INTERNATIONAL STANDARD

ISO 13399-1

First edition
2006-02-15

Cutting tool data representation and exchange —

Part 1: Overview, fundamental principles and general information model

Représentation et échange des données relatives aux outils coupants —

Partie 1: Vue d'ensemble, principes fondamentaux et modèle général d'informations

Reference number
ISO 13399-1:2006(E)

PDF disclaimer

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

Adobe is a trademark of Adobe Systems Incorporated.

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

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

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

ISO 13399-1 was prepared by Technical Committee ISO/TC 29, *Small tools*.

ISO 13399 consists of the following parts, under the general title *Cutting tool data representation and exchange*:

- *Part 1: Overview, fundamental principles and general information model*
- *Part 2: Reference dictionary for the cutting items* [Technical Specification]
- *Part 100: Definitions, principles and methods for reference dictionaries* [Technical Specification]

The following parts are under preparation:

- *Part 3: Reference dictionary for the tool items* [Technical Specification]
- *Part 4: Reference dictionary of terms for adaptive items* [Technical Specification]
- *Part 50: Reference dictionary for reference systems and common concepts* [Technical Specification]
- *Part 60: Reference dictionary for connection systems* [Technical Specification]

Assembly items is to form the subject of a future Part 5.

Introduction

ISO 13399 provides the means to achieve an electronic representation of cutting tool data by providing the information structure needed to describe various data about cutting tools and cutting tool assemblies. It is intended to facilitate the use, manipulation and exchange of cutting tool data within and between manufacturing, distribution, and usage.

A cutting tool with defined cutting edges is used on a machine tool to remove workpiece material through a shearing action at the cutting edge(s) of the tool. Cutting tool data are characteristics of the cutting tool and its use that must be known and evaluated in order to make manufacturing decisions and to perform manufacturing operations.

ISO 13399 includes the data representation of everything between the workpiece and the machine tool. Information about inserts, solid tools (e.g. solid drill and solid endmill), assembled tools (e.g. boring bars, indexable drills and indexable milling cutters), adaptors (e.g. milling arbor and chucks), components (e.g. shims, screws and clamps) or any combination of the above can be exchanged.

The cutting tool data described include, but are not limited to, geometrical and dimensional data, identification and designation data, miscellaneous and spare part data, cutting material data, and component connectivity.

The use of the tool information model established by ISO 13399 will provide increased productivity for the user in the same way as do the tools. The effective management of tool information will improve the management of the tools themselves. Use of the tool information model will enable the identification of the “right” tool in every operation — from tool purchase, through planning, set-up in machine-tools, maintenance and reuse of the tools — with short lead times and with high reliability and product quality. Tool users will benefit from improved support from the tool vendors who will be able to provide a standard information product to accompany the tool products. Computer interfaces for information exchange will be more efficient.

The representation of product data is defined by ISO 10303, which specifies the computer-interpretable representation of product information and the exchange of product data. The objective is to provide a neutral mechanism capable of describing products throughout their life cycle. This mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases, and as a basis for archiving (ISO 10303-1).

This part of ISO 13399 uses the following resources according to ISO 10303:

a) the EXPRESS language defined in ISO 10303-11;

b) the file format for data exchange defined in ISO 10303-21 and ISO 10303-28;

c) the integrated resources given in parts ISO 10303-40 to ISO 10303-56.

ISO 13399 is intended for use by manufacturers, tool vendors or producers, and developers of manufacturing software, among others. It provides a common structure for exchanging data about cutting tools (see Figure 1), and is intended to allow or improve several capabilities, including

— provision of a common set of definitions for use in describing cutting tools and cutting tool assemblies,

— the integration and sharing of cutting tool and assembly data between software applications,

— direct import of vendor cutting tool data into customer databases or applications, and

— a reduction in the level of effort required for manufacturers to maintain accurate and current cutting tool information from multiple sources and for multiple applications.

