require two geometric_tolerance_relationship entities: one relating the top (“relating”) and middle frames (“related”), the second relating the middle (“relating”) and the bottom (“related”) frames. These relationships will allow the receiver to reconstruct the semantic relationships as well as the visual representation.

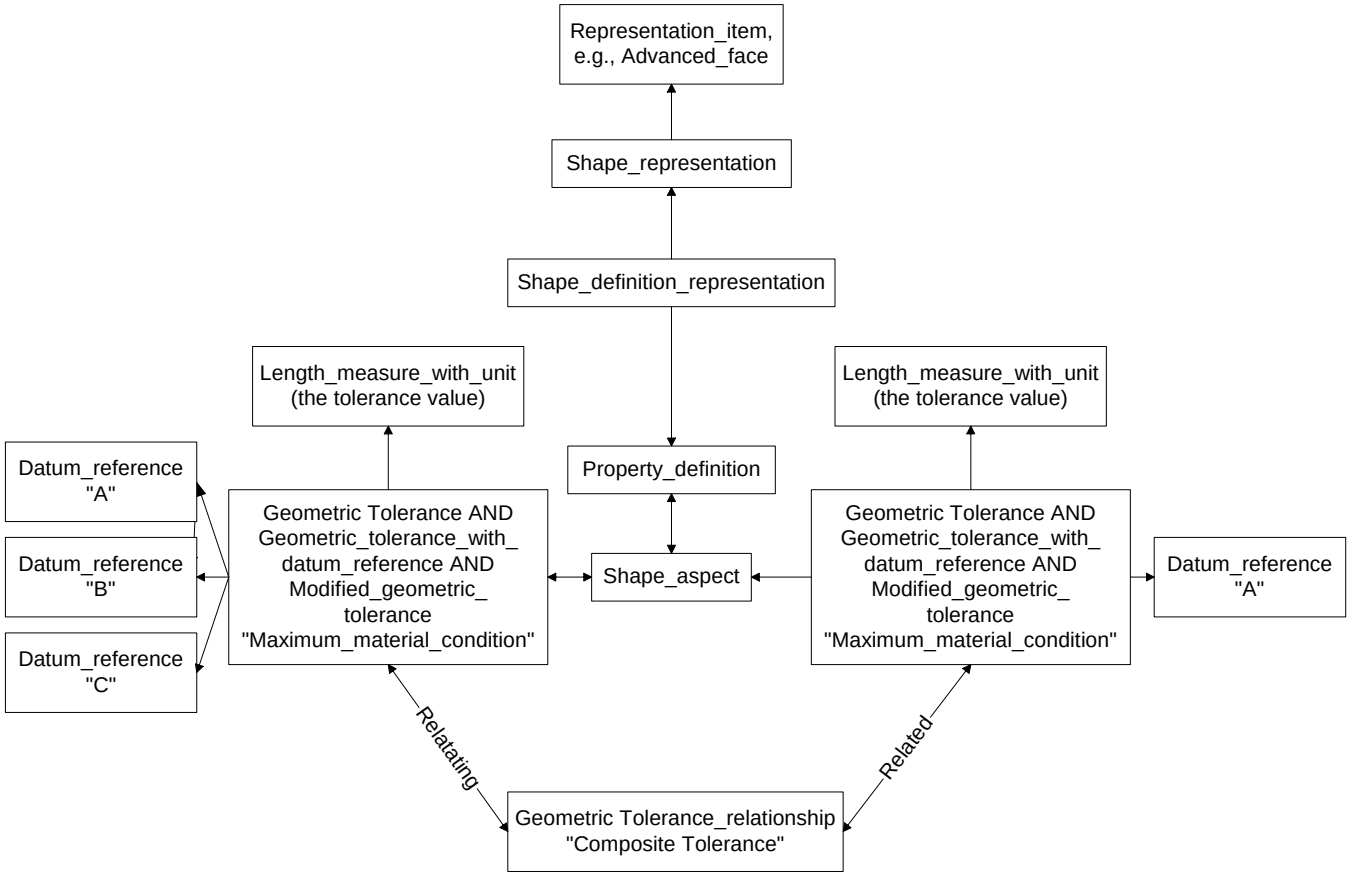


Figure 9 Structure of Composite Tolerance

3.7 Tolerances on assemblies (Under construction)

In some engineering situations, tolerances are associated to the assembly level product. The following sections examine different cases of toleranced assemblies.

3.7.1 Toleranced placement of a component

The placement and/or orientation of a component in an assembly might be controlled by a tolerance. In this case, the tolerance needs to be associated with the instance of the component, i.e., the next_assembly_usage_occurrence (or other usage_occurrance). The tolerance is associated with a

shape_aspect which is related to the NAUO through a product_definition_shape. This provides the link to the instance of the component. When the tolerance is associated with a specific feature on the component, this shape_aspect is related to the shape_aspect in the component that defines the shape of the feature. See Figure 10 for details.

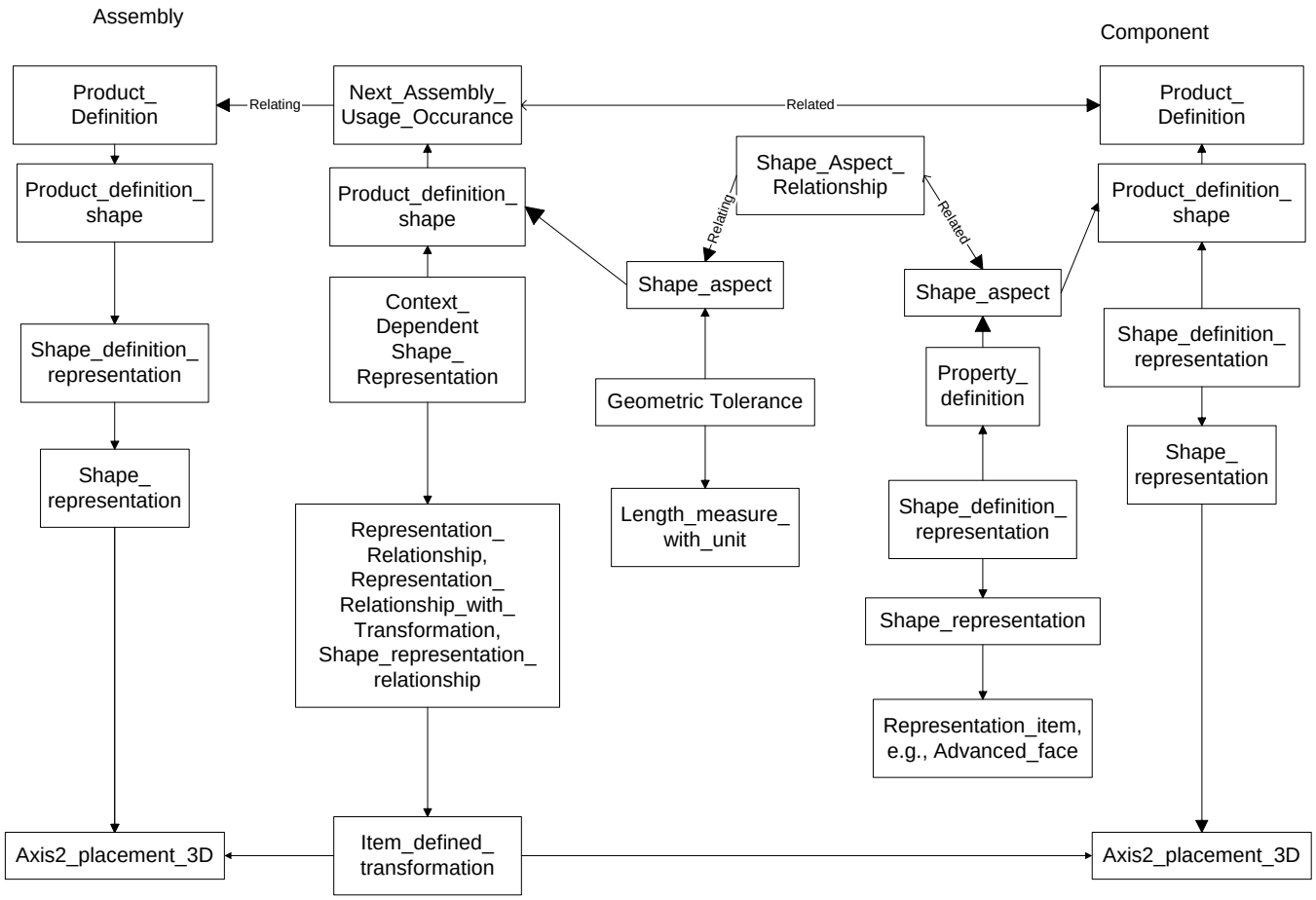


Figure 10 Tolerance on component instance

3.7.2 Assembly features

Another case is a tolerance on a feature that exists in the context of the assembley. An example is a hole drilled on assembly. In this case the shape of the toleranced feature, e.g., the hole, is defined as part of the shape-representation of the assembly. A shape_aspect defining the shape of the feature is associated with the product_definition_shape of the assembly. The tolerance is associated with this shape-aspect. See Figure 11 for details.

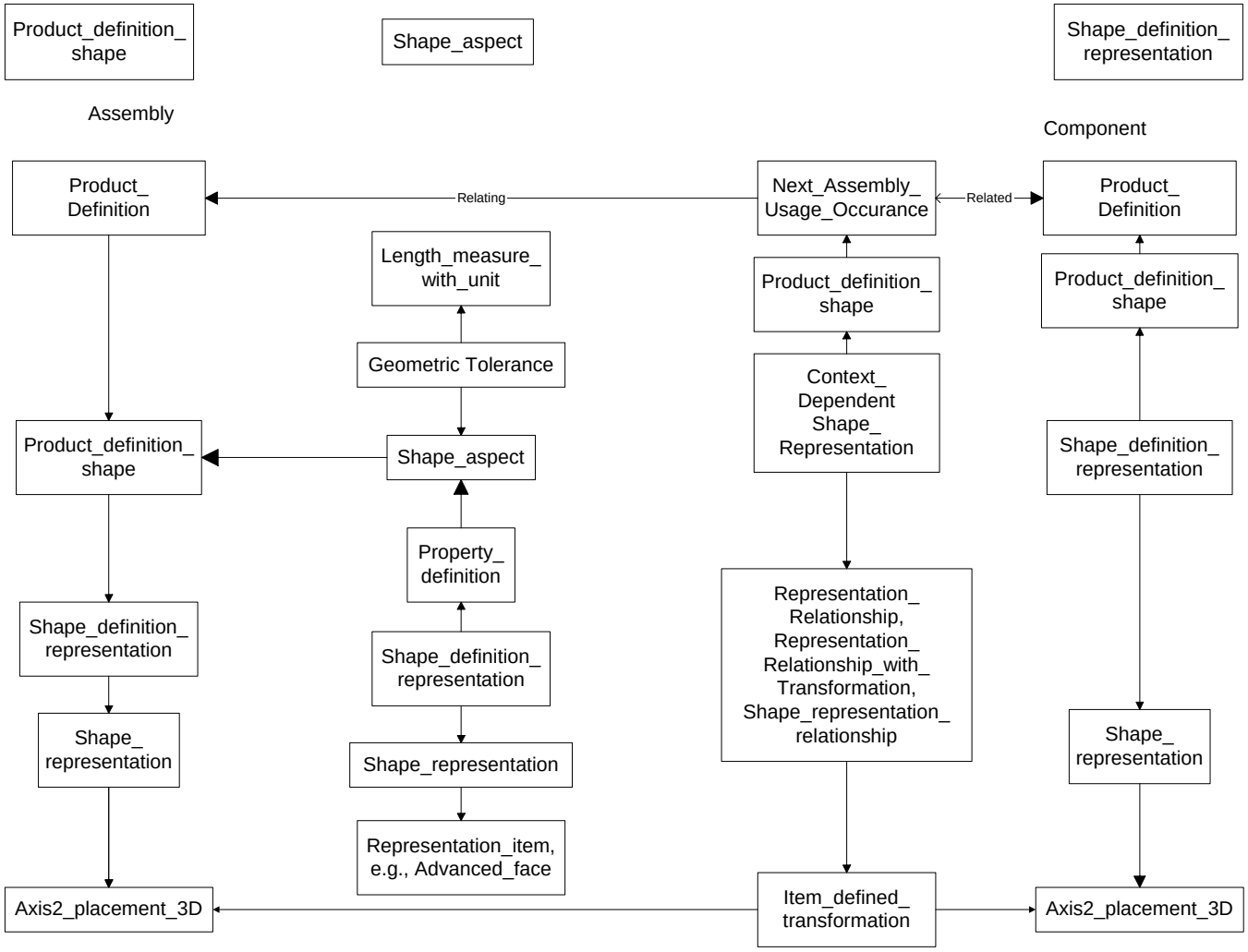


Figure 11 Tolerance on Assembly Feature

3.8 Choice of dimensioning standards

Different dimensioning standards define the visual representation and interpretation of the symbology of the tolerances. In order for the receiving system to create the correct visual representation of the tolerance, the dimensioning standard must be known. This information is captured as an applied_document_reference which applies the referenced document, i.e., the dimensioning standard, to the product_definition of the part. This is shown in Figure 12.

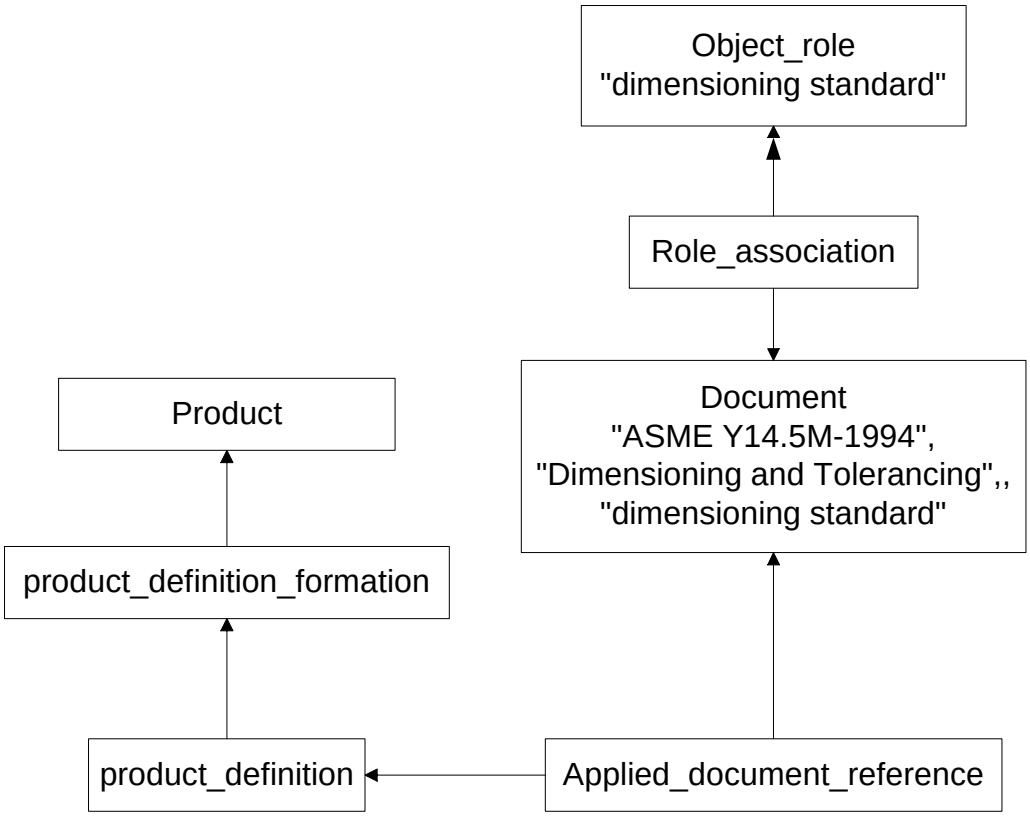


Figure 12 Choice of dimensioning standard

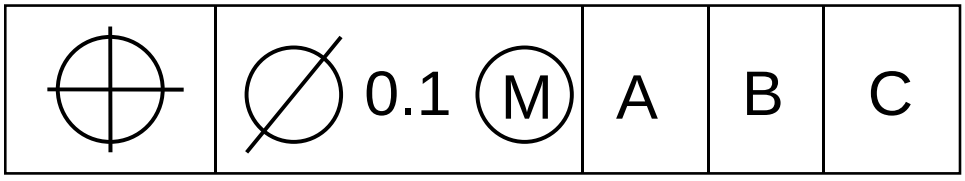
The following standard strings should be used for the document identifier:

“ASME Y14.41-2003”
“ASME Y14.5M-1994”
“ISO 5459-1981”

Per the tolerancing harmonization project, the object role for the document has the string value of "dimensioning standard".

4 Part 21 Example of Typical Tolerance

Consider an example of a location tolerance on a hole whose visual representation as a control frame is:



Interpreting this in STEP (reading the control frame left to right) the location tolerance is a position_tolerance, locating a cylindrical feature, with a tolerance value of 0.1, with a modifier condition of maximum material condition, referencing datums A, B and C in that order.

Using the AIC 519 schema, this would be a complex entity instance comprising a

geometric_tolerance AND
position_tolerance AND
modified_geometric_tolerance AND
geometric_tolerance_with_datum_reference

In the Part 21 file, this would appear as:

```
#1000=(GEOMETRIC_TOLERANCE ('position', 'Positional tolerance for hole 1', 1045, #1008)
        GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE ((#1002, #1003, #1004))
        MODIFIED_GEOMETRIC_TOLERANCE (.MAXIMUM_MATERIAL_CONDITION.)
        POSITIONAL_TOLERANCE ());
```

The hole feature would be defined by:

```
#1008=SHAPE_ASPECT('Hole1', 'One inch top hole', #40, .T.);
#1018=PROPERTY_DEFINITION('Feature shape','Shape of feature',#1008);
#1009=SHAPE_DEFINITION_REPRESENTATION(#1018,#1001) ;
#1001=SHAPE_REPRESENTATION('Representation of hole',(#93),#49) ;
```

where #93 is the ADVANCED_FACE for the inside surface of the hole.

The datum references A, B and C would be referenced by:

```
#1002=DATUM_REFERENCE(1,#1005);
#1003=DATUM_REFERENCE(2,#1006);
#1004=DATUM_REFERENCE(3,#1007);

The DATUM entities are defined and related to the model geometry via:

#1005=DATUM('Datum A','Datum A',#40,.F.,'A');
#1058=PROPERTY_DEFINITION('Datum geometry','Geometric plane of datum',#1005);
#1059=SHAPE_DEFINITION_REPRESENTATION(#1058,#1061) ;
#1061=SHAPE_REPRESENTATION('Representation of datum plane',(#98),#49) ;

#1006=DATUM('Datum B','Datum B',#40,.F.,'B');
#1068=PROPERTY_DEFINITION('Datum geometry','Geometric plane of datum',#1006);
#1069=SHAPE_DEFINITION_REPRESENTATION(#1068,#1071) ;
#1071=SHAPE_REPRESENTATION('Representation of datum plane',(#214),#49) ;

#1007=DATUM('Datum C','Datum C',#40,.F.,'C');
#1078=PROPERTY_DEFINITION('Datum geometry','Geometric plane of datum',#1007);
#1079=SHAPE_DEFINITION_REPRESENTATION(#1078,#1081) ;
#1081=SHAPE_REPRESENTATION('Representation of datum plane',(#190),#49) ;
```

```
where #98, #214 and #190 are the plane entities representing the datum planes in
the B-rep solid model.
```

5 Usage

This section contains the recommendation on population of the values for the tolerance entities.

Since the geometric_tolerance entity is the supertype for the tolerance entities, most of the attributes are inherited from it. The recommended values for the geometric_tolerance entity are:

Name	Explanation
Geometric_tolerance	
name	One of the following strings: angularity circular runout coaxiality concentricity cylindricity flatness parallelism perpendicularity position linear profile surface profile roundness straightness symmetry total runout
description	No requirement defined – instantiate this attribute with an empty string.
magnitude	The measure_with_unit that contains the tolerance value..
toleranced_shape_aspect	The shape_aspect the tolerance is associated to. The type of shape_aspect varies depending on the type of tolerance.

5.1 Angularity

The tolerance is related to the shape_aspect referencing the advanced_face whose underlying geometry is planar that is being toleranced to the datum.

5.2 Circular runout

The tolerance is related to the shape_aspect referencing the advanced_face(s) whose underlying geometry are the cylindrical_surface(s) that are being toleranced.

5.3 Circularity

See Roundness, Section 5.13

5.4 Coaxiality

This is the ISO term for concentricity. See 5.5.

5.5 Concentricity

The tolerance is related to the shape_aspect referencing the advanced_face(s) whose underlying geometry are the cylindrical_surface(s) that are being toleranced.

5.6 Cylindricity

The tolerance is related to the shape_aspect referencing the advanced_face(s) whose underlying geometry are the cylindrical_surface(s) that are being toleranced.

5.7 Flatness

The tolerance is related to the shape_aspect referencing the advanced_face whose underlying geometry is planar that is being toleranced to the datum.

5.8 Parallelism

The tolerance is related to the shape_aspect referencing the advanced_face whose underlying geometry is planar that is being toleranced to the datum.

5.9 Perpendicularity

The tolerance is related to the shape_aspect referencing the advanced_face whose underlying geometry is planar that is being toleranced to the datum.

5.10 Position

Position tolerances can be applied to several types of features. When the feature is a hole, the tolerance is related to the shape_aspect referencing the advanced_face(s) whose underlying geometry is the cylindrical_surface that is being toleranced. For other features, such as tabs, the tolerance is related to the composite_shape_aspect that defines the feature.

5.11 Profile of a line

The tolerance is applied to a shape_aspect that is related to another shape_aspect containing a plane or an intersection_curve. The two shape_aspect entities are related via a shape_aspect_relationship whose name attribute is 'affected plane association', when the related shape-aspect corresponds to the reference plane, or 'resulting intersection curve association', when the related shape-aspect corresponds to one of the intersection curves.

5.12 Profile of a surface

The tolerance is applied to a shape_aspect that references an advanced_face whose underlying geometry is planar that is being toleranced.

5.13 Roundness

The tolerance is related to the shape_aspect referencing the advanced_face whose underlying geometry is the cylindrical_surface that is being toleranced. Roundness is the ISO term for what ASME Y14.5 refers to as circularity.

5.14 Straightness

The tolerance is applied to a shape_aspect that references an advanced_face whose underlying geometry is planar that is being toleranced.

5.15 Symmetry

Where the tolerance is applied to a cylinder, the tolerance entity is related to a shape_aspect referencing the advanced_face(s) whose underlying geometry is the cylindrical_surface. Where the tolerance is applied to opposing faces, the tolerance entity is related to a shape_aspect referencing the advanced_faces whose underlying geometry are the opposing planar faces.

5.16 Total runout

The tolerance is applied to a shape_aspect that references an advanced_face whose underlying geometry is the surface that is being toleranced. The surface may be cylindrical or planar.

6 Entity Definitions

```
ENTITY angularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) < 3;
END_ENTITY;

ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 2;
END_ENTITY;

ENTITY coaxiality_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 2;
END_ENTITY;

ENTITY common_datum
  SUBTYPE OF (composite_shape_aspect, datum);
WHERE
  WR1: SIZEOF (SELF.component_relationships) = 2;
  WR2: SIZEOF ( QUERY ( sar <* SELF.component_relationships| NOT
(('AIC_GEOMETRIC_TOLERANCES.DATUM' IN TYPEOF (sar.related_shape_aspect)) AND NOT
('AIC_GEOMETRIC_TOLERANCES.COMMON_DATUM' IN TYPEOF (sar.related_shape_aspect)))
)) = 0;
END_ENTITY;

ENTITY concentricity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) = 1;
END_ENTITY;

ENTITY cylindricity_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: NOT ('AIC_GEOMETRIC_TOLERANCES.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF));
END_ENTITY;

ENTITY flatness_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
```

```
  WR1: NOT ('AIC_GEOMETRIC_TOLERANCES.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF));
END_ENTITY;

ENTITY geometric_tolerance;
  name : label;
  description : text;
  magnitude : measure_with_unit;
  toleranced_shape_aspect : shape_aspect;
WHERE
  WR1: ('NUMBER' IN TYPEOF (magnitude\measure_with_unit.value_component)) AND
(magnitude\measure_with_unit.value_component >= 0.0);
END_ENTITY;
ENTITY line_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: NOT ('AIC_GEOMETRIC_TOLERANCES.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF)) OR ( SIZEOF
(SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
  WR2: SIZEOF ( QUERY ( sar <* USEDIN (SELF\geometric_tolerance.toleranced_shape_aspect,
'AIC_GEOMETRIC_TOLERANCES.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')| (sar.name IN [ 'affected plane
association', 'resulting intersection curve association' ]) )) = 1;
END_ENTITY;

ENTITY parallelism_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) < 3;
END_ENTITY;

ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3;
END_ENTITY;

ENTITY position_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: NOT ('AIC_GEOMETRIC_TOLERANCES.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF)) OR ( SIZEOF
(SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
END_ENTITY;

ENTITY roundness_tolerance
```

```
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: NOT ('AIC_GEOMETRIC_TOLERANCES.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF));
END_ENTITY;

ENTITY straightness_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: NOT ('AIC_GEOMETRIC_TOLERANCES.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF));
END_ENTITY;

ENTITY surface_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
WHERE
  WR1: NOT ('AIC_GEOMETRIC_TOLERANCES.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF)) OR ( SIZEOF
(SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
END_ENTITY;

ENTITY symmetry_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3;
END_ENTITY;

ENTITY total_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  WR1: SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 2;
END_ENTITY;
```

7 Sample Part 21 File

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('CATIA V5 STEP Exchange'),'2;1');

FILE_NAME('PatternTol.stp','2004-06-15T21:29:40+00:00',('none'),('none'),'CATIA
Version 5 Release 13 (IN-9)','CATIA V5 STEP AP203','none');

FILE_SCHEMA(('Configuration_control_3d_design_ed2_mim_lf'));

ENDSEC;
DATA;
```

```
#1=APPLICATION_CONTEXT('configuration controlled 3D design of mechanical parts and
assemblies') ;
#2=MECHANICAL_CONTEXT(' ',#1,'mechanical') ;
#3=DESIGN_CONTEXT(' ',#1,'design') ;
#4=APPLICATION_PROTOCOL_DEFINITION('draft international
standard','Configuration_control_3d_design_ed2_mim_lf',2004,#1) ;
#5=PRODUCT('Pos Tol','','',(#2)) ;
#7=PRODUCT_CATEGORY('part',$) ;
#9=PRODUCT_CATEGORY_RELATIONSHIP(' ',' ',#7,#8) ;
#6=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE('',' ',#5,.NOT_KNOWN.) ;
#8=PRODUCT_RELATED_PRODUCT_CATEGORY('detail',$,(#5)) ;
#10=COORDINATED_UNIVERSAL_TIME_OFFSET(0,0,.AHEAD.) ;
#11=CALENDAR_DATE(2004,15,6) ;
#12=LOCAL_TIME(14,29,39.,#10) ;
#13=DATE_AND_TIME(#11,#12) ;
#14=PRODUCT_DEFINITION(' ',' ',#6,#3) ;
#16=SECURITY_CLASSIFICATION(' ','Because',#15) ;
#15=SECURITY_CLASSIFICATION_LEVEL('unclassified') ;
#17=DATE_TIME_ROLE('classification_date') ;
#18=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#13,#17,(#16)) ;
#19=APPROVAL_ROLE('APPROVER') ;
#20=APPROVAL_STATUS('not_yet_approved') ;
#21=APPROVAL(#20,' ') ;
#22=PERSON('Pers1 ','Step ','Joe ',$,$,$) ;
#23=ORGANIZATION('Org1 ','Generic Airplane Company ','Yet another jet manufacturer
') ;
#24=PERSONAL_ADDRESS('Mail Stop 1 ','123 ','Sesame Street','PO Box
101','Anytown','Central','98124','USA','206-555-1234','206-555-
5678','help@company.com','206-555-9876',(#22),' ') ;
#25=PERSON_AND_ORGANIZATION(#22,#23) ;
#27=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#26,(#16)) ;
#28=DATE_TIME_ROLE('creation_date') ;
#29=APPLIED_DATE_AND_TIME_ASSIGNMENT(#13,#28,(#14)) ;
#30=APPLIED_APPROVAL_ASSIGNMENT(#21,(#6,#14)) ;
#32=APPROVAL_DATE_TIME(#13,#21) ;
#33=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#34,(#6)) ;
#35=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#36,(#6,#14)) ;
#37=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#25,#38,(#5)) ;
#26=PERSON_AND_ORGANIZATION_ROLE('classification_officer') ;
#34=PERSON_AND_ORGANIZATION_ROLE('design_supplier') ;
#36=PERSON_AND_ORGANIZATION_ROLE('creator') ;
#38=PERSON_AND_ORGANIZATION_ROLE('design_owner') ;
#39=APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT(#16,(#6)) ;
#31=APPROVAL_PERSON_ORGANIZATION(#25,#21,#19) ;
#40=PRODUCT_DEFINITION_SHAPE(' ',' ',#14) ;
#41=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.)) ;
#42=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.)) ;
#44=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT()) ;
#48=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.000196850393701),#47,'TOL_CRV','
CONFUSED CURVE UNCERTAINTY') ;
#45=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(25.4),#41) ;
#47=(CONVERSION_BASED_UNIT('INCH',#45)LENGTH_UNIT()NAMED_UNIT(#46)) ;
#49=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#48))G
LOBAL_UNIT_ASSIGNED_CONTEXT((#47,#42,#44))REPRESENTATION_CONTEXT(' ',' ')) ;

/* Geometry of Hole 1 */
#55=CARTESIAN_POINT('Axis2P3D Location',(1.,1.,0.99606299213)) ;
```

```
#56=DIRECTION('Axis2P3D Direction',(0.,0.,-0.0393700787402)) ;
#57=DIRECTION('Axis2P3D XDirection',(0.0345504945626,-0.0188750212049,0.)) ;
#58=AXIS2_PLACEMENT_3D('Cylinder Axis2P3D',#55,#56,#57) ;
#59=CYLINDRICAL_SURFACE('generated cylinder',#58,0.500000000002) ;

/* Geometry of Hole 2 */

#270=CARTESIAN_POINT('Axis2P3D
Location',(3.02048023855,0.944704671544,0.99606299213));
#271=DIRECTION('Axis2P3D Direction',(0.,0.,-0.0393700787402)) ;
#272=DIRECTION('Axis2P3D XDirection',(0.0345504945626,-0.0188750212049,0.)) ;
#273=AXIS2_PLACEMENT_3D('Cylinder Axis2P3D',#270,#271,#272) ;
#274=CYLINDRICAL_SURFACE('generated cylinder',#273,0.500000000002) ;


#50=CARTESIAN_POINT(' ',(0.,0.,0.)) ;

#60=CARTESIAN_POINT('Line Origine',(0.561208719057,1.23971276931,0.500000000002)) ;
#64=CARTESIAN_POINT('Vertex',(0.561208719057,1.23971276931,0.)) ;
#66=CARTESIAN_POINT('Vertex',(0.561208719057,1.23971276931,1.)) ;
#69=CARTESIAN_POINT('Axis2P3D Location',(1.,1.,0.)) ;
#73=CARTESIAN_POINT('Vertex',(1.43879128095,0.760287230701,0.)) ;
#76=CARTESIAN_POINT('Line Origine',(1.43879128095,0.760287230701,0.500000000002)) ;
#80=CARTESIAN_POINT('Vertex',(1.43879128095,0.760287230701,1.)) ;
#83=CARTESIAN_POINT('Axis2P3D Location',(1.,1.,1.)) ;
#95=CARTESIAN_POINT('Axis2P3D Location',(0.,0.,1.)) ;
#100=CARTESIAN_POINT('Line Origine',(2.00000000001,0.,1.)) ;
#104=CARTESIAN_POINT('Vertex',(0.,0.,1.)) ;
#106=CARTESIAN_POINT('Vertex',(4.00000000002,0.,1.)) ;
#109=CARTESIAN_POINT('Line Origine',(4.00000000002,1.,1.)) ;
#113=CARTESIAN_POINT('Vertex',(4.00000000002,2.00000000001,1.)) ;
#116=CARTESIAN_POINT('Line Origine',(2.00000000001,2.00000000001,1.)) ;
#120=CARTESIAN_POINT('Vertex',(-1.1189649382E-015,2.00000000001,1.)) ;
#123=CARTESIAN_POINT('Line Origine',(0.,1.,1.)) ;
#134=CARTESIAN_POINT('Axis2P3D Location',(3.02048023855,0.944704671544,1.)) ;
#138=CARTESIAN_POINT('Vertex',(2.5816889576,1.18441744085,1.)) ;
#140=CARTESIAN_POINT('Vertex',(3.45927151949,0.704991902241,1.)) ;
#143=CARTESIAN_POINT('Axis2P3D Location',(3.02048023855,0.944704671544,1.)) ;
#152=CARTESIAN_POINT('Axis2P3D Location',(1.,1.,1.)) ;
#162=CARTESIAN_POINT('Axis2P3D Location',(1.,1.,0.)) ;
#174=CARTESIAN_POINT('Axis2P3D Location',(0.,0.,0.)) ;
#179=CARTESIAN_POINT('Line Origine',(0.,0.,0.500000000002)) ;
#183=CARTESIAN_POINT('Vertex',(0.,0.,0.)) ;
#186=CARTESIAN_POINT('Line Origine',(2.00000000001,0.,0.)) ;
#190=CARTESIAN_POINT('Vertex',(4.00000000002,0.,0.)) ;
#193=CARTESIAN_POINT('Line Origine',(4.00000000002,0.,0.500000000002)) ;
#205=CARTESIAN_POINT('Axis2P3D Location',(4.00000000002,-9.1845476537E-016,0.)) ;
#210=CARTESIAN_POINT('Line Origine',(4.00000000002,1.,0.)) ;
#214=CARTESIAN_POINT('Vertex',(4.00000000002,2.00000000001,0.)) ;
#217=CARTESIAN_POINT('Line Origine',(4.00000000002,2.00000000001,0.500000000002)) ;
#229=CARTESIAN_POINT('Axis2P3D Location',(4.00000000002,2.00000000001,0.)) ;

#234=CARTESIAN_POINT('Line Origine',(2.00000000001,2.00000000001,0.)) ;
#238=CARTESIAN_POINT('Vertex',(-1.1189649382E-015,2.00000000001,0.)) ;
#241=CARTESIAN_POINT('Line Origine',(-1.1189649382E-
015,2.00000000001,0.500000000002)) ;
#253=CARTESIAN_POINT('Axis2P3D Location',(0.,2.00000000001,0.)) ;
```

```
#258=CARTESIAN_POINT('Line Origine',(0.,1.,0.)) ;

#275=CARTESIAN_POINT('Line Origine',(2.5816889576,1.18441744085,0.499999999971)) ;
#279=CARTESIAN_POINT('Vertex',(2.5816889576,1.18441744085,0.)) ;
#282=CARTESIAN_POINT('Axis2P3D Location',(3.02048023855,0.944704671544,0.)) ;
#286=CARTESIAN_POINT('Vertex',(3.45927151949,0.704991902241,0.)) ;
#289=CARTESIAN_POINT('Line Origine',(3.45927151949,0.704991902241,0.499999999971))
;
#301=CARTESIAN_POINT('Axis2P3D Location',(3.02048023855,0.944704671544,0.)) ;
#313=CARTESIAN_POINT('Axis2P3D Location',(0.,0.,0.)) ;

#61=DIRECTION('Vector Direction',(0.,0.,-0.0393700787402)) ;
#70=DIRECTION('Axis2P3D Direction',(0.,0.,-0.0393700787402)) ;
#77=DIRECTION('Vector Direction',(0.,0.,-0.0393700787402)) ;
#84=DIRECTION('Axis2P3D Direction',(0.,0.,-0.0393700787402)) ;
#96=DIRECTION('Axis2P3D Direction',(0.,0.,0.0393700787402)) ;
#97=DIRECTION('Axis2P3D XDirection',(0.0393700787402,0.,0.)) ;
#101=DIRECTION('Vector Direction',(0.0393700787402,0.,0.)) ;
#110=DIRECTION('Vector Direction',(0.,0.0393700787402,0.)) ;
#117=DIRECTION('Vector Direction',(-0.0393700787402,0.,0.)) ;
#124=DIRECTION('Vector Direction',(0.,-0.0393700787402,0.)) ;
#135=DIRECTION('Axis2P3D Direction',(0.,0.,0.0393700787402)) ;
#144=DIRECTION('Axis2P3D Direction',(0.,0.,0.0393700787402)) ;
#153=DIRECTION('Axis2P3D Direction',(0.,0.,0.0393700787402)) ;
#163=DIRECTION('Axis2P3D Direction',(0.,0.,-0.0393700787402)) ;
#175=DIRECTION('Axis2P3D Direction',(0.,-0.0393700787402,0.)) ;
#176=DIRECTION('Axis2P3D XDirection',(0.0393700787402,0.,0.)) ;
#180=DIRECTION('Vector Direction',(0.,0.,0.0393700787402)) ;
#187=DIRECTION('Vector Direction',(0.0393700787402,0.,0.)) ;
#194=DIRECTION('Vector Direction',(0.,0.,0.0393700787402)) ;
#206=DIRECTION('Axis2P3D Direction',(0.0393700787402,0.,0.)) ;
#207=DIRECTION('Axis2P3D XDirection',(0.,0.0393700787402,0.)) ;
#211=DIRECTION('Vector Direction',(0.,0.0393700787402,0.)) ;
#218=DIRECTION('Vector Direction',(0.,0.,0.0393700787402)) ;
#230=DIRECTION('Axis2P3D Direction',(0.,0.0393700787402,0.)) ;
#231=DIRECTION('Axis2P3D XDirection',(-0.0393700787402,0.,0.)) ;
#235=DIRECTION('Vector Direction',(-0.0393700787402,0.,0.)) ;
#242=DIRECTION('Vector Direction',(0.,0.,0.0393700787402)) ;
#254=DIRECTION('Axis2P3D Direction',(-0.0393700787402,0.,0.)) ;
#255=DIRECTION('Axis2P3D XDirection',(0.,-0.0393700787402,0.)) ;
#259=DIRECTION('Vector Direction',(0.,-0.0393700787402,0.)) ;

#276=DIRECTION('Vector Direction',(0.,0.,-0.0393700787402)) ;
#283=DIRECTION('Axis2P3D Direction',(0.,0.,-0.0393700787402)) ;
#290=DIRECTION('Vector Direction',(0.,0.,-0.0393700787402)) ;
#302=DIRECTION('Axis2P3D Direction',(0.,0.,-0.0393700787402)) ;
#314=DIRECTION('Axis2P3D Direction',(0.,0.,0.0393700787402)) ;
#315=DIRECTION('Axis2P3D XDirection',(0.0393700787402,0.,0.)) ;
#51=AXIS2_PLACEMENT_3D(' ',#50,$,$) ;

#71=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#69,#70,$) ;
#85=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#83,#84,$) ;
#98=AXIS2_PLACEMENT_3D('Plane Axis2P3D',#95,#96,#97) ;
#136=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#134,#135,$) ;
#145=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#143,#144,$) ;
#154=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#152,#153,$) ;
#164=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#162,#163,$) ;
```

```
#177=AXIS2_PLACEMENT_3D('Plane Axis2P3D',#174,#175,#176) ;
#208=AXIS2_PLACEMENT_3D('Plane Axis2P3D',#205,#206,#207) ;
#232=AXIS2_PLACEMENT_3D('Plane Axis2P3D',#229,#230,#231) ;
#256=AXIS2_PLACEMENT_3D('Plane Axis2P3D',#253,#254,#255) ;

#284=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#282,#283,$) ;
#303=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#301,#302,$) ;
#316=AXIS2_PLACEMENT_3D('Plane Axis2P3D',#313,#314,#315) ;
#89=ORIENTED_EDGE('',*,*,#68,.F.) ;
#90=ORIENTED_EDGE('',*,*,#75,.F.) ;
#91=ORIENTED_EDGE('',*,*,#82,.T.) ;
#92=ORIENTED_EDGE('',*,*,#87,.T.) ;
#129=ORIENTED_EDGE('',*,*,#108,.T.) ;
#130=ORIENTED_EDGE('',*,*,#115,.T.) ;
#131=ORIENTED_EDGE('',*,*,#122,.T.) ;
#132=ORIENTED_EDGE('',*,*,#127,.T.) ;
#149=ORIENTED_EDGE('',*,*,#142,.F.) ;
#150=ORIENTED_EDGE('',*,*,#147,.F.) ;
#158=ORIENTED_EDGE('',*,*,#156,.F.) ;
#159=ORIENTED_EDGE('',*,*,#87,.F.) ;
#168=ORIENTED_EDGE('',*,*,#82,.F.) ;
#169=ORIENTED_EDGE('',*,*,#166,.F.) ;
#170=ORIENTED_EDGE('',*,*,#68,.T.) ;
#171=ORIENTED_EDGE('',*,*,#156,.T.) ;
#199=ORIENTED_EDGE('',*,*,#185,.F.) ;
#200=ORIENTED_EDGE('',*,*,#192,.T.) ;
#201=ORIENTED_EDGE('',*,*,#197,.T.) ;
#202=ORIENTED_EDGE('',*,*,#108,.F.) ;
#223=ORIENTED_EDGE('',*,*,#197,.F.) ;
#224=ORIENTED_EDGE('',*,*,#216,.T.) ;
#225=ORIENTED_EDGE('',*,*,#221,.T.) ;
#226=ORIENTED_EDGE('',*,*,#115,.F.) ;
#247=ORIENTED_EDGE('',*,*,#221,.F.) ;
#248=ORIENTED_EDGE('',*,*,#240,.T.) ;
#249=ORIENTED_EDGE('',*,*,#245,.T.) ;
#250=ORIENTED_EDGE('',*,*,#122,.F.) ;
#264=ORIENTED_EDGE('',*,*,#245,.F.) ;
#265=ORIENTED_EDGE('',*,*,#262,.T.) ;
#266=ORIENTED_EDGE('',*,*,#185,.T.) ;
#267=ORIENTED_EDGE('',*,*,#127,.F.) ;
#295=ORIENTED_EDGE('',*,*,#281,.F.) ;
#296=ORIENTED_EDGE('',*,*,#288,.F.) ;
#297=ORIENTED_EDGE('',*,*,#293,.T.) ;
#298=ORIENTED_EDGE('',*,*,#147,.T.) ;
#307=ORIENTED_EDGE('',*,*,#293,.F.) ;
#308=ORIENTED_EDGE('',*,*,#305,.F.) ;
#309=ORIENTED_EDGE('',*,*,#281,.T.) ;
#310=ORIENTED_EDGE('',*,*,#142,.T.) ;
#319=ORIENTED_EDGE('',*,*,#262,.F.) ;
#320=ORIENTED_EDGE('',*,*,#240,.F.) ;
#321=ORIENTED_EDGE('',*,*,#216,.F.) ;
#322=ORIENTED_EDGE('',*,*,#192,.F.) ;
#325=ORIENTED_EDGE('',*,*,#288,.T.) ;
#326=ORIENTED_EDGE('',*,*,#305,.T.) ;
#329=ORIENTED_EDGE('',*,*,#75,.T.) ;
#330=ORIENTED_EDGE('',*,*,#166,.T.) ;
#151=FACE_BOUND('',#148,.T.) ;
```

```
#160=FACE_BOUND('',#157,.T.) ;
#327=FACE_BOUND('',#324,.T.) ;
#331=FACE_BOUND('',#328,.T.) ;
#54=CLOSED_SHELL('Closed Shell',(#94,#161,#173,#204,#228,#252,#269,#300,#312,#332))
;
#46=DIMENSIONAL_EXPONENTS(1.,0.,0.,0.,0.,0.,0.) ;
#62=VECTOR('Line Direction',#61,0.0393700787402) ;
#78=VECTOR('Line Direction',#77,0.0393700787402) ;
#102=VECTOR('Line Direction',#101,0.0393700787402) ;
#111=VECTOR('Line Direction',#110,0.0393700787402) ;
#118=VECTOR('Line Direction',#117,0.0393700787402) ;
#125=VECTOR('Line Direction',#124,0.0393700787402) ;
#181=VECTOR('Line Direction',#180,0.0393700787402) ;
#188=VECTOR('Line Direction',#187,0.0393700787402) ;
#195=VECTOR('Line Direction',#194,0.0393700787402) ;
#212=VECTOR('Line Direction',#211,0.0393700787402) ;
#219=VECTOR('Line Direction',#218,0.0393700787402) ;
#236=VECTOR('Line Direction',#235,0.0393700787402) ;
#243=VECTOR('Line Direction',#242,0.0393700787402) ;
#260=VECTOR('Line Direction',#259,0.0393700787402) ;
#277=VECTOR('Line Direction',#276,0.0393700787402) ;
#291=VECTOR('Line Direction',#290,0.0393700787402) ;
#334=ADVANCED_BREP_SHAPE_REPRESENTATION('NONE',(#333),#49) ;
#52=SHAPE_REPRESENTATION(' ',(#51),#49) ;
#94=ADVANCED_FACE('',(#93),#59,.F.) ;
#161=ADVANCED_FACE('',(#133,#151,#160),#99,.T.) ;
#173=ADVANCED_FACE('',(#172),#59,.F.) ;
#204=ADVANCED_FACE('',(#203),#178,.T.) ;
#228=ADVANCED_FACE('',(#227),#209,.T.) ;
#252=ADVANCED_FACE('',(#251),#233,.T.) ;
#269=ADVANCED_FACE('',(#268),#257,.T.) ;
#300=ADVANCED_FACE('',(#299),#274,.F.) ;
#312=ADVANCED_FACE('',(#311),#274,.F.) ;
#332=ADVANCED_FACE('',(#323,#327,#331),#317,.F.) ;
#333=MANIFOLD_SOLID_BREP('PartBody',#54) ;
#72=CIRCLE('generated circle',#71,0.500000000002) ;
#86=CIRCLE('generated circle',#85,0.500000000002) ;
#137=CIRCLE('generated circle',#136,0.500000000002) ;
#146=CIRCLE('generated circle',#145,0.500000000002) ;
#155=CIRCLE('generated circle',#154,0.500000000002) ;
#165=CIRCLE('generated circle',#164,0.500000000002) ;
#285=CIRCLE('generated circle',#284,0.500000000002) ;
#304=CIRCLE('generated circle',#303,0.500000000002) ;
#335=SHAPE_REPRESENTATION_RELATIONSHIP(' ',' ',#52,#334) ;


#68=EDGE_CURVE('',#65,#67,#63,.F.) ;
#75=EDGE_CURVE('',#74,#65,#72,.F.) ;
#82=EDGE_CURVE('',#74,#81,#79,.F.) ;
#87=EDGE_CURVE('',#81,#67,#86,.F.) ;
#108=EDGE_CURVE('',#105,#107,#103,.T.) ;
#115=EDGE_CURVE('',#107,#114,#112,.T.) ;
#122=EDGE_CURVE('',#114,#121,#119,.T.) ;
#127=EDGE_CURVE('',#121,#105,#126,.T.) ;
#142=EDGE_CURVE('',#139,#141,#137,.T.) ;
#147=EDGE_CURVE('',#141,#139,#146,.T.) ;
#156=EDGE_CURVE('',#67,#81,#155,.T.) ;
```

```
#166=EDGE_CURVE('',#65,#74,#165,.F.) ;
#185=EDGE_CURVE('',#184,#105,#182,.T.) ;
#192=EDGE_CURVE('',#184,#191,#189,.T.) ;
#197=EDGE_CURVE('',#191,#107,#196,.T.) ;
#216=EDGE_CURVE('',#191,#215,#213,.T.) ;
#221=EDGE_CURVE('',#215,#114,#220,.T.) ;
#240=EDGE_CURVE('',#215,#239,#237,.T.) ;
#245=EDGE_CURVE('',#239,#121,#244,.T.) ;
#262=EDGE_CURVE('',#239,#184,#261,.T.) ;
#281=EDGE_CURVE('',#280,#139,#278,.F.) ;
#288=EDGE_CURVE('',#287,#280,#285,.F.) ;
#293=EDGE_CURVE('',#287,#141,#292,.F.) ;
#305=EDGE_CURVE('',#280,#287,#304,.F.) ;
#88=EDGE_LOOP('',(#89,#90,#91,#92)) ;
#128=EDGE_LOOP('',(#129,#130,#131,#132)) ;
#148=EDGE_LOOP('',(#149,#150)) ;
#157=EDGE_LOOP('',(#158,#159)) ;
#167=EDGE_LOOP('',(#168,#169,#170,#171)) ;
#198=EDGE_LOOP('',(#199,#200,#201,#202)) ;
#222=EDGE_LOOP('',(#223,#224,#225,#226)) ;
#246=EDGE_LOOP('',(#247,#248,#249,#250)) ;
#263=EDGE_LOOP('',(#264,#265,#266,#267)) ;
#294=EDGE_LOOP('',(#295,#296,#297,#298)) ;
#306=EDGE_LOOP('',(#307,#308,#309,#310)) ;
#318=EDGE_LOOP('',(#319,#320,#321,#322)) ;
#324=EDGE_LOOP('',(#325,#326)) ;
#328=EDGE_LOOP('',(#329,#330)) ;
#93=FACE_OUTER_BOUND('',#88,.T.) ;
#133=FACE_OUTER_BOUND('',#128,.T.) ;
#172=FACE_OUTER_BOUND('',#167,.T.) ;
#203=FACE_OUTER_BOUND('',#198,.T.) ;
#227=FACE_OUTER_BOUND('',#222,.T.) ;
#251=FACE_OUTER_BOUND('',#246,.T.) ;
#268=FACE_OUTER_BOUND('',#263,.T.) ;
#299=FACE_OUTER_BOUND('',#294,.T.) ;
#311=FACE_OUTER_BOUND('',#306,.T.) ;
#323=FACE_OUTER_BOUND('',#318,.T.) ;
#63=LINE('Line',#60,#62) ;
#79=LINE('Line',#76,#78) ;
#103=LINE('Line',#100,#102) ;
#112=LINE('Line',#109,#111) ;
#119=LINE('Line',#116,#118) ;
#126=LINE('Line',#123,#125) ;
#182=LINE('Line',#179,#181) ;
#189=LINE('Line',#186,#188) ;
#196=LINE('Line',#193,#195) ;
#213=LINE('Line',#210,#212) ;
#220=LINE('Line',#217,#219) ;
#237=LINE('Line',#234,#236) ;
#244=LINE('Line',#241,#243) ;
#261=LINE('Line',#258,#260) ;
#278=LINE('Line',#275,#277) ;
#292=LINE('Line',#289,#291) ;
#99=PLANE('Plane',#98) ;
#178=PLANE('Plane',#177) ;
#209=PLANE('Plane',#208) ;
#233=PLANE('Plane',#232) ;
```

```
#257=PLANE('Plane',#256) ;
#317=PLANE('Plane',#316) ;
#43=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.0174532925199),#42) ;
#53=SHAPE_DEFINITION_REPRESENTATION(#40,#52) ;
#65=VERTEX_POINT('',#64) ;
#67=VERTEX_POINT('',#66) ;
#74=VERTEX_POINT('',#73) ;
#81=VERTEX_POINT('',#80) ;
#105=VERTEX_POINT('',#104) ;
#107=VERTEX_POINT('',#106) ;
#114=VERTEX_POINT('',#113) ;
#121=VERTEX_POINT('',#120) ;
#139=VERTEX_POINT('',#138) ;
#141=VERTEX_POINT('',#140) ;
#184=VERTEX_POINT('',#183) ;
#191=VERTEX_POINT('',#190) ;
#215=VERTEX_POINT('',#214) ;
#239=VERTEX_POINT('',#238) ;
#280=VERTEX_POINT('',#279) ;
#287=VERTEX_POINT('',#286) ;

/* Geometric tolerance entities */

/* Pattern of Position Tolerances */

#1000=(GEOMETRIC_TOLERANCE('Position', 'Pattern of positional tolerances for
holes', #1045, #1060)
       GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#1003, #1004))
       MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERIAL_CONDITION.)
       POSITION_TOLERANCE());

#1045=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.01),#47) ;

#1008=SHAPE_ASPECT('Hole1', 'One inch top hole', #40, .T.);
#1018=PROPERTY_DEFINITION('Feature shape','Shape of feature',#1008);
#1009=SHAPE_DEFINITION_REPRESENTATION(#1018,#1001) ;
#1001=SHAPE_REPRESENTATION('Representation of hole',(#94),#49) ;

#1028=SHAPE_ASPECT('Hole2', 'One inch top hole', #40, .T.);
#1027=PROPERTY_DEFINITION('Feature shape','Shape of feature',#1028);
#1029=SHAPE_DEFINITION_REPRESENTATION(#1027,#1001) ;
#1001=SHAPE_REPRESENTATION('Representation of hole',(#93),#49) ;

#1060=COMPOSITE_SHAPE_ASPECT('Pattern', 'Hole Pattern', #40, .T.,);
#1062=SHAPE_ASPECT_RELATIONSHIP('Pattern composition', 'Composition of Hole
Pattern', #1060, #1008);
#1063=SHAPE_ASPECT_RELATIONSHIP('Pattern composition', 'Composition of Hole
Pattern', #1060, #1028);

/* #1002=DATUM_REFERENCE(1,#1005); */
#1003=DATUM_REFERENCE(2,#1006);
#1004=DATUM_REFERENCE(1,#1007);

/* Datum related to geometric plane of datum */

#1005=DATUM('Datum A','Datum A',#40,.F.,'A');
#1058=PROPERTY_DEFINITION('Datum geometry','Geometric plane of datum',#1005);
```

```
#1059=SHAPE_DEFINITION_REPRESENTATION(#1058,#1061) ;
#1061=SHAPE_REPRESENTATION('Representation of datum plane',(#98),#49) ;

#1006=DATUM('Datum B','Datum B',#40,.F.,'B');
#1068=PROPERTY_DEFINITION('Datum geometry','Geometric plane of datum',#1006);
#1069=SHAPE_DEFINITION_REPRESENTATION(#1068,#1071) ;
#1071=SHAPE_REPRESENTATION('Representation of datum plane',(#214),#49) ;

#1007=DATUM('Datum C','Datum C',#40,.F.,'C');
#1078=PROPERTY_DEFINITION('Datum geometry','Geometric plane of datum',#1007);
#1079=SHAPE_DEFINITION_REPRESENTATION(#1078,#1081) ;
#1081=SHAPE_REPRESENTATION('Representation of datum plane',(#190),#49) ;

/* Datum Feature is topology of actual solid face representing datum plane */

#1010=DATUM_FEATURE('Datum A','Datum A',#40,.T.);
#1038=PROPERTY_DEFINITION('Datum shape','Shape of datum',#1010);
#1039=SHAPE_DEFINITION_REPRESENTATION(#1038,#1040) ;
#1040=SHAPE_REPRESENTATION('Representation of datum feature',(#142),#49) ;

/* #1041=SHAPE_ASPECT('Datum Feature A', 'Face representing datum feature', #1040,
.T.); */


#1011=DATUM_FEATURE('Datum B','Datum B',#40,.T. );
#1048=PROPERTY_DEFINITION('Datum shape','Shape of datum',#1011);
#1049=SHAPE_DEFINITION_REPRESENTATION(#1048,#1050) ;
#1050=SHAPE_REPRESENTATION('Representation of datum feature',(#233),#49) ;

#1012=DATUM_FEATURE('Datum C','Datum C',#40,.T.);
#1098=PROPERTY_DEFINITION('Datum shape','Shape of datum',#1012);
#1099=SHAPE_DEFINITION_REPRESENTATION(#1098,#1100) ;
#1100=SHAPE_REPRESENTATION('Representation of datum feature',(#209),#49);

#1015=SHAPE_ASPECT_RELATIONSHIP('DtoF','Datum to Feature
relationship',#1010,#1005);
#1016=SHAPE_ASPECT_RELATIONSHIP('DtoF','Datum to Feature
relationship',#1011,#1006);
#1017=SHAPE_ASPECT_RELATIONSHIP('DtoF','Datum to Feature
relationship',#1012,#1007);

/* Perpendicularity tolerance */


#2000=PERPENDICULARITY_TOLERANCE('perpendicularity', 'Perpendicularity tolerance
for top face', #2045, #2008,(#2003,#2004)) ;

#2045=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.254),#41) ;

#2008=SHAPE_ASPECT('Top face', 'Top face of part', #40, .T.);
#2018=PROPERTY_DEFINITION('Shape','Shape of Feature',#2008);
#2009=SHAPE_DEFINITION_REPRESENTATION(#2018,#2001) ;
#2001=SHAPE_REPRESENTATION('Representation of face',(#142),#49) ;

#2003=DATUM_REFERENCE(1,#1006);
#2004=DATUM_REFERENCE(2,#1012);
```

```
/*************************************************************************** *
* Instances #3910-#3956 represent the choioce of dimensioning                *
* standards per tolerance harmonization effort.                             *
****************************************************************************/

#3910=APPLIED_DOCUMENT_REFERENCE(#3911, "ASME Y14.41-2003", (#14));

#3911=DOCUMENT("ASME Y14.41-200", "DIGITAL PRODUCT DEFINITION DATA
PRACTICES",,"dimensioning standard");

#3912=OBJECT_ROLE("dimensioning standard",);


#3913=ROLE_ASSOCIATION(#3912, #3910);


ENDSEC;
END-ISO-10303-21;
```

8 AP203 Edition 2 Long Form

```
(*
 $Id: mim_lf.exp,v 1.14 2004/12/03 17:18:07 darla Exp $
 ISO TC184/SC4/WG3 N1587 - ISO/TS 10303-403 Configuration_control_3d_design_ed2 -
EXPRESS MIM
 Supersedes ISO TC184/SC4/WG3 N1202
*)

(*
================================================================================
== *)
(* Long form schema generated by The EXPRESS Data Manager  compiler version
9.5B.2004.01.23*)
(* Wed Dec 01 11:06:08 2004
*)
(* The schema is converted from ISO10303 P11-2003 to ISO10303 P11-1994
*)
(*
================================================================================
== *)

SCHEMA Configuration_control_3d_design_ed2_mim_lf;


CONSTANT

  deprecated_constructed_data_types : SET OF STRING :=
   ['approved_item',
    'certified_item',
    'change_request_item',
    'contracted_item',
    'classified_item',
    'date_time_item',
```

```
    'person_organization_item',
    'specified_item',
    'start_request_item',
    'work_item'];

  deprecated_entity_data_types : SET OF STRING :=
   ['cc_design_approval',
    'cc_design_certification',
    'cc_design_contract',
    'cc_design_date_and_time_assignment',
    'cc_design_person_and_organization_assignment',
    'cc_design_security_classification',
    'cc_design_specification_reference',
    'change',
    'change_request',
    'design_context',
    'design_make_from_relationship',
    'mechanical_context',
    'start_request',
    'start_work',
    'supplied_part_relationship'];

  deprecated_interfaced_data_types : SET OF STRING :=
   ['document_with_class',
    'ordinal_date',
    'product_definition_formation_with_specified_source',
    'week_of_year_and_day_date'];

  dummy_gri : geometric_representation_item :=  representation_item('')||
                                    geometric_representation_item();

    dummy_tri : topological_representation_item := representation_item('')||
                    topological_representation_item();

END_CONSTANT;

(* Pruned unused type: action_item_203e2  *)


TYPE action_items = SELECT (
   configuration_effectivity,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_relationship);
END_TYPE;


TYPE action_request_item = SELECT (
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_relationship,
   property_definition);
END_TYPE;
```

```
(* Pruned unused type: action_request_item_203e2  *)


TYPE ahead_or_behind = ENUMERATION OF
   (ahead,
    exact,
    behind);
END_TYPE;

(* Pruned unused type: aliasable_item  *)


TYPE amount_of_substance_measure = REAL;
END_TYPE;

TYPE angle_relator = ENUMERATION OF
  (equal,
   large,
   small);
END_TYPE;

(* Pruned unused type: appearance_representation_identification_item  *)


(* Pruned unused type: appearance_representation_version_item  *)


TYPE approval_item = SELECT (
   action,
   alternate_product_relationship,
   applied_action_assignment,
   assembly_component_usage_substitute,
   certification,
   configuration_effectivity,
   contract,
   document_file,
   executed_action,
   product,
   product_definition,
   product_definition_formation,
   product_definition_relationship,
   versioned_action_request);
END_TYPE;


(* Pruned unused type: approval_item_203e2  *)


TYPE approved_item = SELECT
   (certification,
    change,
    change_request,
    configuration_effectivity,
    configuration_item,
    contract,
    product,
    security_classification,
```

```
    start_request,
    start_work);
END_TYPE;

TYPE area_measure = REAL;
END_TYPE;

TYPE area_or_view = SELECT
  (presentation_area,
   presentation_view);
END_TYPE;

TYPE attribute_language_item = SELECT (
   alternate_product_relationship,
   application_context,
   applied_certification_assignment,
   applied_document_reference,

   applied_document_usage_constraint_assignment,
   applied_external_identification_assignment,
   applied_identification_assignment,
   applied_organizational_project_assignment,
   applied_security_classification_assignment,
   approval,
   approval_relationship,
   approval_status,
   assembly_component_usage_substitute,
   attribute_value_assignment,
   certification,
   certification_type,
   configuration_design,
   configuration_item,
   contract,
   date_role,
   date_time_role,
   descriptive_representation_item,
   document_relationship,
   document_usage_role,
   effectivity,
   effectivity_relationship,
   event_occurrence,
   external_source,
   general_property,
   general_property_relationship,
   geometric_representation_item,
   geometric_tolerance,
   identification_role,
   make_from_usage_option,
   mapped_item,
   multi_language_attribute_assignment,
   object_role,
   organization_relationship,
   organization_role,
   organizational_project,
   organizational_project_relationship,
   organizational_project_role,
   person_and_organization,
```

```
   person_and_organization_role,
   product,
   product_category,
   product_concept,
   product_concept_relationship,
   product_definition,
   product_definition_context,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_relationship,
   product_definition_shape,
   product_related_product_category,
   property_definition,
   representation,
   security_classification,
   security_classification_assignment,
   shape_aspect,
   shape_aspect_relationship,
   shape_representation,
   time_interval_role,
   topological_representation_item,
   uncertainty_measure_with_unit,
   uncertainty_qualifier);
END_TYPE;


(* Pruned unused type: attribute_language_item_203e2  *)


TYPE attribute_type = SELECT
   (label,
    text);
END_TYPE;

 TYPE axis2_placement = SELECT
   (axis2_placement_2d,
    axis2_placement_3d);
 END_TYPE;

 TYPE b_spline_curve_form = ENUMERATION OF
   (polyline_form,
    circular_arc,
    elliptic_arc,
    parabolic_arc,
    hyperbolic_arc,
    unspecified);
 END_TYPE;

 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;

TYPE boolean_operand = SELECT (
   solid_model,

   half_space_solid,
   csg_primitive,
   boolean_result);
END_TYPE;


 TYPE boolean_operator = ENUMERATION OF
   (union,
    intersection,
    difference);
 END_TYPE;

TYPE box_characteristic_select = SELECT
  (box_height,
   box_width,

   box_slant_angle,
   box_rotate_angle);
END_TYPE;

TYPE box_height = positive_ratio_measure;
END_TYPE;

TYPE box_rotate_angle = plane_angle_measure;
END_TYPE;

TYPE  box_slant_angle = plane_angle_measure;
END_TYPE;

TYPE box_width = positive_ratio_measure;
END_TYPE;

(* Pruned unused type: cc3dpa_groupable_item  *)


TYPE celsius_temperature_measure = REAL;
END_TYPE;

TYPE central_or_parallel = ENUMERATION OF
  (central,
   parallel);
END_TYPE;

TYPE certification_item = SELECT (
   product_definition_formation_relationship);
END_TYPE;


(* Pruned unused type: certification_item_203e2  *)
```

```
TYPE certified_item = SELECT
   (supplied_part_relationship);
END_TYPE;

TYPE change_request_item = SELECT
   (product_definition_formation);
END_TYPE;

TYPE character_spacing_select = SELECT (
   length_measure,
   ratio_measure,
   measure_with_unit,
   descriptive_measure);
END_TYPE;


TYPE character_style_select = SELECT
  (character_glyph_style_stroke,
   character_glyph_style_outline,
   text_style_for_defined_font);
END_TYPE;

TYPE characterized_action_definition = SELECT (
   action,
   action_method,
   action_relationship);
END_TYPE;


TYPE characterized_definition = SELECT
   (characterized_object,
    characterized_product_definition,
    shape_definition);
END_TYPE;

TYPE characterized_product_definition = SELECT
   (product_definition,
    product_definition_relationship);
END_TYPE;

TYPE classification_item = SELECT (
   action,
   action_directive,
   action_method,
   action_property,
   action_relationship,
   action_request_solution,
   action_request_status,
   address,
   alternate_product_relationship,
   applied_action_assignment,
   applied_action_request_assignment,
   applied_approval_assignment,
   applied_certification_assignment,
   applied_contract_assignment,
```

```
applied_date_and_time_assignment,
applied_date_assignment,
applied_document_reference,
applied_document_usage_constraint_assignment,
applied_effectivity_assignment,
applied_event_occurrence_assignment,
applied_external_identification_assignment,
applied_identification_assignment,
applied_organization_assignment,
applied_organizational_project_assignment,
applied_person_and_organization_assignment,
applied_security_classification_assignment,
approval,
approval_person_organization,
approval_relationship,
approval_status,
assembly_component_usage_substitute,
calendar_date,
certification,
class,
configuration_item,
context_dependent_unit,
contract,
conversion_based_unit,
date_and_time,
date_and_time_assignment,
date_assignment,
derived_unit,
descriptive_representation_item,
directed_action,
document_file,
document_relationship,
effectivity,
event_occurrence,
executed_action,
general_property,
general_property_relationship,
group,
identification_assignment,
language,
measure_representation_item,
measure_with_unit,
multi_language_attribute_assignment,
named_unit,
organization,
organization_relationship,
organizational_address,
organizational_project,
organizational_project_relationship,
person,
person_and_organization_address,
product,
product_category,
product_concept,
product_definition,
product_definition_context,
product_definition_formation,
```

```
   product_definition_formation_relationship,
   product_definition_relationship,
   property_definition,
   property_definition_representation,
   representation,
   representation_context,
   representation_item,
   security_classification,
   uncertainty_measure_with_unit,
   versioned_action_request);
END_TYPE;


TYPE classified_item = SELECT
   (assembly_component_usage,
    product_definition_formation);
END_TYPE;

(* Pruned unused type: complete_membership_select  *)


TYPE compound_item_definition = SELECT
  (list_representation_item,
   set_representation_item);
END_TYPE;

TYPE configuration_design_item = SELECT
   (product_definition,
    product_definition_formation);
END_TYPE;

TYPE context_dependent_measure = REAL;
END_TYPE;

TYPE contract_item = SELECT (
   executed_action,
   product_definition_formation);
END_TYPE;


(* Pruned unused type: contract_item_203e2  *)


TYPE contracted_item = SELECT
   (product_definition_formation);
END_TYPE;

TYPE count_measure = NUMBER;
END_TYPE;

TYPE csg_primitive = SELECT (
   sphere,
   block,
   right_angular_wedge,
   torus,
   right_circular_cone,
   right_circular_cylinder);
```

```
END_TYPE;


 TYPE csg_select = SELECT
   (boolean_result,
    csg_primitive);
 END_TYPE;

TYPE curve_font_or_scaled_curve_font_select = SELECT (
   curve_style_font_select);
END_TYPE;


 TYPE curve_on_surface = SELECT
   (pcurve,
    surface_curve,
    composite_curve_on_surface);
 END_TYPE;

TYPE curve_or_annotation_curve_occurrence = SELECT
  (curve,
   annotation_curve_occurrence);
END_TYPE;

TYPE curve_or_render = SELECT (
   curve_style);
END_TYPE;


TYPE curve_style_font_select = SELECT
  (curve_style_font,
   pre_defined_curve_font,
   externally_defined_curve_font);
END_TYPE;

TYPE date_and_time_item = SELECT (
   action,
   applied_action_assignment,
   applied_organization_assignment,
   applied_person_and_organization_assignment,
   applied_security_classification_assignment,
   approval_person_organization,
   certification,
   contract,
   document_file,
   event_occurrence,
   executed_action,
   organizational_project,
   product_definition,
   product_definition_formation,
   product_definition_relationship,
   security_classification,
   versioned_action_request);
END_TYPE;


(* Pruned unused type: date_and_time_item_203e2  *)
```

```
(* Pruned unused type: date_and_time_item_approval  *)


TYPE date_item = SELECT (
   action,
   applied_action_assignment,
   applied_organization_assignment,
   applied_person_and_organization_assignment,
   applied_security_classification_assignment,
   approval_person_organization,
   certification,
   contract,
   document_file,
   event_occurrence,
   organizational_project,
   product_definition,
   product_definition_formation,
   product_definition_relationship,
   security_classification,
   versioned_action_request);
END_TYPE;


(* Pruned unused type: date_item_203e2  *)


(* Pruned unused type: date_item_approval  *)


TYPE date_time_item = SELECT
   (approval_person_organization,
    certification,
    change,
    change_request,
    contract,
    product_definition,
    security_classification,
    start_request,
    start_work);
END_TYPE;

TYPE date_time_or_event_occurrence = SELECT
   (date_time_select,
    event_occurrence);
END_TYPE;

TYPE date_time_select = SELECT
   (date,
    date_and_time,
    local_time);
END_TYPE;

TYPE day_in_month_number = INTEGER;
WHERE
  WR1: {1 <= SELF <= 31};
```

```
END_TYPE;

TYPE day_in_week_number = INTEGER;
WHERE
  WR1: { 1 <= SELF <= 7 };
END_TYPE;

TYPE day_in_year_number = INTEGER;
WHERE
  WR1: {1 <= SELF <= 366};
END_TYPE;

TYPE defined_glyph_select = SELECT
  (pre_defined_character_glyph,
   externally_defined_character_glyph);
END_TYPE;

TYPE defined_symbol_select = SELECT
  (pre_defined_symbol,
   externally_defined_symbol);
END_TYPE;

TYPE derived_property_select = SELECT (
   action_property,
   property_definition);
END_TYPE;


TYPE description_attribute_select = SELECT (
   action_request_solution,
   application_context,
   approval_role,
   configuration_design,
   context_dependent_shape_representation,
   date_role,
   date_time_role,
   effectivity,
   external_source,
   organization_role,
   person_and_organization,
   person_and_organization_role,
   property_definition_representation,
   representation);
END_TYPE;


TYPE descriptive_measure = STRING;
END_TYPE;

TYPE dimension_count = INTEGER;
WHERE
  WR1: SELF > 0;
END_TYPE;

TYPE dimension_extent_usage = ENUMERATION OF
  (origin,
   target);
```

```
END_TYPE;

(* Pruned unused type: dimension_identification_item  *)


(* Pruned unused type: dimension_tolerance_document_reference_item  *)


TYPE dimensional_characteristic = SELECT (dimensional_location,
                                          dimensional_size);
END_TYPE;

(* Pruned unused type: dimensioning_standard_document_reference_item  *)


TYPE direction_count_select = SELECT
  (u_direction_count,
   v_direction_count);
END_TYPE;

(* Pruned unused type: dm_aliasable_item  *)


(* Pruned unused type: dm_approval_item  *)


(* Pruned unused type: dm_attribute_language_item  *)


(* Pruned unused type: dm_contract_item  *)


(* Pruned unused type: dm_date_and_time_item  *)


(* Pruned unused type: dm_date_item  *)


(* Pruned unused type: dm_identification_item  *)


(* Pruned unused type: dm_multi_language_attribute_item  *)


(* Pruned unused type: dm_organization_item  *)


(* Pruned unused type: dm_person_and_organization_item  *)


(* Pruned unused type: dm_security_classification_item  *)


(* Pruned unused type: document_location_select  *)


TYPE document_reference_item = SELECT (
```

```
   action_method,
   dimensional_size,
   executed_action,
   externally_defined_dimension_definition,
   product,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_relationship,
   shape_aspect,
   shape_aspect_relationship,
   versioned_action_request);
END_TYPE;


(* Pruned unused type: document_reference_item_203e2  *)


TYPE draughting_callout_element = SELECT
  (annotation_text_occurrence,
   annotation_symbol_occurrence,
   annotation_curve_occurrence);
END_TYPE;

TYPE effectivity_item = SELECT (
   assembly_component_usage_substitute,
   product,
   product_definition,
   product_definition_formation,
   product_definition_relationship);
END_TYPE;


(* Pruned unused type: effectivity_item_203e2  *)



TYPE electric_current_measure = REAL;
END_TYPE;

(* Pruned unused type: event_occurrence_date_and_time_item  *)


(* Pruned unused type: event_occurrence_date_item  *)


TYPE event_occurrence_item = SELECT (
   organizational_project);
END_TYPE;


(* Pruned unused type: event_occurrence_item_203e2  *)


(* Pruned unused type: external_class_name_item  *)
```

```
TYPE external_identification_item = SELECT (
   document_file,
   product_definition);
END_TYPE;


(* Pruned unused type: external_identification_item_203e2  *)


(* Pruned unused type: file_identification_item  *)


(* Pruned unused type: file_location_select  *)


(* Pruned unused type: file_version_item  *)


TYPE fill_area_style_tile_shape_select = SELECT
  (fill_area_style_tile_curve_with_style,
   fill_area_style_tile_coloured_region,
   fill_area_style_tile_symbol_with_style,
   pre_defined_tile,
   externally_defined_tile);
END_TYPE;

TYPE fill_style_select = SELECT (
   fill_area_style_colour,
   externally_defined_tile_style,
   fill_area_style_tiles,
   externally_defined_hatch_style,
   fill_area_style_hatching);
END_TYPE;


TYPE font_select = SELECT
  (pre_defined_text_font,
   externally_defined_text_font);
END_TYPE;

TYPE founded_item_select = SELECT
  (founded_item,
   representation_item);
END_TYPE;

 TYPE geometric_set_select = SELECT
   (point,
    curve,
    surface);
 END_TYPE;

TYPE groupable_item = SELECT (
   geometric_representation_item,
   mapped_item,
   representation_relationship_with_transformation,
   shape_aspect,
   styled_item);
```

```
   WHERE
  WR1: NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GROUP' IN TYPEOF(SELF));

END_TYPE;

TYPE hour_in_day = INTEGER;
WHERE
  WR1: { 0 <= SELF < 24 };
END_TYPE;

TYPE id_attribute_select = SELECT
   (action,
    address,
    application_context,
    group,
    organizational_project,
    product_category,
    property_definition,
    representation,
    shape_aspect,
    shape_aspect_relationship);
END_TYPE;

TYPE identification_item = SELECT (
   approval_status,
   contract,
   dimensional_size,
   document_file,
   draughting_model,
   general_property,
   mechanical_design_geometric_presentation_representation,
   organization,
   product,
   product_definition,
   product_definition_formation,
   security_classification,
   security_classification_level,
   shape_aspect_relationship,
   shape_representation);
END_TYPE;


(* Pruned unused type: identification_item_203e2  *)


TYPE identifier = STRING;
END_TYPE;

(* Pruned unused type: idrm_marked_item  *)


TYPE invisibility_context = SELECT (
   presentation_representation,
   presentation_set);
END_TYPE;
```

```
TYPE invisible_item = SELECT
  (styled_item,
   presentation_layer_assignment,
   representation);
END_TYPE;

 TYPE knot_type = ENUMERATION OF
   (uniform_knots,
    quasi_uniform_knots,
    piecewise_bezier_knots,
    unspecified);
 END_TYPE;

TYPE label = STRING;
END_TYPE;


TYPE layered_item = SELECT
  (presentation_representation,
   representation_item);
END_TYPE;

TYPE length_measure = REAL;
END_TYPE;

TYPE limit_condition = ENUMERATION OF
  (maximum_material_condition,
   least_material_condition,
   regardless_of_feature_size);
END_TYPE;

 TYPE list_of_reversible_topology_item =
                                LIST [0:?] of reversible_topology_item;
 END_TYPE;

TYPE list_representation_item = LIST [1:?] OF representation_item;
END_TYPE;

TYPE luminous_intensity_measure = REAL;
END_TYPE;

TYPE marker_select = SELECT
  (marker_type,
   pre_defined_marker);
END_TYPE;

TYPE marker_type = ENUMERATION OF
  (dot,
   x,
   plus,
   asterisk,
   ring,
   square,
   triangle);
END_TYPE;

TYPE mass_measure = REAL;
```

```
END_TYPE;

TYPE measure_value = SELECT
   (amount_of_substance_measure,
    area_measure,
    celsius_temperature_measure,
    context_dependent_measure,
    count_measure,
    descriptive_measure,
    electric_current_measure,
    length_measure,
    luminous_intensity_measure,
    mass_measure,
    numeric_measure,
    parameter_value,
    plane_angle_measure,
    positive_length_measure,
    positive_plane_angle_measure,
    positive_ratio_measure,
    ratio_measure,
    solid_angle_measure,
    thermodynamic_temperature_measure,
    time_measure,
    volume_measure);
END_TYPE;

TYPE minute_in_hour = INTEGER;
WHERE
  WR1: { 0 <= SELF <= 59 };
END_TYPE;

TYPE month_in_year_number = INTEGER;
WHERE
  WR1: { 1 <= SELF <= 12 };
END_TYPE;

(* Pruned unused type: mri_aliasable_item  *)


(* Pruned unused type: mri_approval_item  *)


(* Pruned unused type: mri_attribute_language_item  *)


(* Pruned unused type: mri_date_and_time_item  *)


(* Pruned unused type: mri_date_item  *)


(* Pruned unused type: mri_identification_item  *)


(* Pruned unused type: mri_multi_language_attribute_item  *)
```

```
(* Pruned unused type: mri_organization_item  *)


(* Pruned unused type: mri_person_and_organization_item  *)


TYPE multi_language_attribute_item = SELECT (
   alternate_product_relationship,
   application_context,
   applied_certification_assignment,
   applied_document_reference,
   applied_document_usage_constraint_assignment,
   applied_external_identification_assignment,
   applied_identification_assignment,
   applied_organizational_project_assignment,
   approval,
   approval_relationship,
   approval_status,
   assembly_component_usage_substitute,
   attribute_value_assignment,
   certification,
   certification_type,
   colour,
   configuration_design,
   configuration_item,
   contract,
   date_role,
   date_time_role,
   descriptive_representation_item,
   document_relationship,
   document_usage_role,
   effectivity,
   effectivity_relationship,
   event_occurrence,
   external_source,
   general_property,
   general_property_relationship,
   geometric_representation_item,
   geometric_tolerance,
   identification_role,
   make_from_usage_option,
   mapped_item,
   object_role,
   organization_relationship,
   organization_role,
   organizational_project,
   organizational_project_relationship,
   organizational_project_role,
   person_and_organization,
   person_and_organization_role,
   product,
   product_category,
   product_concept,
   product_concept_relationship,
   product_definition,
   product_definition_context,
   product_definition_formation,
```

```
   product_definition_formation_relationship,
   product_definition_relationship,
   product_definition_shape,
   product_related_product_category,
   property_definition,
   representation,
   representation_relationship,
   security_classification,
   security_classification_assignment,
   shape_aspect,
   shape_aspect_relationship,
   shape_representation,
   time_interval_role,
   topological_representation_item,
   uncertainty_measure_with_unit);
END_TYPE;


(* Pruned unused type: multi_language_attribute_item_203e2  *)


TYPE name_attribute_select = SELECT (
   action_request_solution,
   address,
   configuration_design,
   context_dependent_shape_representation,
   derived_unit,
   effectivity,
   person_and_organization,
   product_definition,
   property_definition_representation);
END_TYPE;


TYPE name_item = SELECT (
   assembly_component_usage,
   external_class_library);
END_TYPE;


TYPE null_style = ENUMERATION OF
  (null);
END_TYPE;

TYPE numeric_measure = NUMBER;
END_TYPE;

TYPE organization_item = SELECT (
   action,
   alternate_product_relationship,
   applied_action_assignment,
   applied_identification_assignment,
   applied_security_classification_assignment,
   approval,
   assembly_component_usage_substitute,
   certification,
   configuration_item,
```

```
   contract,
   document_file,
   executed_action,
   general_property,
   organizational_project,
   product,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   property_definition,
   security_classification,
   shape_representation,
   versioned_action_request);
END_TYPE;


(* Pruned unused type: organization_item_203e2  *)


(* Pruned unused type: organizational_project_item_203e2  *)


TYPE parameter_value = REAL;
END_TYPE;

 TYPE pcurve_or_surface = SELECT
   (pcurve,
    surface);
 END_TYPE;

(* Pruned unused type: pdm_action_items  *)


(* Pruned unused type: pdm_action_request_item  *)


(* Pruned unused type: pdm_approval_item  *)


(* Pruned unused type: pdm_attribute_language_item  *)


(* Pruned unused type: pdm_certification_item  *)


(* Pruned unused type: pdm_classification_item  *)


(* Pruned unused type: pdm_date_and_time_item  *)


(* Pruned unused type: pdm_date_item  *)


(* Pruned unused type: pdm_document_reference_item  *)
```

```
(* Pruned unused type: pdm_effectivity_item  *)

(* Pruned unused type: pdm_event_occurrence_item  *)

(* Pruned unused type: pdm_external_class_name_item  *)

(* Pruned unused type: pdm_identification_item  *)

(* Pruned unused type: pdm_multi_language_attribute_item  *)

(* Pruned unused type: pdm_organization_item  *)

(* Pruned unused type: pdm_person_and_organization_item  *)

(* Pruned unused type: pdm_project_item  *)

(* Pruned unused type: pdm_security_classification_item  *)

TYPE person_and_organization_item = SELECT (
   action,
   alternate_product_relationship,
   applied_action_assignment,
   applied_identification_assignment,
   applied_security_classification_assignment,
   approval,
   assembly_component_usage_substitute,
   certification,
   configuration_item,
   contract,
   document_file,
   executed_action,
   general_property,
   organizational_project,
   product,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   property_definition,
   security_classification,
   shape_representation,
   versioned_action_request);
END_TYPE;

(* Pruned unused type: person_and_organization_item_203e2  *)

TYPE person_organization_item = SELECT
```

```
   (change,
    change_request,
    configuration_item,
    contract,
    product,
    product_definition,
    product_definition_formation,
    security_classification,
    start_request,
    start_work);
END_TYPE;

TYPE person_organization_select = SELECT
   (organization,
    person,
    person_and_organization);
END_TYPE;

TYPE plane_angle_measure = REAL;
END_TYPE;

TYPE positive_length_measure = length_measure;
WHERE
  WR1: SELF > 0.0;
END_TYPE;

TYPE positive_plane_angle_measure = plane_angle_measure;
WHERE
  WR1: SELF > 0.0;
END_TYPE;

TYPE positive_ratio_measure = ratio_measure;
WHERE
  WR1: SELF > 0.0;
END_TYPE;

 TYPE preferred_surface_curve_representation = ENUMERATION OF
   (curve_3d,
    pcurve_s1,
    pcurve_s2);
 END_TYPE;

TYPE presentable_text = STRING;
END_TYPE;

(* Pruned unused type: presentation_representation_select  *)


TYPE presentation_size_assignment_select = SELECT
  (presentation_view,
   presentation_area,
   area_in_set);
END_TYPE;

TYPE presentation_style_select = SELECT (
   point_style,
   curve_style,
```

```
   surface_style_usage,
   symbol_style,
   fill_area_style,
   text_style,
   null_style);
END_TYPE;


TYPE product_or_formation_or_definition = SELECT
   (product,
    product_definition_formation,
    product_definition);
END_TYPE;

(* Pruned unused type: project_as_date_and_time_item  *)


(* Pruned unused type: project_as_date_item  *)


TYPE project_item = SELECT (
   executed_action,
   product_concept);
END_TYPE;


TYPE ratio_measure = REAL;
END_TYPE;

(* Pruned unused type: representation_identification_item  *)


(* Pruned unused type: representation_version_item  *)


TYPE represented_definition = SELECT (
   general_property,
   property_definition,
   shape_aspect,
   shape_aspect_relationship);
END_TYPE;


 TYPE reversible_topology = SELECT
           (reversible_topology_item,
            list_of_reversible_topology_item,
            set_of_reversible_topology_item);
 END_TYPE;

 TYPE reversible_topology_item = SELECT
   (edge,
    path,
    face,
    face_bound,
    closed_shell,
    open_shell);
 END_TYPE;
```

```
TYPE role_select = SELECT (
   action_assignment,
   action_request_assignment,
   approval_assignment,
   approval_date_time,
   certification_assignment,
   contract_assignment,
   document_reference,
   effectivity_assignment,
   group_assignment,
   name_assignment,
   security_classification_assignment);
END_TYPE;


TYPE second_in_minute = REAL;
WHERE
  WR1: { 0 <= SELF <= 60.0 };
END_TYPE;

TYPE security_classification_item = SELECT (
   assembly_component_usage,
   document_file,
   product,
   product_definition,
   product_definition_formation,
   product_definition_usage);
END_TYPE;


(* Pruned unused type: security_classification_item_203e2  *)


  TYPE set_of_reversible_topology_item =
                       SET [0:?] of reversible_topology_item;
  END_TYPE;

TYPE set_representation_item = SET [1:?] OF representation_item;
END_TYPE;

TYPE shape_definition = SELECT
   (product_definition_shape,
    shape_aspect,
    shape_aspect_relationship);
END_TYPE;

(* Pruned unused type: shape_tolerance_select  *)


 TYPE shell = SELECT
   (vertex_shell,
    wire_shell,
    open_shell,
    closed_shell);
 END_TYPE;
```

```
TYPE si_prefix = ENUMERATION OF
   (exa,
    peta,
    tera,
    giga,
    mega,
    kilo,
    hecto,
    deca,
    deci,
    centi,
    milli,
    micro,
    nano,
    pico,
    femto,
    atto);
END_TYPE;

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;

TYPE size_select = SELECT (
   positive_length_measure,
   measure_with_unit,
   descriptive_measure);
END_TYPE;
```

```
TYPE solid_angle_measure = REAL;
END_TYPE;

TYPE source = ENUMERATION OF
   (made,
    bought,
    not_known);
END_TYPE;

TYPE source_item = SELECT (
   identifier);
END_TYPE;


TYPE specified_item = SELECT
   (product_definition,
    shape_aspect);
END_TYPE;

(* Pruned unused type: spr_document_reference_item  *)


(* Pruned unused type: spr_organization_item  *)


(* Pruned unused type: spr_person_and_organization_item  *)


(* Pruned unused type: start_request_item  *)


TYPE style_context_select = SELECT
  (group,
   presentation_layer_assignment,
   representation,
   representation_item,
   presentation_set);
END_TYPE;

(* Pruned unused type: supported_item  *)


(* Pruned unused type: surface_boundary  *)


(* Pruned unused type: surface_model  *)


TYPE surface_side = ENUMERATION OF
  (positive,
   negative,
   both);
END_TYPE;

TYPE surface_side_style_select = SELECT (
   surface_side_style);
```

```
END_TYPE;


TYPE surface_style_element_select = SELECT (
   surface_style_fill_area,
   surface_style_boundary,
   surface_style_silhouette,
   surface_style_segmentation_curve,
   surface_style_control_grid,
   surface_style_parameter_line);
END_TYPE;


TYPE symbol_style_select = SELECT (
   symbol_colour);
END_TYPE;


(* Pruned unused type: terminator  *)


TYPE text = STRING;
END_TYPE;

TYPE text_alignment = label;
END_TYPE;

TYPE text_delineation = label;
END_TYPE;

TYPE text_or_character = SELECT
  (annotation_text,
   annotation_text_character,
   defined_character_glyph,
   composite_text,
   text_literal);
END_TYPE;

TYPE text_path = ENUMERATION OF
  (left,
   right,
   up,
   down);
END_TYPE;

TYPE thermodynamic_temperature_measure = REAL;
END_TYPE;

TYPE time_interval_item = SELECT (
   time_interval_based_effectivity);
END_TYPE;


(* Pruned unused type: time_interval_item_203e2  *)


TYPE time_measure = REAL;
```

```
END_TYPE;

TYPE tolerance_method_definition = SELECT
  (tolerance_value,
   limits_and_fits);
END_TYPE;

TYPE transformation = SELECT
  (item_defined_transformation,

   functionally_defined_transformation);
END_TYPE;

 TYPE transition_code = ENUMERATION OF
   (discontinuous,
    continuous,
    cont_same_gradient,
    cont_same_gradient_same_curvature);
 END_TYPE;

 TYPE trimming_preference = ENUMERATION OF
   (cartesian,
    parameter,
    unspecified);
 END_TYPE;

 TYPE trimming_select = SELECT
   (cartesian_point,
    parameter_value);
 END_TYPE;

TYPE u_direction_count = INTEGER;
WHERE
  WR1: SELF > 1;
END_TYPE;

TYPE unit = SELECT
   (derived_unit,
    named_unit);
END_TYPE;

TYPE v_direction_count = INTEGER;
WHERE
  WR1: SELF > 1;
END_TYPE;

TYPE value_qualifier = SELECT
  (precision_qualifier,
   type_qualifier,
   uncertainty_qualifier);
END_TYPE;

 TYPE vector_or_direction = SELECT
   (vector,
    direction);
 END_TYPE;
```

```
(* Pruned unused type: versionable_item  *)


TYPE volume_measure = REAL;
END_TYPE;

TYPE week_in_year_number = INTEGER;
WHERE
  WR1: { 1 <= SELF <= 53 };
END_TYPE;

(* Pruned unused type: wireframe_model  *)


TYPE work_item = SELECT
   (product_definition_formation);
END_TYPE;

TYPE year_number = INTEGER;
END_TYPE;

ENTITY action;
  name : label;
  description : OPTIONAL text;
  chosen_method : action_method;
DERIVE
  id : identifier := get_id_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;

END_ENTITY;

ENTITY action_assignment
   ABSTRACT SUPERTYPE;
  assigned_action : action;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;

ENTITY action_directive;
  name : label;
  description : OPTIONAL text;
  analysis : text;
  comment : text;
  requests : SET[1:?] OF versioned_action_request;
END_ENTITY;

ENTITY action_method;
  name : label;
  description : OPTIONAL text;
  consequence : text;
  purpose : text;
```

```
END_ENTITY;

ENTITY action_property;
  name        : label;
  description : text;
  definition  : characterized_action_definition;
END_ENTITY;

ENTITY action_property_representation;
  name           : label;
  description    : text;
  property       : action_property;
  representation : representation;
END_ENTITY;

ENTITY action_relationship;
  name : label;
  description : OPTIONAL text;
  relating_action : action;
  related_action : action;
END_ENTITY;

ENTITY action_request_assignment
   ABSTRACT SUPERTYPE;
  assigned_action_request : versioned_action_request;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;

ENTITY action_request_solution;
  method : action_method;
  request : versioned_action_request;
DERIVE
  description : text := get_description_value (SELF);
  name : label := get_name_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;

END_ENTITY;

ENTITY action_request_status;
  status : label;
  assigned_request : versioned_action_request;
END_ENTITY;

ENTITY action_status;
  status : label;
  assigned_action : executed_action;
END_ENTITY;
```

```
ENTITY address;
  internal_location : OPTIONAL label;
  street_number : OPTIONAL label;
  street : OPTIONAL label;
  postal_box : OPTIONAL label;
  town : OPTIONAL label;
  region : OPTIONAL label;
  postal_code : OPTIONAL label;
  country : OPTIONAL label;
  facsimile_number : OPTIONAL label;
  telephone_number : OPTIONAL label;
  electronic_mail_address : OPTIONAL label;
  telex_number : OPTIONAL label;
DERIVE
  name : label := get_name_value(SELF);
  url : identifier := get_id_value(SELF);
WHERE
  WR1: EXISTS(internal_location) OR EXISTS(street_number) OR EXISTS(street) OR
EXISTS(postal_box) OR EXISTS(town) OR EXISTS(region) OR EXISTS(postal_code) OR
EXISTS(country) OR EXISTS(facsimile_number) OR EXISTS(telephone_number) OR
EXISTS(electronic_mail_address) OR EXISTS(telex_number);
END_ENTITY;

ENTITY advanced_brep_shape_representation
 SUBTYPE OF ( shape_representation );
 WHERE
WR1 :
SIZEOF (
QUERY ( it <* SELF.items| NOT ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MANIFOLD_SOLID_BREP',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACETED_BREP',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D' ] * TYPEOF (it)) =
1) )) = 0;
WR2 :
SIZEOF (
QUERY ( it <* SELF.items| ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MANIFOLD_SOLID_BREP',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' ] * TYPEOF (it)) = 1) )) >
0;
WR3 :
SIZEOF (
QUERY ( msb <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MANIFOLD_SOLID_BREP' IN TYPEOF (it))
)| NOT ( SIZEOF (
QUERY ( csh <* msb_shells(msb)| NOT ( SIZEOF (
QUERY ( fcs <* csh\connected_face_set.cfs_faces| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF (fcs)) )) =
0) )) = 0) )) = 0;
WR4 :
SIZEOF (
QUERY ( msb <*
QUERY ( it <* items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MANIFOLD_SOLID_BREP' IN TYPEOF (it))
)| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_CLOSED_SHELL' IN TYPEOF
(msb\manifold_solid_brep.outer)) )) = 0;
```

```
WR5 :
SIZEOF (
QUERY ( brv <*
QUERY ( it <* items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BREP_WITH_VOIDS'
IN TYPEOF (it)) )| NOT ( SIZEOF (
QUERY ( csh <* brv\brep_with_voids.voids| csh\oriented_closed_shell.orientation ))
= 0) )) = 0;
WR6 :
SIZEOF (
QUERY ( mi <*
QUERY ( it <* items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' IN
TYPEOF (it)) )| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_BREP_SHAPE_REPRESENTATION' IN
TYPEOF (mi\mapped_item.mapping_source.mapped_representation)) )) = 0;

 END_ENTITY;

ENTITY advanced_face
 SUBTYPE OF ( face_surface );
 WHERE
WR1 :
SIZEOF ([ 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ELEMENTARY_SURFACE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_SURFACE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SWEPT_SURFACE' ] * TYPEOF
(face_geometry)) = 1;
WR2 :
SIZEOF (
QUERY ( elp_fbnds <*
QUERY ( bnds <* bounds| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN
TYPEOF (bnds.bound)) )| NOT ( SIZEOF (
QUERY ( oe <* elp_fbnds.bound\path.edge_list| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_CURVE' IN TYPEOF
(oe\oriented_edge.edge_element)) )) = 0) )) = 0;
WR3 :
SIZEOF (
QUERY ( elp_fbnds <*
QUERY ( bnds <* bounds| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN
TYPEOF (bnds.bound)) )| NOT ( SIZEOF (
QUERY ( oe <* elp_fbnds.bound\path.edge_list| NOT ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONIC',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_CURVE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_CURVE' ] * TYPEOF
(oe.edge_element\edge_curve.edge_geometry)) = 1) )) = 0) )) = 0;
WR4 :
SIZEOF (
QUERY ( elp_fbnds <*
QUERY ( bnds <* bounds| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN
TYPEOF (bnds.bound)) )| NOT ( SIZEOF (
QUERY ( oe <* elp_fbnds.bound\path.edge_list| NOT
((('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
(oe\edge.edge_start)) AND
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT' IN TYPEOF
(oe\edge.edge_start\vertex_point.vertex_geometry))) AND
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
(oe\edge.edge_end)) AND
```

```
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT' IN TYPEOF
(oe\edge.edge_end\vertex_point.vertex_geometry)))) )) = 0) )) = 0;
WR5 :
SIZEOF (
QUERY ( elp_fbnds <*
QUERY ( bnds <* bounds| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN
TYPEOF (bnds.bound)) )| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_PATH'
IN TYPEOF (elp_fbnds.bound)) )) = 0;
WR6 : NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SWEPT_SURFACE' IN TYPEOF
(face_geometry)) OR ( SIZEOF ([ 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONIC',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_CURVE' ] * TYPEOF
(face_geometry\swept_surface.swept_curve)) = 1);
WR7 :
SIZEOF (
QUERY ( vlp_fbnds <*
QUERY ( bnds <* bounds| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_LOOP'
IN TYPEOF (bnds.bound)) )| NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
(vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex)) AND
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT' IN TYPEOF
(vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex\vertex_point.vertex_geometry)))
)) = 0;
WR8 :
SIZEOF (
QUERY ( bnd <* bounds| NOT ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_LOOP' ] * TYPEOF (bnd.bound)) =
1) )) = 0;
WR9 :
SIZEOF (
QUERY ( elp_fbnds <*
QUERY ( bnds <* bounds| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN
TYPEOF (bnds.bound)) )| NOT ( SIZEOF (
QUERY ( oe <* elp_fbnds.bound\path.edge_list|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_CURVE' IN TYPEOF
(oe\oriented_edge.edge_element\edge_curve.edge_geometry)) AND NOT ( SIZEOF (
QUERY ( sc_ag <*
oe.edge_element\edge_curve.edge_geometry\surface_curve.associated_geometry| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN TYPEOF (sc_ag)) )) = 0) ))
= 0) )) = 0;
WR10 : (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SWEPT_SURFACE' IN TYPEOF
(face_geometry)) OR (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN
TYPEOF (face_geometry\swept_surface.swept_curve)) OR ( SIZEOF
(face_geometry\swept_surface.swept_curve\polyline.points) >= 3))) AND ( SIZEOF (
QUERY ( elp_fbnds <*
QUERY ( bnds <* bounds| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN
TYPEOF (bnds.bound)) )| NOT ( SIZEOF (
QUERY ( oe <* elp_fbnds.bound\path.edge_list|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN TYPEOF
(oe\oriented_edge.edge_element\edge_curve.edge_geometry)) AND NOT ( SIZEOF
(oe\oriented_edge.edge_element\edge_curve.edge_geometry\polyline.points) >= 3) )) =
0) )) = 0);

 END_ENTITY;
```

```
ENTITY alternate_product_relationship;
  name        : label;
  definition  : OPTIONAL text;
  alternate   : product;
  base        : product;
  basis       : text;
UNIQUE
  UR1: alternate, base;
WHERE
  WR1: alternate :<>: base;
END_ENTITY;

ENTITY amount_of_substance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AMOUNT_OF_SUBSTANCE_UNIT' IN
TYPEOF (SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY amount_of_substance_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 = 1.0) AND
(SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;

ENTITY angular_dimension
  SUBTYPE OF (dimension_curve_directed_callout);
END_ENTITY;

ENTITY angular_location
  SUBTYPE OF (dimensional_location);
  angle_selection : angle_relator;
END_ENTITY;

ENTITY angular_size
  SUBTYPE OF (dimensional_size);
  angle_selection : angle_relator;
END_ENTITY;

ENTITY angularity_tolerance
 SUBTYPE OF ( geometric_tolerance_with_datum_reference );
 WHERE
WR1 :
SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) < 3;
 END_ENTITY;

ENTITY annotation_curve_occurrence
  SUBTYPE OF (annotation_occurrence);
WHERE
```

```
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE' IN TYPEOF
(SELF\styled_item.item);

END_ENTITY;

ENTITY annotation_fill_area
  SUBTYPE OF (geometric_representation_item);
  boundaries : SET [1:?] OF curve;
END_ENTITY;

ENTITY annotation_fill_area_occurrence
  SUBTYPE OF (annotation_occurrence);
  fill_style_target : point;
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_FILL_AREA' IN

         TYPEOF (SELF.item);
END_ENTITY;

ENTITY annotation_occurrence
  SUPERTYPE OF (ONEOF (
              ANNOTATION_POINT_OCCURRENCE,
              ANNOTATION_CURVE_OCCURRENCE,
              ANNOTATION_FILL_AREA_OCCURRENCE,
              ANNOTATION_TEXT_OCCURRENCE,
              ANNOTATION_SYMBOL_OCCURRENCE))
  SUBTYPE OF (styled_item);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_ITEM'
IN

          TYPEOF (SELF);
END_ENTITY;

ENTITY annotation_occurrence_associativity
  SUBTYPE OF (annotation_occurrence_relationship);
WHERE
  WR1:  SIZEOF (TYPEOF (SELF.related_annotation_occurrence) *

['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_FILL_AREA_OCCURRENCE',
               'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROJECTION_CURVE',
               'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_CURVE']) = 1;

END_ENTITY;

ENTITY annotation_occurrence_relationship;
  name                           : label;
  description                    : text;
  relating_annotation_occurrence : annotation_occurrence;
  related_annotation_occurrence  : annotation_occurrence;
END_ENTITY;

ENTITY annotation_point_occurrence
  SUBTYPE OF (annotation_occurrence);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT' IN TYPEOF
(SELF\styled_item.item);
```

```
END_ENTITY;

ENTITY annotation_subfigure_occurrence
  SUBTYPE OF (annotation_symbol_occurrence);
WHERE
  WR1: SIZEOF (QUERY (sty <* SELF.styles |
         NOT (SIZEOF (sty.styles) = 1)
       )) = 0;
  WR2: SIZEOF (QUERY (sty <* SELF.styles |
         NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.NULL_STYLE'
               IN TYPEOF (sty.styles[1]))       ))=0;
  WR3: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL'
           IN TYPEOF (SELF.item));
  WR4:
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_SUBFIGURE_REPRESENTATION'

           IN TYPEOF
           (SELF.item\mapped_item.mapping_source.mapped_representation));
END_ENTITY;

ENTITY annotation_symbol
  SUBTYPE OF(mapped_item);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMBOL_REPRESENTATION_MAP' IN
         TYPEOF (SELF\mapped_item.mapping_source);
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMBOL_TARGET' IN
         TYPEOF (SELF\mapped_item.mapping_target);
  WR3: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_ITEM'
IN

         TYPEOF (SELF);
END_ENTITY;

ENTITY annotation_symbol_occurrence
  SUBTYPE OF (annotation_occurrence);
WHERE
  WR1: SIZEOF(
         ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL',
          'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEFINED_SYMBOL'] *

         TYPEOF(SELF\styled_item.item)) > 0;
END_ENTITY;

ENTITY annotation_table
  SUBTYPE OF(annotation_symbol);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TABLE_REPRESENTATION' IN

         TYPEOF (SELF\mapped_item.mapping_source.mapped_representation);
END_ENTITY;

ENTITY annotation_table_occurrence
  SUBTYPE OF (annotation_symbol_occurrence);
WHERE
  WR1: SIZEOF (
         ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TABLE',
```

```
          'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEFINED_TABLE'] *

         TYPEOF (SELF\styled_item.item)) > 0;
END_ENTITY;

ENTITY annotation_text
  SUBTYPE OF (mapped_item);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT' IN
       TYPEOF( SELF\mapped_item.mapping_target);
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_STRING_REPRESENTATION' IN
       TYPEOF( SELF\mapped_item.mapping_source.mapped_representation);
  WR3: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_ITEM'
IN

       TYPEOF( SELF);
END_ENTITY;

ENTITY annotation_text_character
  SUBTYPE OF (mapped_item);
  alignment : text_alignment;
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CHARACTER_GLYPH_SYMBOL' IN
         TYPEOF (SELF\mapped_item.mapping_source.mapped_representation);
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT' IN
         TYPEOF (SELF\mapped_item.mapping_target);
  WR3: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_ITEM'
IN

         TYPEOF (SELF);
END_ENTITY;

ENTITY annotation_text_occurrence
  SUBTYPE OF (annotation_occurrence);
WHERE
  WR1: SIZEOF (
         ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL',
          'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT',
          'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_CHARACTER',
          'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEFINED_CHARACTER_GLYPH',
          'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'] *

         TYPEOF (SELF\styled_item.item)) > 0;
END_ENTITY;

ENTITY annotation_text_with_associated_curves
  SUBTYPE OF (annotation_text);
  associated_curves : SET[1:?] of curve;
END_ENTITY;

ENTITY annotation_text_with_blanking_box
  SUBTYPE OF (annotation_text);
  blanking : planar_box;
END_ENTITY;

ENTITY annotation_text_with_delineation
  SUBTYPE OF (annotation_text);
```

```
  delineation : text_delineation;
END_ENTITY;

ENTITY annotation_text_with_extent
  SUBTYPE OF (annotation_text);
  extent : planar_extent;
END_ENTITY;

ENTITY apex
  SUBTYPE OF (derived_shape_aspect);
END_ENTITY;

ENTITY application_context;
  application : label;
DERIVE
  description : text := get_description_value (SELF);
  id : identifier := get_id_value (SELF);
INVERSE
  context_elements : SET[1:?] OF application_context_element FOR
frame_of_reference;
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;

END_ENTITY;

ENTITY application_context_element
  SUPERTYPE OF (ONEOF (
              PRODUCT_CONCEPT_CONTEXT,
              PRODUCT_CONTEXT,
              PRODUCT_DEFINITION_CONTEXT));
  name : label;
  frame_of_reference : application_context;
END_ENTITY;

ENTITY application_context_relationship;
  name : label;
  description : OPTIONAL text;
  relating_context : application_context;
  related_context : application_context;
END_ENTITY;

ENTITY application_protocol_definition;
  status : label;
  application_interpreted_model_schema_name : label;
  application_protocol_year : year_number;
  application : application_context;
END_ENTITY;

ENTITY applied_action_assignment
SUBTYPE OF (action_assignment);
  items : SET [1 : ?] OF action_items;
END_ENTITY;

ENTITY applied_action_request_assignment
```

```
SUBTYPE OF (action_request_assignment);
  items : SET [1 : ?] OF action_request_item;
END_ENTITY;

ENTITY applied_approval_assignment
  SUBTYPE OF (approval_assignment);
  items : SET[1:?] OF approval_item;
END_ENTITY;

ENTITY applied_certification_assignment
  SUBTYPE OF (certification_assignment);
  items : SET[1:?] OF certification_item;
END_ENTITY;

ENTITY applied_classification_assignment
  SUBTYPE OF (classification_assignment);
  items : SET[1:?] OF classification_item;
END_ENTITY;

ENTITY applied_contract_assignment
  SUBTYPE OF (contract_assignment);
  items : SET [1:?] OF contract_item;
END_ENTITY;

ENTITY applied_date_and_time_assignment
  SUBTYPE OF (date_and_time_assignment);
  items : SET [1:?] OF date_and_time_item;
END_ENTITY;

ENTITY applied_date_assignment
  SUBTYPE OF (date_assignment);
  items : SET [1:?] OF date_item;
END_ENTITY;

ENTITY applied_document_reference
  SUBTYPE OF (document_reference);
  items : SET[1:?] OF document_reference_item;
END_ENTITY;

ENTITY applied_document_usage_constraint_assignment
  SUBTYPE OF (document_usage_constraint_assignment);
  items : SET[1:?] OF document_reference_item;
END_ENTITY;

ENTITY applied_effectivity_assignment
  SUBTYPE OF (effectivity_assignment);
  items : SET[1:?] OF effectivity_item;
END_ENTITY;

ENTITY applied_event_occurrence_assignment
  SUBTYPE OF (event_occurrence_assignment);
  items : SET[1:?] OF event_occurrence_item;
END_ENTITY;

ENTITY applied_external_identification_assignment
SUBTYPE OF (external_identification_assignment);
  items : SET [1 : ?] OF external_identification_item;
```

```
END_ENTITY;

ENTITY applied_group_assignment
  SUBTYPE OF (group_assignment);
  items : SET[1:?] OF groupable_item;
END_ENTITY;

ENTITY applied_identification_assignment
  SUBTYPE OF (identification_assignment);
  items : SET[1:?] OF identification_item;
END_ENTITY;

ENTITY applied_name_assignment
  SUBTYPE OF (name_assignment);
  item : name_item;
END_ENTITY;

ENTITY applied_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF organization_item;
END_ENTITY;

ENTITY applied_organizational_project_assignment
  SUBTYPE OF (organizational_project_assignment);
  items : SET[1:?] OF project_item;
END_ENTITY;

ENTITY applied_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF person_and_organization_item;
END_ENTITY;

ENTITY applied_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
  items : SET [1:?] OF security_classification_item;
END_ENTITY;

ENTITY applied_time_interval_assignment
  SUBTYPE OF (time_interval_assignment);
  items : SET[0:?] OF time_interval_item;
END_ENTITY;

ENTITY approval;
  status : approval_status;
  level : label;
END_ENTITY;

ENTITY approval_assignment
   ABSTRACT SUPERTYPE;
  assigned_approval : approval;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;
```

```
ENTITY approval_date_time;
  date_time : date_time_select;
  dated_approval : approval;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;

ENTITY approval_person_organization;
  person_organization : person_organization_select;
  authorized_approval : approval;
  role : approval_role;
END_ENTITY;

ENTITY approval_relationship;
  name : label;
  description : OPTIONAL text;
  relating_approval : approval;
  related_approval : approval;
END_ENTITY;

ENTITY approval_role;
  role : label;
DERIVE
  description : text := get_description_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;

END_ENTITY;

ENTITY approval_status;
  name : label;
END_ENTITY;

ENTITY area_in_set;
  area   : presentation_area;
  in_set : presentation_set;
END_ENTITY;

ENTITY area_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AREA_UNIT' IN TYPEOF
(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY area_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 2.0) 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 assembly_component_usage
  SUPERTYPE OF (ONEOF (
              NEXT_ASSEMBLY_USAGE_OCCURRENCE,
              SPECIFIED_HIGHER_USAGE_OCCURRENCE,
              PROMISSORY_USAGE_OCCURRENCE))
  SUBTYPE OF (product_definition_usage);
  reference_designator : OPTIONAL identifier;
END_ENTITY;

ENTITY assembly_component_usage_substitute;
  name               : label;
  definition         : OPTIONAL text;
  base               : assembly_component_usage;
  substitute         : assembly_component_usage;
UNIQUE
  UR1: base,substitute;
WHERE
  WR1: base.relating_product_definition :=:
       substitute.relating_product_definition;
  WR2: base :<>: substitute;
END_ENTITY;

ENTITY attribute_classification_assignment
   ABSTRACT SUPERTYPE;
  assigned_class : group;
  attribute_name : label;
  role : classification_role;
END_ENTITY;

ENTITY attribute_language_assignment
SUBTYPE OF (attribute_classification_assignment);
  SELF\attribute_classification_assignment.assigned_class: language;
  items : SET [1:?] OF attribute_language_item;
WHERE
WR1: SELF\attribute_classification_assignment.role.name IN ['primary',
'translated'];
WR2: SELF\attribute_classification_assignment.attribute_name<> '';
END_ENTITY;

ENTITY attribute_value_assignment
   ABSTRACT SUPERTYPE;
  attribute_name : label;
  attribute_value : attribute_type;
  role : attribute_value_role;
END_ENTITY;

ENTITY attribute_value_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

```
ENTITY axis1_placement
SUBTYPE OF (placement);
  axis     : OPTIONAL direction;
DERIVE
  z : direction := NVL(normalise(axis), dummy_gri ||
                               direction([0.0,0.0,1.0]));
WHERE
  WR1: SELF\geometric_representation_item.dim  = 3;
END_ENTITY;

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;

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 > 0.0);
END_ENTITY;

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: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.UNIFORM_CURVE' IN
TYPEOF(self)) OR
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.QUASI_UNIFORM_CURVE' IN
TYPEOF(self)) OR
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BEZIER_CURVE' IN TYPEOF(self))
OR
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_CURVE_WITH_KNOTS' IN
TYPEOF(self));

 END_ENTITY;

 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;

 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: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.UNIFORM_SURFACE' IN
TYPEOF(SELF)) OR
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.QUASI_UNIFORM_SURFACE' IN
TYPEOF(SELF)) OR
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BEZIER_SURFACE' IN
TYPEOF(SELF)) OR
```

```
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_SURFACE_WITH_KNOTS'
IN TYPEOF(SELF));

 END_ENTITY;

 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;

ENTITY background_colour
  SUBTYPE OF (colour);
  presentation : area_or_view;
UNIQUE
  UR1:  presentation;
END_ENTITY;

 ENTITY bezier_curve
   SUBTYPE OF (b_spline_curve);
 END_ENTITY;

 ENTITY bezier_surface
   SUBTYPE OF (b_spline_surface);
 END_ENTITY;

 ENTITY block
   SUBTYPE OF (geometric_representation_item);
   position : axis2_placement_3d;
   x        : positive_length_measure;
   y        : positive_length_measure;
   z        : positive_length_measure;
 END_ENTITY;

 ENTITY boolean_result
   SUBTYPE OF (geometric_representation_item);
   operator        : boolean_operator;
   first_operand   :  boolean_operand;
   second_operand  :  boolean_operand;
 END_ENTITY;

 ENTITY boundary_curve
```

```
  SUBTYPE OF (composite_curve_on_surface);
WHERE
  WR1: SELF\composite_curve.closed_curve;
END_ENTITY;

ENTITY bounded_curve
  SUPERTYPE OF (ONEOF (
              POLYLINE,
              B_SPLINE_CURVE,
              TRIMMED_CURVE,
              BOUNDED_PCURVE,
              BOUNDED_SURFACE_CURVE,
              COMPOSITE_CURVE))
 SUBTYPE OF (curve);
END_ENTITY;

ENTITY bounded_pcurve
  SUBTYPE OF (pcurve, bounded_curve);
 WHERE
   WR1: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BOUNDED_CURVE' IN

                  TYPEOF(SELF\pcurve.reference_to_curve.items[1]));
 END_ENTITY;

ENTITY bounded_surface
  SUPERTYPE OF (ONEOF (
              B_SPLINE_SURFACE,
              RECTANGULAR_TRIMMED_SURFACE,
              CURVE_BOUNDED_SURFACE,
              RECTANGULAR_COMPOSITE_SURFACE))
 SUBTYPE OF (surface);
END_ENTITY;

ENTITY bounded_surface_curve
  SUBTYPE OF (surface_curve, bounded_curve);
WHERE
  WR1: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BOUNDED_CURVE' IN

            TYPEOF(SELF\surface_curve.curve_3d));
END_ENTITY;

ENTITY box_domain
  SUBTYPE OF (founded_item);
  corner  : cartesian_point;
  xlength : positive_length_measure;
  ylength : positive_length_measure;
  zlength : positive_length_measure;
WHERE
  WR1: SIZEOF(QUERY(item <* USEDIN(SELF,'')|
            NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BOXED_HALF_SPACE'

                   IN TYPEOF(item)))) = 0;
END_ENTITY;

ENTITY boxed_half_space
  SUBTYPE OF(half_space_solid);
  enclosure : box_domain;
```

```
 END_ENTITY;

 ENTITY brep_with_voids
   SUBTYPE OF (manifold_solid_brep);
   voids : SET [1:?] OF oriented_closed_shell;
 END_ENTITY;

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;

ENTITY camera_image
  SUBTYPE OF (mapped_item);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CAMERA_USAGE'
       IN TYPEOF (SELF\mapped_item.mapping_source);
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLANAR_BOX'
       IN TYPEOF (SELF\mapped_item.mapping_target);
  WR3: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_ITEM'

       IN TYPEOF (SELF);
END_ENTITY;

ENTITY camera_image_3d_with_scale
  SUBTYPE OF (camera_image);
DERIVE
  scale: positive_ratio_measure := ((SELF\mapped_item.mapping_target\
         planar_extent.size_in_x) / (SELF\mapped_item.mapping_source.
         mapping_origin\camera_model_d3.perspective_of_volume.view_window.
         size_in_x));
WHERE
  WR1: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CAMERA_MODEL_D3'
       IN TYPEOF (SELF\mapped_item.mapping_source.mapping_origin));
  WR2: aspect_ratio(SELF\mapped_item.mapping_target) =
       aspect_ratio(SELF\mapped_item.mapping_source.mapping_origin\
       camera_model_d3.perspective_of_volume.view_window);
  WR3: SELF\mapped_item.mapping_source.mapping_origin\camera_model_d3.
       perspective_of_volume.front_plane_clipping
       AND
       SELF\mapped_item.mapping_source.mapping_origin\camera_model_d3.
       perspective_of_volume.view_volume_sides_clipping;
  WR4: (SELF\mapped_item.mapping_target\planar_extent.size_in_x > 0)
       AND
       (SELF\mapped_item.mapping_target\planar_extent.size_in_y > 0);
  WR5: (SELF\mapped_item.mapping_source.mapping_origin\camera_model_d3.
       perspective_of_volume.view_window.size_in_x > 0)
       AND
       (SELF\mapped_item.mapping_source.mapping_origin\camera_model_d3.
       perspective_of_volume.view_window.size_in_y > 0);
  WR6: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
       'AXIS2_PLACEMENT_2D' IN TYPEOF (SELF\mapped_item.
       mapping_target\planar_box.placement))
       AND NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
```

```
       'AXIS2_PLACEMENT_3D' IN TYPEOF (SELF\mapped_item.
       mapping_target\planar_box.placement));
END_ENTITY;

ENTITY camera_model

  SUBTYPE OF (geometric_representation_item);
WHERE
  WR1: (SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                              'ITEM_DEFINED_TRANSFORMATION.' +
                              'TRANSFORM_ITEM_1')) +
        SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                              'REPRESENTATION_MAP.MAPPING_ORIGIN'))
       ) > 0;
  WR2: SIZEOF(USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+

                          'STYLED_ITEM.ITEM')) = 0;
END_ENTITY;

ENTITY camera_model_d3
  SUBTYPE OF (camera_model);
  view_reference_system : axis2_placement_3d;
  perspective_of_volume : view_volume;
WHERE
  WR1: (dot_product (SELF.view_reference_system.p[3],
         SELF.perspective_of_volume.view_window.placement.p[3]) = 1.0)
         AND
       (SELF.view_reference_system.location.coordinates[3] =
        SELF.perspective_of_volume.view_window.
             placement.location.coordinates[3]);
  WR2: SELF\geometric_representation_item.dim = 3;
END_ENTITY;

ENTITY camera_usage
  SUBTYPE OF (representation_map);
WHERE

  WR1: NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRESENTATION_REPRESENTATION'
       IN TYPEOF(SELF\representation_map.mapped_representation));
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CAMERA_MODEL'

       IN TYPEOF (SELF\representation_map.mapping_origin);
END_ENTITY;

 ENTITY cartesian_point

  SUBTYPE OF (point);
    coordinates  : LIST [1:3] OF length_measure;
 END_ENTITY;

 ENTITY cartesian_transformation_operator
   SUPERTYPE OF (ONEOF (
              CARTESIAN_TRANSFORMATION_OPERATOR_2D,
              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, 1.0);
 WHERE
   WR1: scl > 0.0;
 END_ENTITY;

 ENTITY cartesian_transformation_operator_2d
   SUBTYPE OF (cartesian_transformation_operator);
 DERIVE
   u : LIST[2:2] OF direction :=
       base_axis(2,SELF\cartesian_transformation_operator.axis1,
                   SELF\cartesian_transformation_operator.axis2,?);
 WHERE
   WR1: SELF\geometric_representation_item.dim = 2;
 END_ENTITY;

 ENTITY cartesian_transformation_operator_3d
   SUBTYPE OF (cartesian_transformation_operator);
   axis3 : OPTIONAL direction;
 DERIVE
   u     : LIST[3:3] OF direction
         := base_axis(3,SELF\cartesian_transformation_operator.axis1,
                        SELF\cartesian_transformation_operator.axis2,axis3);
 WHERE
   WR1: SELF\geometric_representation_item.dim = 3;
 END_ENTITY;

ENTITY cc_design_approval
  SUBTYPE OF (approval_assignment);
  items : SET[1:?] OF approved_item;
END_ENTITY;

ENTITY cc_design_certification
  SUBTYPE OF (certification_assignment);
  items : SET[1:?] OF certified_item;
END_ENTITY;

ENTITY cc_design_contract
  SUBTYPE OF (contract_assignment);
  items : SET[1:?] OF contracted_item;
END_ENTITY;

ENTITY cc_design_date_and_time_assignment
  SUBTYPE OF (date_and_time_assignment);
  items : SET[1:?] OF date_time_item;
END_ENTITY;

ENTITY cc_design_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET[1:?] OF person_organization_item;
WHERE
  WR1: cc_design_person_and_organization_correlation(SELF);
```

```
END_ENTITY;

ENTITY cc_design_security_classification
  SUBTYPE OF (security_classification_assignment);
  items : SET[1:?] OF classified_item;
END_ENTITY;

ENTITY cc_design_specification_reference
  SUBTYPE OF (document_reference);
  items : SET[1:?] OF specified_item;
END_ENTITY;

ENTITY celsius_temperature_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.THERMODYNAMIC_TEMPERATURE_UNIT'
IN TYPEOF (SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY centre_of_symmetry
   SUBTYPE OF (derived_shape_aspect);
WHERE
    WR1: SIZEOF
         (QUERY(sadr<*SELF\derived_shape_aspect.deriving_relationships|
    NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMMETRIC_SHAPE_ASPECT'

     IN TYPEOF
     (sadr\shape_aspect_relationship.related_shape_aspect))))=0;
END_ENTITY;

ENTITY certification;

  name : label;
  purpose : text;
  kind : certification_type;
END_ENTITY;

ENTITY certification_assignment
   ABSTRACT SUPERTYPE;
  assigned_certification : certification;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;

ENTITY certification_type;
  description : label;
END_ENTITY;

ENTITY change
  SUBTYPE OF (action_assignment);
  items : SET[1:?] OF work_item;
END_ENTITY;
```

```
ENTITY change_request
  SUBTYPE OF (action_request_assignment);
  items : SET[1:?] OF change_request_item;
END_ENTITY;

ENTITY character_glyph_style_outline;
  outline_style : curve_style;
END_ENTITY;

ENTITY character_glyph_style_stroke;
  stroke_style : curve_style;
END_ENTITY;

ENTITY characterized_object;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

 ENTITY circle
   SUBTYPE OF (conic);
   radius   : positive_length_measure;
 END_ENTITY;


ENTITY circular_runout_tolerance
 SUBTYPE OF ( geometric_tolerance_with_datum_reference );
 WHERE
WR1 :
SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 2;
 END_ENTITY;

ENTITY class
  SUBTYPE OF (group);
END_ENTITY;

ENTITY class_by_extension
  SUBTYPE OF (class);
END_ENTITY;

ENTITY class_by_intension
  SUBTYPE OF (class);
END_ENTITY;

ENTITY classification_assignment
   ABSTRACT SUPERTYPE;
  assigned_class : group;
  role : classification_role;
END_ENTITY;

ENTITY classification_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

 ENTITY closed_shell
   SUBTYPE OF (connected_face_set);
```

```
 END_ENTITY;

ENTITY coaxiality_tolerance
 SUBTYPE OF ( geometric_tolerance_with_datum_reference );
 WHERE
WR1 :
SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 2;
 END_ENTITY;

ENTITY colour;
END_ENTITY;

ENTITY colour_rgb
  SUBTYPE OF (colour_specification);
  red   : REAL;
  green : REAL;
  blue  : REAL;
WHERE
  WR1: {0.0 <= red <= 1.0};
  WR2: {0.0 <= green <= 1.0};
  WR3: {0.0 <= blue <= 1.0};
END_ENTITY;

ENTITY colour_specification
  SUBTYPE OF (colour);
  name : label;
END_ENTITY;

ENTITY common_datum
 SUBTYPE OF ( composite_shape_aspect , datum );
 WHERE
WR1 :
SIZEOF (SELF.component_relationships) = 2;
WR2 :
SIZEOF (
QUERY ( sar <* SELF.component_relationships| NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DATUM' IN TYPEOF
(sar.related_shape_aspect)) AND NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMMON_DATUM' IN TYPEOF
(sar.related_shape_aspect))) )) = 0;

 END_ENTITY;

 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;
```

```
 ENTITY composite_curve_on_surface
   SUPERTYPE OF (
               BOUNDARY_CURVE)
  SUBTYPE OF (composite_curve);

 DERIVE
   basis_surface : SET[0:2] OF surface :=
                 get_basis_surface(SELF);
 WHERE
   WR1: SIZEOF(basis_surface) > 0;
   WR2: constraints_composite_curve_on_surface(SELF);
 END_ENTITY;

 ENTITY composite_curve_segment
 SUBTYPE OF (founded_item);
   transition    : transition_code;
   same_sense    : BOOLEAN;
   parent_curve  : curve;
 INVERSE
   using_curves  : BAG[1:?] OF composite_curve FOR segments;
 WHERE
   WR1 : ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BOUNDED_CURVE' IN
TYPEOF(parent_curve));

 END_ENTITY;

ENTITY composite_shape_aspect
  SUBTYPE OF (shape_aspect);
INVERSE
  component_relationships : SET [2:?] OF shape_aspect_relationship
                            FOR relating_shape_aspect;
END_ENTITY;

ENTITY composite_text
  SUBTYPE OF (geometric_representation_item);
  collected_text : SET[2:?] of text_or_character;
WHERE
  WR1: acyclic_composite_text( SELF, SELF.collected_text);
END_ENTITY;

ENTITY composite_text_with_associated_curves
  SUBTYPE OF (composite_text);
  associated_curves : SET[1:?] of curve;
END_ENTITY;

ENTITY composite_text_with_blanking_box
  SUBTYPE OF (composite_text);
  blanking : planar_box;
END_ENTITY;

ENTITY composite_text_with_delineation
  SUBTYPE OF (composite_text);
  delineation : text_delineation;
END_ENTITY;

ENTITY composite_text_with_extent
```

```
  SUBTYPE OF (composite_text);
  extent : planar_extent;
END_ENTITY;

ENTITY compound_representation_item
  SUBTYPE OF (representation_item);
  item_element : compound_item_definition;
END_ENTITY;

ENTITY concentricity_tolerance
 SUBTYPE OF ( geometric_tolerance_with_datum_reference );
 WHERE
WR1 :
SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) = 1;
 END_ENTITY;

ENTITY configuration_design;
  configuration : configuration_item;
  design        : configuration_design_item;
DERIVE
  name          : label := get_name_value (SELF);
  description   : text := get_description_value (SELF);
UNIQUE
  UR1: configuration, design;
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                             'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

                             'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY;

ENTITY configuration_effectivity
  SUBTYPE OF (product_definition_effectivity);
  configuration : configuration_design;
UNIQUE
  UR1: configuration,
       usage,
       id;
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_DEFINITION_USAGE' IN

        TYPEOF (SELF\product_definition_effectivity.usage);
END_ENTITY;

ENTITY configuration_item;
   id                  :  identifier;
   name                :  label;
   description         :  OPTIONAL text;
   item_concept        :  product_concept;
   purpose             :  OPTIONAL label;
END_ENTITY;

 ENTITY conic
   SUPERTYPE OF (ONEOF (
              CIRCLE,
              ELLIPSE,
```

```
              HYPERBOLA,
              PARABOLA))
  SUBTYPE OF (curve);
    position: axis2_placement;
 END_ENTITY;

 ENTITY
 conical_surface
   SUBTYPE OF (elementary_surface);
   radius     : length_measure;
   semi_angle : plane_angle_measure;
 WHERE
   WR1: radius >= 0.0;
 END_ENTITY;

 ENTITY connected_edge_set
   SUBTYPE OF (topological_representation_item);
   ces_edges : SET [1:?] OF edge;
 END_ENTITY;

 ENTITY connected_face_set
   SUPERTYPE OF (ONEOF (
              CLOSED_SHELL,
              OPEN_SHELL))
  SUBTYPE OF (topological_representation_item);
   cfs_faces : SET [1:?] OF face;
 END_ENTITY;

ENTITY constructive_geometry_representation
  SUBTYPE OF (representation);
WHERE
  WR1:
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_CONTEXT' IN
TYPEOF(SELF.context_of_items)) AND ({2 <=
SELF.context_of_items\geometric_representation_context. coordinate_space_dimension
<= 3});
  WR2: SIZEOF( QUERY( cgr_i <* SELF.items |
SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLACEMENT',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE_SURFACE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT'] * TYPEOF(cgr_i)) <> 1 ))
= 0;
  WR3: SIZEOF( USEDIN( SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'REPRESENTATION_RELATIONSHIP.REP_2') ) > 0;
  WR4: SIZEOF( USEDIN( SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'REPRESENTATION_MAP.MAPPED_REPRESENTATION') ) = 0;

END_ENTITY;

ENTITY constructive_geometry_representation_relationship
  SUBTYPE OF (representation_relationship);
WHERE
```

```
  WR1: (SELF.rep_1.context_of_items :=: SELF.rep_2.context_of_items) AND
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_CONTEXT' IN
TYPEOF(SELF.rep_1.context_of_items));
  WR2:
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONSTRUCTIVE_GEOMETRY_REPRESENTATION'
IN TYPEOF(SELF.rep_2);
  WR3: SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_REPRESENTATION',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONSTRUCTIVE_GEOMETRY_REPRESENTATION']
* TYPEOF(SELF.rep_1)) = 1;
  WR4: NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION' IN TYPEOF(SELF));

END_ENTITY;

ENTITY context_dependent_invisibility
  SUBTYPE OF (invisibility);
  presentation_context : invisibility_context;
END_ENTITY;

ENTITY context_dependent_shape_representation;
  representation_relation : shape_representation_relationship;
  represented_product_relation : product_definition_shape;
DERIVE
  description : text := get_description_value (SELF);
  name : label := get_name_value (SELF);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_DEFINITION_RELATIONSHIP'
IN TYPEOF (SELF.represented_product_relation.definition);
  WR2: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  WR3: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;

END_ENTITY;


ENTITY context_dependent_unit
  SUBTYPE OF (named_unit);
  name : label;
END_ENTITY;

ENTITY contract;
  name : label;
  purpose : text;
  kind : contract_type;
END_ENTITY;

ENTITY contract_assignment
   ABSTRACT SUPERTYPE;
  assigned_contract : contract;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;
```

```
ENTITY contract_type;
  description : label;
END_ENTITY;

ENTITY conversion_based_unit
  SUBTYPE OF (named_unit);
  name : label;
  conversion_factor : measure_with_unit;
END_ENTITY;

ENTITY coordinated_universal_time_offset;
  hour_offset : INTEGER;
  minute_offset : OPTIONAL INTEGER;
  sense : ahead_or_behind;
DERIVE
  actual_minute_offset : INTEGER := NVL(minute_offset,0);
WHERE
  WR1: { 0 <= hour_offset < 24 };
  WR2: { 0 <= actual_minute_offset <= 59 };
  WR3: NOT (((hour_offset <> 0) OR (actual_minute_offset <>0)) AND (sense =
exact));
END_ENTITY;

ENTITY csg_shape_representation
 SUBTYPE OF ( shape_representation );
 WHERE
WR1 :
SELF.context_of_items\geometric_representation_context.coordinate_space_dimension =
3;
WR2 :
SIZEOF (
QUERY ( it <* SELF.items| ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CSG_SOLID',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SOLID_REPLICA',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REVOLVED_FACE_SOLID',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EXTRUDED_FACE_SOLID' ] * TYPEOF (it))
<> 1) )) = 0;
WR3 :
SIZEOF (
QUERY ( mi <*
QUERY ( it <* SELF.items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
IN TYPEOF (it)) )| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CSG_SHAPE_REPRESENTATION' IN TYPEOF
(mi\mapped_item.mapping_source.mapped_representation)) )) = 0;
WR4 :
SIZEOF (
QUERY ( sr <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SOLID_REPLICA' IN TYPEOF (it)) )| (
SIZEOF ([ 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CSG_SOLID',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REVOLVED_FACE_SOLID',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EXTRUDED_FACE_SOLID' ] * TYPEOF
(sr\solid_replica.parent_solid)) = 0) )) = 0;
WR5 :
```

```
SIZEOF (
QUERY ( it <* SELF.items| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D' IN TYPEOF (it)) ))
> 0;

 END_ENTITY;

 ENTITY csg_solid
   SUBTYPE OF (solid_model);
   tree_root_expression : csg_select;
 END_ENTITY;

ENTITY curve
   SUPERTYPE OF (ONEOF (
               LINE,
               CONIC,
               PCURVE,
               SURFACE_CURVE,
               OFFSET_CURVE_2D,
               OFFSET_CURVE_3D,
               CURVE_REPLICA))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY;

 ENTITY curve_bounded_surface
    SUBTYPE OF (bounded_surface);
    basis_surface    : surface;
    boundaries       : SET [1:?] OF boundary_curve;

    implicit_outer   : BOOLEAN;
  WHERE
    WR1: (NOT implicit_outer) OR
         (SIZEOF (QUERY (temp <* boundaries |
           'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OUTER_BOUNDARY_CURVE' IN
TYPEOF(temp))) = 0);
    WR2: (NOT(implicit_outer)) OR
                ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BOUNDED_SURFACE' IN
TYPEOF(basis_surface));
    WR3: SIZEOF(QUERY(temp <* boundaries |
                  'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OUTER_BOUNDARY_CURVE'
IN

                                          TYPEOF(temp))) <= 1;
    WR4: SIZEOF(QUERY(temp <* boundaries |
            (temp\composite_curve_on_surface.basis_surface [1] <>
                                          SELF.basis_surface))) = 0;
 END_ENTITY;

ENTITY curve_dimension
  SUBTYPE OF (dimension_curve_directed_callout);
END_ENTITY;

 ENTITY curve_replica
   SUBTYPE OF (curve);
   parent_curve   : curve;
   transformation : cartesian_transformation_operator;
 WHERE
```

```
   WR1: transformation.dim = parent_curve.dim;
   WR2: acyclic_curve_replica (SELF, parent_curve);
 END_ENTITY;

ENTITY curve_style;
  name         : label;
  curve_font   : curve_font_or_scaled_curve_font_select;
  curve_width  : size_select;
  curve_colour : colour;
END_ENTITY;


ENTITY curve_style_font;
  name         : label;
  pattern_list : LIST [1:?] OF curve_style_font_pattern;
END_ENTITY;

ENTITY curve_style_font_pattern;
  visible_segment_length   : positive_length_measure;
  invisible_segment_length : positive_length_measure;
END_ENTITY;

 ENTITY cylindrical_surface
   SUBTYPE OF (elementary_surface);
   radius : positive_length_measure;
 END_ENTITY;

ENTITY cylindricity_tolerance
 SUBTYPE OF ( geometric_tolerance );
 WHERE
WR1 : NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF));

 END_ENTITY;

ENTITY date
  SUPERTYPE OF (ONEOF (
              CALENDAR_DATE,
              ORDINAL_DATE,
              WEEK_OF_YEAR_AND_DAY_DATE));
  year_component : year_number;
END_ENTITY;

ENTITY date_and_time;
  date_component : date;
  time_component : local_time;
END_ENTITY;

ENTITY date_and_time_assignment
   ABSTRACT SUPERTYPE;
  assigned_date_and_time : date_and_time;
  role : date_time_role;
END_ENTITY;

ENTITY date_assignment
   ABSTRACT SUPERTYPE;
  assigned_date : date;
```

```
  role : date_role;
END_ENTITY;

ENTITY date_role;
  name : label;
DERIVE
  description : text := get_description_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;

END_ENTITY;

ENTITY date_time_role;
  name : label;
DERIVE
  description : text := get_description_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;

END_ENTITY;

ENTITY dated_effectivity
  SUBTYPE OF (effectivity);
  effectivity_end_date : OPTIONAL date_time_or_event_occurrence;
  effectivity_start_date : date_time_or_event_occurrence;
END_ENTITY;

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\datum.established_by_relationships |
       SIZEOF (TYPEOF(x\shape_aspect_relationship.relating_shape_aspect)*
       ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DATUM_FEATURE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DATUM_TARGET']) <> 1))=0;

END_ENTITY;

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,
     'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_ASPECT_RELATIONSHIP.'+
       'RELATING_SHAPE_ASPECT'))
       | NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DATUM' IN TYPEOF

       (sar\shape_aspect_relationship.related_shape_aspect))))=0;
  WR2: SELF\shape_aspect.product_definitional = TRUE;
END_ENTITY;
```

```
ENTITY datum_feature_callout
  SUBTYPE OF (draughting_callout);
END_ENTITY;

ENTITY datum_reference;
  precedence       : INTEGER;
  referenced_datum : datum;
WHERE
  WR1: precedence > 0;
END_ENTITY;

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,
     'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT'))
      | NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DATUM' IN TYPEOF

      (sar\shape_aspect_relationship.related_shape_aspect))))=0;
   WR2: SELF\shape_aspect.product_definitional = TRUE;
END_ENTITY;

ENTITY datum_target_callout
  SUBTYPE OF (draughting_callout);
END_ENTITY;

ENTITY default_tolerance_table
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF( QUERY( i <* SELF.items |
NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEFAULT_TOLERANCE_TABLE_CELL' IN
TYPEOF(i)) )) = 0;
  WR2: (SIZEOF( QUERY( rr <*
USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'REPRESENTATION_RELATIONSHIP.REP_1') | rr.name < 'general tolerance definition' ))
= 0) AND (SIZEOF( QUERY( rr <*
USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'REPRESENTATION_RELATIONSHIP.REP_1') | (rr.name = 'general tolerance definition')
AND (rr.rep_2.name < 'default tolerance') )) = 0) AND (SIZEOF(
USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'REPRESENTATION_RELATIONSHIP.REP_2') ) = 0);

END_ENTITY;

ENTITY default_tolerance_table_cell
  SUBTYPE OF (compound_representation_item);
WHERE
  WR1: SIZEOF(QUERY( x <* USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'
+ 'REPRESENTATION.ITEMS') | 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DEFAULT_TOLERANCE_TABLE' IN TYPEOF(x)))=1 ;
```

```
  WR2:
default_tolerance_table_cell_wr2(SELF\compound_representation_item.item_element);
  WR3:
default_tolerance_table_cell_wr3(SELF\compound_representation_item.item_element);
  WR4:
default_tolerance_table_cell_wr4(SELF\compound_representation_item.item_element);
  WR5:
default_tolerance_table_cell_wr5(SELF\compound_representation_item.item_element);
END_ENTITY;

ENTITY defined_character_glyph
  SUBTYPE OF(geometric_representation_item);
  definition : defined_glyph_select;
  placement  : axis2_placement;
END_ENTITY;

ENTITY defined_symbol
  SUBTYPE OF(geometric_representation_item);
  definition : defined_symbol_select;
  target     : symbol_target;
END_ENTITY;

ENTITY defined_table
  SUBTYPE OF(defined_symbol);
END_ENTITY;

ENTITY definitional_representation
  SUBTYPE OF (representation);
WHERE
  WR1:
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PARAMETRIC_REPRESENTATION_CONTEXT' IN

        TYPEOF (SELF\representation.context_of_items );
END_ENTITY;

 ENTITY degenerate_pcurve
   SUBTYPE OF (point);
   basis_surface:  surface;
   reference_to_curve : definitional_representation;
 WHERE
   WR1: SIZEOF(reference_to_curve\representation.items) = 1;
   WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE' IN TYPEOF

                    (reference_to_curve\representation.items[1]);
   WR3: reference_to_curve\representation.
                   items[1]\geometric_representation_item.dim =2;
 END_ENTITY;

 ENTITY degenerate_toroidal_surface
   SUBTYPE OF (toroidal_surface);
   select_outer : BOOLEAN;
 WHERE
  WR1: major_radius <   minor_radius;
 END_ENTITY;

ENTITY derived_shape_aspect
  SUPERTYPE OF (ONEOF (
```

```
              APEX,
              CENTRE_OF_SYMMETRY,
              GEOMETRIC_ALIGNMENT,
              GEOMETRIC_INTERSECTION,
              PARALLEL_OFFSET,
              PERPENDICULAR_TO,
              EXTENSION,
              TANGENT))
  SUBTYPE OF (shape_aspect);
INVERSE
  deriving_relationships : SET [1:?] OF
          shape_aspect_relationship FOR relating_shape_aspect;
WHERE
   WR1: SIZEOF (QUERY (dr <*
          SELF\derived_shape_aspect.deriving_relationships |
          NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

               'SHAPE_ASPECT_DERIVING_RELATIONSHIP'
          IN TYPEOF (dr)))) = 0;
END_ENTITY;

ENTITY derived_unit;
  elements : SET[1:?] OF derived_unit_element;
DERIVE
  name : label := get_name_value (SELF);
WHERE
  WR1: (SIZEOF (elements) > 1) OR ((SIZEOF (elements) = 1) AND
(elements[1].exponent <> 1.0));
  WR2: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;

END_ENTITY;

ENTITY derived_unit_element;
  unit : named_unit;
  exponent : REAL;
END_ENTITY;

ENTITY description_attribute;
  attribute_value : text;
  described_item : description_attribute_select;
END_ENTITY;

ENTITY descriptive_representation_item
  SUBTYPE OF (representation_item);
  description : text;
END_ENTITY;

ENTITY design_context
  SUBTYPE OF (product_definition_context);
WHERE
  WR1: SELF.life_cycle_stage = 'design';
END_ENTITY;

ENTITY design_make_from_relationship
  SUBTYPE OF (product_definition_relationship);
END_ENTITY;
```

```
ENTITY diameter_dimension
  SUBTYPE OF (dimension_curve_directed_callout);
END_ENTITY;

ENTITY dimension_callout_component_relationship
  SUBTYPE OF (draughting_callout_relationship);
WHERE
  WR1: SELF.name IN ['prefix', 'suffix'];
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STRUCTURED_DIMENSION_CALLOUT'
          IN TYPEOF (SELF.relating_draughting_callout);
  WR3: SIZEOF (TYPEOF (SELF.related_draughting_callout) *
       ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_DIRECTED_CALLOUT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROJECTION_DIRECTED_CALLOUT',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIMENSION_CURVE_DIRECTED_CALLOUT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STRUCTURED_DIMENSION_CALLOUT'])
= 0;
  WR4: SELF.related_draughting_callout.contents *
       SELF.relating_draughting_callout.contents =
       SELF.related_draughting_callout.contents;
  WR5: ((SELF.name = 'prefix') AND
       (SIZEOF (QUERY (ato <* QUERY (con <*
                 SELF.related_draughting_callout.contents |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
             IN TYPEOF(con))) |
         NOT (ato.name = 'prefix text')
       )) = 0));

  WR6: ((SELF.name = 'suffix') AND
       (SIZEOF (QUERY (ato <* QUERY (con <*
                 SELF.related_draughting_callout.contents |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'

             IN TYPEOF(con))) |
         NOT (ato.name = 'suffix text')
       )) = 0));
END_ENTITY;

ENTITY dimension_callout_relationship
  SUBTYPE OF (draughting_callout_relationship);
WHERE
  WR1: SELF.name IN ['primary', 'secondary'];
  WR2: SIZEOF (TYPEOF (SELF.relating_draughting_callout) *
       ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANGULAR_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIAMETER_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_DIRECTED_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINEAR_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORDINATE_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.RADIUS_DIMENSION'])>=1;
  WR3: SIZEOF (TYPEOF (SELF.related_draughting_callout) *

['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIMENSION_CURVE_DIRECTED_CALLOUT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROJECTION_DIRECTED_CALLOUT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_DIRECTED_CALLOUT']) = 0;
```

```
  WR4: SELF.related_draughting_callout.contents *
       SELF.relating_draughting_callout.contents =
       SELF.related_draughting_callout.contents;
END_ENTITY;

ENTITY dimension_curve
  SUBTYPE OF (annotation_curve_occurrence);
WHERE
  wr1: (SIZEOF(
          QUERY(dct <* USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                'TERMINATOR_SYMBOL.ANNOTATED_CURVE')
               | (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                  'DIMENSION_CURVE_TERMINATOR' IN TYPEOF(dct))
                  ))
          ) <= 2);
   wr2:  SIZEOF(
            QUERY( dcdc <*
USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                   'DRAUGHTING_CALLOUT.CONTENTS') |
                   ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                    'DIMENSION_CURVE_DIRECTED_CALLOUT' IN TYPEOF(dcdc)))
        )>= 1;
   wr3: (SIZEOF(
            QUERY(dct1 <* USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'
+
                  'TERMINATOR_SYMBOL.ANNOTATED_CURVE')
                | (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                   'DIMENSION_CURVE_TERMINATOR' IN TYPEOF(dct1))
                   AND (dct1.role = origin)))
         ) <= 1)
     AND
        (SIZEOF(
            QUERY (dct2 <*
USEDIN(SELF,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                   'TERMINATOR_SYMBOL.ANNOTATED_CURVE')
                 | (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+

                   'DIMENSION_CURVE_TERMINATOR' IN TYPEOF(dct2))
                   AND (dct2.role = target)))
       ) <= 1);
END_ENTITY;

ENTITY dimension_curve_directed_callout
  SUBTYPE OF (draughting_callout);
WHERE
  WR1: SIZEOF(QUERY(d_c<*SELF\draughting_callout.contents |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIMENSION_CURVE' IN
(TYPEOF(d_c))))=1;

  WR2: SIZEOF(SELF\draughting_callout.contents) >= 2;
END_ENTITY;

ENTITY dimension_curve_terminator
  SUBTYPE OF (terminator_symbol);
  role  : dimension_extent_usage;
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIMENSION_CURVE' IN TYPEOF
```

```
       (SELF\terminator_symbol.annotated_curve);
END_ENTITY;

ENTITY dimension_pair
  SUBTYPE OF (draughting_callout_relationship);
WHERE
  WR1: SELF.name IN ['chained', 'parallel'];
  WR2: SIZEOF (TYPEOF (SELF.relating_draughting_callout) *
       ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANGULAR_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIAMETER_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINEAR_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORDINATE_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.RADIUS_DIMENSION'])=1;
  WR3: SIZEOF (TYPEOF (SELF.related_draughting_callout) *
       ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANGULAR_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIAMETER_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINEAR_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORDINATE_DIMENSION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.RADIUS_DIMENSION'])=1;

END_ENTITY;

ENTITY dimension_related_tolerance_zone_element;
  related_dimension : dimensional_location;
  related_element   : tolerance_zone_definition;
END_ENTITY;

ENTITY dimension_text_associativity
  SUBTYPE OF (text_literal, mapped_item);
WHERE
  WR1:
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_DIMENSION_REPRESENTATION'
          IN TYPEOF (SELF\mapped_item.
                       mapping_source.mapped_representation));
  WR2:  ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_CALLOUT'
          IN TYPEOF (SELF\mapped_item.mapping_target));
  WR3:  SIZEOF (QUERY (ato <* QUERY (si <*
                USEDIN (SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM.ITEM') |

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
                    IN TYPEOF(si))) |
                NOT (SIZEOF( QUERY (dc <*
                   USEDIN (ato, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                                'DRAUGHTING_CALLOUT.CONTENTS') |

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_CALLOUT'

                   IN TYPEOF (dc)))
                  * [SELF\mapped_item.mapping_target]) = 1)
                )) = 0;
END_ENTITY;

ENTITY dimensional_characteristic_representation;
```

```
  dimension      : dimensional_characteristic;
  representation : shape_dimension_representation;
END_ENTITY;

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;

ENTITY dimensional_location
  SUPERTYPE OF (ONEOF (
              ANGULAR_LOCATION,
              DIMENSIONAL_LOCATION_WITH_PATH))
  SUBTYPE OF (shape_aspect_relationship);
END_ENTITY;

ENTITY dimensional_location_with_path
  SUBTYPE OF (dimensional_location);
  path : shape_aspect;
END_ENTITY;

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;

ENTITY dimensional_size_with_path
  SUBTYPE OF (dimensional_size);
  path : shape_aspect;
END_ENTITY;

ENTITY directed_action
  SUBTYPE OF (executed_action);
  directive : action_directive;
END_ENTITY;

ENTITY directed_dimensional_location
  SUBTYPE OF (dimensional_location);
END_ENTITY;

 ENTITY direction
   SUBTYPE OF (geometric_representation_item);
   direction_ratios : LIST [2:3] OF REAL;
 WHERE
   WR1: SIZEOF(QUERY(tmp <* direction_ratios | tmp <> 0.0)) > 0;
 END_ENTITY;
```

```
ENTITY document;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  kind : document_type;
INVERSE
  representation_types : SET[0:?] OF document_representation_type FOR
represented_document;
END_ENTITY;

ENTITY document_file
SUBTYPE OF (document, characterized_object);
WHERE
  WR1: SELF\characterized_object.name = '';
  WR2: NOT EXISTS(SELF\characterized_object.description);
  WR3: SIZEOF( QUERY( drt <* SELF\document.representation_types |
               drt.name IN ['digital','physical'])) = 1;
END_ENTITY;

ENTITY document_product_association;
  name : label;
  description : OPTIONAL text;
  relating_document : document;
  related_product : product_or_formation_or_definition;
END_ENTITY;

ENTITY document_product_equivalence
  SUBTYPE OF (document_product_association);
WHERE
  WR1: SELF\document_product_association.name = 'equivalence';
  WR2: NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT' IN
TYPEOF(SELF\document_product_association.related_product)) OR
((SELF\document_product_association.relating_document.kind. product_data_type =
'configuration controlled document') AND (SIZEOF( QUERY( prpc <*
USEDIN(SELF\document_product_association.related_product,'CONFIGURATION_CONTROL_3D_
DESIGN_ED2_MIM_LF.' + 'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | prpc.name =
'document' )) = 1));
  WR3:
NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_DEFINITION_FORMATION' IN
TYPEOF(SELF.related_product)) OR
((SELF\document_product_association.relating_document.kind.product_data_type =
'configuration controlled document version') AND (SIZEOF( QUERY( prpc <*
USEDIN(SELF.related_product\product_definition_formation.of_product,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUC
TS') | prpc.name = 'document')) = 1));
  WR4: NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_DEFINITION' IN
TYPEOF(SELF.related_product)) OR
((SELF\document_product_association.relating_document.kind.product_data_type =
'configuration controlled document definition') AND (SIZEOF( QUERY( prpc <*
USEDIN(SELF\document_product_association.related_product\product_definition.formati
on.of_product,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUC
TS') | prpc.name = 'document' )) = 1));

END_ENTITY;

ENTITY document_reference
```

```
   ABSTRACT SUPERTYPE;
  assigned_document : document;
  source : label;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;

ENTITY document_relationship;
  name : label;
  description : OPTIONAL text;
  relating_document : document;
  related_document : document;
END_ENTITY;

ENTITY document_representation_type;
  name : label;
  represented_document : document;
END_ENTITY;

ENTITY document_type;
  product_data_type : label;
END_ENTITY;

ENTITY document_usage_constraint;
  source : document;
  subject_element : label;
  subject_element_value : text;
END_ENTITY;

ENTITY document_usage_constraint_assignment
   ABSTRACT SUPERTYPE;
  assigned_document_usage : document_usage_constraint;
  role : document_usage_role;
END_ENTITY;

ENTITY document_usage_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY document_with_class
  SUBTYPE OF (document);
  class : identifier;
END_ENTITY;

ENTITY draughting_annotation_occurrence
  SUBTYPE OF (annotation_occurrence);
WHERE
  WR1:      -- curve_has_curve_style:
    (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_CURVE_OCCURRENCE'
            IN TYPEOF (SELF))) OR
    (SIZEOF (QUERY (sty <* SELF.styles |
       NOT ((SIZEOF (sty.styles) = 1)
```

```
        AND ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE'
                IN TYPEOF (sty.styles[1])))) = 0);
  WR2:      -- fill_area_has_fill_style:
    (NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_FILL_AREA_OCCURRENCE'
           IN TYPEOF (SELF))) OR    (SIZEOF (QUERY (sty <* SELF.styles |
      NOT ((SIZEOF (sty.styles) = 1)
        AND ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FILL_AREA_STYLE'
                IN TYPEOF (sty.styles[1])))) = 0);
  WR3:      -- styled_fill_boundaries:
    (NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_FILL_AREA_OCCURRENCE'
           IN TYPEOF (SELF))) OR    (SIZEOF (QUERY (bound <*
                    SELF.item\annotation_fill_area.boundaries |
       NOT (SIZEOF (QUERY (si <*
                  USEDIN (bound, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                                 'STYLED_ITEM.ITEM') |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
             'ANNOTATION_CURVE_OCCURRENCE' IN TYPEOF (si)))) > 0))) = 0);
  WR4:      -- symbol_has_symbol_style:
    (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL_OCCURRENCE'
           IN TYPEOF (SELF))) OR    (SIZEOF (QUERY (sty <* SELF.styles |
       NOT ((SIZEOF (sty.styles) = 1)           AND
           (SIZEOF (TYPEOF (sty.styles[1]) *
               ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMBOL_STYLE',
                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.NULL_STYLE']) = 1)) ))
= 0);
  WR5:      -- allowable_symbol_representations:
    (NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL_OCCURRENCE'
            IN TYPEOF (SELF)) AND
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL'
            IN TYPEOF(SELF.item)))) OR
         (SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                   'DRAUGHTING_SYMBOL_REPRESENTATION',
                  'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                  'DRAUGHTING_SUBFIGURE_REPRESENTATION'] *
          TYPEOF (SELF.item\mapped_item.mapping_source.

                   mapped_representation)) = 1);
  WR6:      -- text_has_text_style:
    (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
           IN TYPEOF (SELF))) OR
    (SIZEOF (QUERY (sty <* SELF.styles |
       NOT ((SIZEOF (sty.styles) = 1)
        AND ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_STYLE'
                IN TYPEOF (sty.styles[1])))) = 0);
  WR7:      -- allowable_text:
     (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF (SELF))) OR
         (SIZEOF (TYPEOF(SELF.item) *
             ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT',
              'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL']) = 1);
  WR8:      -- text_not_nested:
    (NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF (SELF)) AND
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'
```

```
            IN TYPEOF (SELF.item)))) OR    (SIZEOF (QUERY (tl <*
                  SELF.item\composite_text.collected_text |
            NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL'
            IN TYPEOF (tl)) )) = 0);
  WR9:      -- text_alignment_literals:
    (NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF (SELF)) AND
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL'
            IN TYPEOF (SELF.item)))) OR (SELF.item\text_literal.alignment
            IN ['baseline left', 'baseline centre', 'baseline right']);
  WR10:     -- text_alignment_composites:
    (NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF (SELF)) AND
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'
            IN TYPEOF (SELF.item)))) OR
         (SIZEOF (QUERY (tl <* QUERY (text <* SELF.
                  item\composite_text.collected_text
            |('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL' IN
TYPEOF(text))) |
          NOT (tl\text_literal.alignment IN
         ['baseline left', 'baseline centre', 'baseline right']) )) = 0);
  WR11:     -- single_text_alignment:
    NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF(SELF)) AND
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'
            IN TYPEOF (SELF.item))) OR check_text_alignment(SELF.item);
  WR12:     -- single_text_font:
    NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF(SELF)) AND
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'
            IN TYPEOF (SELF.item))) OR check_text_font(SELF.item);
  WR13:     -- allowable_text_literals:
    (NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF(SELF)) AND
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'
              IN TYPEOF (SELF.item)))) OR
         (SIZEOF (QUERY (tl <* QUERY (text <*
            SELF.item\composite_text.collected_text |
            ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL' IN TYPEOF
(text))) |
            NOT (SIZEOF (TYPEOF(tl) *
                 ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                  'TEXT_LITERAL_WITH_BLANKING_BOX',
                  'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                  'TEXT_LITERAL_WITH_ASSOCIATED_CURVES']) = 0) )) = 0);
  WR14:     -- styled_text_literal_curves:
    (NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF (SELF)) AND

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL_WITH_ASSOCIATED_CURVES'
            IN TYPEOF (SELF.item)))) OR
    (SIZEOF (QUERY (crv <*
                  SELF.item\text_literal_with_associated_curves.
                  associated_curves |
      NOT (SIZEOF (QUERY (si <*  USEDIN (crv,
            'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM.ITEM') |
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_CURVE_OCCURRENCE'
```

```
           IN TYPEOF (si)) )) > 0) )) = 0);
  WR15:     -- styled_composite_text_curves:
    (NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
            IN TYPEOF (SELF)) AND

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT_WITH_ASSOCIATED_CURVES'
            IN TYPEOF (SELF.item)))) OR
          (SIZEOF (QUERY (crv <*
                SELF.item\composite_text_with_associated_curves.
                associated_curves |
           NOT (SIZEOF (QUERY (si <*  USEDIN (crv,
                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM.ITEM') |

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_CURVE_OCCURRENCE'
          IN TYPEOF (si)) )) > 0) )) = 0);
WR16: -- curve_style_has_width:
    SIZEOF (QUERY (cs <* QUERY (sty <* SELF.styles |
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE' IN TYPEOF
(sty.styles[1])))
      | NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LENGTH_MEASURE_WITH_UNIT'
        IN TYPEOF (cs.styles[1]\curve_style.curve_width)) AND
       ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POSITIVE_LENGTH_MEASURE'
        IN TYPEOF (cs.styles[1]\curve_style.
        curve_width\measure_with_unit.value_component))))) = 0;
WR17: -- tiling_constraints:
    SIZEOF (QUERY (fas <* QUERY (sty <* SELF.styles |
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FILL_AREA_STYLE'
         IN TYPEOF (sty.styles[1]))) |
      NOT ((SIZEOF (QUERY (fs <* fas.styles[1]\fill_area_style.fill_styles
                    |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FILL_AREA_STYLE_TILES'
                      IN TYPEOF (fs)))) <= 1)
       AND (SIZEOF (QUERY (fst <* QUERY (fs <*
                          fas.styles[1]\fill_area_style.fill_styles |
             ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FILL_AREA_STYLE_TILES'
                IN TYPEOF (fs))) |
                NOT (SIZEOF (fst\fill_area_style_tiles.tiles) = 1)
          )) = 0))
    )) = 0;
  WR18:     -- hatching_constraints:
    SIZEOF (QUERY (fas <* QUERY (sty <* SELF.styles |
           ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FILL_AREA_STYLE'
            IN TYPEOF (sty.styles[1]))) |

             NOT (SIZEOF (QUERY (fsh <* QUERY (fs <*
                         fas.styles[1]\fill_area_style.fill_styles |
             ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FILL_AREA_STYLE_HATCHING'
               IN TYPEOF (fs))) |
        NOT (fsh\fill_area_style_hatching.point_of_reference_hatch_line :=:
             fsh\fill_area_style_hatching.pattern_start)  )) = 0)  )) = 0;
  WR19:     -- text_style_constraint:
    SIZEOF (QUERY (ts <* QUERY (sty <* SELF.styles |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_STYLE'
         IN TYPEOF(sty.styles[1]))) |
             NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                  'TEXT_STYLE_WITH_BOX_CHARACTERISTICS'
             IN TYPEOF (ts.styles[1])))) = 0;
```

```
  WR20:     -- text_style_characteristics:
    SIZEOF (QUERY (ts <* QUERY (sty <* SELF.styles |

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_STYLE_WITH_BOX_CHARACTERISTICS'

             IN TYPEOF (sty.styles[1]))) |
           NOT (SIZEOF (ts.styles[1]\text_style_with_box_characteristics.
                characteristics) = 4) )) = 0;
END_ENTITY;

ENTITY draughting_callout
  SUBTYPE OF (geometric_representation_item);
  contents : SET [1:?] OF draughting_callout_element;
END_ENTITY;

ENTITY draughting_callout_relationship;
  name : label;
  description : text;
  relating_draughting_callout : draughting_callout;
  related_draughting_callout : draughting_callout;
END_ENTITY;

ENTITY draughting_elements
  SUBTYPE OF (draughting_callout);
WHERE
  WR1: SIZEOF (QUERY (l_c <* QUERY (con <* SELF.contents |
               ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_CURVE' IN
TYPEOF(con))) |
               NOT (SIZEOF (QUERY (ldc <* USEDIN (l_c,
                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DRAUGHTING_CALLOUT.CONTENTS') |
               ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_DIRECTED_CALLOUT'
               IN TYPEOF (ldc)))) <= 1)))=0;
  WR2: NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIMENSION_CURVE_DIRECTED_CALLOUT'
               IN TYPEOF(SELF)) OR
               (SIZEOF (QUERY (con <* SELF.contents |
               ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROJECTION_CURVE' IN
                TYPEOF (con)))) <= 2);
  WR3: SIZEOF (QUERY (rc <* USEDIN (SELF,
               'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_CALLOUT_' +
               'RELATIONSHIP.RELATING_DRAUGHTING_CALLOUT') |
               ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                'DIMENSION_CALLOUT_RELATIONSHIP' IN TYPEOF (rc)) AND
               (rc.name = 'primary') )) <= 1;
  WR4: SIZEOF (QUERY (rc <* USEDIN (SELF,
               'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_CALLOUT_' +
               'RELATIONSHIP.RELATING_DRAUGHTING_CALLOUT') |
                ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                 'DIMENSION_CALLOUT_RELATIONSHIP' IN TYPEOF (rc)) AND
                 (rc.name = 'secondary') )) <= 1;
  WR5: SIZEOF (QUERY (sec <* QUERY (rc <* USEDIN (SELF,
                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_CALLOUT_' +
                'RELATIONSHIP.RELATING_DRAUGHTING_CALLOUT') |
                ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                 'DIMENSION_CALLOUT_RELATIONSHIP' IN TYPEOF (rc)) AND
                 (rc.name = 'secondary') ) |
```

```
         NOT (SIZEOF (QUERY (prim <* USEDIN (SELF,
               'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_CALLOUT_' +
               'RELATIONSHIP.RELATING_DRAUGHTING_CALLOUT') |
               ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

                'DIMENSION_CALLOUT_RELATIONSHIP' IN TYPEOF (prim)) AND
                (prim.name = 'primary') )) = 1))) = 0;
END_ENTITY;

ENTITY draughting_model
  SUBTYPE OF (representation);
UNIQUE
  UR1:  SELF\representation.name;
WHERE
  WR1:  SIZEOF (QUERY (it <* SELF.items |
                 NOT (SIZEOF
(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CAMERA_MODEL'] *
                       TYPEOF (it)) = 1
               ))) = 0;
  WR2:  SIZEOF (QUERY (mi <* QUERY (it <* SELF.items |
                 ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' IN
TYPEOF(it))) |
                 NOT (
                   SIZEOF
(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_REPRESENTATION',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_MODEL'] *
                       TYPEOF (mi\mapped_item.mapping_source.
                              mapped_representation)) = 1
               ))) = 0;
  WR3:  SIZEOF (QUERY (smi <* QUERY (si <* QUERY (it <* SELF.items |
                 ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM' IN
TYPEOF(it))) |
                 ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' IN
                     TYPEOF(si\styled_item.item))) |
                 (NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_REPRESENTATION' IN
                          TYPEOF(smi\styled_item.item\mapped_item.
                                 mapping_source.mapped_representation))
                    AND
                     (SIZEOF (QUERY (sty <* smi\styled_item.styles |
                       (NOT (SIZEOF (QUERY (psa <* sty.styles |
                              (NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE'

                                    IN TYPEOF(psa))))) = 1
                       )))) = 1)))
               )) = 0;
END_ENTITY;

ENTITY draughting_pre_defined_colour
```

```
  SUBTYPE OF (pre_defined_colour);
WHERE
  WR1: SELF.name IN
      ['red',
       'green',
       'blue',
       'yellow',
       'magenta',
       'cyan',
       'black',
       'white'];
END_ENTITY;

ENTITY draughting_pre_defined_curve_font
  SUBTYPE OF (pre_defined_curve_font);
WHERE
  WR1: SELF.name IN
       ['continuous',
        'chain',
        'chain double dash',
        'dashed',
        'dotted'];
END_ENTITY;

ENTITY draughting_subfigure_representation
  SUBTYPE OF (symbol_representation);
WHERE
  WR1: SIZEOF (QUERY (item <* SELF\representation.items |
         NOT (SIZEOF
(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_OCCURRENCE',
              'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_CALLOUT',
              'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT']
               * TYPEOF (item)) = 1))) = 0;
  WR2: SIZEOF (QUERY (item <* SELF\representation.items |
         SIZEOF
(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_OCCURRENCE',
                  'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_CALLOUT']
*
           TYPEOF (item)) = 1)) >= 1;
  WR3: SIZEOF (QUERY (srm <* QUERY (rm <*
         USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                       'REPRESENTATION_MAP.MAPPED_REPRESENTATION') |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMBOL_REPRESENTATION_MAP'
            IN TYPEOF(rm))) |
           NOT (SIZEOF (QUERY (a_s <* QUERY (mi <* srm.map_usage |
           ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL' IN
TYPEOF(mi)))
         | NOT (SIZEOF (QUERY (aso <*
           USEDIN (a_s, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                        'STYLED_ITEM.ITEM') |
         NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SUBFIGURE_OCCURRENCE'

           IN TYPEOF(aso)))) = 0))) = 0))) > 0;
  WR4: NOT (acyclic_mapped_item_usage (SELF));
  WR5: SIZEOF (SELF.context_of_items.representations_in_context) = 1;
END_ENTITY;
```

```
ENTITY draughting_symbol_representation
  SUBTYPE of (symbol_representation);
UNIQUE
  UR1: SELF\representation.name;
WHERE
  WR1:
    SIZEOF (QUERY (item <* SELF\representation.items |
         NOT (SIZEOF (TYPEOF (item) *

['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_CURVE_OCCURRENCE',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL_OCCURRENCE',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_FILL_AREA_OCCURRENCE',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE',
             'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT']) = 1)
       )) = 0;
  WR2:
    SIZEOF (QUERY (item <* SELF\representation.items |
         (SIZEOF (TYPEOF (item) *

['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_CURVE_OCCURRENCE',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL_OCCURRENCE',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_FILL_AREA_OCCURRENCE',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE']) = 1)
       )) >= 1;
  WR3:
    SIZEOF (QUERY (item <* SELF\representation.items |

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SUBFIGURE_OCCURRENCE'
          IN TYPEOF (item))) = 0;
  WR4:
    SIZEOF (QUERY (srm <* QUERY (rm <*
          USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                        'REPRESENTATION_MAP.MAPPED_REPRESENTATION') |
          ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMBOL_REPRESENTATION_MAP'
             IN TYPEOF(rm))) |
         NOT (SIZEOF (QUERY (a_s <* QUERY (mi <* srm.map_usage |
           ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL' IN
TYPEOF(mi))) |
         NOT (SIZEOF (QUERY(aso <*
             USEDIN(a_s, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                         'STYLED_ITEM.ITEM') |
             NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_SYMBOL_OCCURRENCE'

                   IN TYPEOF(aso))
          )) = 0) )) = 0) )) > 0;
  WR5:
    NOT (acyclic_mapped_item_usage (SELF));
  WR6:
    SIZEOF (SELF.context_of_items.representations_in_context) = 1;
```

```
END_ENTITY;

ENTITY draughting_text_literal_with_delineation
  SUBTYPE OF (text_literal_with_delineation);
WHERE
  WR1: SELF.delineation IN ['underline', 'overline'];
END_ENTITY;

 ENTITY edge
   SUPERTYPE OF (ONEOF (
              EDGE_CURVE,
              ORIENTED_EDGE))
  SUBTYPE OF (topological_representation_item);
   edge_start : vertex;
   edge_end   : vertex;
 END_ENTITY;

 ENTITY edge_based_wireframe_model
   SUBTYPE OF (geometric_representation_item);
   ebwm_boundary  : SET [1:?] OF connected_edge_set;
 END_ENTITY;

ENTITY edge_based_wireframe_shape_representation
 SUBTYPE OF ( shape_representation );
 WHERE
WR1 :
SIZEOF (
QUERY ( it <* SELF.items| NOT ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_BASED_WIREFRAME_MODEL',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D' ] * TYPEOF (it)) =
1) )) = 0;
WR2 :
SIZEOF (
QUERY ( it <* SELF.items| ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_BASED_WIREFRAME_MODEL',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' ] * TYPEOF (it)) = 1) ))
>= 1;
WR3 :
SIZEOF (
QUERY ( ebwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( eb <* ebwm\edge_based_wireframe_model.ebwm_boundary| NOT ( SIZEOF (
QUERY ( edges <* eb.ces_edges| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_CURVE' IN TYPEOF (edges)) )) = 0)
)) = 0) )) = 0;
WR4 :
SIZEOF (
QUERY ( ebwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( eb <* ebwm\edge_based_wireframe_model.ebwm_boundary| NOT ( SIZEOF (
QUERY ( pline_edges <*
```

```
QUERY ( edges <* eb.ces_edges|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN TYPEOF
(edges\edge_curve.edge_geometry)) )| NOT ( SIZEOF
(pline_edges\edge_curve.edge_geometry\polyline.points) > 2) )) = 0) )) = 0) )) = 0;
WR5 :
SIZEOF (
QUERY ( ebwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( eb <* ebwm\edge_based_wireframe_model.ebwm_boundary| NOT ( SIZEOF (
QUERY ( edges <* eb.ces_edges| NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
(edges.edge_start)) AND ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT'
IN TYPEOF (edges.edge_end))) )) = 0) )) = 0) )) = 0;
WR6 :
SIZEOF (
QUERY ( ebwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( eb <* ebwm\edge_based_wireframe_model.ebwm_boundary| NOT ( SIZEOF (
QUERY ( edges <* eb.ces_edges| NOT
valid_wireframe_edge_curve(edges\edge_curve.edge_geometry) )) = 0) )) = 0) )) = 0;
WR7 :
SIZEOF (
QUERY ( ebwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( eb <* ebwm\edge_based_wireframe_model.ebwm_boundary| NOT ( SIZEOF (
QUERY ( edges <* eb.ces_edges| NOT
(valid_wireframe_vertex_point(edges.edge_start\vertex_point.vertex_geometry) AND
valid_wireframe_vertex_point(edges.edge_end\vertex_point.vertex_geometry)) )) = 0)
)) = 0) )) = 0;
WR8 :
SIZEOF (
QUERY ( mi <*
QUERY ( it <* SELF.items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
IN TYPEOF (it)) )| NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION' IN TYPEOF
(mi\mapped_item.mapping_source.mapped_representation)) )) = 0;

WR9 :
SELF.context_of_items\geometric_representation_context.coordinate_space_dimension =
3;
 END_ENTITY;

 ENTITY edge_curve
   SUBTYPE OF(edge,geometric_representation_item);
   edge_geometry : curve;
   same_sense    : BOOLEAN;
 END_ENTITY;

 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;

ENTITY effectivity
  SUPERTYPE OF (ONEOF (
              SERIAL_NUMBERED_EFFECTIVITY,
              DATED_EFFECTIVITY,
              LOT_EFFECTIVITY,
              TIME_INTERVAL_BASED_EFFECTIVITY));
  id : identifier;
DERIVE
  name : label := get_name_value(SELF);
  description : text := get_description_value(SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  WR2: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;

END_ENTITY;

ENTITY effectivity_assignment
   ABSTRACT SUPERTYPE;
  assigned_effectivity : effectivity;
DERIVE
  role : object_role := get_role(SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;

ENTITY effectivity_relationship;
  name : label;
  description : OPTIONAL text;

  related_effectivity : effectivity;
  relating_effectivity : effectivity;
END_ENTITY;

ENTITY electric_current_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ELECTRIC_CURRENT_UNIT' IN TYPEOF
(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY electric_current_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 = 1.0) 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 elementary_surface
   SUPERTYPE OF (ONEOF (
               PLANE,
               CYLINDRICAL_SURFACE,
               CONICAL_SURFACE,
               SPHERICAL_SURFACE,
               TOROIDAL_SURFACE))
  SUBTYPE OF (surface);
   position : axis2_placement_3d;
 END_ENTITY;

 ENTITY ellipse
   SUBTYPE OF (conic);
   semi_axis_1 : positive_length_measure;
   semi_axis_2 : positive_length_measure;
 END_ENTITY;

 ENTITY evaluated_degenerate_pcurve
   SUBTYPE OF (degenerate_pcurve);
   equivalent_point : cartesian_point;
 END_ENTITY;

ENTITY event_occurrence;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY event_occurrence_assignment
   ABSTRACT SUPERTYPE;
  assigned_event_occurrence : event_occurrence;
  role : event_occurrence_role;
END_ENTITY;

ENTITY event_occurrence_relationship;
  name : label;
  description : OPTIONAL text;
  relating_event : event_occurrence;
  related_event : event_occurrence;
END_ENTITY;

ENTITY event_occurrence_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY executed_action
  SUBTYPE OF (action);
END_ENTITY;

ENTITY extension
```

```
  SUBTYPE OF (derived_shape_aspect);
WHERE
  WR1: SIZEOF (SELF\derived_shape_aspect.deriving_relationships)= 1;
END_ENTITY;

ENTITY external_class_library
  SUBTYPE OF (external_source);
END_ENTITY;

ENTITY external_identification_assignment
   ABSTRACT SUPERTYPE
  SUBTYPE OF (identification_assignment);
  source : external_source;
END_ENTITY;

ENTITY external_source;
  source_id : source_item;
DERIVE
  description : text := get_description_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;

END_ENTITY;

ENTITY externally_defined_character_glyph
  SUBTYPE OF (externally_defined_item);
END_ENTITY;

ENTITY externally_defined_class
  SUBTYPE OF (class, externally_defined_item);
END_ENTITY;

 ENTITY externally_defined_colour
  SUBTYPE OF (colour_specification,externally_defined_item);
END_ENTITY;

ENTITY externally_defined_curve_font
  SUBTYPE OF (externally_defined_item);
END_ENTITY;

ENTITY externally_defined_dimension_definition
  SUBTYPE OF (dimensional_size, externally_defined_item);
WHERE
  WR1: (SELF\externally_defined_item.item_id = 'external size dimension') AND
(SELF\externally_defined_item.source.source_id = 'external size dimension
specification');
  WR2: 1 >= SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
(adr.assigned_document.description = 'external size dimension specification') ));

END_ENTITY;

ENTITY externally_defined_hatch_style
  SUBTYPE OF (externally_defined_item, geometric_representation_item);
END_ENTITY;
```

```
ENTITY externally_defined_item;
  item_id : source_item;
  source : external_source;
END_ENTITY;

ENTITY externally_defined_marker
  SUBTYPE OF (externally_defined_symbol, pre_defined_marker);
END_ENTITY;

ENTITY externally_defined_symbol
  SUBTYPE OF (externally_defined_item);
END_ENTITY;

ENTITY externally_defined_terminator_symbol
  SUBTYPE OF (externally_defined_symbol);
END_ENTITY;

ENTITY externally_defined_text_font
  SUBTYPE OF (externally_defined_item);
END_ENTITY;

ENTITY externally_defined_tile
  SUBTYPE OF (externally_defined_item);
END_ENTITY;

ENTITY externally_defined_tile_style
  SUBTYPE OF (externally_defined_item, geometric_representation_item);
END_ENTITY;

 ENTITY extruded_face_solid
   SUBTYPE OF (swept_face_solid);
   extruded_direction : direction;
   depth              : positive_length_measure;
 WHERE
   WR1: dot_product(
        (SELF\swept_face_solid.swept_face.face_geometry\
        elementary_surface.position.p[3]), extruded_direction) <> 0.0;
 END_ENTITY;

 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 |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE_OUTER_BOUND' IN

                                                TYPEOF(temp))) <= 1;
 END_ENTITY;

 ENTITY face_bound
   SUBTYPE OF(topological_representation_item);
   bound       :  loop;
   orientation :  BOOLEAN;
```

```
 END_ENTITY;

    ENTITY face_outer_bound
    SUBTYPE OF (face_bound);
    END_ENTITY;

 ENTITY face_surface
   SUBTYPE OF(face,geometric_representation_item);
   face_geometry :  surface;
   same_sense    :  BOOLEAN;
 WHERE
   WR1: NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_SURFACE' IN
TYPEOF(face_geometry));

 END_ENTITY;

 ENTITY faceted_brep
   SUBTYPE OF (manifold_solid_brep);
 END_ENTITY;

ENTITY faceted_brep_shape_representation
 SUBTYPE OF ( shape_representation );
 WHERE
WR1 :
SIZEOF (
QUERY ( it <* items| NOT ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACETED_BREP',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D' ] * TYPEOF (it)) =
1) )) = 0;
WR2 :
SIZEOF (
QUERY ( it <* items| ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACETED_BREP',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' ] * TYPEOF (it)) = 1) )) >
0;
WR3 :
SIZEOF (
QUERY ( fbrep <*
QUERY ( it <* items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACETED_BREP' IN
TYPEOF (it)) )| NOT ( SIZEOF (
QUERY ( csh <* msb_shells(fbrep)| NOT ( SIZEOF (
QUERY ( fcs <* csh\connected_face_set.cfs_faces| NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE_SURFACE' IN TYPEOF (fcs)) AND
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLANE' IN TYPEOF
(fcs\face_surface.face_geometry)) AND
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT' IN TYPEOF
(fcs\face_surface.face_geometry\elementary_surface.position.location)))) )) = 0) ))
= 0) )) = 0;
WR4 :
SIZEOF (
QUERY ( fbrep <*
QUERY ( it <* items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACETED_BREP' IN
TYPEOF (it)) )| NOT ( SIZEOF (
QUERY ( csh <* msb_shells(fbrep)| NOT ( SIZEOF (
QUERY ( fcs <* csh\connected_face_set.cfs_faces| NOT ( SIZEOF (
```

```
QUERY ( bnds <* fcs.bounds|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE_OUTER_BOUND' IN TYPEOF (bnds)) ))
= 1) )) = 0) )) = 0) )) = 0;
WR5 :
SIZEOF (
QUERY ( msb <*
QUERY ( it <* items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MANIFOLD_SOLID_BREP' IN TYPEOF (it))
)| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_CLOSED_SHELL' IN TYPEOF
(msb\manifold_solid_brep.outer)) )) = 0;
WR6 :
SIZEOF (
QUERY ( brv <*
QUERY ( it <* items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BREP_WITH_VOIDS'
IN TYPEOF (it)) )| NOT ( SIZEOF (
QUERY ( csh <* brv\brep_with_voids.voids| csh\oriented_closed_shell.orientation ))
= 0) )) = 0;
WR7 :
SIZEOF (
QUERY ( mi <*
QUERY ( it <* items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' IN
TYPEOF (it)) )| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACETED_BREP_SHAPE_REPRESENTATION' IN
TYPEOF (mi\mapped_item.mapping_source.mapped_representation)) )) = 0;

 END_ENTITY;

ENTITY fill_area_style;
  name       : label;
  fill_styles : SET [1:?] OF fill_style_select;
WHERE
  WR1: SIZEOF(QUERY(fill_style <* SELF.fill_styles |
         'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+

         'FILL_AREA_STYLE_COLOUR' IN
         TYPEOF(fill_style)
         )) <= 1;
END_ENTITY;

ENTITY fill_area_style_colour;
  name        : label;
  fill_colour : colour;
END_ENTITY;

ENTITY fill_area_style_hatching
  SUBTYPE OF (geometric_representation_item);
  hatch_line_appearance         : curve_style;
  start_of_next_hatch_line      : one_direction_repeat_factor;
  point_of_reference_hatch_line : cartesian_point;
  pattern_start                 : cartesian_point;
  hatch_line_angle              : plane_angle_measure;
END_ENTITY;

ENTITY fill_area_style_tile_coloured_region
  SUBTYPE OF (geometric_representation_item);
  closed_curve  : curve_or_annotation_curve_occurrence;
  region_colour : colour;
```

```
END_ENTITY;

ENTITY fill_area_style_tile_curve_with_style
  SUBTYPE OF (geometric_representation_item);
  styled_curve : annotation_curve_occurrence;
END_ENTITY;

ENTITY fill_area_style_tile_symbol_with_style
  SUBTYPE OF (geometric_representation_item);
  symbol : annotation_symbol_occurrence;
END_ENTITY;

ENTITY fill_area_style_tiles
  SUBTYPE OF (geometric_representation_item);
  tiling_pattern : two_direction_repeat_factor;
  tiles          : SET [1:?] OF fill_area_style_tile_shape_select;
  tiling_scale   : positive_ratio_measure;
END_ENTITY;

ENTITY flatness_tolerance
 SUBTYPE OF ( geometric_tolerance );
 WHERE
WR1 : NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF));

 END_ENTITY;

ENTITY founded_item;
END_ENTITY;

ENTITY functionally_defined_transformation;
  name        : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY general_property;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY general_property_association;
  name : label;
  description : OPTIONAL text;
  base_definition : general_property;
  derived_definition : derived_property_select;
WHERE
  WR1: SIZEOF( USEDIN( derived_definition,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' + 'GENERAL_PROPERTY_ASSOCIATION.' +
'DERIVED_DEFINITION')) = 1;

  WR2: derived_definition.name = base_definition.name;
END_ENTITY;

ENTITY general_property_relationship;
  name : label;
  description : OPTIONAL text;
```

```
  relating_property : general_property;
  related_property : general_property;
END_ENTITY;

ENTITY geometric_alignment
  SUBTYPE OF (derived_shape_aspect);
WHERE
  WR1: SIZEOF (SELF\derived_shape_aspect.deriving_relationships)> 1;
END_ENTITY;

 ENTITY geometric_curve_set
   SUBTYPE OF (geometric_set);
 WHERE
   WR1: SIZEOF(QUERY(temp <* SELF\geometric_set.elements |
                           'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE' IN
TYPEOF(temp))) = 0;

 END_ENTITY;

ENTITY geometric_intersection
  SUBTYPE OF (derived_shape_aspect);
WHERE
  WR1: SIZEOF (SELF\derived_shape_aspect.deriving_relationships)> 1;
END_ENTITY;

ENTITY geometric_representation_context
  SUBTYPE OF (representation_context);
  coordinate_space_dimension : dimension_count;
END_ENTITY;

  ENTITY geometric_representation_item
  SUPERTYPE OF (ONEOF (
              POINT,
              DIRECTION,
              VECTOR,
              PLACEMENT,
              CARTESIAN_TRANSFORMATION_OPERATOR,
              CURVE,
              SURFACE,
              EDGE_CURVE,
              FACE_SURFACE,
              POLY_LOOP,
              VERTEX_POINT,
              SOLID_MODEL,
              BOOLEAN_RESULT,
              SPHERE,
              RIGHT_CIRCULAR_CONE,
              RIGHT_CIRCULAR_CYLINDER,
              TORUS,
              BLOCK,
              RIGHT_ANGULAR_WEDGE,
              HALF_SPACE_SOLID,
              SHELL_BASED_SURFACE_MODEL,
              SHELL_BASED_WIREFRAME_MODEL,
              EDGE_BASED_WIREFRAME_MODEL,
              GEOMETRIC_SET))
  SUBTYPE OF (representation_item);
```

```
  DERIVE
    dim : dimension_count := dimension_of(SELF);
  WHERE
  WR1: SIZEOF (QUERY (using_rep <* using_representations (SELF) |
      NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_CONTEXT' IN

      TYPEOF (using_rep.context_of_items)))) = 0;
  END_ENTITY;

 ENTITY geometric_set

  SUBTYPE OF (geometric_representation_item);
   elements :  SET [1:?] OF geometric_set_select;
 END_ENTITY;

ENTITY geometric_tolerance;
  name                    : label;
  description             : text;
  magnitude               : measure_with_unit;
  toleranced_shape_aspect : shape_aspect;
WHERE
  WR1: ('NUMBER' IN TYPEOF
       (magnitude\measure_with_unit.value_component)) AND
       (magnitude\measure_with_unit.value_component >= 0.0);
END_ENTITY;

ENTITY geometric_tolerance_relationship;
                                  name                         :label;
                                  description                  :text;
                                  relating_geometric_tolerance :
geometric_tolerance;
                                  related_geometric_tolerance  :
geometric_tolerance;
END_ENTITY;

ENTITY geometric_tolerance_with_datum_reference
   SUBTYPE OF (geometric_tolerance);
   datum_system : SET [1:?] OF  datum_reference;
END_ENTITY;

ENTITY geometric_tolerance_with_defined_unit
  SUBTYPE OF (geometric_tolerance);
  unit_size : measure_with_unit;
WHERE
  WR1: ('NUMBER' IN TYPEOF
       (unit_size\measure_with_unit.value_component)) AND
       (unit_size\measure_with_unit.value_component > 0.0);
END_ENTITY;

ENTITY geometrical_tolerance_callout
  SUBTYPE OF (draughting_callout);
END_ENTITY;

ENTITY geometrically_bounded_surface_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
```

```
  WR1: SIZEOF(QUERY(it <* SELF.items | NOT
(SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_SET',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) =
1))) = 0;
  WR2: SIZEOF(QUERY(it <* SELF.items |
SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_SET',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'] * TYPEOF(it)) = 1)) > 0;
  WR3: SIZEOF(QUERY(mi <* QUERY(it <* SELF.items |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' IN TYPEOF(it)) | NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION' IN
TYPEOF(mi\mapped_item.mapping_source.mapped_representation)) AND
(SIZEOF(QUERY(mr_it <* mi\mapped_item.mapping_source.mapped_representation.items |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_SET' IN TYPEOF(mr_it)))) >
0)))) = 0;
  WR4: SIZEOF(QUERY(gs <* QUERY(it <* SELF.items |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_SET' IN TYPEOF(it)) | NOT
(SIZEOF(QUERY(pnt <* QUERY(gsel <* gs\geometric_set.elements |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT' IN TYPEOF(gsel)) | NOT
(gbsf_check_point(pnt)))) = 0))) = 0;
  WR5: SIZEOF(QUERY(gs <* QUERY(it <* SELF.items |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_SET' IN TYPEOF(it)) | NOT
(SIZEOF(QUERY(cv <* QUERY(gsel <* gs\geometric_set.elements |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE' IN TYPEOF(gsel)) | NOT
(gbsf_check_curve(cv)))) = 0))) = 0;
  WR6: SIZEOF(QUERY(gs <* QUERY(it <* SELF.items |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_SET' IN TYPEOF(it)) | NOT
(SIZEOF(QUERY(sf <* QUERY(gsel <* gs\geometric_set.elements |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE' IN TYPEOF(gsel)) | NOT
(gbsf_check_surface(sf)))) = 0))) = 0;
  WR7: SIZEOF(QUERY(gs <* QUERY(it <* SELF.items |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_SET' IN TYPEOF(it)) |
SIZEOF(QUERY(gsel <* gs\geometric_set.elements |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE' IN TYPEOF(gsel))) > 0)) > 0;

END_ENTITY;

ENTITY geometrically_bounded_wireframe_shape_representation
 SUBTYPE OF ( shape_representation );
 WHERE
WR1 :
SIZEOF (
QUERY ( it <* SELF.items| NOT ( SIZEOF ( TYPEOF (it) * [
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_CURVE_SET',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' ]) = 1) )) = 0;
WR2 :

SIZEOF (
QUERY ( it <* SELF.items| ( SIZEOF ( TYPEOF (it) * [
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_CURVE_SET',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' ]) = 1) )) >= 1;
WR3 :
SIZEOF (
QUERY ( gcs <*
```

```
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_CURVE_SET' IN TYPEOF (it))
)| NOT ( SIZEOF (
QUERY ( crv <*
QUERY ( elem <* gcs\geometric_set.elements|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE' IN TYPEOF (elem)) )| NOT
valid_geometrically_bounded_wf_curve(crv) )) = 0) )) = 0;
WR4 :
SIZEOF (
QUERY ( gcs <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_CURVE_SET' IN TYPEOF (it))
)| NOT ( SIZEOF (
QUERY ( pnts <*
QUERY ( elem <* gcs\geometric_set.elements|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT' IN TYPEOF (elem)) )| NOT
valid_geometrically_bounded_wf_point(pnts) )) = 0) )) = 0;
WR5 :
SIZEOF (
QUERY ( gcs <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_CURVE_SET' IN TYPEOF (it))
)| NOT ( SIZEOF (
QUERY ( cnc <*
QUERY ( elem <* gcs\geometric_set.elements|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONIC' IN TYPEOF (elem)) )| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D' IN TYPEOF
(cnc\conic.position)) )) = 0) )) = 0;
WR6 :
SIZEOF (
QUERY ( gcs <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_CURVE_SET' IN TYPEOF (it))
)| NOT ( SIZEOF (
QUERY ( pline <*
QUERY ( elem <* gcs\geometric_set.elements|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN TYPEOF (elem)) )| NOT (
SIZEOF (pline\polyline.points) > 2) )) = 0) )) = 0;
WR7 :
SIZEOF (
QUERY ( mi <*
QUERY ( it <* SELF.items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
IN TYPEOF (it)) )| NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION' IN TYPEOF
(mi\mapped_item.mapping_source.mapped_representation)) )) = 0;

 END_ENTITY;

ENTITY global_uncertainty_assigned_context
  SUBTYPE OF (representation_context);
  uncertainty : SET [1:?] OF uncertainty_measure_with_unit;
END_ENTITY;

ENTITY global_unit_assigned_context
  SUBTYPE OF (representation_context);
  units : SET[1:?] OF unit;
END_ENTITY;
```

```
ENTITY group;
  name : label;
  description : OPTIONAL text;
DERIVE
  id : identifier := get_id_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;

END_ENTITY;

ENTITY group_assignment
   ABSTRACT SUPERTYPE;
  assigned_group : group;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;

ENTITY group_relationship;
  name : label;
  description : OPTIONAL text;
  relating_group : group;
  related_group : group;
END_ENTITY;

 ENTITY half_space_solid
   SUBTYPE OF(geometric_representation_item);
   base_surface   : surface;
   agreement_flag : BOOLEAN;
 END_ENTITY;

 ENTITY hyperbola
   SUBTYPE OF (conic);
   semi_axis      : positive_length_measure;
   semi_imag_axis : positive_length_measure;
 END_ENTITY;

ENTITY id_attribute;
  attribute_value : identifier;
  identified_item : id_attribute_select;
END_ENTITY;

ENTITY identification_assignment
   ABSTRACT SUPERTYPE;
  assigned_id : identifier;
  role : identification_role;
END_ENTITY;

ENTITY identification_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

```
 ENTITY intersection_curve
   SUBTYPE OF (surface_curve);
 WHERE
   WR1: SIZEOF(SELF\surface_curve.associated_geometry) = 2;
   WR2: associated_surface(SELF\surface_curve.associated_geometry[1]) <>
             associated_surface(SELF\surface_curve.associated_geometry[2]);
 END_ENTITY;

ENTITY invisibility;
  invisible_items : SET [1:?] OF invisible_item;
END_ENTITY;

ENTITY item_defined_transformation;
  name             : label;
  description      : OPTIONAL text;
  transform_item_1 : representation_item;
  transform_item_2 : representation_item;
END_ENTITY;

ENTITY language
SUBTYPE OF (group);
WHERE
WR1: SELF\group.name <> '';
END_ENTITY;

ENTITY leader_curve
  SUBTYPE OF (annotation_curve_occurrence);
WHERE
  WR1: SIZEOF(
          QUERY(ldc <* USEDIN( SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'
+
                       'DRAUGHTING_CALLOUT.CONTENTS')
                   |   'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

                       'LEADER_DIRECTED_CALLOUT'  IN TYPEOF(ldc))) >= 1;
END_ENTITY;

ENTITY leader_directed_callout
  SUBTYPE OF (draughting_callout);
WHERE
  WR1: SIZEOF (QUERY (l_1 <* SELF\draughting_callout.contents |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_CURVE' IN (TYPEOF(l_1))))
>= 1;

  WR2: SIZEOF(SELF\draughting_callout.contents) >=2;
END_ENTITY;

ENTITY leader_directed_dimension
  SUBTYPE OF (leader_directed_callout);
WHERE
  WR1: SIZEOF (QUERY (con <* SELF.contents |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_CURVE' IN TYPEOF
(con)))=1;

END_ENTITY;
```

```
ENTITY leader_terminator
  SUBTYPE OF (terminator_symbol);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_CURVE' IN TYPEOF

       (SELF\terminator_symbol.annotated_curve);
END_ENTITY;

ENTITY length_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LENGTH_UNIT' IN TYPEOF
(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY length_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 1.0) 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 limits_and_fits;
  form_variance   : label;
  zone_variance   : label;
  grade           : label;
  source          : text;
END_ENTITY;

 ENTITY line
   SUBTYPE OF (curve);
   pnt : cartesian_point;
   dir : vector;
 WHERE
   WR1: dir.dim  = pnt.dim;
 END_ENTITY;

ENTITY line_profile_tolerance
 SUBTYPE OF ( geometric_tolerance );
 WHERE
WR1 : NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF)) OR ( SIZEOF
(SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);
WR2 :
SIZEOF (
QUERY ( sar <* USEDIN (SELF\geometric_tolerance.toleranced_shape_aspect,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')| (sar.name IN [ 'affected plane
association', 'resulting intersection curve association' ]) )) = 1;

 END_ENTITY;
```

```
ENTITY linear_dimension
  SUBTYPE OF (dimension_curve_directed_callout);
END_ENTITY;

ENTITY local_time;
  hour_component : hour_in_day;
  minute_component : OPTIONAL minute_in_hour;
  second_component : OPTIONAL second_in_minute;
  zone : coordinated_universal_time_offset;
WHERE
  WR1: valid_time (SELF);
END_ENTITY;

 ENTITY loop
   SUPERTYPE OF (ONEOF (
               VERTEX_LOOP,
               EDGE_LOOP,
               POLY_LOOP))
  SUBTYPE OF (topological_representation_item);
 END_ENTITY;

ENTITY lot_effectivity
  SUBTYPE OF (effectivity);
  effectivity_lot_id : identifier;
  effectivity_lot_size : measure_with_unit;
END_ENTITY;

ENTITY luminous_intensity_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LUMINOUS_INTENSITY_UNIT' IN
TYPEOF(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY luminous_intensity_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 = 1.0);
END_ENTITY;

ENTITY make_from_usage_option
  SUBTYPE OF (product_definition_usage);
  ranking           : INTEGER;
  ranking_rationale : text;
  quantity          : measure_with_unit;
WHERE
  WR1: (NOT ('NUMBER' IN TYPEOF(quantity.value_component)))
       OR (quantity.value_component > 0);
END_ENTITY;
```

```
 ENTITY manifold_solid_brep
   SUBTYPE OF (solid_model);
   outer : closed_shell;
 END_ENTITY;

ENTITY manifold_surface_shape_representation
   SUBTYPE OF (shape_representation);
WHERE
  WR1: SIZEOF (QUERY (it <* SELF.items |
       NOT (SIZEOF
(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D'] * TYPEOF
(it)) = 1))) = 0;
  WR2: SIZEOF (QUERY (it <* SELF.items |
       SIZEOF
(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'] * TYPEOF (it)) =
1)) > 0;
  WR3: SIZEOF (QUERY (mi <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' IN TYPEOF (it)) |
       NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MANIFOLD_SURFACE_SHAPE_REPRESENTATION
'
       IN TYPEOF (mi\mapped_item.mapping_source.mapped_representation))
       AND
       (SIZEOF(QUERY (mr_it <*
       mi\mapped_item.mapping_source.mapped_representation.items |
       ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL'
       IN TYPEOF (mr_it)))) > 0 )))) = 0;
  WR4: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (sh <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OPEN_SHELL',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_CLOSED_SHELL',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CLOSED_SHELL']
       * TYPEOF (sh)) = 1))) = 0))) = 0;
  WR5: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE_SURFACE' IN TYPEOF
(fa)) )) = 0)))
       = 0))) = 0;
  WR6: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
```

```
       OR
       (msf_surface_check(fa\face_surface.face_geometry))))) = 0)))
       = 0))) = 0;
  WR7: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
       OR
       (SIZEOF (QUERY (bnds <* fa.bounds |
       NOT (SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_LOOP']
       * TYPEOF (bnds.bound)) = 1))) = 0)))) = 0))) = 0))) = 0;
  WR8: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items|
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
       OR
       (SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF
(bnds.bound)) |
       NOT (SIZEOF (QUERY (oe <* elp_fbnds\path.edge_list |
       NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_CURVE' IN TYPEOF
       (oe.edge_element)))) = 0))) = 0)))) = 0))) = 0))) = 0;
  WR9: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
       OR
       (SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF
(bnds.bound)) |
       NOT (SIZEOF (QUERY (oe_cv <* QUERY (oe <*
       elp_fbnds\path.edge_list |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_CURVE' IN TYPEOF
(oe.edge_element)) |
       NOT (SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_CURVE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONIC',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_REPLICA',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_CURVE_3D',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_CURVE'] *
       TYPEOF (oe_cv.edge_element\edge_curve.edge_geometry))
       = 1))) = 0))) = 0)))) = 0))) = 0))) = 0;
```

```
 WR10: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
       OR
       (SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF
(bnds.bound)) |
       NOT (SIZEOF (QUERY (oe <* elp_fbnds\path.edge_list |
       NOT (msf_curve_check (oe.edge_element\edge_curve.edge_geometry))))
       = 0))) = 0)))) = 0))) = 0))) = 0;
 WR11: SIZEOF (QUERY(sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
       OR
       (SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF
(bnds.bound)) |
       NOT (SIZEOF (QUERY (oe <* elp_fbnds\path.edge_list|
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
       (oe.edge_element.edge_start))
       AND
       ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN
       TYPEOF (oe.edge_element.edge_end)))))
       = 0))) = 0)))) = 0))) = 0))) = 0;
 WR12: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
       OR
       (SIZEOF (QUERY (elp_fbnds <* QUERY (bnds <* fa.bounds |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF
(bnds.bound)) |
       NOT (SIZEOF (QUERY (oe <* elp_fbnds\path.edge_list |
       NOT ((SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEGENERATE_PCURVE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_CURVE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_SURFACE'] * TYPEOF
       (oe.edge_element.edge_start\vertex_point.vertex_geometry)) = 1)
       AND
       (SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEGENERATE_PCURVE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_CURVE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_SURFACE'] * TYPEOF
```

```
       (oe.edge_element.edge_end\vertex_point.vertex_geometry)) = 1
       )))) = 0))) = 0)))) = 0))) = 0))) = 0;
 WR13: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
       OR
       (SIZEOF (QUERY (vlp_fbnds <* QUERY (bnds <* fa.bounds |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_LOOP' IN TYPEOF
(bnds.bound)) |
       NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
       (vlp_fbnds\vertex_loop.loop_vertex)))) = 0)))) = 0)))
       = 0))) = 0;
 WR14: SIZEOF (QUERY (sbsm <* QUERY (it <* SELF.items |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_SURFACE_MODEL' IN
TYPEOF (it)) |
       NOT (SIZEOF (QUERY (cfs <*
       sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF (QUERY (fa <* cfs\connected_face_set.cfs_faces |
       NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ADVANCED_FACE' IN TYPEOF
(fa))
       OR
       (SIZEOF (QUERY (vlp_fbnds <* QUERY (bnds <* fa.bounds |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_LOOP' IN TYPEOF
(bnds.bound)) |
       NOT (SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEGENERATE_PCURVE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_CURVE',
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_SURFACE'] * TYPEOF

       (vlp_fbnds\vertex_loop.loop_vertex\vertex_point.vertex_geometry))
       = 1))) = 0)))) = 0))) = 0))) = 0;
END_ENTITY;

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;

ENTITY mass_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MASS_UNIT' IN TYPEOF
(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY mass_unit
  SUBTYPE OF (named_unit);
WHERE
```

```
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
(SELF\named_unit.dimensions.mass_exponent = 1.0) 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 measure_qualification;
  name : label;
  description : text;
  qualified_measure : measure_with_unit;
  qualifiers : SET [1:?] OF value_qualifier;
WHERE
  WR1: SIZEOF(QUERY(temp <* qualifiers |
             'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRECISION_QUALIFIER'

             IN TYPEOF(temp))) < 2;
END_ENTITY;

ENTITY measure_representation_item
  SUBTYPE OF (representation_item, measure_with_unit);
END_ENTITY;

ENTITY measure_with_unit
  SUPERTYPE OF (ONEOF (
              LENGTH_MEASURE_WITH_UNIT,
              MASS_MEASURE_WITH_UNIT,
              TIME_MEASURE_WITH_UNIT,
              ELECTRIC_CURRENT_MEASURE_WITH_UNIT,
              THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT,
              CELSIUS_TEMPERATURE_MEASURE_WITH_UNIT,
              AMOUNT_OF_SUBSTANCE_MEASURE_WITH_UNIT,
              LUMINOUS_INTENSITY_MEASURE_WITH_UNIT,
              PLANE_ANGLE_MEASURE_WITH_UNIT,
              SOLID_ANGLE_MEASURE_WITH_UNIT,
              AREA_MEASURE_WITH_UNIT,
              VOLUME_MEASURE_WITH_UNIT,
              RATIO_MEASURE_WITH_UNIT));
  value_component : measure_value;
  unit_component : unit;
WHERE
  WR1: valid_units (SELF);
END_ENTITY;

ENTITY mechanical_context
  SUBTYPE OF (product_context);
WHERE
  WR1: SELF.discipline_type = 'mechanical';
END_ENTITY;

ENTITY mechanical_design_geometric_presentation_area
  SUBTYPE OF (presentation_area);
WHERE
  WR1:  -- only presentation_views or axis2_placements in
        -- mechanical_design_geometric_presentation_area
```

```
      SIZEOF(QUERY(it1 <* SELF.items |
      NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT'
      IN TYPEOF(it1))
      OR
      (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
      IN TYPEOF(it1)) AND
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRESENTATION_VIEW'
      IN TYPEOF
      (it1\mapped_item.mapping_source.mapped_representation)))))) = 0;
WR2:  -- only mechanical_design_geometric_presentation_representation
      -- via camera_image_3d_with_scale or axis2_placements in
      -- presentation_views
      SIZEOF(QUERY(pv <* QUERY(mi1 <* QUERY(it1 <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
      IN TYPEOF(it1)) |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRESENTATION_VIEW'
      IN TYPEOF
      (mi1\mapped_item.mapping_source.mapped_representation)) |
      -- search in all presentation_views for axis2_placements and
      -- mapped_items and for the subtype of mapped_item
      -- camera_image_3d_with_scale; the latter shall reference
      -- a mechanical_design_geometric_presentation_representation;
      -- the supertype mapped_item shall reference presentation_view.
      NOT (SIZEOF(QUERY(it2 <* pv\mapped_item.mapping_source.
      mapped_representation\representation.items |
      NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT'
      IN TYPEOF(it2))
      OR
      (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
      IN TYPEOF(it2)) AND NOT
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'CAMERA_IMAGE_3D_WITH_SCALE' IN TYPEOF(it2))) AND NOT (
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRESENTATION_VIEW'
      IN TYPEOF
      (it2\mapped_item.mapping_source.mapped_representation)))
      OR
      (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'CAMERA_IMAGE_3D_WITH_SCALE' IN TYPEOF(it2))
      AND NOT (
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION'
      IN TYPEOF (it2\mapped_item.mapping_source.mapped_representation) ))
      ))) = 0))) = 0;
WR3:  (SIZEOF(QUERY(ps <* USEDIN (SELF\presentation_area,
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'PRESENTATION_SIZE.UNIT') | ((ps.size\planar_extent.size_in_x <= 0)
      OR
      (ps.size\planar_extent.size_in_y <= 0)))) = 0)
      AND
      (SIZEOF(QUERY(ais <* USEDIN (SELF\presentation_area,
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'AREA_IN_SET.AREA') |
      (SIZEOF(QUERY(ps <* USEDIN (ais,
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'PRESENTATION_SIZE.UNIT') |
      ((ps.size\planar_extent.size_in_x <= 0)
      OR
```

```
        (ps.size\planar_extent.size_in_y <= 0)))) > 0))) = 0);
  WR4:  (SIZEOF(QUERY(ps <* USEDIN (SELF\presentation_area,
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'PRESENTATION_SIZE.UNIT') |
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'AXIS2_PLACEMENT_2D' IN TYPEOF (ps.size.placement)))) = 1)
        AND
        (SIZEOF(QUERY(ps <* USEDIN (SELF\presentation_area,
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'PRESENTATION_SIZE.UNIT') |
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'AXIS2_PLACEMENT_3D' IN TYPEOF (ps.size.placement)))) = 0)
        OR
        ((SIZEOF(QUERY(ais <* USEDIN (SELF\presentation_area,
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'AREA_IN_SET.AREA') |
        (SIZEOF(QUERY(ps <* USEDIN (ais,
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'PRESENTATION_SIZE.UNIT') |
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'AXIS2_PLACEMENT_2D' IN TYPEOF (ps.size.placement)))) = 1))) = 1)
        AND
        (SIZEOF(QUERY(ais <* USEDIN (SELF\presentation_area,
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'AREA_IN_SET.AREA') |
        (SIZEOF(QUERY(ps <* USEDIN (ais,
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'PRESENTATION_SIZE.UNIT') |
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

        'AXIS2_PLACEMENT_3D' IN TYPEOF (ps.size.placement)))) = 0))) = 1));
END_ENTITY;

ENTITY mechanical_design_geometric_presentation_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF(QUERY(it <* SELF.items |
        NOT (SIZEOF(
        ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CAMERA_MODEL_D3']
        * TYPEOF(it)) = 1))) = 0;
  WR2:  -- only shape_representations and
        -- mechanical_design_geometric_presentation_representations
        -- shall be referenced from mapped_items
        SIZEOF(QUERY(mi <* QUERY(it <* SELF.items |
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
        IN TYPEOF(it))) | NOT (SIZEOF(
        ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

        'SHAPE_REPRESENTATION',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION']
        * TYPEOF(mi\mapped_item.mapping_source.mapped_representation))
        = 1))) = 0;
  WR3:  -- a mapped_item that is styled shall reference a
```

```
      -- shape_representation
      SIZEOF(QUERY(smi <* QUERY(si <* QUERY(it <* SELF.items |
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it))) |
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
      IN TYPEOF(si\styled_item.item))) | NOT (
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SHAPE_REPRESENTATION' IN TYPEOF (smi\styled_item.
      item\mapped_item.mapping_source.mapped_representation))) )) = 0;
WR4:  SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(pss <* psa.styles | NOT (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_STYLE',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE']
      * TYPEOF(pss)) = 1))) = 0))) = 0))) = 0;
WR5:  SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) |
      NOT (SIZEOF(QUERY(psbc <* QUERY(psa <* si\styled_item.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'PRESENTATION_STYLE_BY_CONTEXT' IN TYPEOF(psa)) | NOT (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'REPRESENTATION_ITEM',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REPRESENTATION']
      * TYPEOF(psbc\presentation_style_by_context.style_context))
      = 1))) = 0))) = 0;
WR6:  SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'

      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(ps <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_STYLE'
      IN TYPEOF(pss)) | NOT
      (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'POSITIVE_LENGTH_MEASURE' IN TYPEOF (ps\point_style.marker_size))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COLOUR_RGB',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_COLOUR']
      * TYPEOF(ps\point_style.marker_colour))
      = 1)))) = 0))) = 0))) = 0;
WR7:  SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(cs <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE'
      IN TYPEOF(pss)) | NOT((SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COLOUR_RGB',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_COLOUR']
      * TYPEOF(cs\curve_style.curve_colour)) = 1)
      AND
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'POSITIVE_LENGTH_MEASURE' IN TYPEOF (cs\curve_style.curve_width))
      AND (SIZEOF(
```

```
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE_FONT',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_CURVE_FONT']
      * TYPEOF(cs\curve_style.curve_font)) = 1)))) = 0))) = 0))) = 0;
WR8:  SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE'
      IN TYPEOF(pss)) |
      NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_SIDE_STYLE' IN TYPEOF
      (ssu\surface_style_usage.style)))) = 0))) = 0))) = 0;
WR9:  SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE'
      IN TYPEOF(pss)) | NOT (SIZEOF(QUERY(sses <*
      ssu\surface_style_usage.style\surface_side_style.styles |
      NOT (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_PARAMETER_LINE',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_CONTROL_GRID',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_SILHOUETTE',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_SEGMENTATION_CURVE',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_FILL_AREA',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_BOUNDARY']
      * TYPEOF(sses)) = 1))) = 0))) = 0))) = 0))) = 0;
WR10: SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE'
      IN TYPEOF(pss)) | NOT (SIZEOF(QUERY(sspl <* QUERY(sses <*
      ssu\surface_style_usage.style\surface_side_style.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_PARAMETER_LINE' IN TYPEOF(sses)) |
      NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE'
      IN TYPEOF
      (sspl\surface_style_parameter_line.style_of_parameter_lines))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COLOUR_RGB',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_COLOUR']
      * TYPEOF(sspl\surface_style_parameter_line.
      style_of_parameter_lines\curve_style.curve_colour)) = 1)
      AND (
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'POSITIVE_LENGTH_MEASURE' IN TYPEOF
      (sspl\surface_style_parameter_line.
      style_of_parameter_lines\curve_style.curve_width))
```

```
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE_FONT',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
      'DRAUGHTING_PRE_DEFINED_CURVE_FONT']
      * TYPEOF(sspl\surface_style_parameter_line.
      style_of_parameter_lines\curve_style.curve_font)) = 1))))
      = 0))) = 0))) = 0))) = 0;
WR11: SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE'
      IN TYPEOF(pss)) | NOT (SIZEOF(QUERY(sscg <* QUERY(sses <*
      ssu\surface_style_usage.style\surface_side_style.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_CONTROL_GRID' IN TYPEOF(sses)) |
      NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE'
      IN TYPEOF (sscg\surface_style_control_grid.style_of_control_grid))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COLOUR_RGB',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_COLOUR']
      * TYPEOF(sscg\surface_style_control_grid.
      style_of_control_grid\curve_style.curve_colour)) = 1)
      AND
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'POSITIVE_LENGTH_MEASURE' IN TYPEOF
      (sscg\surface_style_control_grid.
      style_of_control_grid\curve_style.curve_width))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE_FONT',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_CURVE_FONT']
      * TYPEOF(sscg\surface_style_control_grid.
      style_of_control_grid\curve_style.curve_font)) = 1))))
      = 0))) = 0))) = 0))) = 0;
WR12: SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) |
      NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE'
      IN TYPEOF(pss)) | NOT (SIZEOF(QUERY(sssh <* QUERY(sses <*
      ssu\surface_style_usage.style\surface_side_style.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_SILHOUETTE' IN TYPEOF(sses)) |
      NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE'
      IN TYPEOF (sssh\surface_style_silhouette.style_of_silhouette))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COLOUR_RGB',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_COLOUR']
      * TYPEOF(sssh\surface_style_silhouette.
      style_of_silhouette\curve_style.curve_colour)) = 1)
      AND
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'POSITIVE_LENGTH_MEASURE' IN TYPEOF
```

```
      (sssh\surface_style_silhouette.style_of_silhouette\curve_style.
      curve_width))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE_FONT',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_CURVE_FONT']
      * TYPEOF(sssh\surface_style_silhouette.
      style_of_silhouette\curve_style.curve_font)) = 1))))
      = 0))) = 0))) = 0))) = 0;
WR13: SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE'
      IN TYPEOF(pss)) | NOT (SIZEOF(QUERY(sssc <* QUERY(sses <*
      ssu\surface_style_usage.style\surface_side_style.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_SEGMENTATION_CURVE' IN TYPEOF(sses)) |
      NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE'

      IN TYPEOF
      (sssc\surface_style_segmentation_curve.style_of_segmentation_curve))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COLOUR_RGB',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_COLOUR']
      * TYPEOF(sssc\surface_style_segmentation_curve.
      style_of_segmentation_curve\curve_style.curve_colour)) = 1)
      AND
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'POSITIVE_LENGTH_MEASURE' IN TYPEOF
      (sssc\surface_style_segmentation_curve.
      style_of_segmentation_curve\curve_style.curve_width))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE_FONT',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_CURVE_FONT']
      * TYPEOF(sssc\surface_style_segmentation_curve.
      style_of_segmentation_curve\curve_style.curve_font)) = 1))))
      = 0))) = 0))) = 0))) = 0;
WR14: SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM'
      IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa <* si\styled_item.styles |
      NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE'
      IN TYPEOF(pss)) | NOT (SIZEOF(QUERY(ssbd <* QUERY(sses <*
      ssu\surface_style_usage.style\surface_side_style.styles |
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'SURFACE_STYLE_BOUNDARY' IN TYPEOF(sses)) |
      NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE'
      IN TYPEOF (ssbd\surface_style_boundary.style_of_boundary))
      AND (SIZEOF(
      ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COLOUR_RGB',
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
      'DRAUGHTING_PRE_DEFINED_COLOUR']
      * TYPEOF(ssbd\surface_style_boundary.
      style_of_boundary\curve_style.curve_colour)) = 1)
```

```
        AND
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
        'POSITIVE_LENGTH_MEASURE' IN TYPEOF (ssbd\surface_style_boundary.
        style_of_boundary\curve_style.curve_width))
        AND (SIZEOF(
        ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE_FONT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

        'DRAUGHTING_PRE_DEFINED_CURVE_FONT']
        * TYPEOF(ssbd\surface_style_boundary.
        style_of_boundary\curve_style.curve_font)) = 1)))) = 0)))
        = 0))) = 0))) = 0;
END_ENTITY;

ENTITY modified_geometric_tolerance
  SUBTYPE OF (geometric_tolerance);
  modifier : limit_condition;
END_ENTITY;

ENTITY multi_language_attribute_assignment
SUBTYPE OF (attribute_value_assignment);
  items : SET [1:?] OF multi_language_attribute_item;

DERIVE
  translation_language : language :=
language_indication[1]\attribute_classification_assignment.assigned_class;

INVERSE
  language_indication: SET[1:1] OF attribute_language_assignment FOR items;

WHERE
WR1: (SELF\attribute_value_assignment.role.name = 'alternate language');
WR2: SIZEOF( QUERY( ala <* language_indication |
   (ala\attribute_classification_assignment.attribute_name = 'attribute_value') AND
   (ala\attribute_classification_assignment.role.name='translated') )) = 1 ;
WR3: SELF\attribute_value_assignment.attribute_name <> '' ;
WR4: SIZEOF(QUERY(ci <* items |
SIZEOF(QUERY(ata <* USEDIN(ci,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITE
MS') |
(ata\attribute_value_assignment.attribute_name =
SELF\attribute_value_assignment.attribute_name) AND
(ata.translation_language :=: translation_language) ))>1 )) =0;
WR5: SIZEOF(QUERY(ci <* items |
SIZEOF(QUERY(ata <* USEDIN(ci,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ATTRIBUTE_LANGUAGE_ASSIGNMENT.ITEMS') |

 (ata\attribute_classification_assignment.role.name='primary') AND
 (ata\attribute_classification_assignment.attribute_name=
SELF\attribute_value_assignment.attribute_name) AND
 (ata\attribute_classification_assignment.assigned_class :=: translation_language)
))>0 )) =0;
END_ENTITY;

ENTITY name_assignment
   ABSTRACT SUPERTYPE;
  assigned_name : label;
```

```
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1 ;

END_ENTITY;

ENTITY name_attribute;
  attribute_value : label;
  named_item : name_attribute_select;
END_ENTITY;

ENTITY named_unit
  SUPERTYPE OF (ONEOF (
              SI_UNIT,
              CONVERSION_BASED_UNIT,
              CONTEXT_DEPENDENT_UNIT)
              ANDOR ONEOF (
              LENGTH_UNIT,
              MASS_UNIT,
              TIME_UNIT,
              ELECTRIC_CURRENT_UNIT,
              THERMODYNAMIC_TEMPERATURE_UNIT,
              AMOUNT_OF_SUBSTANCE_UNIT,
              LUMINOUS_INTENSITY_UNIT,
              PLANE_ANGLE_UNIT,
              SOLID_ANGLE_UNIT,
              AREA_UNIT,
              VOLUME_UNIT,
              RATIO_UNIT));
  dimensions : dimensional_exponents;
END_ENTITY;

ENTITY next_assembly_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
END_ENTITY;

ENTITY object_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

 ENTITY offset_curve_2d
   SUBTYPE OF (curve);
   basis_curve    : curve;
   distance       : length_measure;
   self_intersect : LOGICAL;
 WHERE
   WR1: basis_curve.dim = 2;
 END_ENTITY;

 ENTITY offset_curve_3d
   SUBTYPE OF (curve);
   basis_curve    : curve;
   distance       : length_measure;
   self_intersect : LOGICAL;
```

```
   ref_direction  : direction;
 WHERE
   WR1 : (basis_curve.dim = 3) AND (ref_direction.dim = 3);
 END_ENTITY;

 ENTITY offset_surface
   SUBTYPE OF (surface);
   basis_surface  : surface;
   distance       : length_measure;
   self_intersect : LOGICAL;
 END_ENTITY;

ENTITY one_direction_repeat_factor
  SUBTYPE OF (geometric_representation_item);
  repeat_factor : vector;
END_ENTITY;

 ENTITY open_shell
   SUBTYPE OF (connected_face_set);
 END_ENTITY;

ENTITY ordinal_date
  SUBTYPE OF (date);
  day_component : day_in_year_number;
WHERE
  WR1: (NOT leap_year(SELF.year_component) AND { 1 <= day_component <= 365 }) OR
(leap_year(SELF.year_component) AND { 1 <= day_component <= 366 });

END_ENTITY;

ENTITY ordinate_dimension
   SUBTYPE OF (projection_directed_callout);
END_ENTITY;

ENTITY organization;
  id : OPTIONAL identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY organization_assignment
   ABSTRACT SUPERTYPE;
  assigned_organization : organization;
  role : organization_role;
END_ENTITY;

ENTITY organization_relationship;
  name : label;
  description : OPTIONAL text;
  relating_organization : organization;
  related_organization : organization;
END_ENTITY;

ENTITY organization_role;
  name : label;
DERIVE
  description : text := get_description_value(SELF);
```

```
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;

END_ENTITY;

ENTITY organizational_address
  SUBTYPE OF (address);
  organizations : SET[1:?] OF organization;
  description : OPTIONAL text;
END_ENTITY;

ENTITY organizational_project;
  name : label;
  description : OPTIONAL text;
  responsible_organizations : SET[1:?] OF organization;
DERIVE
  id : identifier := get_id_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;

END_ENTITY;

ENTITY organizational_project_assignment
   ABSTRACT SUPERTYPE;
  assigned_organizational_project : organizational_project;
  role : organizational_project_role;
END_ENTITY;

ENTITY organizational_project_relationship;
  name : label;
  description : OPTIONAL text;
  relating_organizational_project : organizational_project;
  related_organizational_project : organizational_project;
END_ENTITY;

ENTITY organizational_project_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

 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 ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_CLOSED_SHELL'

                IN TYPEOF (SELF.closed_shell_element));
 END_ENTITY;

 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 ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_EDGE' IN TYPEOF
(SELF.edge_element));

END_ENTITY;

ENTITY oriented_face
  SUBTYPE OF (face);
  face_element : face;
  orientation  : BOOLEAN;
DERIVE
  SELF\face.bounds : SET[1:?] OF face_bound
         := conditional_reverse(SELF.orientation,SELF.face_element.bounds);
WHERE
  WR1: NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_FACE' IN TYPEOF
(SELF.face_element));

END_ENTITY;

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 ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_OPEN_SHELL'

               IN TYPEOF (SELF.open_shell_element));
END_ENTITY;

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 ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_PATH' IN TYPEOF
(SELF.path_element));

END_ENTITY;
```

```
 ENTITY oriented_surface
   SUBTYPE OF (surface);
   orientation : BOOLEAN;
 END_ENTITY;

 ENTITY outer_boundary_curve
   SUBTYPE OF (boundary_curve);
 END_ENTITY;

ENTITY over_riding_styled_item
  SUBTYPE OF (styled_item);
  over_ridden_style : styled_item;
END_ENTITY;

 ENTITY parabola
   SUBTYPE OF (conic);
   focal_dist : length_measure;
 WHERE
   WR1: focal_dist <> 0.0;
 END_ENTITY;

ENTITY parallel_offset
  SUBTYPE OF (derived_shape_aspect);
  offset               : measure_with_unit;
WHERE
  WR1: SIZEOF (SELF\derived_shape_aspect.deriving_relationships)= 1;
END_ENTITY;

ENTITY parallelism_tolerance
 SUBTYPE OF ( geometric_tolerance_with_datum_reference );
 WHERE
WR1 :
SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) < 3;
 END_ENTITY;

ENTITY parametric_representation_context
  SUBTYPE OF (representation_context);
END_ENTITY;

 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;

 ENTITY pcurve
   SUBTYPE OF (curve);
   basis_surface      : surface;
   reference_to_curve : definitional_representation;
 WHERE
   WR1: SIZEOF(reference_to_curve\representation.items) = 1;
   WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE' IN TYPEOF
```

```
                    (reference_to_curve\representation.items[1]);
   WR3: reference_to_curve\representation.items[1]\
                                geometric_representation_item.dim =2;
 END_ENTITY;

ENTITY perpendicular_to
  SUBTYPE OF (derived_shape_aspect);
WHERE
  WR1: SIZEOF (SELF\derived_shape_aspect.deriving_relationships)= 1;
END_ENTITY;

ENTITY perpendicularity_tolerance
 SUBTYPE OF ( geometric_tolerance_with_datum_reference );
 WHERE
WR1 :
SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3;
 END_ENTITY;

ENTITY person;
  id : identifier;
  last_name : OPTIONAL label;
  first_name : OPTIONAL label;
  middle_names : OPTIONAL LIST[1:?] OF label;
  prefix_titles : OPTIONAL LIST[1:?] OF label;
  suffix_titles : OPTIONAL LIST[1:?] OF label;
WHERE
  WR1: EXISTS(last_name) OR EXISTS(first_name);
END_ENTITY;

ENTITY person_and_organization;
  the_person : person;
  the_organization : organization;
DERIVE
  name : label := get_name_value (SELF);
  description : text := get_description_value(SELF);

WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;

END_ENTITY;

ENTITY person_and_organization_address
  SUBTYPE OF (organizational_address, personal_address);
  SELF\organizational_address.organizations : SET[1:1] OF organization;
  SELF\personal_address.people : SET[1:1] OF person;
WHERE
  WR1: SIZEOF(QUERY(pao <* USEDIN (SELF\personal_address.people[1],
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PERSON_AND_ORGANIZATION.THE_PERSON') |
pao.the_organization :=: SELF\organizational_address.organizations[1])) = 1;

END_ENTITY;

ENTITY person_and_organization_assignment
   ABSTRACT SUPERTYPE;
```

```
  assigned_person_and_organization : person_and_organization;
  role : person_and_organization_role;
END_ENTITY;

ENTITY person_and_organization_role;
  name : label;
DERIVE
  description : text := get_description_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;

END_ENTITY;

ENTITY personal_address
  SUBTYPE OF (address);
  people : SET[1:?] OF person;
  description : OPTIONAL text;
END_ENTITY;

ENTITY placed_datum_target_feature
  SUBTYPE OF (datum_target);
DERIVE
  representation_associations : SET[0:?] OF property_definition_representation :=
get_shape_aspect_property_definition_representations(SELF);
WHERE
  WR1: SELF.description IN ['point','line','rectangle','circle', 'circular line'];
  WR2: SIZEOF (QUERY (pdr <* representation_associations |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF (pdr.used_representation) )) = 1;

  WR3: valid_datum_target_parameters(SELF);
END_ENTITY;

 ENTITY placement
   SUPERTYPE OF (ONEOF (
              AXIS1_PLACEMENT,
              AXIS2_PLACEMENT_2D,
              AXIS2_PLACEMENT_3D))
  SUBTYPE OF (geometric_representation_item);
   location : cartesian_point;
 END_ENTITY;

ENTITY planar_box
  SUBTYPE OF (planar_extent);
  placement:  axis2_placement;
END_ENTITY;

ENTITY planar_extent
  SUBTYPE OF (geometric_representation_item);
   size_in_x : length_measure;
   size_in_y : length_measure;
END_ENTITY;

 ENTITY plane
 SUBTYPE OF (elementary_surface);
 END_ENTITY;
```

```
ENTITY plane_angle_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLANE_ANGLE_UNIT' IN TYPEOF
(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY plane_angle_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  plus_minus_tolerance;
  range                : tolerance_method_definition;
  toleranced_dimension : dimensional_characteristic;
UNIQUE
  UR1: toleranced_dimension;
END_ENTITY;

ENTITY point
   SUPERTYPE OF (ONEOF (
               CARTESIAN_POINT,
               POINT_ON_CURVE,
               POINT_ON_SURFACE,
               POINT_REPLICA,
               DEGENERATE_PCURVE))
  SUBTYPE OF (geometric_representation_item);
 END_ENTITY;

 ENTITY point_on_curve
   SUBTYPE OF (point);
   basis_curve     : curve;
   point_parameter : parameter_value;
 END_ENTITY;

 ENTITY point_on_surface
   SUBTYPE OF (point);
   basis_surface     : surface;
   point_parameter_u : parameter_value;
   point_parameter_v : parameter_value;
 END_ENTITY;

 ENTITY point_replica
   SUBTYPE OF (point);
   parent_pt      : point;
   transformation : cartesian_transformation_operator;
 WHERE
   WR1: transformation.dim = parent_pt.dim;
```

```
   WR2: acyclic_point_replica (SELF,parent_pt);
 END_ENTITY;

ENTITY point_style;
  name          : label;
  marker        : marker_select;
  marker_size   : size_select;
  marker_colour : colour;
END_ENTITY;

 ENTITY poly_loop
   SUBTYPE OF (loop,geometric_representation_item);
   polygon : LIST [3:?] OF UNIQUE cartesian_point;
 END_ENTITY;

 ENTITY polyline
   SUBTYPE OF (bounded_curve);
   points : LIST [2:?] OF cartesian_point;
 END_ENTITY;

ENTITY position_tolerance
 SUBTYPE OF ( geometric_tolerance );
 WHERE
WR1 : NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF)) OR ( SIZEOF
(SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);

 END_ENTITY;

ENTITY pre_defined_character_glyph
  SUBTYPE OF (pre_defined_item);
END_ENTITY;

ENTITY pre_defined_colour
  SUBTYPE OF (pre_defined_item, colour);
END_ENTITY;

ENTITY pre_defined_curve_font
  SUBTYPE OF (pre_defined_item);
END_ENTITY;

ENTITY pre_defined_item;
  name : label;
END_ENTITY;

ENTITY pre_defined_marker
  SUBTYPE OF (pre_defined_item);
END_ENTITY;

ENTITY pre_defined_symbol
  SUBTYPE OF (pre_defined_item);
END_ENTITY;

ENTITY pre_defined_terminator_symbol
  SUBTYPE OF (pre_defined_symbol);
WHERE
```

```
  WR1: SELF.name IN ['blanked arrow', 'blanked box', 'blanked dot', 'dimension
origin', 'filled arrow', 'filled box', 'filled dot', 'integral symbol', 'open
arrow', 'slash', 'unfilled arrow'];
END_ENTITY;

ENTITY pre_defined_text_font
  SUBTYPE OF (pre_defined_item);
END_ENTITY;

ENTITY pre_defined_tile
  SUBTYPE OF (pre_defined_item);
END_ENTITY;

ENTITY precision_qualifier;
  precision_value : INTEGER;
END_ENTITY;

ENTITY presentation_area
  SUBTYPE OF (presentation_representation);
WHERE
  WR1: ((SIZEOF (QUERY (ais <* USEDIN (SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                                          'AREA_IN_SET.AREA') |
           SIZEOF (USEDIN (ais, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                                'PRESENTATION_SIZE.UNIT')) =1)) > 0) OR
         (SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

                                'PRESENTATION_SIZE.UNIT')) =1));
END_ENTITY;

ENTITY presentation_layer_assignment;
  name           : label;
  description    : text;
  assigned_items : SET [1:?] OF layered_item;
END_ENTITY;

ENTITY presentation_representation
  SUBTYPE OF (representation);
WHERE
  WR1: SELF\representation.
            context_of_items\geometric_representation_context.
            coordinate_space_dimension = 2;
  WR2:
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRIC_REPRESENTATION_CONTEXT'

       IN TYPEOF (SELF\representation.context_of_items);
END_ENTITY;

ENTITY presentation_set;
INVERSE
  areas : SET [1:?] OF area_in_set FOR in_set;
END_ENTITY;

ENTITY presentation_size;
  unit : presentation_size_assignment_select;
  size : planar_box;
WHERE
```

```
  WR1: (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRESENTATION_REPRESENTATION'
         IN TYPEOF (SELF.unit)) AND
         item_in_context (SELF.size,
                          SELF.unit\representation.context_of_items)
       )
            OR
       (
        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AREA_IN_SET'

          IN TYPEOF (SELF.unit)) AND
        (SIZEOF (QUERY ( ais <* SELF.unit\area_in_set.in_set.areas |
                NOT item_in_context (SELF.size, ais.area\representation.
                                                context_of_items) )) = 0)
       );
END_ENTITY;

ENTITY presentation_style_assignment;
  styles : SET [1:?] OF presentation_style_select;
WHERE
  WR1: SIZEOF (QUERY (style1 <* SELF.styles |
         NOT (SIZEOF (QUERY (style2 <* (SELF.styles - style1) |
           NOT ((TYPEOF (style1) <> TYPEOF (style2)) OR
             (SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                       'SURFACE_STYLE_USAGE',
                       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                       'EXTERNALLY_DEFINED_STYLE'] *
                     TYPEOF (style1)) = 1)
           ))) = 0
         ))) = 0;
  WR2: SIZEOF (QUERY (style1 <* SELF.styles |
         'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_STYLE_USAGE' IN

         TYPEOF(style1)
         )) <= 2;
END_ENTITY;

ENTITY presentation_style_by_context
  SUBTYPE OF (presentation_style_assignment);
  style_context : style_context_select;
END_ENTITY;

ENTITY presentation_view
  SUBTYPE OF (presentation_representation);
END_ENTITY;

ENTITY product;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  frame_of_reference : SET[1:?] OF product_context;
END_ENTITY;

ENTITY product_category;
  name : label;
  description : OPTIONAL text;
DERIVE
  id : identifier := get_id_value (SELF);
```

```
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;

END_ENTITY;

ENTITY product_category_relationship;
  name : label;
  description : OPTIONAL text;
  category : product_category;
  sub_category : product_category;
WHERE
  WR1: acyclic_product_category_relationship (SELF, [SELF.sub_category]);
END_ENTITY;

ENTITY product_concept;
  id                  : identifier;
  name                : label;
  description         : OPTIONAL text;

  market_context      : product_concept_context;
UNIQUE
  UR1: id;
END_ENTITY;

ENTITY product_concept_context
  SUBTYPE OF (application_context_element);
  market_segment_type : label;
END_ENTITY;

ENTITY product_concept_relationship;
  name                     : label;
  description              : OPTIONAL text;
  relating_product_concept : product_concept;
  related_product_concept  : product_concept;
END_ENTITY;

ENTITY product_context
  SUBTYPE OF (application_context_element);
  discipline_type : label;
END_ENTITY;

ENTITY product_definition;
  id : identifier;
  description : OPTIONAL text;
  formation : product_definition_formation;
  frame_of_reference : product_definition_context;
DERIVE
  name : label := get_name_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;

END_ENTITY;

ENTITY product_definition_context
  SUBTYPE OF (application_context_element);
```

```
  life_cycle_stage : label;

END_ENTITY;

ENTITY product_definition_context_association;
  definition : product_definition;
  frame_of_reference : product_definition_context;
  role : product_definition_context_role;
END_ENTITY;

ENTITY product_definition_context_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY product_definition_effectivity
  SUBTYPE OF (effectivity);
  usage : product_definition_relationship;
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY')) = 0;

END_ENTITY;

ENTITY product_definition_formation;
  id : identifier;
  description : OPTIONAL text;
  of_product : product;
UNIQUE
  UR1: id, of_product;
END_ENTITY;

ENTITY product_definition_formation_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_product_definition_formation : product_definition_formation;
  related_product_definition_formation : product_definition_formation;
END_ENTITY;

ENTITY product_definition_formation_with_specified_source
  SUBTYPE OF (product_definition_formation);
  make_or_buy : source;
END_ENTITY;

ENTITY product_definition_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_product_definition : product_definition;
  related_product_definition : product_definition;
END_ENTITY;

ENTITY product_definition_shape
    SUBTYPE OF (property_definition);
  UNIQUE
    UR1: SELF\property_definition.definition;
```

```
  WHERE

    WR1:
SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CHARACTERIZED_PRODUCT_DEFINITIO
N', 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CHARACTERIZED_OBJECT'] *
TYPEOF(SELF\property_definition.definition)) > 0;

END_ENTITY;

ENTITY product_definition_usage
  SUPERTYPE OF (ONEOF (
              MAKE_FROM_USAGE_OPTION,
              ASSEMBLY_COMPONENT_USAGE))
  SUBTYPE OF (product_definition_relationship);
UNIQUE
  UR1: SELF\product_definition_relationship.id,
       SELF\product_definition_relationship.relating_product_definition,
       SELF\product_definition_relationship.related_product_definition;
WHERE
  WR1:  acyclic_product_definition_relationship
         (SELF,
         [SELF\product_definition_relationship.related_product_definition],
         'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_DEFINITION_USAGE');

END_ENTITY;

ENTITY product_definition_with_associated_documents
  SUBTYPE OF (product_definition);
  documentation_ids : SET[1:?] OF document;
END_ENTITY;

ENTITY product_related_product_category
  SUBTYPE OF (product_category);
  products : SET[1:?] OF product;
END_ENTITY;

ENTITY projected_zone_definition
  SUBTYPE OF (tolerance_zone_definition);
  projection_end   : shape_aspect;
  projected_length : measure_with_unit;
WHERE
  WR1: ('NUMBER' IN TYPEOF
       (projected_length\measure_with_unit.value_component)) AND
       (projected_length\measure_with_unit.value_component > 0.0);
  WR2: (derive_dimensional_exponents
       (projected_length\measure_with_unit.unit_component)=
        dimensional_exponents(1,0,0,0,0,0,0));
END_ENTITY;

ENTITY projection_curve
  SUBTYPE OF (annotation_curve_occurrence);
END_ENTITY;

ENTITY projection_directed_callout
  SUBTYPE OF (draughting_callout);
WHERE
  WR1: SIZEOF(QUERY(p_1<*SELF\draughting_callout.contents |
```

```
      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROJECTION_CURVE' IN
(TYPEOF(p_1)))=1;

  WR2: SIZEOF(SELF\draughting_callout.contents) >=2;
END_ENTITY;

ENTITY promissory_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
END_ENTITY;

ENTITY property_definition;
  name : label;
  description : OPTIONAL text;
  definition : characterized_definition;
DERIVE
  id : identifier := get_id_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;

END_ENTITY;

ENTITY property_definition_representation;
  definition : represented_definition;
  used_representation : representation;
DERIVE
  description : text := get_description_value (SELF);
  name : label := get_name_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;

END_ENTITY;

ENTITY qualified_representation_item
  SUBTYPE OF (representation_item);
  qualifiers : SET [1:?] OF value_qualifier;
WHERE
  WR1: SIZEOF(QUERY(temp <* qualifiers |
              'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRECISION_QUALIFIER'

              IN TYPEOF(temp))) < 2;
END_ENTITY;

ENTITY quantified_assembly_component_usage
  SUBTYPE OF (assembly_component_usage);
  quantity   : measure_with_unit;
WHERE
  WR1: (NOT ('NUMBER' IN TYPEOF(quantity.value_component)))
       OR (quantity.value_component > 0);
END_ENTITY;

 ENTITY quasi_uniform_curve
   SUBTYPE OF (b_spline_curve);
 END_ENTITY;
```

```
 ENTITY quasi_uniform_surface
   SUBTYPE OF (b_spline_surface);
 END_ENTITY;

ENTITY radius_dimension
  SUBTYPE OF (dimension_curve_directed_callout);
WHERE
  WR1: SIZEOF (QUERY (con <* SELF.contents |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROJECTION_CURVE' IN TYPEOF
(con)))<=1;

END_ENTITY;

ENTITY ratio_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.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;

 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;

 ENTITY rectangular_composite_surface
   SUBTYPE OF (bounded_surface);
   segments      : LIST [1:?] OF LIST [1:?] OF surface_patch;
 DERIVE
   n_u : INTEGER := SIZEOF(segments);
   n_v : INTEGER := SIZEOF(segments[1]);
 WHERE
   WR1: SIZEOF(QUERY (s <* segments | n_v <> SIZEOF (s))) = 0;
   WR2: constraints_rectangular_composite_surface(SELF);
 END_ENTITY;


 ENTITY rectangular_trimmed_surface
   SUBTYPE OF (bounded_surface);
   basis_surface : surface;
   u1            : parameter_value;
   u2            : parameter_value;
   v1            : parameter_value;
   v2            : parameter_value;
   usense        : BOOLEAN;
   vsense        : BOOLEAN;
 WHERE
   WR1: u1 <> u2;
   WR2: v1 <> v2;
   WR3: (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ELEMENTARY_SURFACE' IN
TYPEOF(basis_surface))
       AND  (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLANE' IN
TYPEOF(basis_surface)))) OR
      ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_OF_REVOLUTION' IN
TYPEOF(basis_surface))
          OR (usense = (u2 > u1));
   WR4: (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SPHERICAL_SURFACE' IN
TYPEOF(basis_surface))
          OR
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TOROIDAL_SURFACE' IN
TYPEOF(basis_surface)))

          OR (vsense = (v2 > v1));
 END_ENTITY;

ENTITY referenced_modified_datum
  SUBTYPE OF (datum_reference);
  modifier : limit_condition;
END_ENTITY;

ENTITY relative_event_occurrence
  SUBTYPE OF (event_occurrence);
  base_event : event_occurrence;
```

```
  offset : time_measure_with_unit;
END_ENTITY;

 ENTITY reparametrised_composite_curve_segment
   SUBTYPE OF (composite_curve_segment);
   param_length : parameter_value;
 WHERE
   WR1: param_length > 0.0;
 END_ENTITY;

ENTITY representation;
  name             : label;
  items            : SET[1:?] OF representation_item;
  context_of_items : representation_context;
DERIVE
  id               : identifier := get_id_value (SELF);
  description      : text := get_description_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                             'ID_ATTRIBUTE.IDENTIFIED_ITEM'))
       <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

                             'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM'))
       <= 1;
 END_ENTITY;

ENTITY representation_context;
  context_identifier : identifier;
  context_type       : text;
INVERSE
  representations_in_context : SET [1:?] OF representation
    FOR context_of_items;
END_ENTITY;

ENTITY representation_item;
  name : label;
WHERE
  WR1: SIZEOF(using_representations(SELF)) > 0;
END_ENTITY;

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;

ENTITY representation_relationship;
  name        : label;
  description : OPTIONAL text;
  rep_1       : representation;
  rep_2       : representation;
END_ENTITY;
```

```
ENTITY representation_relationship_with_transformation
  SUBTYPE OF (representation_relationship);
  transformation_operator : transformation;
WHERE
  WR1:
   SELF\representation_relationship.rep_1.context_of_items
   :<>: SELF\representation_relationship.rep_2.context_of_items;
END_ENTITY;

ENTITY revolved_face_solid
   SUBTYPE OF (swept_face_solid);
   axis  : axis1_placement;
   angle : plane_angle_measure;
 DERIVE
  axis_line : line := representation_item('')||
                    geometric_representation_item()|| curve()||
                    line(axis.location, representation_item('')||
                    geometric_representation_item()||
                    vector(axis.z, 1.0));
 END_ENTITY;

 ENTITY right_angular_wedge
   SUBTYPE OF (geometric_representation_item);
   position : axis2_placement_3d;
   x        : positive_length_measure;
   y        : positive_length_measure;
   z        : positive_length_measure;
   ltx      : length_measure;
 WHERE
   WR1: ((0.0 <= ltx) AND (ltx < x));
 END_ENTITY;

 ENTITY right_circular_cone
   SUBTYPE OF (geometric_representation_item);
   position   : axis1_placement;
   height     : positive_length_measure;
   radius     : length_measure;
   semi_angle : plane_angle_measure;
 WHERE
   WR1: radius >= 0.0;
 END_ENTITY;

 ENTITY right_circular_cylinder
   SUBTYPE OF (geometric_representation_item);
   position   : axis1_placement;
   height     : positive_length_measure;
   radius     : positive_length_measure;
 END_ENTITY;

ENTITY role_association;
  role : object_role;
  item_with_role : role_select;
END_ENTITY;

ENTITY roundness_tolerance
 SUBTYPE OF ( geometric_tolerance );
```

```
 WHERE
WR1 : NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF));

 END_ENTITY;

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 runout_zone_orientation_reference_direction
  SUBTYPE OF (runout_zone_orientation);
  orientation_defining_relationship: shape_aspect_relationship;
END_ENTITY;

 ENTITY seam_curve
   SUBTYPE OF (surface_curve);
 WHERE
   WR1: SIZEOF(SELF\surface_curve.associated_geometry) = 2;
   WR2: associated_surface(SELF\surface_curve.associated_geometry[1]) =
           associated_surface(SELF\surface_curve.associated_geometry[2]);
   WR3: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN
           TYPEOF(SELF\surface_curve.associated_geometry[1]);
   WR4: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN

            TYPEOF(SELF\surface_curve.associated_geometry[2]);
 END_ENTITY;

ENTITY security_classification;
  name : label;
  purpose : text;
  security_level : security_classification_level;
END_ENTITY;

ENTITY security_classification_assignment
   ABSTRACT SUPERTYPE;
  assigned_security_classification : security_classification;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;

END_ENTITY;

ENTITY security_classification_level;
  name : label;
END_ENTITY;

ENTITY serial_numbered_effectivity
  SUBTYPE OF (effectivity);
  effectivity_start_id : identifier;
```

```
  effectivity_end_id : OPTIONAL identifier;
END_ENTITY;

ENTITY shape_aspect;
  name : label;
  description : OPTIONAL text;
  of_shape : product_definition_shape;
  product_definitional : LOGICAL;
DERIVE
  id : identifier := get_id_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;

END_ENTITY;

ENTITY shape_aspect_associativity
  SUBTYPE OF (shape_aspect_relationship);
WHERE
  WR1:  SELF.relating_shape_aspect.product_definitional;
  WR2:  NOT (SELF.related_shape_aspect.product_definitional);
  WR3:  SIZEOF (QUERY (pd <* USEDIN (SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                 'PROPERTY_DEFINITION.DEFINITION') |
          NOT (SIZEOF (QUERY (sdr <* QUERY (pdr <* USEDIN (pd,
                          'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_DEFINITION_REPRESENTATION'
                   IN TYPEOF (pdr))) |
            NOT (SIZEOF (QUERY (it <* sdr.used_representation.items |
              NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE' IN
TYPEOF(it))
              AND
              (SIZEOF (QUERY (aco <* USEDIN (it,

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STYLED_ITEM.ITEM') |
               SIZEOF (TYPEOF (aco) *
                     ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_CURVE',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROJECTION_CURVE']) = 1)) +
               SIZEOF (USEDIN (it, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                      'ANNOTATION_FILL_AREA.BOUNDARIES')) >= 1)
            )) = 1)
          )) = 0)
        )) = 0;
  WR4:  SIZEOF (QUERY (pd <* USEDIN (SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                 'PROPERTY_DEFINITION.DEFINITION') |
          NOT (SIZEOF (QUERY (sdr <* QUERY (pdr <* USEDIN (pd,
                          'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_DEFINITION_REPRESENTATION'

                   IN TYPEOF (pdr))) |
            NOT (check_associative_shape_aspects(sdr))
```

```
          )) = 0)
        )) = 0;
END_ENTITY;

ENTITY shape_aspect_deriving_relationship
       SUBTYPE OF (shape_aspect_relationship);
     WHERE
       WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DERIVED_SHAPE_ASPECT' IN

TYPEOF
            (SELF\SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT);
     END_ENTITY;

ENTITY shape_aspect_relationship;
  name : label;
  description : OPTIONAL text;
  relating_shape_aspect : shape_aspect;
  related_shape_aspect : shape_aspect;
DERIVE
  id : identifier := get_id_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;

END_ENTITY;

ENTITY shape_definition_representation
  SUBTYPE OF (property_definition_representation);
WHERE
  WR1: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF (SELF.definition)) OR
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_DEFINITION' IN TYPEOF
(SELF.definition.definition));
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_REPRESENTATION' IN
TYPEOF(SELF.used_representation);

END_ENTITY;

ENTITY shape_dimension_representation
                SUBTYPE OF (shape_representation);
  WHERE
   WR1: SIZEOF (QUERY (temp <* SELF\representation.items |
    NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM'

        IN TYPEOF (temp)))) = 0;
   WR2: SIZEOF (SELF\representation.items) <= 3;
   WR3: SIZEOF (QUERY (pos_mri <* QUERY (real_mri <*
        SELF\representation.items | 'REAL' IN TYPEOF
        (real_mri\measure_with_unit.value_component) ) |
     NOT  (pos_mri\measure_with_unit.value_component > 0.0 ))) = 0;
END_ENTITY;

ENTITY shape_representation
  SUBTYPE OF (representation);
END_ENTITY;

ENTITY shape_representation_relationship
```

```
  SUBTYPE OF (representation_relationship);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_REPRESENTATION' IN
(TYPEOF(SELF\representation_relationship.rep_1) +
TYPEOF(SELF\representation_relationship.rep_2));

END_ENTITY;

ENTITY shape_representation_with_parameters
  SUBTYPE OF (shape_representation);
WHERE
  WR1: SIZEOF( QUERY( i <* SELF.items |
SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLACEMENT',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DESCRIPTIVE_REPRESENTATION_ITEM'] *
TYPEOF(i)) = 1 )) = SIZEOF(SELF.items);

END_ENTITY;

 ENTITY shell_based_surface_model
   SUBTYPE OF (geometric_representation_item);
   sbsm_boundary  : SET [1:?] OF shell;
 WHERE
   WR1: constraints_geometry_shell_based_surface_model(SELF);
 END_ENTITY;

 ENTITY shell_based_wireframe_model
   SUBTYPE OF (geometric_representation_item);
   sbwm_boundary  : SET [1:?] OF shell;
 WHERE
   WR1: constraints_geometry_shell_based_wireframe_model(SELF);
 END_ENTITY;

ENTITY shell_based_wireframe_shape_representation
 SUBTYPE OF ( shape_representation );
 WHERE
WR1 :
SIZEOF (
QUERY ( it <* SELF.items| NOT ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT_3D' ] * TYPEOF (it)) =
1) )) = 0;
WR2 :
SIZEOF (
QUERY ( it <* SELF.items| ( SIZEOF ([
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' ] * TYPEOF (it)) = 1) ))
>= 1;
WR3 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( ws <*
```

```
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.WIRE_SHELL' IN TYPEOF (sb)) )| NOT (
SIZEOF (
QUERY ( eloop <*
QUERY ( wsb <* ws\wire_shell.wire_shell_extent|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF (wsb)) )| NOT (
SIZEOF (
QUERY ( el <* eloop\path.edge_list| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_CURVE' IN TYPEOF
(el.edge_element)) )) = 0) )) = 0) )) = 0) )) = 0;
WR4 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( ws <*
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.WIRE_SHELL' IN TYPEOF (sb)) )| NOT (
SIZEOF (
QUERY ( eloop <*
QUERY ( wsb <* ws\wire_shell.wire_shell_extent|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF (wsb)) )| NOT (
SIZEOF (
QUERY ( pline_el <*
QUERY ( el <* eloop\path.edge_list|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN TYPEOF
(el.edge_element\edge_curve.edge_geometry)) )| NOT ( SIZEOF
(pline_el.edge_element\edge_curve.edge_geometry\polyline.points) > 2) )) = 0) )) =
0) )) = 0) )) = 0;
WR5 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( ws <*
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.WIRE_SHELL' IN TYPEOF (sb)) )| NOT (
SIZEOF (
QUERY ( eloop <*
QUERY ( wsb <* ws\wire_shell.wire_shell_extent|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF (wsb)) )| NOT (
SIZEOF (
QUERY ( el <* eloop\path.edge_list| NOT
valid_wireframe_edge_curve(el.edge_element\edge_curve.edge_geometry) )) = 0) )) =
0) )) = 0) )) = 0;
WR6 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( ws <*
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.WIRE_SHELL' IN TYPEOF (sb)) )| NOT (
SIZEOF (
```

```
QUERY ( eloop <*
QUERY ( wsb <* ws\wire_shell.wire_shell_extent|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF (wsb)) )| NOT (
SIZEOF (
QUERY ( el <* eloop\path.edge_list| NOT
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
(el.edge_element.edge_start)) AND
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
(el.edge_element.edge_end))) )) = 0) )) = 0) )) = 0) )) = 0;
WR7 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( ws <*
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.WIRE_SHELL' IN TYPEOF (sb)) )| NOT (
SIZEOF (
QUERY ( eloop <*
QUERY ( wsb <* ws\wire_shell.wire_shell_extent|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE_LOOP' IN TYPEOF (wsb)) )| NOT (
SIZEOF (
QUERY ( el <* eloop\path.edge_list| NOT
(valid_wireframe_vertex_point(el.edge_element.edge_start\vertex_point.vertex_geomet
ry) AND
valid_wireframe_vertex_point(el.edge_element.edge_end\vertex_point.vertex_geometry)
) )) = 0) )) = 0) )) = 0) )) = 0;
WR8 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( ws <*
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.WIRE_SHELL' IN TYPEOF (sb)) )| NOT (
SIZEOF (
QUERY ( vloop <*
QUERY ( wsb <* ws\wire_shell.wire_shell_extent|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_LOOP' IN TYPEOF (wsb)) )| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
(vloop\vertex_loop.loop_vertex)) )) = 0) )) = 0) )) = 0;
WR9 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( ws <*
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.WIRE_SHELL' IN TYPEOF (sb)) )| NOT (
SIZEOF (
QUERY ( vloop <*
QUERY ( wsb <* ws\wire_shell.wire_shell_extent|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_LOOP' IN TYPEOF (wsb)) )| NOT
```

```
valid_wireframe_vertex_point(vloop\vertex_loop.loop_vertex\vertex_point.vertex_geom
etry) )) = 0) )) = 0) )) = 0;
WR10 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( vs <*
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_SHELL' IN TYPEOF (sb)) )| NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_POINT' IN TYPEOF
(vs\vertex_shell.vertex_shell_extent.loop_vertex)) )) = 0) )) = 0;
WR11 :
SIZEOF (
QUERY ( sbwm <*
QUERY ( it <* SELF.items|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL_BASED_WIREFRAME_MODEL' IN TYPEOF
(it)) )| NOT ( SIZEOF (
QUERY ( vs <*
QUERY ( sb <* sbwm\shell_based_wireframe_model.sbwm_boundary|
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_SHELL' IN TYPEOF (sb)) )| NOT
valid_wireframe_vertex_point(vs\vertex_shell.vertex_shell_extent.loop_vertex\vertex
_point.vertex_geometry) )) = 0) )) = 0;
WR12 :
SIZEOF (
QUERY ( mi <*
QUERY ( it <* SELF.items| ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
IN TYPEOF (it)) )| NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION' IN TYPEOF
(mi\mapped_item.mapping_source.mapped_representation)) )) = 0;

WR13 :
SELF.context_of_items\geometric_representation_context.coordinate_space_dimension =
3;
 END_ENTITY;

ENTITY si_unit
  SUBTYPE OF (named_unit);
  prefix : OPTIONAL si_prefix;
  name : si_unit_name;
DERIVE
  SELF\named_unit.dimensions : dimensional_exponents := dimensions_for_si_unit
(name);
END_ENTITY;

ENTITY solid_angle_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SOLID_ANGLE_UNIT' IN TYPEOF
(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY solid_angle_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 solid_model
   SUPERTYPE OF (ONEOF (
               CSG_SOLID,
               MANIFOLD_SOLID_BREP,
               SWEPT_FACE_SOLID,
               SOLID_REPLICA))
  SUBTYPE OF (geometric_representation_item);
 END_ENTITY;

 ENTITY solid_replica
   SUBTYPE OF (solid_model);
   parent_solid   : solid_model;
   transformation : cartesian_transformation_operator_3d;

 WHERE
   WR1: acyclic_solid_replica(SELF, parent_solid);
   WR2: parent_solid\geometric_representation_item.dim = 3;
 END_ENTITY;

ENTITY specified_higher_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
  upper_usage   : assembly_component_usage;
  next_usage    : next_assembly_usage_occurrence;
UNIQUE
  UR1: upper_usage, next_usage;
WHERE
  WR1: SELF :<>: upper_usage;
  WR2: SELF\product_definition_relationship.relating_product_definition
       :=: upper_usage.relating_product_definition;
  WR3: SELF\product_definition_relationship.related_product_definition
       :=: next_usage.related_product_definition;
  WR4: (upper_usage.related_product_definition :=:
       next_usage.relating_product_definition) OR
       (SIZEOF (QUERY (pdr <* USEDIN (upper_usage.related_product_definition,

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRODUCT_DEFINITION_RELATIONSHIP.' +
       'RELATED_PRODUCT_DEFINITION') |
        pdr.relating_product_definition :=:
        next_usage.relating_product_definition)) = 1);
  WR5: SIZEOF
(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.NEXT_ASSEMBLY_USAGE_OCCURRENCE',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SPECIFIED_HIGHER_USAGE_OCCURRENCE']

       * TYPEOF(upper_usage)) = 1;
END_ENTITY;

 ENTITY sphere
```

```
   SUBTYPE OF (geometric_representation_item);
   radius : positive_length_measure;
   centre : point;
 END_ENTITY;

 ENTITY spherical_surface
   SUBTYPE OF (elementary_surface);
   radius   : positive_length_measure;
 END_ENTITY;

ENTITY standard_uncertainty

  SUBTYPE OF (uncertainty_qualifier);
  uncertainty_value : REAL;
END_ENTITY;

ENTITY start_request
  SUBTYPE OF (action_request_assignment);
  items : SET[1:?] OF change_request_item;
END_ENTITY;

ENTITY start_work
  SUBTYPE OF (action_assignment);
  items : SET[1:?] OF work_item;
END_ENTITY;

ENTITY straightness_tolerance
 SUBTYPE OF ( geometric_tolerance );
 WHERE
WR1 : NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF));

 END_ENTITY;

ENTITY structured_dimension_callout
  SUBTYPE OF (draughting_callout);
WHERE
  WR1: SIZEOF (TYPEOF (SELF) *
       ['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DATUM_FEATURE_CALLOUT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DATUM_TARGET_CALLOUT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GEOMETRICAL_TOLERANCE_CALLOUT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LEADER_DIRECTED_CALLOUT',
        'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROJECTION_DIRECTED_CALLOUT',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIMENSION_CURVE_DIRECTED_CALLOUT']) =
0;
  WR2: SIZEOF (QUERY (ato <* QUERY (con <* SELF.contents |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
             IN TYPEOF (con))) |
         NOT (ato.name IN
             ['dimension value', 'tolerance value', 'unit text',
              'prefix text', 'suffix text']))) = 0;
  WR3: SIZEOF (QUERY (ato <* QUERY (con <* SELF.contents |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
             IN TYPEOF (con))) |
         (ato.name = 'dimension value')
       )) >= 1;
```

```
  WR4: SIZEOF (QUERY (dcr <* USEDIN (SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
         'DRAUGHTING_CALLOUT_RELATIONSHIP.' +
         'RELATING_DRAUGHTING_CALLOUT') |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
          'DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP' IN TYPEOF (dcr)) AND
         (dcr.name = 'prefix') )) <= 1;
  WR5: SIZEOF (QUERY (dcr <* USEDIN (SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
         'DRAUGHTING_CALLOUT_RELATIONSHIP.' +
         'RELATING_DRAUGHTING_CALLOUT') |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
          'DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP' IN TYPEOF (dcr)) AND
         (dcr.name = 'suffix') )) <= 1;
  WR6: NOT((SIZEOF (QUERY (ato <* QUERY (con <* SELF.contents |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
             IN TYPEOF(con)) ) |
         (ato.name = 'prefix text')
       )) > 0)) OR
       (SIZEOF (QUERY (dcr <* USEDIN (SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                              'DRAUGHTING_CALLOUT_RELATIONSHIP.' +
                              'RELATING_DRAUGHTING_CALLOUT') |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
          'DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP' IN TYPEOF (dcr)) AND
         (dcr.name = 'prefix') )) = 1);
  WR7: NOT(SIZEOF (QUERY (ato <* QUERY (con <* SELF.contents |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'
             IN TYPEOF(con))) |
         (ato.name = 'suffix text')
       )) > 0) OR
       (SIZEOF (QUERY (dcr <* USEDIN (SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                              'DRAUGHTING_CALLOUT_RELATIONSHIP.' +
                              'RELATING_DRAUGHTING_CALLOUT') |
         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

          'DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP' IN TYPEOF (dcr)) AND
           (dcr.name = 'suffix') )) = 1);
END_ENTITY;

ENTITY styled_item
  SUBTYPE OF (representation_item);
  styles :  SET [1:?] OF presentation_style_assignment;
  item   :  representation_item;
WHERE
  WR1: (SIZEOF(SELF.styles) = 1)
              XOR
       (SIZEOF(QUERY(pres_style <* SELF.styles |
         NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

              'PRESENTATION_STYLE_BY_CONTEXT' IN
         TYPEOF(pres_style))
         )) = 0);
END_ENTITY;

ENTITY supplied_part_relationship
```

```
  SUBTYPE OF (product_definition_relationship);
END_ENTITY;

 ENTITY surface
   SUPERTYPE OF (ONEOF (
               ELEMENTARY_SURFACE,
               SWEPT_SURFACE,
               BOUNDED_SURFACE,
               OFFSET_SURFACE,
               SURFACE_REPLICA))
  SUBTYPE OF (geometric_representation_item);
 END_ENTITY;

 ENTITY surface_curve
   SUPERTYPE OF (ONEOF (
               INTERSECTION_CURVE,
               SEAM_CURVE)
               ANDOR
               BOUNDED_SURFACE_CURVE)
  SUBTYPE OF (curve);
   curve_3d              : curve;
   associated_geometry   : LIST[1:2] OF pcurve_or_surface;
   master_representation : preferred_surface_curve_representation;
 DERIVE
   basis_surface         : SET[1:2] OF surface
                         := get_basis_surface(SELF);
 WHERE
   WR1: curve_3d.dim = 3;
   WR2: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN
TYPEOF(associated_geometry[1])) OR
                        (master_representation <> pcurve_s1);
   WR3: ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN
TYPEOF(associated_geometry[2])) OR
                        (master_representation <> pcurve_s2);
   WR4: NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN
TYPEOF(curve_3d));

 END_ENTITY;

 ENTITY surface_of_linear_extrusion
   SUBTYPE OF (swept_surface);
   extrusion_axis      : vector;
 END_ENTITY;

 ENTITY surface_of_revolution
  SUBTYPE OF (swept_surface);
   axis_position       : axis1_placement;
 DERIVE
   axis_line : line := representation_item('')||
                      geometric_representation_item()|| curve()||
                      line(axis_position.location, representation_item('')||
                      geometric_representation_item()||
                      vector(axis_position.z, 1.0));
 END_ENTITY;

 ENTITY surface_patch
 SUBTYPE OF (founded_item);
```

```
   parent_surface : bounded_surface;
   u_transition   : transition_code;
   v_transition   : transition_code;
   u_sense        : BOOLEAN;
   v_sense        : BOOLEAN;
 INVERSE
   using_surfaces : BAG[1:?] OF rectangular_composite_surface FOR segments;
 WHERE
   WR1: (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_BOUNDED_SURFACE'

                 IN TYPEOF(parent_surface)));
 END_ENTITY;

ENTITY surface_profile_tolerance
 SUBTYPE OF ( geometric_tolerance );
 WHERE
WR1 : NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF)) OR ( SIZEOF
(SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3);

 END_ENTITY;

 ENTITY surface_replica
   SUBTYPE OF (surface);
   parent_surface : surface;
   transformation : cartesian_transformation_operator_3d;
 WHERE
   WR1: acyclic_surface_replica(SELF, parent_surface);
 END_ENTITY;

ENTITY surface_side_style;
  name   : label;
  styles : SET [1:7] OF surface_style_element_select;
WHERE
  WR1: SIZEOF(QUERY( style1 <* SELF.styles |
         SIZEOF(QUERY( style2 <* SELF.styles - style1 |
           TYPEOF(style1) = TYPEOF(style2)
           )) > 0
         )) = 0;
END_ENTITY;

ENTITY surface_style_boundary;
  style_of_boundary : curve_or_render;
END_ENTITY;

ENTITY surface_style_control_grid;
  style_of_control_grid : curve_or_render;
END_ENTITY;

ENTITY surface_style_fill_area;
  fill_area : fill_area_style;
END_ENTITY;

ENTITY surface_style_parameter_line;
  style_of_parameter_lines : curve_or_render;
  direction_counts         : SET [1:2] OF direction_count_select;
WHERE
```

```
  WR1: (HIINDEX(SELF.direction_counts) = 1)
                        XOR
       (TYPEOF(SELF.direction_counts[1]) <>
          TYPEOF(SELF.direction_counts[2]));
END_ENTITY;

ENTITY surface_style_segmentation_curve;
  style_of_segmentation_curve : curve_or_render;
END_ENTITY;

ENTITY surface_style_silhouette;
  style_of_silhouette : curve_or_render;
END_ENTITY;

ENTITY surface_style_usage;
  side  : surface_side;
  style : surface_side_style_select;
END_ENTITY;

 ENTITY swept_face_solid
   SUPERTYPE OF (ONEOF (
               EXTRUDED_FACE_SOLID,
               REVOLVED_FACE_SOLID))
  SUBTYPE OF (solid_model);
   swept_face :  face_surface;
 WHERE
   WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLANE' IN
TYPEOF(swept_face.face_geometry);

 END_ENTITY;

 ENTITY swept_surface
   SUPERTYPE OF (ONEOF (
               SURFACE_OF_LINEAR_EXTRUSION,
               SURFACE_OF_REVOLUTION))
  SUBTYPE OF (surface);
   swept_curve : curve;
 END_ENTITY;

ENTITY symbol_colour;
  colour_of_symbol : colour;
END_ENTITY;

ENTITY symbol_representation
  SUBTYPE OF (representation);
END_ENTITY;

ENTITY symbol_representation_map
  SUBTYPE OF (representation_map);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMBOL_REPRESENTATION' IN
         TYPEOF (SELF\representation_map.mapped_representation);
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT' IN

         TYPEOF (SELF\representation_map.mapping_origin);
END_ENTITY;
```

```
ENTITY symbol_representation_relationship
  SUBTYPE OF (representation_relationship_with_transformation);
WHERE
  WR1: acyclic_symbol_representation_relationship (SELF,

[SELF\representation_relationship.
                                                        rep_2]);
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMBOL_REPRESENTATION' IN
         TYPEOF (SELF\representation_relationship.rep_1);
  WR3: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMBOL_REPRESENTATION'IN

          TYPEOF (SELF\representation_relationship.rep_2);
END_ENTITY;

ENTITY symbol_representation_with_blanking_box
  SUBTYPE OF (symbol_representation);
  blanking : planar_box;
WHERE
  WR1: item_in_context (SELF.blanking, SELF\representation.context_of_items);
END_ENTITY;

ENTITY symbol_style;
  name            : label;
  style_of_symbol : symbol_style_select;
END_ENTITY;

ENTITY symbol_target
  SUBTYPE OF (geometric_representation_item);
  placement         : axis2_placement;
  x_scale           : positive_ratio_measure;
  y_scale           : positive_ratio_measure;
END_ENTITY;

ENTITY symmetric_shape_aspect
  SUBTYPE OF (shape_aspect);
INVERSE
  basis_relationships : SET [1:?] OF shape_aspect_relationship
                        FOR relating_shape_aspect;
WHERE
  WR1: SIZEOF (QUERY (x<*SELF\symmetric_shape_aspect.basis_relationships |
       'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CENTRE_OF_SYMMETRY' IN TYPEOF

       (x\shape_aspect_relationship.related_shape_aspect)))>=1;
END_ENTITY;

ENTITY symmetry_tolerance
 SUBTYPE OF ( geometric_tolerance_with_datum_reference );
 WHERE
WR1 :
SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 3;
 END_ENTITY;

ENTITY table_record_field_representation
  SUBTYPE OF (symbol_representation);
WHERE
  WR1: (SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                            'REPRESENTATION_RELATIONSHIP.REP_2')) > 0)
```

```
                        OR
       (SIZEOF(QUERY( map_item <* USEDIN(SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                                            'REPRESENTATION_MAP.'+
                                            'MAPPED_REPRESENTATION') |
         SIZEOF(QUERY( mi <* USEDIN(map_item,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                                           'MAPPED_ITEM.'+
                                           'MAPPING_SOURCE') |

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+

                                         'TABLE_RECORD_REPRESENTATION' IN
             TYPEOF (using_representations (mi)) )) > 0))
                    > 0);
END_ENTITY;

ENTITY table_record_field_representation_with_clipping_box
  SUBTYPE OF (table_record_field_representation);
  clipping_box : planar_box;
WHERE
   WR1: item_in_context (SELF.clipping_box,
                         SELF\representation.context_of_items);
END_ENTITY;

ENTITY table_record_representation
  SUBTYPE OF (symbol_representation);
WHERE
  WR1: (SIZEOF(USEDIN(SELF, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                            'REPRESENTATION_RELATIONSHIP.REP_2')) > 0)
                        OR
       (SIZEOF(QUERY( map_item <* USEDIN(SELF,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                                            'REPRESENTATION_MAP.'+
                                            'MAPPED_REPRESENTATION') |
         SIZEOF(QUERY( mi <* USEDIN(map_item,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                                           'MAPPED_ITEM.'+
                                           'MAPPING_SOURCE') |

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+

                                           'TABLE_REPRESENTATION' IN
             TYPEOF (using_representations (mi)) )) > 0))
                    > 0);
END_ENTITY;

ENTITY table_representation
  SUBTYPE OF (symbol_representation);
END_ENTITY;

ENTITY table_representation_relationship
  SUBTYPE OF (symbol_representation_relationship);
WHERE
  WR1: NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TABLE_RECORD_REPRESENTATION' IN
            TYPEOF (SELF\representation_relationship.rep_1))
```

```
        XOR

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TABLE_RECORD_FIELD_REPRESENTATION' IN
        TYPEOF (SELF\representation_relationship.rep_2));
  WR2: NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TABLE_REPRESENTATION' IN
            TYPEOF (SELF\representation_relationship.rep_1))
        XOR
       ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TABLE_RECORD_REPRESENTATION' IN
        TYPEOF (SELF\representation_relationship.rep_2));
  WR3: NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TABLE_RECORD_FIELD_REPRESENTATION' IN
           TYPEOF (SELF\representation_relationship.rep_1))
        XOR

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TABLE_RECORD_FIELD_REPRESENTATION' IN

        TYPEOF (SELF\representation_relationship.rep_2));
END_ENTITY;

ENTITY table_text_relationship
  SUBTYPE OF (annotation_occurrence_relationship);
  field : table_record_field_representation;
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TABLE_OCCURRENCE'
       IN TYPEOF (SELF\annotation_occurrence_relationship.
                  relating_annotation_occurrence);
  WR2: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TABLE'
       IN TYPEOF (SELF\annotation_occurrence_relationship.
                  relating_annotation_occurrence\styled_item.item);
  WR3: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_OCCURRENCE'

       IN TYPEOF (SELF\annotation_occurrence_relationship.
                  related_annotation_occurrence);
  WR4: field_in_table (SELF.field,
                       SELF\annotation_occurrence_relationship.
                       relating_annotation_occurrence);
END_ENTITY;

ENTITY tangent
  SUBTYPE OF (derived_shape_aspect);
WHERE
  WR1: SIZEOF (SELF\derived_shape_aspect.deriving_relationships)= 1;
END_ENTITY;

ENTITY terminator_symbol
  SUBTYPE OF (annotation_symbol_occurrence);
  annotated_curve : annotation_curve_occurrence;
END_ENTITY;

ENTITY text_literal
  SUBTYPE OF (geometric_representation_item);
  literal   : presentable_text;
  placement : axis2_placement;
  alignment : text_alignment;
  path      : text_path;
  font      : font_select;
END_ENTITY;
```

```
ENTITY text_literal_with_associated_curves
  SUBTYPE OF (text_literal);
  associated_curves : SET[1:?] of curve;
END_ENTITY;

ENTITY text_literal_with_blanking_box
  SUBTYPE OF (text_literal);
  blanking : planar_box;
END_ENTITY;

ENTITY text_literal_with_delineation
  SUBTYPE OF (text_literal);
  delineation : text_delineation;
END_ENTITY;

ENTITY text_literal_with_extent
  SUBTYPE OF (text_literal);
  extent : planar_extent;
END_ENTITY;

ENTITY text_string_representation
  SUBTYPE OF (representation);
WHERE
  WR1: SIZEOF (
         QUERY (item <* SELF\representation.items |
           SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL',
                    'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_CHARACTER',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEFINED_CHARACTER_GLYPH',
                    'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT',
                    'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT'] *
TYPEOF (item)) = 0
         )) = 0;
  WR2: SIZEOF (
         QUERY (item <* SELF\representation.items |
           NOT (SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT_CHARACTER',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEFINED_CHARACTER_GLYPH',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'] *
                 TYPEOF (item)) = 0)
         )) >= 1;
  WR3: SIZEOF (
         QUERY (a2p <*
           QUERY (item <* SELF\representation.items |
             'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AXIS2_PLACEMENT' IN TYPEOF
(item)) |
           NOT ((SIZEOF (
             QUERY (at <*
               QUERY (item <* SELF\representation.items |
```

```
                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                'ANNOTATION_TEXT' IN TYPEOF (item)) |
              (at\mapped_item.mapping_target :=: a2p))) >= 1) OR
          (SIZEOF (
            QUERY (atc <*
              QUERY (item <* SELF\representation.items |
                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +

                'ANNOTATION_TEXT_CHARACTER' IN TYPEOF (item)) |
              (atc\mapped_item.mapping_target :=: a2p))) >= 1)
        ))) = 0;
END_ENTITY;

ENTITY text_style;
  name                 : label;
  character_appearance : character_style_select;
END_ENTITY;

ENTITY text_style_for_defined_font;
  text_colour : colour;
END_ENTITY;

ENTITY text_style_with_box_characteristics
  SUBTYPE OF (text_style);
  characteristics : SET [1:4] OF box_characteristic_select;
WHERE
  WR1: SIZEOF( QUERY( c1 <* SELF.characteristics |
         SIZEOF( QUERY( c2 <* SELF.characteristics - c1 |
           TYPEOF (c1) = TYPEOF (c2)
          )) > 0
       )) = 0;
END_ENTITY;

ENTITY text_style_with_mirror
  SUBTYPE OF (text_style);
  mirror_placement : axis2_placement;
END_ENTITY;

ENTITY text_style_with_spacing
  SUBTYPE OF (text_style);
  character_spacing : character_spacing_select;
END_ENTITY;

ENTITY thermodynamic_temperature_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.THERMODYNAMIC_TEMPERATURE_UNIT'
IN TYPEOF (SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY thermodynamic_temperature_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 = 1.0) AND
(SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
(SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;

ENTITY time_interval;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY time_interval_assignment
   ABSTRACT SUPERTYPE;
  assigned_time_interval : time_interval;
  role : time_interval_role;
END_ENTITY;

ENTITY time_interval_based_effectivity
  SUBTYPE OF (effectivity);
  effectivity_period : time_interval;
END_ENTITY;

ENTITY time_interval_relationship;
  name : label;
  description : OPTIONAL text;
  relating_time_interval : time_interval;
  related_time_interval : time_interval;
END_ENTITY;

ENTITY time_interval_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;

ENTITY time_interval_with_bounds
  SUBTYPE OF (time_interval);
  primary_bound : OPTIONAL date_time_or_event_occurrence;
  secondary_bound : OPTIONAL date_time_or_event_occurrence;
  duration : OPTIONAL time_measure_with_unit;
WHERE
  WR1: NOT (EXISTS(secondary_bound) AND EXISTS(duration));
  WR2: EXISTS(primary_bound) OR EXISTS(secondary_bound);
END_ENTITY;

ENTITY time_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TIME_UNIT' IN TYPEOF
(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY time_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 = 1.0) 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 tolerance_value;
   lower_bound : measure_with_unit;
   upper_bound : measure_with_unit;
WHERE
   WR1: upper_bound\measure_with_unit.value_component >
        lower_bound\measure_with_unit.value_component;
   WR2: upper_bound\measure_with_unit.unit_component =
        lower_bound\measure_with_unit.unit_component;
END_ENTITY;

ENTITY tolerance_zone
  SUBTYPE OF (shape_aspect);
  defining_tolerance : SET [1:?] OF geometric_tolerance;
  form               : tolerance_zone_form;
END_ENTITY;

ENTITY tolerance_zone_definition
  SUPERTYPE OF (ONEOF (
              PROJECTED_ZONE_DEFINITION,
              RUNOUT_ZONE_DEFINITION));
  zone : tolerance_zone;
  boundaries: SET [1:?] OF shape_aspect;
END_ENTITY;

ENTITY tolerance_zone_form;
  name : label;
END_ENTITY;

 ENTITY topological_representation_item
   SUPERTYPE OF (ONEOF (
              VERTEX,
              EDGE,
              FACE_BOUND,
              FACE,
              VERTEX_SHELL,
              WIRE_SHELL,
              CONNECTED_EDGE_SET,
              CONNECTED_FACE_SET,(
              LOOP
              ANDOR
              PATH)))
  SUBTYPE OF (representation_item);
 END_ENTITY;

 ENTITY toroidal_surface
   SUBTYPE OF (elementary_surface);
   major_radius : positive_length_measure;
   minor_radius : positive_length_measure;
```

```
 END_ENTITY;

 ENTITY torus
   SUBTYPE OF (geometric_representation_item);
   position     : axis1_placement;
   major_radius : positive_length_measure;
   minor_radius : positive_length_measure;
 WHERE
   WR1: major_radius > minor_radius;
 END_ENTITY;

ENTITY total_runout_tolerance
 SUBTYPE OF ( geometric_tolerance_with_datum_reference );
 WHERE
WR1 :
SIZEOF (SELF\geometric_tolerance_with_datum_reference.datum_system) <= 2;
 END_ENTITY;

 ENTITY trimmed_curve
   SUBTYPE OF (bounded_curve);
   basis_curve           : curve;
   trim_1                : SET[1:2] OF trimming_select;
   trim_2                : SET[1:2] OF trimming_select;
   sense_agreement       : BOOLEAN;
   master_representation : trimming_preference;
 WHERE
   WR1: (HIINDEX(trim_1) = 1) OR (TYPEOF(trim_1[1]) <> TYPEOF(trim_1[2]));
   WR2: (HIINDEX(trim_2) = 1) OR (TYPEOF(trim_2[1]) <> TYPEOF(trim_2[2]));
 END_ENTITY;

ENTITY two_direction_repeat_factor
  SUBTYPE OF (one_direction_repeat_factor);
  second_repeat_factor : vector;
END_ENTITY;

ENTITY type_qualifier;
  name : label;
END_ENTITY;

ENTITY uncertainty_assigned_representation
  SUBTYPE OF (representation);
  uncertainty : SET [1:?] OF uncertainty_measure_with_unit;
END_ENTITY;

ENTITY uncertainty_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  name        : label;
  description : OPTIONAL text;
WHERE
  WR1: valid_measure_value (SELF\measure_with_unit.value_component);
END_ENTITY;

ENTITY uncertainty_qualifier
  ;
  measure_name : label;
  description  : text;
END_ENTITY;
```

```
 ENTITY uniform_curve
   SUBTYPE OF (b_spline_curve);
 END_ENTITY;

 ENTITY uniform_surface

   SUBTYPE OF (b_spline_surface);
 END_ENTITY;

ENTITY user_defined_curve_font
  SUBTYPE OF (curve_style_font, mapped_item);
END_ENTITY;

ENTITY user_defined_marker
  SUBTYPE OF (mapped_item, pre_defined_marker);
END_ENTITY;

ENTITY user_defined_terminator_symbol
  SUBTYPE OF (mapped_item, pre_defined_symbol);
END_ENTITY;

ENTITY value_range
  SUBTYPE OF (compound_representation_item);
WHERE
  WR1: ( 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+ 'SET_REPRESENTATION_ITEM'
IN TYPEOF ( item_element ) ) AND value_range_wr1 ( item_element );

  WR2: value_range_wr2 ( item_element );
  WR3: value_range_wr3 ( item_element );
END_ENTITY;

ENTITY value_representation_item
SUBTYPE OF (representation_item);
  value_component : measure_value;
WHERE
  WR1: SIZEOF (QUERY (rep <* using_representations (SELF) |
       NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.GLOBAL_UNIT_ASSIGNED_CONTEXT'

       IN TYPEOF (rep.context_of_items)
       ))) = 0;
END_ENTITY;

 ENTITY vector
   SUBTYPE OF (geometric_representation_item);
   orientation : direction;
   magnitude   : length_measure;
 WHERE
   WR1 : magnitude >= 0.0;
 END_ENTITY;

ENTITY vector_style
  SUBTYPE OF (curve_style, pre_defined_terminator_symbol);
END_ENTITY;

ENTITY versioned_action_request;
```

```
  id : identifier;
  version : label;
  purpose : text;
  description : OPTIONAL text;
END_ENTITY;

 ENTITY vertex
   SUBTYPE OF (topological_representation_item);
 END_ENTITY;

 ENTITY vertex_loop
   SUBTYPE OF (loop);
   loop_vertex : vertex;
 END_ENTITY;

 ENTITY vertex_point
 SUBTYPE OF(vertex,geometric_representation_item);
   vertex_geometry : point;
 END_ENTITY;

 ENTITY vertex_shell
   SUBTYPE OF (topological_representation_item);
   vertex_shell_extent : vertex_loop;
 END_ENTITY;

ENTITY view_volume
  SUBTYPE OF (founded_item);
  projection_type            : central_or_parallel;
  projection_point           : cartesian_point;
  view_plane_distance        : length_measure;
  front_plane_distance       : length_measure;
  front_plane_clipping       : BOOLEAN;
  back_plane_distance        : length_measure;
  back_plane_clipping        : BOOLEAN;
  view_volume_sides_clipping : BOOLEAN;
  view_window                : planar_box;
END_ENTITY;


ENTITY volume_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VOLUME_UNIT' IN TYPEOF
(SELF\measure_with_unit.unit_component);

END_ENTITY;

ENTITY volume_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 3.000000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.000000) AND
(SELF\named_unit.dimensions.time_exponent = 0.000000) AND
(SELF\named_unit.dimensions.electric_current_exponent = 0.000000) AND
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.000000) AND
(SELF\named_unit.dimensions.amount_of_substance_exponent = 0.000000) AND
(SELF\named_unit.dimensions.luminous_intensity_exponent = 0.000000);
```

```
END_ENTITY;

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;

 ENTITY wire_shell
   SUBTYPE OF (topological_representation_item);
   wire_shell_extent : SET [1:?] OF loop;
 WHERE
   WR1: NOT mixed_loop_type_set(wire_shell_extent);
 END_ENTITY;

FUNCTION acyclic_composite_text(start_composite : composite_text;
                                child_text : SET [1:?] OF
                                text_or_character) : LOGICAL;

  LOCAL
   i : INTEGER;
   local_composite_text : SET [0:?] OF composite_text;
   local_annotation_text : SET [0:?] OF annotation_text;
   local_children : SET [0:?] OF text_or_character;
  END_LOCAL;

  local_composite_text := QUERY (child <* child_text |

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'
                           IN TYPEOF (child)));

  IF (SIZEOF (local_composite_text) > 0) THEN
    REPEAT i := 1 TO HIINDEX (local_composite_text);
      IF (start_composite :=: local_composite_text[i]) THEN
        RETURN (FALSE);
      END_IF;
    END_REPEAT;
  END_IF;

  local_children := child_text;

  IF (SIZEOF (local_composite_text)) > 0 THEN
    REPEAT i := 1 TO HIINDEX (local_composite_text);
      local_children := local_children +
                        local_composite_text[i].collected_text;
    END_REPEAT;
  END_IF;


  local_annotation_text := QUERY (child <* child_text |

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT'
                           IN TYPEOF (child)));

  IF (SIZEOF (local_annotation_text) > 0) THEN
    REPEAT i := 1 TO HIINDEX (local_annotation_text);
      local_children := local_children +
```

```
      QUERY (item <* local_annotation_text[i]\mapped_item.
                     mapping_source.mapped_representation.items |
        SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANNOTATION_TEXT',
                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_TEXT'] *

                TYPEOF(item)) > 0);
    END_REPEAT;
  END_IF;

  IF (local_children :<>: child_text) THEN
    RETURN (acyclic_composite_text (start_composite, local_children));
  ELSE
    RETURN (TRUE);
  END_IF;


 END_FUNCTION;

 FUNCTION acyclic_curve_replica(rep : curve_replica; parent : curve)
                                                    : BOOLEAN;
   IF NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_REPLICA') IN
TYPEOF(parent)) THEN

      RETURN (TRUE);
   END_IF;
 (* Return TRUE if the parent is not of type curve_replica *)
   IF (parent :=: rep) THEN
      RETURN (FALSE);
  (* Return FALSE if the parent is the same curve_replica, otherwise,
   call function again with the parents own parent_curve.     *)
    ELSE
    RETURN(acyclic_curve_replica(rep,
                parent\curve_replica.parent_curve));
    END_IF;

  END_FUNCTION;

FUNCTION acyclic_mapped_item_usage (rep: representation) : BOOLEAN;
  -- returns TRUE if the representation contains one or more mapped_items
  -- that are a mapping of the representation itself
  LOCAL
    items : SET OF representation_item;
  END_LOCAL;

  items := QUERY (item <* rep.items |
           'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' IN TYPEOF
(item));

  IF SIZEOF (items) = 0
  THEN
    RETURN (FALSE);
  ELSE
    REPEAT i := 1 TO HIINDEX (items);
      IF items[i]\mapped_item.mapping_source.mapped_representation :=: rep
      THEN
        RETURN (TRUE);
```

```
      ELSE

        RETURN (acyclic_mapped_item_usage(items[i]\
                  mapped_item.mapping_source.mapped_representation));
      END_IF;
    END_REPEAT;
  RETURN (FALSE);
  END_IF;

END_FUNCTION;

FUNCTION acyclic_mapped_representation
  (parent_set   : SET OF representation;
   children_set : SET OF representation_item) : BOOLEAN;
  LOCAL
    x,y : SET OF representation_item;
  END_LOCAL;
  -- Determine the subset of children_set that are mapped_items
  x := QUERY(z <* children_set |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
       IN TYPEOF(z));
  -- Determine that the subset has elements
  IF SIZEOF(x) > 0 THEN
    -- Check each element of the set
    REPEAT i := 1 TO HIINDEX(x);
      -- If the selected element maps a representation in the
      -- parent_set, then return false
      IF x[i]\mapped_item.mapping_source.mapped_representation
        IN parent_set THEN
        RETURN (FALSE);
      END_IF;
      -- Recursive check of the items of mapped_representation
      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;
  -- Determine the subset of children_set that are not
  -- mapped_items
  x := children_set - x;
  -- Determine that the subset has elements
  IF SIZEOF(x) > 0 THEN
    -- For each element of the set:
    REPEAT i := 1 TO HIINDEX(x);
      -- Determine the set of representation_items referenced
      y := QUERY(z <* bag_to_set( USEDIN(x[i], '')) |
           'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REPRESENTATION_ITEM' IN
TYPEOF(z));

      -- Recursively check for an offending mapped_item
      -- Return false for any errors encountered
      IF NOT acyclic_mapped_representation(parent_set, y) THEN
        RETURN (FALSE);
      END_IF;
```

```
    END_REPEAT;
  END_IF;
  -- Return true when all elements are checked and
  -- no error conditions found
  RETURN (TRUE);

END_FUNCTION;

 FUNCTION acyclic_point_replica(rep : point_replica; parent : point)
                                                 : BOOLEAN;
   IF NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_REPLICA') IN
TYPEOF(parent)) THEN

      RETURN (TRUE);
   END_IF;
 (* Return TRUE if the parent is not of type point_replica *)
   IF (parent :=: rep) THEN
      RETURN (FALSE);
  (* Return FALSE if the parent is the same point_replica, otherwise,
   call function again with the parents own parent_pt.     *)
    ELSE RETURN(acyclic_point_replica(rep, parent\point_replica.parent_pt));
    END_IF;

  END_FUNCTION;

FUNCTION acyclic_product_category_relationship
 (relation : product_category_relationship; children : SET OF product_category) :
BOOLEAN;
LOCAL
      x : SET OF product_category_relationship;
      local_children : SET OF product_category;
    END_LOCAL;

    REPEAT i := 1 TO HIINDEX(children);
      IF relation.category :=: children[i] THEN
        RETURN (FALSE);
      END_IF;
    END_REPEAT;
    x := bag_to_set(USEDIN(relation.category,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY'));

    local_children := children + relation.category;
    IF SIZEOF(x) > 0 THEN
      REPEAT i := 1 TO HIINDEX(x);
        IF NOT acyclic_product_category_relationship(x[i], local_children) THEN
          RETURN (FALSE);
        END_IF;
      END_REPEAT;
    END_IF;
    RETURN (TRUE);

END_FUNCTION;

FUNCTION acyclic_product_definition_relationship
 (relation : product_definition_relationship; relatives : SET[1:?] OF
product_definition; specific_relation : STRING) : BOOLEAN;
```

```
LOCAL
      x : SET OF product_definition_relationship;
    END_LOCAL;

    IF relation.relating_product_definition IN relatives THEN
      RETURN (FALSE);
    END_IF;
    x := QUERY(pd <* bag_to_set(USEDIN(relation.relating_product_definition,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' + 'PRODUCT_DEFINITION_RELATIONSHIP.'
+ 'RELATED_PRODUCT_DEFINITION')) | specific_relation IN TYPEOF(pd));

    REPEAT i := 1 TO HIINDEX(x);
      IF NOT acyclic_product_definition_relationship(x[i], relatives +
relation.relating_product_definition, specific_relation) THEN
        RETURN (FALSE);
      END_IF;
    END_REPEAT;
RETURN (TRUE);

END_FUNCTION;

 FUNCTION acyclic_solid_replica(rep : solid_replica; parent :
                    solid_model) : BOOLEAN;
   IF NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SOLID_REPLICA') IN
TYPEOF(parent)) THEN

      RETURN (TRUE);
   END_IF;
 (* Return TRUE if the parent is not of type solid_replica. *)
   IF (parent :=: rep) THEN
      RETURN (FALSE);
  (* Return FALSE if the parent is the same solid_replica, otherwise,
   call function again with the parents own parent_solid.     *)
    ELSE RETURN(acyclic_solid_replica(rep,
                      parent\solid_replica.parent_solid));
    END_IF;

  END_FUNCTION;

 FUNCTION acyclic_surface_replica(rep : surface_replica; parent : surface)
                                                 : BOOLEAN;
   IF NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_REPLICA') IN
TYPEOF(parent)) THEN

      RETURN (TRUE);
   END_IF;
 (* Return TRUE if the parent is not of type surface_replica *)
   IF (parent :=: rep) THEN
      RETURN (FALSE);
  (* Return FALSE if the parent is the same surface_replica, otherwise,
   call function again with the parents own parent_surface.     *)
    ELSE RETURN(acyclic_surface_replica(rep,
                    parent\surface_replica.parent_surface));
    END_IF;

  END_FUNCTION;
```

```
FUNCTION acyclic_symbol_representation_relationship
  (relation : symbol_representation_relationship;
   children : SET OF symbol_representation ) : BOOLEAN;
  LOCAL
    x : SET OF symbol_representation_relationship;
    local_children : SET OF symbol_representation;
  END_LOCAL;

  REPEAT i:=1 TO HIINDEX(children);
    IF relation\representation_relationship.rep_1 :=: children[i] THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;

  x := bag_to_set (USEDIN ( relation\representation_relationship.rep_1,
                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+

                'REPRESENTATION_RELATIONSHIP.'+ 'REP_2'));
  local_children := children + relation\representation_relationship.rep_1;

  IF SIZEOF (x) > 0 THEN
    REPEAT i:=1 TO HIINDEX (x);
      IF NOT acyclic_symbol_representation_relationship(x[i] ,
                                                local_children) THEN
        RETURN (FALSE);
      END_IF;
    END_REPEAT;
  END_IF;

  RETURN (TRUE);


END_FUNCTION;

FUNCTION aspect_ratio (p : planar_box) : positive_ratio_measure;
(* if the dimensions of the planar_box are greater than zero,
      compute the aspect ratio and return the resulting value. *)
   IF (p.size_in_x > 0.) AND (p.size_in_y > 0.) THEN
      RETURN (p.size_in_x / p.size_in_y);
   ELSE
      RETURN (?);
   END_IF;

END_FUNCTION;

 FUNCTION associated_surface(arg : pcurve_or_surface) : surface;
   LOCAL
     surf : surface;
   END_LOCAL;

   IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN TYPEOF(arg) THEN

     surf := arg.basis_surface;
   ELSE
     surf := arg;
   END_IF;
   RETURN(surf);
```

```
 END_FUNCTION;

FUNCTION bag_to_set (the_bag : BAG OF GENERIC:intype):SET OF GENERIC:intype;
    LOCAL
      the_set : SET OF GENERIC:intype := [];
    END_LOCAL;

    IF SIZEOF(the_bag) > 0 THEN
      REPEAT i := 1 TO HIINDEX(the_bag);
        the_set := the_set + the_bag[i];
      END_REPEAT;
    END_IF;
    RETURN (the_set);

END_FUNCTION;

FUNCTION base_axis(dim : INTEGER; axis1, axis2, axis3 : direction) :
                                                    LIST [2:3] OF
direction;
  LOCAL
    u      : LIST [2:3] OF direction;
    factor : REAL;
    d1, d2 : direction;
  END_LOCAL;

  IF (dim = 3) THEN
    d1 := NVL(normalise(axis3),  dummy_gri || direction([0.0,0.0,1.0]));
    d2 := first_proj_axis(d1,axis1);
    u := [d2, second_proj_axis(d1,d2,axis2), d1];
  ELSE
     IF EXISTS(axis1) THEN
      d1 := normalise(axis1);
      u := [d1, orthogonal_complement(d1)];
      IF EXISTS(axis2) THEN
        factor := dot_product(axis2,u[2]);
        IF (factor < 0.0) THEN
          u[2].direction_ratios[1] := -u[2].direction_ratios[1];
          u[2].direction_ratios[2] := -u[2].direction_ratios[2];
        END_IF;
      END_IF;
    ELSE
      IF EXISTS(axis2) THEN
        d1 := normalise(axis2);
        u := [orthogonal_complement(d1), d1];
        u[1].direction_ratios[1] := -u[1].direction_ratios[1];
        u[1].direction_ratios[2] := -u[1].direction_ratios[2];
      ELSE
        u := [dummy_gri || direction([1.0, 0.0]), dummy_gri ||
                                                  direction([0.0, 1.0])];
      END_IF;
    END_IF;
  END_IF;
  RETURN(u);

END_FUNCTION;
```

```
  FUNCTION boolean_choose (b : boolean;
           choice1, choice2 : generic : item)  : generic : item;

    IF b THEN
      RETURN (choice1);
    ELSE
      RETURN (choice2);
    END_IF;

  END_FUNCTION;

 FUNCTION build_2axes(ref_direction : direction) : LIST [2:2] OF direction;
  LOCAL
    d : direction := NVL(normalise(ref_direction),
                         dummy_gri || direction([1.0,0.0]));
  END_LOCAL;

  RETURN([d, orthogonal_complement(d)]);

 END_FUNCTION;

 FUNCTION build_axes(axis, ref_direction : direction) :
                                          LIST [3:3] OF direction;
  LOCAL
    d1, d2 : direction;
  END_LOCAL;
 d1 := NVL(normalise(axis), dummy_gri || direction([0.0,0.0,1.0]));
 d2 := first_proj_axis(d1, ref_direction);
 RETURN([d2, normalise(cross_product(d1,d2)).orientation, d1]);

 END_FUNCTION;

FUNCTION cc_design_person_and_organization_correlation
 (e : cc_design_person_and_organization_assignment) : BOOLEAN;
  LOCAL
    po_role : STRING;
  END_LOCAL;
    po_role := e\person_and_organization_assignment.role.name;
  CASE po_role OF
    'request_recipient'      : IF SIZEOF (e.items) <>
                                 SIZEOF (QUERY (x <* e.items |
                                 SIZEOF(['CONFIG_CONTROL_DESIGN.' +
                                 'CHANGE_REQUEST',
                                 'CONFIG_CONTROL_DESIGN.' +
                                 'START_REQUEST'] *
                                 TYPEOF (x)) = 1))
                                 THEN RETURN(FALSE);
                               END_IF;
    'initiator'              : IF SIZEOF (e.items) <>
                                 SIZEOF (QUERY (x <* e.items |
                                 SIZEOF(['CONFIG_CONTROL_DESIGN.' +
                                 'CHANGE_REQUEST',
                                 'CONFIG_CONTROL_DESIGN.' +
                                 'START_REQUEST',
                                 'CONFIG_CONTROL_DESIGN.' +
                                 'START_WORK',
                                 'CONFIG_CONTROL_DESIGN.' +
```

```
                              'CHANGE'] *
                              TYPEOF (x)) = 1))
                              THEN RETURN(FALSE);
                            END_IF;
  'creator'                : IF SIZEOF (e.items) <>
                              SIZEOF (QUERY (x <* e.items |
                              SIZEOF (['CONFIG_CONTROL_DESIGN.' +
                              'PRODUCT_DEFINITION_FORMATION',
                              'CONFIG_CONTROL_DESIGN.' +
                              'PRODUCT_DEFINITION'] *
                              TYPEOF (x)) = 1))
                              THEN RETURN (FALSE);
                            END_IF;
  'part_supplier'          : IF SIZEOF (e.items) <>
                              SIZEOF (QUERY (x <* e.items |
                              'CONFIG_CONTROL_DESIGN.' +
                              'PRODUCT_DEFINITION_FORMATION'
                              IN TYPEOF (x)))
                              THEN RETURN(FALSE);
                            END_IF;
  'design_supplier'        : IF SIZEOF (e.items) <>
                              SIZEOF (QUERY (x <* e.items |
                              'CONFIG_CONTROL_DESIGN.' +
                              'PRODUCT_DEFINITION_FORMATION'
                              IN TYPEOF (x)))
                              THEN RETURN(FALSE);
                            END_IF;
  'design_owner'           : IF SIZEOF (e.items) <>
                              SIZEOF (QUERY (x <* e.items |
                              'CONFIG_CONTROL_DESIGN.PRODUCT'
                              IN TYPEOF (x)))
                              THEN RETURN(FALSE);
                            END_IF;
  'configuration_manager'  : IF SIZEOF (e.items) <>
                              SIZEOF (QUERY (x <* e.items |
                              'CONFIG_CONTROL_DESIGN.' +
                              'CONFIGURATION_ITEM'
                              IN TYPEOF (x)))
                              THEN RETURN(FALSE);
                            END_IF;
  'contractor'             : IF SIZEOF (e.items) <>
                              SIZEOF (QUERY (x <* e.items |
                              'CONFIG_CONTROL_DESIGN.CONTRACT'
                              IN TYPEOF (x)))
                              THEN RETURN(FALSE);
                            END_IF;
  'classification_officer' : IF SIZEOF (e.items) <>
                              SIZEOF (QUERY (x <* e.items |
                              'CONFIG_CONTROL_DESIGN.' +
                              'SECURITY_CLASSIFICATION'
                              IN TYPEOF (x))) THEN
                              RETURN(FALSE);
                            END_IF;
  OTHERWISE : RETURN(TRUE);
END_CASE;
RETURN (TRUE);
```

```
  END_FUNCTION;

FUNCTION check_associative_shape_aspects
  (sdr : shape_definition_representation) : BOOLEAN;
  LOCAL
    sr1, sr2 : SET OF representation;
    dm, pv   : SET OF representation;
  END_LOCAL;

  -- Check if the representation contains only two items
  IF (SIZEOF (sdr.used_representation.items) <> 2) THEN
     RETURN (FALSE);
  END_IF;

  -- Find the representations that use each item
  sr1 := using_representations(sdr.used_representation.items[1]);
  sr2 := using_representations(sdr.used_representation.items[2]);

  -- Find the representations in which item 1 representations are mapped
  REPEAT i := 1 TO HIINDEX(sr1);
    dm := representations_mapped_into(sr1[i]);
    REPEAT j := 1 TO HIINDEX(dm);
      IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_MODEL' IN
TYPEOF(dm[j]))
      THEN
        IF (dm[j] IN sr2) THEN RETURN (TRUE);
        END_IF;
        -- Find the views that the draughting model is mapped into
        pv := representations_mapped_into(dm[j]);
        REPEAT k := 1 TO HIINDEX(pv);
          IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRESENTATION_VIEW' IN
TYPEOF(pv[k]))
          THEN
            IF (pv[k] IN sr2) THEN RETURN (TRUE);
            END_IF;
          END_IF;
        END_REPEAT;
      END_IF;
    END_REPEAT;
  END_REPEAT;

  -- Find the representations in which item 2 representations are mapped
  REPEAT i := 1 TO HIINDEX(sr2);
    dm := representations_mapped_into(sr2[i]);
    REPEAT j := 1 TO HIINDEX(dm);
      IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DRAUGHTING_MODEL' IN
TYPEOF(dm[j]))
      THEN
        IF (dm[j] IN sr1) THEN RETURN (TRUE);
        END_IF;
        -- Find the views that the draughting model is mapped into
        pv := representations_mapped_into(dm[j]);
        REPEAT k := 1 TO HIINDEX(pv);
          IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PRESENTATION_VIEW' IN
TYPEOF(pv[k]))

          THEN
```

```
            IF (pv[k] IN sr1) THEN RETURN (TRUE);
            END_IF;
          END_IF;
        END_REPEAT;
      END_IF;
    END_REPEAT;
  END_REPEAT;
  RETURN(FALSE);

END_FUNCTION;

FUNCTION check_text_alignment (ct : composite_text) : BOOLEAN;

  LOCAL
    a : SET OF text_alignment := [];
  END_LOCAL;

  -- create a set of all the alignments
  REPEAT i := 1 TO HIINDEX (ct.collected_text);
    a := a + [ct.collected_text[i]\text_literal.alignment];
  END_REPEAT;

  -- if there is more than one element in the set
  -- then not all alignments were the same
  RETURN (SIZEOF(a) = 1);

END_FUNCTION;

FUNCTION check_text_font (ct : composite_text) : BOOLEAN;

  LOCAL
    f : SET OF font_select := [];
  END_LOCAL;

  -- build a set of all the fonts
  REPEAT i := 1 TO HIINDEX (ct.collected_text);
    f := f + [ct.collected_text[i]\text_literal.font];
  END_REPEAT;

  -- if there is more than one element in the set
  -- then not all fonts were the same
  RETURN (SIZEOF(f) <= 1);

END_FUNCTION;

 FUNCTION closed_shell_reversed (a_shell : closed_shell) :
                                        oriented_closed_shell;
  LOCAL
    the_reverse : oriented_closed_shell;
  END_LOCAL;
   IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_CLOSED_SHELL' IN TYPEOF
(a_shell) ) THEN

      the_reverse := dummy_tri ||
                   connected_face_set (
                      a_shell\connected_face_set.cfs_faces) ||
                   closed_shell () || oriented_closed_shell(
```

```
                    a_shell\oriented_closed_shell.closed_shell_element,
                      NOT(a_shell\oriented_closed_shell.orientation));
  ELSE
     the_reverse := dummy_tri ||
               connected_face_set (
                 a_shell\connected_face_set.cfs_faces) ||
               closed_shell () || oriented_closed_shell (a_shell, FALSE);
  END_IF;
  RETURN (the_reverse);

END_FUNCTION;

FUNCTION conditional_reverse (p       : BOOLEAN;
                              an_item : reversible_topology)
                                      : reversible_topology;
  IF p THEN
    RETURN (an_item);
  ELSE
    RETURN (topology_reversed (an_item));
  END_IF;

END_FUNCTION;

FUNCTION constraints_composite_curve_on_surface
               (c: composite_curve_on_surface) : BOOLEAN;
  LOCAL
    n_segments : INTEGER := SIZEOF(c.segments);
  END_LOCAL;

  REPEAT k := 1 TO n_segments;
    IF (NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN
          TYPEOF(c\composite_curve.segments[k].parent_curve))) AND
       (NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_CURVE' IN
          TYPEOF(c\composite_curve.segments[k].parent_curve))) AND

(NOT('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_CURVE_ON_SURFACE' IN

          TYPEOF(c\composite_curve.segments[k].parent_curve)))  THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN(TRUE);

END_FUNCTION;

FUNCTION constraints_geometry_shell_based_surface_model
            (m: shell_based_surface_model): BOOLEAN;
  LOCAL
    result : BOOLEAN := TRUE;
  END_LOCAL;

  REPEAT j := 1 TO SIZEOF(m.sbsm_boundary);
    IF (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OPEN_SHELL' IN
                    TYPEOF(m.sbsm_boundary[j])) AND
       (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CLOSED_SHELL' IN

          TYPEOF(m.sbsm_boundary[j]))))
```

```
    THEN
      result := FALSE;
      RETURN(result);
      (* A surface model is composed of OPEN_ and CLOSED_SHELLs. *)
    END_IF;
  END_REPEAT;
  RETURN(result);

END_FUNCTION;

FUNCTION constraints_geometry_shell_based_wireframe_model
                      (m : shell_based_wireframe_model) : BOOLEAN;
  LOCAL
    result : BOOLEAN := TRUE;
  END_LOCAL;

  REPEAT j := 1 TO SIZEOF(m.sbwm_boundary);
    IF (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.WIRE_SHELL' IN
                  TYPEOF(m.sbwm_boundary[j])) AND
       (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VERTEX_SHELL' IN

                    TYPEOF(m.sbwm_boundary[j]))))
    THEN
      result := FALSE;
      RETURN(result);
      (* A wireframe model is composed of WIRE_ and VERTEX_SHELLs *)
    END_IF;
  END_REPEAT;
  RETURN(result);

END_FUNCTION;

FUNCTION constraints_param_b_spline(degree, up_knots, up_cp : INTEGER;
                                                 knot_mult : LIST OF INTEGER;
                       knots : LIST OF parameter_value) : BOOLEAN;
  LOCAL
    result  : BOOLEAN := TRUE;
    k, sum  : INTEGER;
  END_LOCAL;

  (* Find sum of knot multiplicities. *)
  sum := knot_mult[1];

  REPEAT i := 2 TO up_knots;
    sum := sum + knot_mult[i];
  END_REPEAT;

  (* Check limits holding for all B-spline parametrisations *)
  IF (degree < 1) OR (up_knots < 2) OR (up_cp < degree) OR
        (sum <> (degree + up_cp + 2)) THEN
    result := FALSE;
    RETURN(result);
  END_IF;

  k := knot_mult[1];

  IF (k < 1) OR (k > degree + 1) THEN
```

```
    result := FALSE;
    RETURN(result);
  END_IF;

  REPEAT i := 2 TO up_knots;
    IF (knot_mult[i] < 1) OR (knots[i] <= knots[i-1]) THEN
      result := FALSE;
      RETURN(result);
    END_IF;

    k := knot_mult[i];

    IF (i < up_knots) AND (k > degree) THEN
      result := FALSE;
      RETURN(result);
    END_IF;

    IF (i = up_knots) AND (k > degree + 1) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
  END_REPEAT;
  RETURN(result);

END_FUNCTION;

  FUNCTION constraints_rectangular_composite_surface
          (s : rectangular_composite_surface) : BOOLEAN;

  (* Check the surface types *)
    REPEAT i := 1 TO s.n_u;
      REPEAT j := 1 TO s.n_v;
        IF NOT (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_SURFACE' IN
TYPEOF
                    (s.segments[i][j].parent_surface)) OR

('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.RECTANGULAR_TRIMMED_SURFACE' IN TYPEOF

                    (s.segments[i][j].parent_surface))) THEN
          RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_REPEAT;

  (* Check the transition codes, omitting the last row or column *)
  REPEAT i := 1 TO s.n_u-1;
    REPEAT j := 1 TO s.n_v;
      IF s.segments[i][j].u_transition = discontinuous THEN
        RETURN(FALSE);
      END_IF;
    END_REPEAT;
  END_REPEAT;

  REPEAT i := 1 TO s.n_u;
    REPEAT j := 1 TO s.n_v-1;
      IF s.segments[i][j].v_transition = discontinuous THEN
        RETURN(FALSE);
```

```
      END_IF;
    END_REPEAT;
  END_REPEAT;
  RETURN(TRUE);

END_FUNCTION;

FUNCTION cross_product (arg1, arg2 : direction) : vector;
  LOCAL
    mag    : REAL;
    res    : direction;
    v1,v2  : LIST[3:3] OF REAL;
    result : vector;
  END_LOCAL;

  IF ( NOT EXISTS (arg1) OR (arg1.dim = 2)) OR
     ( NOT EXISTS (arg2) OR (arg2.dim = 2)) THEN
    RETURN(?);
  ELSE
    BEGIN
      v1  := normalise(arg1).direction_ratios;
      v2  := normalise(arg2).direction_ratios;
      res := dummy_gri || direction([(v1[2]*v2[3] - v1[3]*v2[2]),
            (v1[3]*v2[1] - v1[1]*v2[3]), (v1[1]*v2[2] - v1[2]*v2[1])]);
      mag := 0.0;
      REPEAT i := 1 TO 3;
        mag := mag + res.direction_ratios[i]*res.direction_ratios[i];
      END_REPEAT;
      IF (mag > 0.0) THEN
        result := dummy_gri || vector(res, SQRT(mag));
      ELSE
        result := dummy_gri || vector(arg1, 0.0);
      END_IF;
      RETURN(result);
    END;
  END_IF;

END_FUNCTION;

 FUNCTION curve_weights_positive(b: rational_b_spline_curve) : BOOLEAN;
   LOCAL
     result : BOOLEAN := TRUE;
   END_LOCAL;

   REPEAT i := 0 TO b.upper_index_on_control_points;
     IF b.weights[i] <= 0.0  THEN
       result := FALSE;
       RETURN(result);
     END_IF;
   END_REPEAT;
   RETURN(result);

 END_FUNCTION;

FUNCTION default_tolerance_table_cell_wr2
 (agg : compound_item_definition) : BOOLEAN;
BEGIN
```

```
IF SIZEOF(agg) <= 5 THEN
  RETURN(TRUE);
ELSE
  RETURN(FALSE);
END_IF;
END;

      END_FUNCTION;

FUNCTION default_tolerance_table_cell_wr3
 (agg : compound_item_definition) : BOOLEAN;

BEGIN
IF (SIZEOF(QUERY ( i <* agg |
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF(i))
  AND (i\representation_item.name = 'significant number of digits')) )) = 1) OR
((SIZEOF(QUERY ( i <* agg |
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF(i)) AND
  (i\representation_item.name = 'lower limit')) )) = 1) AND
(SIZEOF( QUERY ( i <* agg |
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF(i)) AND

  (i\representation_item.name = 'upper limit')) )) = 1)) THEN
  RETURN(TRUE);
ELSE
  RETURN(FALSE);
END_IF;
END;

      END_FUNCTION;

FUNCTION default_tolerance_table_cell_wr4
 (agg : compound_item_definition) : BOOLEAN;
BEGIN
IF (SIZEOF(QUERY ( i <* agg |
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF(i)) AND
  (i\representation_item.name = 'plus minus tolerance value')) )) = 1) OR
((SIZEOF(QUERY ( i <* agg |
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF(i)) AND (
  i\representation_item.name = 'lower tolerance value')) )) = 1) AND
(SIZEOF( QUERY ( i <* agg |
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN
TYPEOF(i)) AND (

  i\representation_item.name = 'upper tolerance value')) )) = 1)) THEN
  RETURN(TRUE);
ELSE
  RETURN(FALSE);
END_IF;
END;

      END_FUNCTION;
```

```
FUNCTION default_tolerance_table_cell_wr5
 (agg : compound_item_definition) : BOOLEAN;
BEGIN
IF (SIZEOF(QUERY ( i <* agg |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)) )) <= 1) AND
(SIZEOF(QUERY ( i <* agg |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)) )) =
  SIZEOF(QUERY ( i <* agg |
(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)) AND

  (i\representation_item.name = 'cell description'))) ))
THEN
  RETURN(TRUE);
ELSE
  RETURN(FALSE);
END_IF;
END;

      END_FUNCTION;

FUNCTION derive_dimensional_exponents
 (x : unit) : dimensional_exponents;
  LOCAL
    result : dimensional_exponents := dimensional_exponents(0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0);
  END_LOCAL;

  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DERIVED_UNIT' IN TYPEOF(x) THEN

    REPEAT i := LOINDEX(x.elements) TO HIINDEX(x.elements);
      result.length_exponent := result.length_exponent + (x.elements[i].exponent *
x.elements[i].unit.dimensions.length_exponent);
      result.mass_exponent := result.mass_exponent + (x.elements[i].exponent *
x.elements[i].unit.dimensions.mass_exponent);
      result.time_exponent := result.time_exponent + (x.elements[i].exponent *
x.elements[i].unit.dimensions.time_exponent);
      result.electric_current_exponent := result.electric_current_exponent +
(x.elements[i].exponent * x.elements[i].unit.dimensions.electric_current_exponent);
      result.thermodynamic_temperature_exponent :=
result.thermodynamic_temperature_exponent + (x.elements[i].exponent *
x.elements[i].unit.dimensions.thermodynamic_temperature_exponent);
      result.amount_of_substance_exponent := result.amount_of_substance_exponent +
(x.elements[i].exponent *
x.elements[i].unit.dimensions.amount_of_substance_exponent);
      result.luminous_intensity_exponent := result.luminous_intensity_exponent +
(x.elements[i].exponent *
x.elements[i].unit.dimensions.luminous_intensity_exponent);
    END_REPEAT;
  ELSE
    result := x.dimensions;
  END_IF;
  RETURN (result);
```

```
END_FUNCTION;

FUNCTION dimension_of(item : geometric_representation_item) :
  dimension_count;
  LOCAL
    x   : SET OF representation;
    y   : representation_context;
    dim : dimension_count;
  END_LOCAL;
  -- For cartesian_point, direction, or vector dimension is determined by
  -- counting components.
    IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT' IN TYPEOF(item)
THEN
       dim := SIZEOF(item\cartesian_point.coordinates);
       RETURN(dim);
    END_IF;
    IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DIRECTION' IN TYPEOF(item) THEN
       dim := SIZEOF(item\direction.direction_ratios);
       RETURN(dim);
    END_IF;
    IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VECTOR' IN TYPEOF(item) THEN

       dim := SIZEOF(item\vector.orientation\direction.direction_ratios);
       RETURN(dim);
    END_IF;
  -- For all other types of geometric_representation_item dim is obtained
  -- via context.
  -- Find the set of representation in which the item is used.

  x := using_representations(item);

  -- Determines the dimension_count of the
  -- geometric_representation_context. Note that the
  -- RULE compatible_dimension ensures that the context_of_items
  -- is of type geometric_representation_context and has
  -- the same dimension_count for all values of x.
  -- The SET x is non-empty since this is required by WR1 of
  -- representation_item.
    y := x[1].context_of_items;
    dim := y\geometric_representation_context.coordinate_space_dimension;
    RETURN (dim);


END_FUNCTION;

FUNCTION dimensions_for_si_unit
 (n : si_unit_name) : dimensional_exponents;
 CASE n OF
    metre          : RETURN (dimensional_exponents
                          (1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0));
    gram           : RETURN (dimensional_exponents
                         (0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0));
    second         : RETURN (dimensional_exponents
                         (0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0));
    ampere         : RETURN (dimensional_exponents
                         (0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0));
    kelvin         : RETURN (dimensional_exponents
```

```
                        (0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0));
    mole           : RETURN (dimensional_exponents
                        (0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0));
    candela        : RETURN (dimensional_exponents
                        (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0));
    radian         : RETURN (dimensional_exponents
                          (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0));
    steradian      : RETURN (dimensional_exponents
                        (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0));
    hertz          : RETURN (dimensional_exponents
                        (0.0, 0.0, -1.0, 0.0, 0.0, 0.0, 0.0));
    newton         : RETURN (dimensional_exponents
                        (1.0, 1.0, -2.0, 0.0, 0.0, 0.0, 0.0));
    pascal         : RETURN (dimensional_exponents
                        (-1.0, 1.0, -2.0, 0.0, 0.0, 0.0, 0.0));
    joule          : RETURN (dimensional_exponents
                        (2.0, 1.0, -2.0, 0.0, 0.0, 0.0, 0.0));
    watt           : RETURN (dimensional_exponents
                        (2.0, 1.0, -3.0, 0.0, 0.0, 0.0, 0.0));
    coulomb        : RETURN (dimensional_exponents
                        (0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0));
    volt           : RETURN (dimensional_exponents
                        (2.0, 1.0, -3.0, -1.0, 0.0, 0.0, 0.0));
    farad          : RETURN (dimensional_exponents
                        (-2.0, -1.0, 4.0, 1.0, 0.0, 0.0, 0.0));
    ohm            : RETURN (dimensional_exponents
                        (2.0, 1.0, -3.0, -2.0, 0.0, 0.0, 0.0));
    siemens        : RETURN (dimensional_exponents
                        (-2.0, -1.0, 3.0, 2.0, 0.0, 0.0, 0.0));
    weber          : RETURN (dimensional_exponents
                        (2.0, 1.0, -2.0, -1.0, 0.0, 0.0, 0.0));
    tesla          : RETURN (dimensional_exponents
                        (0.0, 1.0, -2.0, -1.0, 0.0, 0.0, 0.0));
    henry          : RETURN (dimensional_exponents
                        (2.0, 1.0, -2.0, -2.0, 0.0, 0.0, 0.0));
    degree_Celsius : RETURN (dimensional_exponents
                        (0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0));
    lumen          : RETURN (dimensional_exponents
                        (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0));
    lux            : RETURN (dimensional_exponents
                        (-2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0));
    becquerel      : RETURN (dimensional_exponents
                        (0.0, 0.0, -1.0, 0.0, 0.0, 0.0, 0.0));
    gray           : RETURN (dimensional_exponents
                        (2.0, 0.0, -2.0, 0.0, 0.0, 0.0, 0.0));
    sievert        : RETURN (dimensional_exponents
                        (2.0, 0.0, -2.0, 0.0, 0.0, 0.0, 0.0));
    OTHERWISE      : RETURN (?);
  END_CASE;

END_FUNCTION;

 FUNCTION dot_product(arg1, arg2 : direction) : REAL;
   LOCAL
     scalar : REAL;
     vec1, vec2: direction;
     ndim : INTEGER;
```

```
  END_LOCAL;

  IF NOT EXISTS (arg1) OR NOT EXISTS (arg2) THEN
    scalar := ?;
    (* When function is called with invalid data an indeterminate result
    is returned *)
  ELSE
    IF (arg1.dim <> arg2.dim) THEN
      scalar := ?;
    (* When function is called with invalid data an indeterminate result
    is returned *)
    ELSE
      BEGIN
        vec1   := normalise(arg1);
        vec2   := normalise(arg2);
        ndim   := arg1.dim;
        scalar := 0.0;
        REPEAT  i := 1 TO ndim;
          scalar := scalar +
                      vec1.direction_ratios[i]*vec2.direction_ratios[i];
        END_REPEAT;
      END;
    END_IF;
  END_IF;
  RETURN (scalar);

END_FUNCTION;

FUNCTION edge_reversed (an_edge : edge) : oriented_edge;
  LOCAL
    the_reverse : oriented_edge;
  END_LOCAL;

  IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_EDGE' IN TYPEOF
(an_edge) ) THEN

    the_reverse  := dummy_tri ||
            edge(an_edge.edge_end, an_edge.edge_start) ||
            oriented_edge(an_edge\oriented_edge.edge_element,
                       NOT (an_edge\oriented_edge.orientation)) ;
  ELSE
    the_reverse := dummy_tri ||
            edge(an_edge.edge_end, an_edge.edge_start) ||
            oriented_edge(an_edge, FALSE);
  END_IF;
  RETURN (the_reverse);

END_FUNCTION;

FUNCTION face_bound_reversed (a_face_bound : face_bound) : face_bound;
  LOCAL
    the_reverse : face_bound ;
  END_LOCAL;
  IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE_OUTER_BOUND' IN TYPEOF
(a_face_bound) ) THEN

    the_reverse := dummy_tri ||
```

```
                      face_bound(a_face_bound\face_bound.bound,
                           NOT (a_face_bound\face_bound.orientation))
                            || face_outer_bound() ;
   ELSE
     the_reverse := dummy_tri ||
         face_bound(a_face_bound.bound, NOT(a_face_bound.orientation));
   END_IF;
  RETURN (the_reverse);

 END_FUNCTION;

 FUNCTION face_reversed (a_face : face) : oriented_face;
   LOCAL
     the_reverse : oriented_face ;
   END_LOCAL;
   IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_FACE' IN TYPEOF
(a_face) ) THEN

     the_reverse := dummy_tri ||
       face(set_of_topology_reversed(a_face.bounds)) ||
          oriented_face(a_face\oriented_face.face_element,
                           NOT (a_face\oriented_face.orientation)) ;
   ELSE
     the_reverse := dummy_tri ||
       face(set_of_topology_reversed(a_face.bounds)) ||
                              oriented_face(a_face, FALSE) ;
   END_IF;
      RETURN (the_reverse);

 END_FUNCTION;

FUNCTION field_in_table (field : table_record_field_representation;
                         table : annotation_table_occurrence): BOOLEAN;
  LOCAL
    table_rep : table_representation;
    symbol_rep_rel_set : SET OF symbol_representation_relationship;
    mapped_item_set : SET OF mapped_item;
    table_record_rep_set : SET OF table_record_representation := [];
  END_LOCAL;

  table_rep := table\styled_item.item\mapped_item.mapping_source.
    mapped_representation;
  mapped_item_set := QUERY(item <* table_rep.items |
                       ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM' IN
                        TYPEOF(item))
                                 AND
                       ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                        'TABLE_RECORD_REPRESENTATION' IN
                        TYPEOF(item\mapped_item.mapping_source.
                                   mapped_representation ))
                     );

  REPEAT i := 1 TO HIINDEX(mapped_item_set);
    table_record_rep_set := table_record_rep_set +
           mapped_item_set[i].mapping_source.mapped_representation;
  END_REPEAT;
```

```
  symbol_rep_rel_set := bag_to_set (USEDIN(table_rep,
                                'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                                'REPRESENTATION_RELATIONSHIP.REP_1'));

  REPEAT i := 1 TO HIINDEX(symbol_rep_rel_set);
     table_record_rep_set := table_record_rep_set +
              symbol_rep_rel_set[i]\representation_relationship.rep_2;
  END_REPEAT;

  IF SIZEOF(QUERY( table_record_rep <* table_record_rep_set |
--               (SIZEOF(QUERY( symbol_rep_rel <* USEDIN(table_record_rep,
--                              'PRESENTATION_DEFINITION_SCHEMA.'+
--                              'SYMBOL_REPRESENTATION_RELATIONSHIP.REP_1') |
--                        symbol_rep_rel\representation_relationship.rep_2 :=: field
              (SIZEOF(QUERY( rep_rel <* USEDIN(table_record_rep,
                                 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'+
                                 'REPRESENTATION_RELATIONSHIP.REP_1') |
                        ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                        'SYMBOL_REPRESENTATION_RELATIONSHIP' IN TYPEOF(rep_rel)) AND
                        (rep_rel.rep_2 :=: field)
                        )) > 0)
                        OR
              (SIZEOF(QUERY(item <* table_record_rep.items |
                         ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MAPPED_ITEM'
IN

                          TYPEOF(item))
                                  AND

                         (field :=: item\mapped_item.mapping_source.
                                    mapped_representation )
                          )) > 0)
            )) = 0 THEN
    RETURN(FALSE);
  END_IF;

  RETURN(TRUE);


END_FUNCTION;

FUNCTION first_proj_axis(z_axis, arg : direction) : direction;
  LOCAL
    x_axis : direction;
    v      : direction;
    z      : direction;
    x_vec  : vector;
  END_LOCAL;

  IF (NOT EXISTS(z_axis)) THEN
    RETURN (?) ;
  ELSE
    z := normalise(z_axis);
    IF NOT EXISTS(arg) THEN
      IF ((z.direction_ratios <> [1.0,0.0,0.0]) AND
          (z.direction_ratios <> [-1.0,0.0,0.0]))  THEN
        v :=  dummy_gri || direction([1.0,0.0,0.0]);
```

```
      ELSE
        v := dummy_gri || direction([0.0,1.0,0.0]);
      END_IF;
    ELSE
      IF  (arg.dim <> 3) THEN
        RETURN (?) ;
      END_IF;
      IF ((cross_product(arg,z).magnitude) = 0.0) THEN
        RETURN (?);
      ELSE
        v := normalise(arg);
      END_IF;
    END_IF;
    x_vec := scalar_times_vector(dot_product(v, z), z);
    x_axis := vector_difference(v, x_vec).orientation;
    x_axis := normalise(x_axis);
  END_IF;
  RETURN(x_axis);

END_FUNCTION;

FUNCTION gbsf_check_curve
 (cv : representation_item) : BOOLEAN;
  IF SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BOUNDED_CURVE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONIC',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_REPLICA',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_CURVE_3D'] * TYPEOF(cv)) > 1
THEN
    RETURN (FALSE);
  END_IF;
  IF SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CIRCLE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ELLIPSE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TRIMMED_CURVE'] * TYPEOF(cv)) = 1 THEN
    RETURN (TRUE);
  ELSE
    IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_CURVE' IN TYPEOF(cv))
AND (cv\b_spline_curve.self_intersect = FALSE) OR (cv\b_spline_curve.self_intersect
= UNKNOWN)) THEN
      RETURN (TRUE);
    ELSE
      IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_CURVE' IN
TYPEOF(cv)) AND (cv\composite_curve.self_intersect = FALSE) OR
(cv\composite_curve.self_intersect = UNKNOWN)) THEN
        RETURN (SIZEOF(QUERY(seg <* cv\composite_curve.segments | NOT
(gbsf_check_curve(seg.parent_curve)))) = 0);
      ELSE
        IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_REPLICA' IN TYPEOF(cv)
THEN
          RETURN (gbsf_check_curve(cv\curve_replica.parent_curve));
        ELSE
          IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_CURVE_3D' IN
TYPEOF(cv)) AND ((cv\offset_curve_3d.self_intersect = FALSE) OR
(cv\offset_curve_3d.self_intersect = UNKNOWN)) AND (NOT
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN
TYPEOF(cv\offset_curve_3d.basis_curve)))) THEN
            RETURN (gbsf_check_curve(cv\offset_curve_3d.basis_curve));
```

```
          ELSE
            IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN TYPEOF(cv)
THEN
              RETURN
((gbsf_check_curve(cv\pcurve.reference_to_curve\representation.items[1])) AND
(gbsf_check_surface(cv\pcurve.basis_surface)));
            ELSE
              IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN
TYPEOF(cv) THEN
                IF (SIZEOF(cv\polyline.points) >= 3) THEN
                  RETURN (TRUE);
                END_IF;
              ELSE
                IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_CURVE' IN
TYPEOF(cv) THEN
                  IF gbsf_check_curve(cv\surface_curve.curve_3d) THEN
                    REPEAT i := 1 TO SIZEOF(cv\surface_curve.associated_geometry);
                      IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE' IN
TYPEOF(cv\surface_curve.associated_geometry[i]) THEN
                        IF NOT
gbsf_check_surface(cv\surface_curve.associated_geometry[i]) THEN
                          RETURN (FALSE);
                        END_IF;
                      ELSE
                        IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN
TYPEOF(cv\surface_curve.associated_geometry[i]) THEN

                        IF NOT
gbsf_check_curve(cv\surface_curve.associated_geometry[i]) THEN
                          RETURN (FALSE);
                        END_IF;
                      END_IF;
                    END_REPEAT;
                    RETURN (TRUE);
                  END_IF;
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);

      END_FUNCTION;

FUNCTION gbsf_check_point
 (pnt : point) : BOOLEAN;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT' IN TYPEOF(pnt)
THEN
    RETURN (TRUE);
  ELSE
    IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_CURVE' IN TYPEOF(pnt)
THEN
      RETURN (gbsf_check_curve(pnt\point_on_curve.basis_curve));
```

```
    ELSE
      IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_SURFACE' IN
TYPEOF(pnt) THEN
        RETURN (gbsf_check_surface(pnt\point_on_surface.basis_surface));
      ELSE
        IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.DEGENERATE_PCURVE' IN
TYPEOF(pnt) THEN

          RETURN
((gbsf_check_curve(pnt\degenerate_pcurve.reference_to_curve\representation.items[1]
)) AND (gbsf_check_surface(pnt\degenerate_pcurve.basis_surface)));
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);

      END_FUNCTION;

FUNCTION gbsf_check_surface
 (sf : surface) : BOOLEAN;
  IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_SURFACE' IN TYPEOF(sf))
AND (sf\b_spline_surface.self_intersect = FALSE) OR
(sf\b_spline_surface.self_intersect = UNKNOWN)) THEN
    RETURN (TRUE);
  ELSE
    IF SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SPHERICAL_SURFACE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TOROIDAL_SURFACE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_BOUNDED_SURFACE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.RECTANGULAR_TRIMMED_SURFACE'] *
TYPEOF(sf)) = 1 THEN
      RETURN (TRUE);
    ELSE
      IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_SURFACE' IN
TYPEOF(sf)) AND (sf\offset_surface.self_intersect = FALSE) OR
(sf\offset_surface.self_intersect = UNKNOWN)) THEN
        RETURN (gbsf_check_surface(sf\offset_surface.basis_surface));
      ELSE
        IF
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.RECTANGULAR_COMPOSITE_SURFACE' IN
TYPEOF(sf) THEN
          REPEAT i := 1 TO SIZEOF(sf\rectangular_composite_surface.segments);
            REPEAT j := 1 TO SIZEOF(sf\rectangular_composite_surface.segments[i]);
              IF NOT
(gbsf_check_surface(sf\rectangular_composite_surface.segments[i][j].parent_surface)
) THEN
                RETURN (FALSE);
              END_IF;
            END_REPEAT;
          END_REPEAT;
          RETURN (TRUE);
        ELSE
          IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_REPLICA' IN
TYPEOF(sf) THEN
            RETURN (gbsf_check_surface(sf\surface_replica.parent_surface));
          ELSE
```

```
            IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_OF_REVOLUTION'
IN TYPEOF(sf) THEN

              RETURN (gbsf_check_curve(sf\swept_surface.swept_curve));
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);

      END_FUNCTION;

FUNCTION get_basis_surface (c : curve_on_surface) : SET[0:2] OF surface;
  LOCAL
    surfs  : SET[0:2] OF surface;
    n      : INTEGER;
  END_LOCAL;
  surfs := [];
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN TYPEOF (c) THEN
    surfs := [c\pcurve.basis_surface];
  ELSE
    IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_CURVE' IN TYPEOF (c)
THEN
      n := SIZEOF(c\surface_curve.associated_geometry);
      REPEAT i := 1 TO n;
      surfs := surfs +
               associated_surface(c\surface_curve.associated_geometry[i]);
      END_REPEAT;
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_CURVE_ON_SURFACE' IN
TYPEOF (c) THEN

   (* For a composite_curve_on_surface the basis_surface is the intersection
    of the basis_surfaces of all the segments. *)
     n := SIZEOF(c\composite_curve.segments);
     surfs := get_basis_surface(
                     c\composite_curve.segments[1].parent_curve);
     IF n > 1 THEN
       REPEAT i := 2 TO n;
         surfs := surfs * get_basis_surface(
                  c\composite_curve.segments[i].parent_curve);
       END_REPEAT;
     END_IF;

  END_IF;
  RETURN(surfs);

END_FUNCTION;

FUNCTION get_description_value
 (obj : description_attribute_select) : text;
  LOCAL
```

```
    description_bag : BAG OF description_attribute := (USEDIN(obj,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' + 'DESCRIPTION_ATTRIBUTE.' +
'DESCRIBED_ITEM'));

  END_LOCAL;

  IF SIZEOF(description_bag) = 1 THEN
    RETURN (description_bag[1].attribute_value);
  ELSE
    RETURN (?);
  END_IF;

END_FUNCTION;

FUNCTION get_id_value
 (obj : id_attribute_select) : identifier;
LOCAL
      id_bag : BAG OF id_attribute := (USEDIN(obj,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' + 'ID_ATTRIBUTE.' +
'IDENTIFIED_ITEM'));

END_LOCAL;
IF SIZEOF(id_bag) = 1 THEN
      RETURN (id_bag[1].attribute_value);
ELSE
      RETURN (?);
END_IF;

END_FUNCTION;

FUNCTION get_name_value
 (obj : name_attribute_select) : label;
  LOCAL
    name_bag : BAG OF name_attribute := (USEDIN(obj,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' + 'NAME_ATTRIBUTE.' + 'NAMED_ITEM'));

  END_LOCAL;

  IF SIZEOF(name_bag) = 1 THEN
    RETURN (name_bag[1].attribute_value);
  ELSE
    RETURN (?);
  END_IF;

END_FUNCTION;

FUNCTION get_role
 (obj : role_select) : object_role;
  LOCAL
    role_bag : BAG OF role_association := (USEDIN(obj,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' + 'ROLE_ASSOCIATION.' +
'ITEM_WITH_ROLE'));

  END_LOCAL;

  IF SIZEOF(role_bag) = 1 THEN
    RETURN (role_bag[1].role);
```

```
  ELSE
    RETURN (?);
  END_IF;

END_FUNCTION;

FUNCTION get_shape_aspect_property_definition_representations
 (s_a_instance : shape_aspect) : SET OF property_definition_representation;
LOCAL
pd_set : SET OF property_definition := [];
pdr_set : SET OF property_definition_representation := [] ;
END_LOCAL;
pd_set := bag_to_set(USEDIN(s_a_instance,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PROPERTY_DEFINITION.DEFINITION'));
IF (SIZEOF(pd_set) < 1) THEN
RETURN (pdr_set);
END_IF;
REPEAT i := 1 TO HIINDEX(pd_set);
pdr_set := pdr_set + (QUERY(pdr <* USEDIN(pd_set[i],
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
'PROPERTY_DEFINITION_REPRESENTATION.' + 'DEFINITION') |
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_DEFINITION_REPRESENTATION' IN
TYPEOF(pdr)));

END_REPEAT;
RETURN (pdr_set);

END_FUNCTION;

FUNCTION item_in_context
  (item  : representation_item;
   cntxt : representation_context) : BOOLEAN;
  LOCAL
    y : BAG OF representation_item;
  END_LOCAL;
  -- If there is one or more representation using both the item
  -- and cntxt return true.
  IF
SIZEOF(USEDIN(item,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REPRESENTATION.ITEMS
')
    * cntxt.representations_in_context) > 0 THEN
    RETURN (TRUE);
    -- Determine the bag of representation_items that reference
    -- item
    ELSE y := QUERY(z <* USEDIN (item , '') |
           'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REPRESENTATION_ITEM' IN
TYPEOF(z));

      -- Ensure that the bag is not empty
      IF SIZEOF(y) > 0 THEN
      -- For each element in the bag
      REPEAT i := 1 TO HIINDEX(y);
        -- Check to see it is an item in the input cntxt.
        IF item_in_context(y[i], cntxt) THEN
          RETURN (TRUE);
        END_IF;
      END_REPEAT;
```

```
    END_IF;
  END_IF;
  -- Return false when all possible branches have been checked
  -- with no success.
  RETURN (FALSE);

END_FUNCTION;

FUNCTION leap_year
 (year : year_number) : BOOLEAN;
  IF ((((year MOD 4) = 0) AND ((year MOD 100) <> 0)) OR ((year MOD 400) = 0)) THEN
    RETURN (TRUE);
  ELSE
    RETURN (FALSE);
  END_IF;

END_FUNCTION;

 FUNCTION list_face_loops(f: face) : LIST[0:?] OF loop;
   LOCAL
     loops : LIST[0:?] OF loop := [];
   END_LOCAL;

   REPEAT i := 1 TO SIZEOF(f.bounds);
     loops := loops +(f.bounds[i].bound);
   END_REPEAT;

   RETURN(loops);

 END_FUNCTION;

 FUNCTION list_of_topology_reversed (a_list
                                    : list_of_reversible_topology_item)
                                    : list_of_reversible_topology_item;
   LOCAL
     the_reverse : list_of_reversible_topology_item;
   END_LOCAL;

   the_reverse := [];
   REPEAT i := 1 TO SIZEOF (a_list);
     the_reverse := topology_reversed (a_list [i]) + the_reverse;
   END_REPEAT;

   RETURN (the_reverse);

 END_FUNCTION;

 FUNCTION list_to_array(lis : LIST [0:?] OF GENERIC : T;
                      low,u : INTEGER) : ARRAY [low:u] OF GENERIC : T;
   LOCAL
     n   : INTEGER;
     res : ARRAY [low:u] OF GENERIC : T;
   END_LOCAL;

   n := SIZEOF(lis);
   IF (n <> (u-low +1)) THEN
     RETURN(?);
```

```
    ELSE
      res := [lis[1] : n];
      REPEAT i := 2 TO n;
        res[low+i-1] := lis[i];
      END_REPEAT;
      RETURN(res);
    END_IF;

  END_FUNCTION;

  FUNCTION list_to_set(l : LIST [0:?] OF GENERIC:T) : SET OF GENERIC:T;
    LOCAL
      s : SET OF GENERIC:T := [];
    END_LOCAL;

    REPEAT i := 1 TO SIZEOF(l);
      s := s + l[i];
    END_REPEAT;

    RETURN(s);

  END_FUNCTION;

  FUNCTION make_array_of_array(lis : LIST[1:?] OF LIST [1:?] OF GENERIC : T;
                               low1, u1, low2, u2 : INTEGER):
                 ARRAY [low1:u1] OF ARRAY [low2:u2] OF GENERIC : T;
    LOCAL
      res   : ARRAY[low1:u1] OF ARRAY [low2:u2] OF GENERIC : T;
    END_LOCAL;

 (* Check input dimensions for consistency *)
    IF (u1-low1+1) <> SIZEOF(lis) THEN
      RETURN (?);
    END_IF;
    IF (u2 - low2 + 1 ) <> SIZEOF(lis[1]) THEN
      RETURN (?) ;
    END_IF;
 (* Initialise res with values from lis[1] *)
    res := [list_to_array(lis[1], low2, u2) : (u1-low1 + 1)];
    REPEAT i := 2 TO HIINDEX(lis);
      IF (u2-low2+1) <> SIZEOF(lis[i]) THEN
        RETURN (?);
      END_IF;
      res[low1+i-1] := list_to_array(lis[i], low2, u2);
    END_REPEAT;

    RETURN (res);

  END_FUNCTION;

  FUNCTION mixed_loop_type_set(l: SET[0:?] OF loop): LOGICAL;
     LOCAL
       poly_loop_type: LOGICAL;
     END_LOCAL;
     IF(SIZEOF(l) <= 1) THEN
       RETURN(FALSE);
     END_IF;
```

```
    poly_loop_type := ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLY_LOOP' IN
TYPEOF(l[1]));
    REPEAT i := 2 TO SIZEOF(l);
      IF(('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLY_LOOP' IN TYPEOF(l[i]))
<> poly_loop_type)

          THEN
          RETURN(TRUE);
       END_IF;
    END_REPEAT;

    RETURN(FALSE);

  END_FUNCTION;

 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;

FUNCTION msf_curve_check (cv : representation_item) : BOOLEAN;

(* This function varifies the validity of a curve in the context of a
   manifold surface model. Representation_items are
   valid input, however, they are supposed to be curves; otherwise
   this function will return false.
*)

(* complex subtypes of curve that are both bounded_curve and
   oneof conic, curve_replica, line, or offset_curve_3d are not
   valid
*)
IF SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.BOUNDED_CURVE',
  'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONIC',
  'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_REPLICA',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE',
  'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_CURVE_3D'] * TYPEOF(cv)) > 1
THEN
  RETURN(FALSE);
END_IF;

(* b_spline_curves shall not self-intersect
 *)
IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_CURVE' IN TYPEOF (cv))
AND
  (cv\b_spline_curve.self_intersect = FALSE)OR
  (cv\b_spline_curve.self_intersect = UNKNOWN)) THEN
  RETURN(TRUE);
ELSE
```

```
  (* conics and lines are valid curve types
   *)
  IF SIZEOF (['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONIC',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE']
    * TYPEOF (cv)) = 1 THEN
    RETURN(TRUE);
  ELSE

    (* a curve_replica shall reference a valid curve
     *)
    IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_REPLICA' IN TYPEOF(cv)
THEN
      RETURN (msf_curve_check(cv\curve_replica.parent_curve));
    ELSE

      (* an offset_curve_3d shall not self-intersect and
         shall reference a valid curve; a polyline is not a
         valid basis_curve
       *)
      IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_CURVE_3D' IN TYPEOF
(cv))
        AND
        ((cv\offset_curve_3d.self_intersect = FALSE) OR
        (cv\offset_curve_3d.self_intersect = UNKNOWN))
        AND
        (NOT ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN TYPEOF
        (cv\offset_curve_3d.basis_curve)))) THEN
        RETURN (msf_curve_check(cv\offset_curve_3d.basis_curve));
      ELSE

        (* a pcurve shall reference a valid curve and a valid
           basis_surface
         *)
        IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN TYPEOF(cv) THEN
          RETURN ((msf_curve_check
          (cv\pcurve.reference_to_curve\representation.items[1])) AND
          (msf_surface_check(cv\pcurve.basis_surface)));
        ELSE

          (* a surface_curve references a curve_3d and one or
             two pcurves or one or two surfaces or one of
             each; all of these references shall be valid
           *)
          IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_CURVE' IN
TYPEOF(cv) THEN

            (* if the curve reference is correct, check also the rest
             *)
            IF msf_curve_check(cv\surface_curve.curve_3d) THEN
              REPEAT i := 1 TO SIZEOF
              (cv\surface_curve.associated_geometry);

                (* do for one or two associated_geometrys:
                 *)
                IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE' IN
                  TYPEOF (cv\surface_curve.associated_geometry[i]) THEN
                  IF NOT msf_surface_check
```

```
                    (cv\surface_curve.associated_geometry[i]) THEN
                    RETURN(FALSE);
                  END_IF;
                ELSE
                  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PCURVE' IN TYPEOF
                    (cv\surface_curve.associated_geometry[i]) THEN
                    IF NOT msf_curve_check
                      (cv\surface_curve.associated_geometry[i]) THEN
                      RETURN(FALSE);
                    END_IF;
                  END_IF;
                END_IF;
              END_REPEAT;
              RETURN(TRUE);
            END_IF;
          ELSE

            (* a polyline shall have at least 3 points
             *)
            IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' IN TYPEOF(cv)
THEN

              IF (SIZEOF (cv\polyline.points) >= 3) THEN RETURN (TRUE);
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
END_IF;
(* FALSE is returned if the input parameter cv is not a valid curve.
 *)
RETURN (FALSE);

END_FUNCTION;

FUNCTION msf_surface_check (surf : surface) : BOOLEAN;

(* This function varifies the validity of a surface in the
   context of a manifold surface model.
*)

  (* elementary_surfaces are valid surface types
   *)
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ELEMENTARY_SURFACE' IN
TYPEOF(surf) THEN
    RETURN(TRUE);
  ELSE

    (* a swept_surface shall have a valid sweeping curve
     *)
    IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SWEPT_SURFACE' IN TYPEOF (surf)
THEN
      RETURN (msf_curve_check(surf\swept_surface.swept_curve));
    ELSE
```

```
      (* an offset_surface shall not self-intersect and shall
         reference a valid surface
       *)
      IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_SURFACE' IN TYPEOF
(surf)) AND
        (surf\offset_surface.self_intersect = FALSE) OR
        (surf\offset_surface.self_intersect = UNKNOWN)) THEN
        RETURN (msf_surface_check(surf\offset_surface.basis_surface));
      ELSE

        (* a surface_replica shall have a valid parent surface
         *)
        IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_REPLICA' IN
TYPEOF(surf) THEN
          RETURN(msf_surface_check(surf\surface_replica.parent_surface));
        ELSE

          (* a b_spline_surface shall not self-intersect
           *)
          IF (('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_SURFACE' IN
TYPEOF(surf)) AND

            (surf\b_spline_surface.self_intersect = FALSE) OR
            (surf\b_spline_surface.self_intersect = UNKNOWN)) THEN
            RETURN(TRUE);
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION;

 FUNCTION normalise (arg : vector_or_direction) : vector_or_direction;
   LOCAL
     ndim   : INTEGER;
     v      : direction;
     result : vector_or_direction;
     vec    : vector;
     mag    : REAL;
   END_LOCAL;

   IF NOT EXISTS (arg) THEN
     result := ?;
 (* When function is called with invalid data a NULL result is returned *)
   ELSE
     ndim := arg.dim;
     IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VECTOR' IN TYPEOF(arg) THEN
       BEGIN
             v := dummy_gri || direction(arg.orientation.direction_ratios);
         IF arg.magnitude = 0.0 THEN
           RETURN(?);
         ELSE
          vec := dummy_gri || vector (v, 1.0);
         END_IF;
       END;
```

```
    ELSE
      v := dummy_gri || direction (arg.direction_ratios);
    END_IF;
    mag := 0.0;
    REPEAT  i := 1 TO ndim;
      mag := mag + v.direction_ratios[i]*v.direction_ratios[i];
    END_REPEAT;
    IF mag > 0.0 THEN
      mag := SQRT(mag);
      REPEAT  i := 1 TO ndim;
        v.direction_ratios[i] := v.direction_ratios[i]/mag;
      END_REPEAT;
      IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VECTOR' IN TYPEOF(arg) THEN

        vec.orientation := v;
        result := vec;
      ELSE
        result := v;
      END_IF;
    ELSE
      RETURN(?);
    END_IF;
  END_IF;
  RETURN (result);

END_FUNCTION;

FUNCTION open_shell_reversed ( a_shell : open_shell) :
                                        oriented_open_shell;
  LOCAL
    the_reverse : oriented_open_shell;
  END_LOCAL;
  IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_OPEN_SHELL' IN TYPEOF
(a_shell) ) THEN

    the_reverse := dummy_tri ||
                 connected_face_set (
                     a_shell\connected_face_set.cfs_faces) ||
                 open_shell () || oriented_open_shell(
                   a_shell\oriented_open_shell.open_shell_element,
                     (NOT (a_shell\oriented_open_shell.orientation)));
  ELSE
    the_reverse := dummy_tri ||
                connected_face_set (
                    a_shell\connected_face_set.cfs_faces) ||
                open_shell () ||  oriented_open_shell (a_shell, FALSE);
  END_IF;
  RETURN (the_reverse);

END_FUNCTION;

FUNCTION orthogonal_complement(vec : direction) : direction;
  LOCAL
    result :  direction ;
  END_LOCAL;

  IF (vec.dim <> 2) OR NOT EXISTS (vec) THEN
```

```
      RETURN(?);
    ELSE
      result := dummy_gri || direction([-vec.direction_ratios[2],
                                        vec.direction_ratios[1]]);
      RETURN(result);
    END_IF;

 END_FUNCTION;

 FUNCTION path_head_to_tail(a_path : path) : LOGICAL;
   LOCAL
     n : INTEGER;
     p : LOGICAL := TRUE;
   END_LOCAL;

     n := SIZEOF (a_path.edge_list);
     REPEAT i := 2 TO n;
       p := p AND (a_path.edge_list[i-1].edge_end :=:
                   a_path.edge_list[i].edge_start);
     END_REPEAT;

     RETURN (p);

 END_FUNCTION;

FUNCTION path_reversed (a_path : path) : oriented_path;
  LOCAL
    the_reverse : oriented_path ;
  END_LOCAL;
  IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ORIENTED_PATH' IN TYPEOF (a_path)
) THEN

    the_reverse := dummy_tri ||
       path(list_of_topology_reversed (a_path.edge_list)) ||
          oriented_path(a_path\oriented_path.path_element,
                        NOT(a_path\oriented_path.orientation)) ;
  ELSE
    the_reverse := dummy_tri ||
                   path(list_of_topology_reversed (a_path.edge_list)) ||
                       oriented_path(a_path, FALSE);
  END_IF;

  RETURN (the_reverse);

END_FUNCTION;

FUNCTION representations_mapped_into
  (rep : representation) : SET of representation;
  LOCAL
    results : SET OF representation := [];
    rm      : SET OF representation_map;
    mi      : SET OF mapped_item := [];
  END_LOCAL;

  -- Find set of representation_maps which specify the representation
  rm := bag_to_set(USEDIN(rep, 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.' +
                   'REPRESENTATION_MAP.MAPPED_REPRESENTATION'));
```

```
  -- Find the set of mapped_items that use each representation_map
  REPEAT i := 1 TO HIINDEX(rm);
    mi := mi + rm[i].map_usage;
  END_REPEAT;
  -- Find the set of representations that use each mapped_item
  REPEAT j := 1 TO HIINDEX(mi);
    results := results + USEDIN(mi[j],'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.'
+

                                      'REPRESENTATION.ITEMS');
  END_REPEAT;
  RETURN(results);

END_FUNCTION;

 FUNCTION scalar_times_vector (scalar : REAL; vec : vector_or_direction)
                                      : vector;
   LOCAL
     v      : direction;
     mag    : REAL;
     result : vector;
   END_LOCAL;

   IF NOT EXISTS (scalar) OR NOT EXISTS (vec) THEN
     RETURN (?) ;
    ELSE
     IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VECTOR' IN TYPEOF (vec) THEN

       v   := dummy_gri || direction(vec.orientation.direction_ratios);
       mag := scalar * vec.magnitude;
     ELSE
       v   := dummy_gri || direction(vec.direction_ratios);
       mag := scalar;
     END_IF;
     IF (mag < 0.0 ) THEN
       REPEAT i := 1 TO SIZEOF(v.direction_ratios);
         v.direction_ratios[i] := -v.direction_ratios[i];
       END_REPEAT;
       mag := -mag;
     END_IF;
     result := dummy_gri || vector(normalise(v), mag);
   END_IF;
   RETURN (result);

 END_FUNCTION;

 FUNCTION second_proj_axis(z_axis, x_axis, arg: direction) : direction;
   LOCAL
     y_axis : vector;
     v      : direction;
     temp   : vector;
   END_LOCAL;

   IF NOT EXISTS(arg) THEN
     v := dummy_gri || direction([0.0,1.0,0.0]);
   ELSE
     v := arg;
```

```
    END_IF;

    temp   := scalar_times_vector(dot_product(v, z_axis), z_axis);
    y_axis := vector_difference(v, temp);
    temp   := scalar_times_vector(dot_product(v, x_axis), x_axis);
    y_axis := vector_difference(y_axis, temp);
    y_axis := normalise(y_axis);
    RETURN(y_axis.orientation);

  END_FUNCTION;

  FUNCTION set_of_topology_reversed (a_set : set_of_reversible_topology_item)
                                          : set_of_reversible_topology_item;
    LOCAL
      the_reverse : set_of_reversible_topology_item;
    END_LOCAL;

    the_reverse := [];
    REPEAT i := 1 TO SIZEOF (a_set);
      the_reverse := the_reverse + topology_reversed (a_set [i]);
    END_REPEAT;

    RETURN (the_reverse);

  END_FUNCTION;

  FUNCTION shell_reversed (a_shell : shell) : shell;
    IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OPEN_SHELL' IN TYPEOF (a_shell)
) THEN
      RETURN (open_shell_reversed (a_shell));
    ELSE
      IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CLOSED_SHELL' IN TYPEOF
(a_shell) ) THEN

        RETURN (closed_shell_reversed (a_shell));
      ELSE
        RETURN (?);
      END_IF;
    END_IF;

  END_FUNCTION;

  FUNCTION surface_weights_positive(b: rational_b_spline_surface) : BOOLEAN;
    LOCAL
      result        : BOOLEAN := TRUE;
    END_LOCAL;

    REPEAT i := 0 TO b.u_upper;
      REPEAT j := 0 TO b.v_upper;
        IF (b.weights[i][j] <= 0.0)  THEN
          result := FALSE;
          RETURN(result);
        END_IF;
      END_REPEAT;
    END_REPEAT;
    RETURN(result);
```

```
 END_FUNCTION;

 FUNCTION topology_reversed (an_item : reversible_topology)
                                     : reversible_topology;

   IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.EDGE' IN TYPEOF (an_item)) THEN
     RETURN (edge_reversed (an_item));
   END_IF;

   IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PATH' IN TYPEOF (an_item)) THEN
     RETURN (path_reversed (an_item));
   END_IF;

   IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE_BOUND' IN TYPEOF (an_item))
THEN
     RETURN (face_bound_reversed (an_item));
   END_IF;

   IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FACE' IN TYPEOF (an_item)) THEN
     RETURN (face_reversed (an_item));
   END_IF;

   IF ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHELL' IN TYPEOF (an_item)) THEN

     RETURN (shell_reversed (an_item));
   END_IF;

   IF ('SET' IN TYPEOF (an_item)) THEN
     RETURN (set_of_topology_reversed (an_item));
   END_IF;

   IF ('LIST' IN TYPEOF (an_item)) THEN
     RETURN (list_of_topology_reversed (an_item));
   END_IF;

   RETURN (?);

 END_FUNCTION;

FUNCTION type_check_function
 (the_type : GENERIC; sub_names : SET OF STRING; criterion : INTEGER) : LOGICAL;
IF (( NOT EXISTS ( the_type ) ) OR (NOT ({0<= criterion <=3})) OR
(SIZEOF ( sub_names ) = 0 ) ) THEN RETURN (UNKNOWN);
ELSE
  CASE criterion OF
       0: RETURN (SIZEOF ( sub_names * TYPEOF (the_type) ) > 0);
       1: RETURN (SIZEOF ( sub_names * TYPEOF (the_type) ) = 0);
       2: RETURN (SIZEOF ( sub_names * TYPEOF (the_type) ) = 1);
       3: RETURN (SIZEOF ( sub_names * TYPEOF (the_type) ) <= 1);
  END_CASE;
END_IF;

END_FUNCTION;

FUNCTION using_items (item : founded_item_select;
                      checked_items: SET OF founded_item_select)
                    : SET OF founded_item_select;
```

```
  LOCAL
    new_check_items    : SET OF founded_item_select;
    result_items       : SET OF founded_item_select;
    next_items         : SET OF founded_item_select;
  END_LOCAL;
  result_items := [];
  new_check_items := checked_items + item;
  -- Find the set of representation_items or founded_items
  -- in which item is used directly.
  next_items := QUERY(z <* bag_to_set( USEDIN(item , '')) |
    ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REPRESENTATION_ITEM' IN TYPEOF(z))
OR
    ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FOUNDED_ITEM'        IN
TYPEOF(z)));

  -- If the set of next_items is not empty;
  IF SIZEOF(next_items) > 0 THEN
    -- For each element in the set, find the using_items recursively
    REPEAT i := 1 TO HIINDEX(next_items);
      -- Check for loop in data model, i.e. one of the next_items
      -- occurred earlier in the set of check_items;
      IF NOT(next_items[i] IN new_check_items) THEN
        result_items := result_items + next_items[i] +
                        using_items(next_items[i],new_check_items);
      END_IF;
    END_REPEAT;
  END_IF;
  -- return the set of representation_items or founded_items
  -- in which the input item is used directly and indirectly.
  RETURN (result_items);

END_FUNCTION;

FUNCTION using_representations (item : founded_item_select)
  : SET OF representation;
  LOCAL
    results            : SET OF representation;
    result_bag         : BAG OF representation;
    intermediate_items : SET OF founded_item_select;
  END_LOCAL;
  -- Find the representations in which the item is used and add to the
  -- results set.
  results := [];
  result_bag :=
USEDIN(item,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REPRESENTATION.ITEMS');
  IF SIZEOF(result_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(result_bag);
      results := results + result_bag[i];
    END_REPEAT;
  END_IF;
  -- Find all representation_items or founded_items
  -- by which item is referenced directly or indirectly.
  intermediate_items := using_items(item,[]);
  -- If the set of intermediate items is not empty;
  IF SIZEOF(intermediate_items) > 0 THEN
    -- For each element in the set, add the
    -- representations of that element.
```

```
    REPEAT i := 1 TO HIINDEX(intermediate_items);
      result_bag := USEDIN(intermediate_items[i],

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.REPRESENTATION.ITEMS');

      IF SIZEOF(result_bag) > 0 THEN
        REPEAT j := 1 TO HIINDEX(result_bag);
          results := results + result_bag[j];
        END_REPEAT;
      END_IF;
    END_REPEAT;
  END_IF;
  -- Return the set of representation in which the input item is

  -- used directly and indirectly (through intervening
  -- representation_items or founded items).
  RETURN (results);

END_FUNCTION;

FUNCTION valid_calendar_date
 (date : calendar_date) : LOGICAL;
  CASE date.month_component OF
    1  : RETURN({ 1 <= date.day_component <= 31 });
    2  : BEGIN
           IF (leap_year(date.year_component)) THEN
             RETURN({ 1 <= date.day_component <= 29 });
           ELSE
             RETURN({ 1 <= date.day_component <= 28 });
           END_IF;
         END;
    3  : RETURN({ 1 <= date.day_component <= 31 });
    4  : RETURN({ 1 <= date.day_component <= 30 });
    5  : RETURN({ 1 <= date.day_component <= 31 });
    6  : RETURN({ 1 <= date.day_component <= 30 });
    7  : RETURN({ 1 <= date.day_component <= 31 });
    8  : RETURN({ 1 <= date.day_component <= 31 });
    9  : RETURN({ 1 <= date.day_component <= 30 });
    10 : RETURN({ 1 <= date.day_component <= 31 });
    11 : RETURN({ 1 <= date.day_component <= 30 });
    12 : RETURN({ 1 <= date.day_component <= 31 });
  END_CASE;
  RETURN (FALSE);

END_FUNCTION;

FUNCTION valid_datum_target_parameters
 (pdf : placed_datum_target_feature) : BOOLEAN;
LOCAL

rep_set : SET OF representation := [] ;

parameter_representations: SET OF representation;
END_LOCAL;


REPEAT i := 1 TO HIINDEX(pdf.representation_associations);
```

```
rep_set := rep_set + pdf.representation_associations[i].used_representation;
END_REPEAT;

parameter_representations := QUERY(rep <* rep_set |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN
TYPEOF(rep)));


IF (SIZEOF( QUERY( srwp <* parameter_representations |
          (SIZEOF( QUERY( i <* srwp.items |
          (i.name='orientation') AND
          ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLACEMENT' IN TYPEOF(i)))) =
1))) <> 1) THEN
   RETURN(FALSE);
END_IF;

CASE pdf\shape_aspect.description OF
'point': RETURN(SIZEOF(QUERY( srwp <* parameter_representations |
              (SIZEOF(srwp.items) = 1))) = 1);

'circle': RETURN((SIZEOF( QUERY( srwp <* parameter_representations |
              (SIZEOF(srwp.items) = 2))) = 1) AND
             (SIZEOF( QUERY( srwp <* parameter_representations |
              (SIZEOF( QUERY( i <* srwp.items |
                (i.name='target diameter') AND

(SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM',

'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LENGTH_MEASURE_WITH_UNIT']*TYPEOF(i)
                ) = 2) )) = 1))) = 1));

'line': RETURN(SIZEOF( QUERY( srwp <* parameter_representations |
              (SIZEOF( QUERY( i <* srwp.items |
                (i.name='target length') AND

(SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM',

      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LENGTH_MEASURE_WITH_UNIT']*TYPEOF
(i)
                ) = 2) )) = 1))) = 1);

'rectangle': RETURN((SIZEOF( QUERY( srwp <* parameter_representations |
              (SIZEOF(srwp.items)= 3))) = 1) AND
             (SIZEOF( QUERY( srwp <* parameter_representations |
              (SIZEOF( QUERY( i <* srwp.items |
                (i.name='target length') AND

(SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM',

      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LENGTH_MEASURE_WITH_UNIT']*TYPEOF
(i)
                ) = 2))) = 1))) = 1) AND
              (SIZEOF( QUERY( srwp <* parameter_representations |
               (SIZEOF( QUERY( i <* srwp.items |
                 (i.name='target width') AND
```

```
(SIZEOF(['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM',

      'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LENGTH_MEASURE_WITH_UNIT']*TYPEOF
(i)

                  ) = 2))) = 1) )) = 1));
OTHERWISE : RETURN(FALSE);
END_CASE;

   END_FUNCTION;

FUNCTION valid_geometrically_bounded_wf_curve
(
crv : curve ) : BOOLEAN ;
 IF SIZEOF ([ 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_CURVE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ELLIPSE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CIRCLE' ] * TYPEOF (crv)) = 1 THEN
 RETURN (TRUE);
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TRIMMED_CURVE' IN TYPEOF (crv) THEN
 IF SIZEOF ([ 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PARABOLA',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.HYPERBOLA' ] * TYPEOF
(crv\trimmed_curve.basis_curve)) = 1 THEN
 RETURN (TRUE);
 ELSE
 RETURN (valid_geometrically_bounded_wf_curve(crv\trimmed_curve.basis_curve));
 END_IF ;
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_CURVE_3D' IN TYPEOF (crv)
THEN
 RETURN (valid_geometrically_bounded_wf_curve(crv\offset_curve_3d.basis_curve));
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_REPLICA' IN TYPEOF (crv) THEN
 RETURN (valid_geometrically_bounded_wf_curve(crv\curve_replica.parent_curve));
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOSITE_CURVE' IN TYPEOF (crv)
THEN

 RETURN ( SIZEOF (
QUERY ( ccs <* crv\composite_curve.segments| NOT
valid_geometrically_bounded_wf_curve(ccs.parent_curve) )) = 0);
 END_IF ;
 END_IF ;
 END_IF ;
 END_IF ;
 END_IF ;
 RETURN (FALSE);

 END_FUNCTION;

FUNCTION valid_geometrically_bounded_wf_point
(
pnt : point ) : BOOLEAN ;
```

```
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT' IN TYPEOF (pnt)
THEN
 RETURN (TRUE);
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_ON_CURVE' IN TYPEOF (pnt)
THEN
 RETURN (valid_geometrically_bounded_wf_curve(pnt\point_on_curve.basis_curve));
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_REPLICA' IN TYPEOF (pnt) THEN

 RETURN (valid_geometrically_bounded_wf_point(pnt\point_replica.parent_pt));
 END_IF ;
 END_IF ;
 END_IF ;
 RETURN (FALSE);

 END_FUNCTION;

FUNCTION valid_measure_value
  (m : measure_value) : BOOLEAN;
  IF ('REAL' IN TYPEOF (m)) THEN
  RETURN (m > 0.0);
  ELSE
    IF ('INTEGER' IN TYPEOF (m)) THEN
    RETURN (m > 0);
    ELSE
      RETURN (TRUE);
    END_IF;
  END_IF;

END_FUNCTION;

FUNCTION valid_time
 (time : local_time) : BOOLEAN;
  IF EXISTS(time.second_component) THEN
    RETURN (EXISTS(time.minute_component));
  ELSE
    RETURN (TRUE);
  END_IF;

END_FUNCTION;

FUNCTION valid_units
 (m : measure_with_unit) : BOOLEAN;
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LENGTH_MEASURE' IN TYPEOF (
m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MASS_MEASURE' IN TYPEOF (
m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
```

```
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TIME_MEASURE' IN TYPEOF (
m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ELECTRIC_CURRENT_MEASURE'
  IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.THERMODYNAMIC_TEMPERATURE_MEASURE'
  IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CELSIUS_TEMPERATURE_MEASURE'
  IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AMOUNT_OF_SUBSTANCE_MEASURE'
  IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LUMINOUS_INTENSITY_MEASURE'
  IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PLANE_ANGLE_MEASURE' IN TYPEOF (
m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SOLID_ANGLE_MEASURE' IN TYPEOF (
m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
```

```
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.AREA_MEASURE' IN TYPEOF (
m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VOLUME_MEASURE' IN TYPEOF (
m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 3.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.RATIO_MEASURE' IN TYPEOF (
m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POSITIVE_LENGTH_MEASURE'
  IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POSITIVE_PLANE_ANGLE_MEASURE'

  IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
       dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  RETURN (TRUE);

END_FUNCTION;

FUNCTION valid_wireframe_edge_curve
(
crv : curve ) : BOOLEAN ;
 IF SIZEOF ([ 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONIC',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.B_SPLINE_CURVE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POLYLINE' ] * TYPEOF (crv)) = 1 THEN
 RETURN (TRUE);
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_REPLICA' IN TYPEOF (crv) THEN
 RETURN (valid_wireframe_edge_curve(crv\curve_replica.parent_curve));
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.OFFSET_CURVE_3D' IN TYPEOF (crv)
THEN

 RETURN (valid_wireframe_edge_curve(crv\offset_curve_3d.basis_curve));
 END_IF ;
```

```
 END_IF ;
 END_IF ;
 RETURN (FALSE);

 END_FUNCTION;

FUNCTION valid_wireframe_vertex_point
(
pnt : point ) : BOOLEAN ;
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CARTESIAN_POINT' IN TYPEOF (pnt)
THEN
 RETURN (TRUE);
 ELSE
 IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POINT_REPLICA' IN TYPEOF (pnt) THEN

 RETURN (valid_wireframe_vertex_point(pnt\point_replica.parent_pt));
 END_IF ;
 END_IF ;
 RETURN (FALSE);

 END_FUNCTION;

FUNCTION value_range_wr1
 (agg : compound_item_definition) : BOOLEAN;
BEGIN
IF (SIZEOF(agg) = 2) AND ((SIZEOF(QUERY (i1 <* agg | (
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN TYPEOF
(i1)))) = 2) OR
(SIZEOF(QUERY (i2 <* agg | (
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VALUE_REPRESENTATION_ITEM' IN TYPEOF

(i2)))) = 2)) THEN
RETURN(TRUE);
ELSE
RETURN(FALSE);
END_IF;
END;

      END_FUNCTION;

FUNCTION value_range_wr2
 (agg : compound_item_definition) : BOOLEAN;
BEGIN
IF (SIZEOF(QUERY (i <* agg | (i.name = 'upper limit'))) = 1)
AND (SIZEOF(QUERY (i <* agg | (i.name = 'lower limit'))) = 1)
THEN
RETURN(TRUE);
ELSE
RETURN(FALSE);
END_IF;
END;

      END_FUNCTION;

FUNCTION value_range_wr3
 (agg : compound_item_definition) : BOOLEAN;
BEGIN
```

```
IF (SIZEOF(QUERY(i1 <* agg |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN TYPEOF
(i1)) AND
(SIZEOF (QUERY (i2 <* agg |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRESENTATION_ITEM' IN TYPEOF
(i2)) AND

(i1 :<>: i2) AND (i1\measure_with_unit.unit_component :=:
i2\measure_with_unit.unit_component))) = 1))) = 2)
THEN
RETURN (TRUE);
ELSE
RETURN (FALSE);
END_IF;
END;

      END_FUNCTION;

 FUNCTION vector_difference(arg1, arg2 : vector_or_direction) : vector;
   LOCAL
     result          : vector;
     res, vec1, vec2 : direction;
     mag, mag1, mag2 : REAL;
     ndim            : INTEGER;
   END_LOCAL;

   IF ((NOT EXISTS (arg1)) OR (NOT EXISTS (arg2))) OR (arg1.dim <> arg2.dim)
       THEN
     RETURN (?) ;
    ELSE
     BEGIN
       IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VECTOR' IN TYPEOF(arg1) THEN
         mag1 := arg1.magnitude;
         vec1 := arg1.orientation;
       ELSE
         mag1 := 1.0;
         vec1 := arg1;
       END_IF;
       IF 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VECTOR' IN TYPEOF(arg2) THEN

         mag2 := arg2.magnitude;

         vec2 := arg2.orientation;
       ELSE
         mag2 := 1.0;
         vec2 := arg2;
       END_IF;
       vec1 := normalise (vec1);
       vec2 := normalise (vec2);
       ndim := SIZEOF(vec1.direction_ratios);
       mag := 0.0;
       res := dummy_gri || direction(vec1.direction_ratios);
       REPEAT i := 1 TO ndim;
         res.direction_ratios[i] := mag1*vec1.direction_ratios[i] +
                                    mag2*vec2.direction_ratios[i];
         mag := mag + (res.direction_ratios[i]*res.direction_ratios[i]);
       END_REPEAT;
```

```
       IF (mag > 0.0 ) THEN
       result := dummy_gri || vector( res, SQRT(mag));
       ELSE
         result := dummy_gri || vector( vec1,  0.0);
       END_IF;
     END;
   END_IF;
   RETURN (result);

 END_FUNCTION;

RULE compatible_dimension FOR
  (cartesian_point,
  direction,
  representation_context,
  geometric_representation_context);
WHERE

  -- ensure that the count of coordinates of each cartesian_point
  -- matches the coordinate_space_dimension of each geometric_context in
  -- which it is geometrically founded
  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;

  -- ensure that the count of direction_ratios of each direction
  -- matches the coordinate_space_dimension of each geometric_context in
  -- which it is geometrically founded
  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;

RULE consistent_uncertainty
      FOR (global_uncertainty_assigned_context,
           qualified_representation_item,
           uncertainty_assigned_representation);
WHERE
WR1: SIZEOF ( QUERY ( guac <* global_uncertainty_assigned_context |
    SIZEOF ( QUERY ( u1 <* guac.uncertainty |
      SIZEOF ( QUERY ( u2 <* guac.uncertainty | u2.name =  u1.name ) ) >1 ) ) >0 )
) =  0 ;
WR2: SIZEOF ( QUERY ( uar <* uncertainty_assigned_representation |
      SIZEOF ( QUERY ( u1<* uar.uncertainty |
      SIZEOF ( QUERY ( u2 <* uar.uncertainty | u2.name =  u1.name ) ) >1 ) ) >0 ) )
=  0;
WR3: SIZEOF ( QUERY ( qri <* qualified_representation_item |
     SIZEOF ( QUERY ( u1 <* qri.qualifiers |
     ('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF' + '.UNCERTAINTY_QUALIFIER' IN
TYPEOF ( u1 ) ) AND
      ( SIZEOF ( QUERY ( u2 <* qri.qualifiers |
     ( 'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF' + '.UNCERTAINTY_QUALIFIER' IN
TYPEOF ( u2 ) ) AND

      ( u2\uncertainty_qualifier.measure_name =
u1\uncertainty_qualifier.measure_name ) )
```

```
     ) >1 ) ) ) >0 ) ) =  0;
END_RULE;

RULE restrict_language_assignment_per_attribute FOR
(attribute_language_assignment);
WHERE
WR1: SIZEOF ( QUERY ( ala1 <* attribute_language_assignment |
      SIZEOF(QUERY( it <* ala1.items |
         SIZEOF ( QUERY ( ala2 <* USEDIN ( it
,'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ATTRIBUTE_LANGUAGE_ASSIGNMENT.ITEMS' )
|

            ( ala1\attribute_classification_assignment.attribute_name =
ala2\attribute_classification_assignment.attribute_name ) AND
            ( ala1\attribute_classification_assignment.assigned_class :=:
ala2\attribute_classification_assignment.assigned_class )
             )) >1
         )) >0
      )) =0;
END_RULE;

RULE styled_curve FOR
(styled_item);
WHERE
  WR1: SIZEOF( QUERY( si <* styled_item |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE' IN TYPEOF (si.item)) AND
(SIZEOF (QUERY (psa <* si.styles | (SIZEOF (QUERY (cs <* psa.styles |
('CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CURVE_STYLE' IN TYPEOF (cs)) )) > 0)
)) <> 1) )) = 0;

END_RULE;

RULE subtype_exclusiveness_geometric_tolerance FOR
(geometric_tolerance);
WHERE
  WR1: SIZEOF(QUERY (gt <* geometric_tolerance | NOT (type_check_function(gt,
['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANGULARITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CIRCULAR_RUNOUT_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COAXIALITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONCENTRICITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CYLINDRICITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FLATNESS_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE_PROFILE_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PARALLELISM_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PERPENDICULARITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POSITION_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ROUNDNESS_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STRAIGHTNESS_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_PROFILE_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMMETRY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TOTAL_RUNOUT_TOLERANCE'] , 3)))) = 0;

END_RULE;

RULE subtype_exclusiveness_representation_item FOR
(representation_item);
WHERE
```

```
  WR1: SIZEOF(QUERY (cri <* representation_item | NOT
(type_check_function(cri,['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.MEASURE_REPRE
SENTATION_ITEM','CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.VALUE_REPRESENTATION_IT
EM','CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COMPOUND_REPRESENTATION_ITEM'] ,
3)))) = 0;

END_RULE;

RULE subtype_mandatory_geometric_tolerance FOR
(geometric_tolerance);
WHERE
  WR1: SIZEOF(QUERY (gt <* geometric_tolerance | NOT (type_check_function(gt,
['CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ANGULARITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CIRCULAR_RUNOUT_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.COAXIALITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CONCENTRICITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.CYLINDRICITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.FLATNESS_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.LINE_PROFILE_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PARALLELISM_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.PERPENDICULARITY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.POSITION_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.ROUNDNESS_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.STRAIGHTNESS_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SURFACE_PROFILE_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.SYMMETRY_TOLERANCE',
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TOTAL_RUNOUT_TOLERANCE'] , 0)))) = 0;

END_RULE;

RULE text_font_usage FOR
(externally_defined_text_font,pre_defined_text_font);
WHERE
  WR1: SIZEOF (QUERY (pdtf <* pre_defined_text_font | SIZEOF (USEDIN (pdtf,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL.FONT')) = 0 )) = 0;
  WR2: SIZEOF (QUERY (edtf <* externally_defined_text_font | SIZEOF (USEDIN (edtf,
'CONFIGURATION_CONTROL_3D_DESIGN_ED2_MIM_LF.TEXT_LITERAL.FONT')) = 0 )) = 0;

END_RULE;

(*** 2003=>1994 conversion: insert a rule to control if instances are legal
according to USE/REF clauses ***)

RULE validate_dependently_instantiable_entity_data_types FOR

(application_context_element,area_in_set,attribute_value_role,bounded_curve,bounded
_surface,camera_image,camera_model_d3,cartesian_transformation_operator,classificat
ion_role,connected_edge_set,contract_type,document_type,document_usage_constraint,e
vent_occurrence_role,local_time,loop,one_direction_repeat_factor,oriented_face,orie
nted_open_shell,oriented_path,planar_box,pre_defined_item,presentation_set,product_
definition_context_role,product_definition_effectivity,runout_zone_orientation,soli
d_model,swept_face_solid,tolerance_zone_form,two_direction_repeat_factor,vertex,vie
w_volume --<list this first and all subsequent relevant referencedentity data types
here>
      );
LOCAL
  number_of_input_instances : INTEGER;
```

```
  previous_in_chain : LIST OF GENERIC := [];
  set_of_input_types : SET OF STRING := [];
  all_instances : SET OF GENERIC := [];
END_LOCAL;

  all_instances := application_context_element + area_in_set + attribute_value_role
+ bounded_curve + bounded_surface + camera_image + camera_model_d3 +
cartesian_transformation_operator + classification_role + connected_edge_set +
contract_type + document_type + document_usage_constraint + event_occurrence_role +
local_time + loop + one_direction_repeat_factor + oriented_face +
oriented_open_shell + oriented_path + planar_box + pre_defined_item +
presentation_set + product_definition_context_role + product_definition_effectivity
+ runout_zone_orientation + solid_model + swept_face_solid + tolerance_zone_form +
two_direction_repeat_factor + vertex + view_volume;--<make a union of all implicit
populations of the FOR-clause>
number_of_input_instances := SIZEOF(all_instances);
(* Collect all type strings of all FOR instances into one set. *)
REPEAT i:=1 TO number_of_input_instances;
  set_of_input_types := set_of_input_types + TYPEOF(all_instances[i]);
END_REPEAT;

WHERE
  WR1: dependently_instantiated(all_instances, set_of_input_types,
                                previous_in_chain);
END_RULE;

FUNCTION dependently_instantiated(
            set_of_input_instances : SET OF GENERIC:igen;
            set_of_input_types     : SET OF STRING;
            previous_in_chain      : LIST OF GENERIC:cgen): BOOLEAN;
(*'dependently_instantiated' To test whether all instances in the
  input set_of_input_instances are referenced by independently
  instantiable instances. If so, this function returns true.
  Set_of_input_types includes the type strings for all input instances.
  The instances in previous_in_chain are used to detect cyclic
  references during recursive calls to this function. The parameter
  lists already tested instances in a chain of references.
*)
LOCAL
  number_of_input_instances     : INTEGER;
  number_of_referring_instances : INTEGER;
  bag_of_referring_instances    : BAG OF GENERIC:igen := [];
  dependently_instantiated_flag : BOOLEAN;
  previous_in_chain_plus        : LIST OF GENERIC:cgen := [];
  result                        : BOOLEAN := true;
  set_of_types                  : SET OF STRING := [];
END_LOCAL;

IF EXISTS(set_of_input_instances) THEN
  number_of_input_instances := SIZEOF(set_of_input_instances);
  (* Add the declared type of bag_of_referring_instances to the set of
     types of the REFERENCEd instances for the subset comparison later.
   *)
  set_of_input_types := set_of_input_types + 'GENERIC';
  REPEAT i:=1 TO number_of_input_instances;
    (* Determine all references to the current input instance. *)
    bag_of_referring_instances := USEDIN (set_of_input_instances[i] , '');
```

```
    IF EXISTS(bag_of_referring_instances) THEN
      number_of_referring_instances := SIZEOF(bag_of_referring_instances);
      dependently_instantiated_flag := false;
      REPEAT j:=1 TO number_of_referring_instances;
        (* Determine the type strings of the current referencing instance.
         *)
        set_of_types := TYPEOF(bag_of_referring_instances[j]);
        (* If the referencing instance is of one of the types of the
           only dependently instantiable select items, the current input
           instance may still be invalidly instantiated.
           Otherwise it is OK, and the next input instance is tested.
         *)
        IF set_of_types <= set_of_input_types THEN -- subset operator
          (* The referring instance is of one of the restricted types.
             However, it may itself be referred to by a valid instance;
             then also the current instance would be valid.
             Thus, call this function recursively with the referring
             instance as input.
             To avoid an infinite loop in case a set of instances
             reference each other in a closed loop, test first whether
             the current referencing instance is in the list of
             previously processed chain members.
           *)
          IF NOT (bag_of_referring_instances[j] IN previous_in_chain) THEN
            previous_in_chain_plus := previous_in_chain +
            set_of_input_instances[i];
            IF dependently_instantiated([bag_of_referring_instances[j]],
              set_of_input_types,
              previous_in_chain_plus) THEN
              dependently_instantiated_flag := true;
              ESCAPE; -- dependently instantiated; next input instance
            ELSE
              (* Not dependently instantiated: go to next referring
              instance. *)
              SKIP;
            END_IF;
          END_IF;
        ELSE
          dependently_instantiated_flag := true;
          ESCAPE; -- dependently instantiated; take next input instance
        END_IF;
      END_REPEAT;
      IF NOT dependently_instantiated_flag THEN
        RETURN(false);
      END_IF;
    ELSE
      RETURN(false); -- not referenced at all => invalidly instantiated
    END_IF;
  END_REPEAT;
ELSE
  RETURN(false); -- no input
END_IF;

RETURN(true);
END_FUNCTION; -- end dependently_instantiated
```

```
END_SCHEMA;
```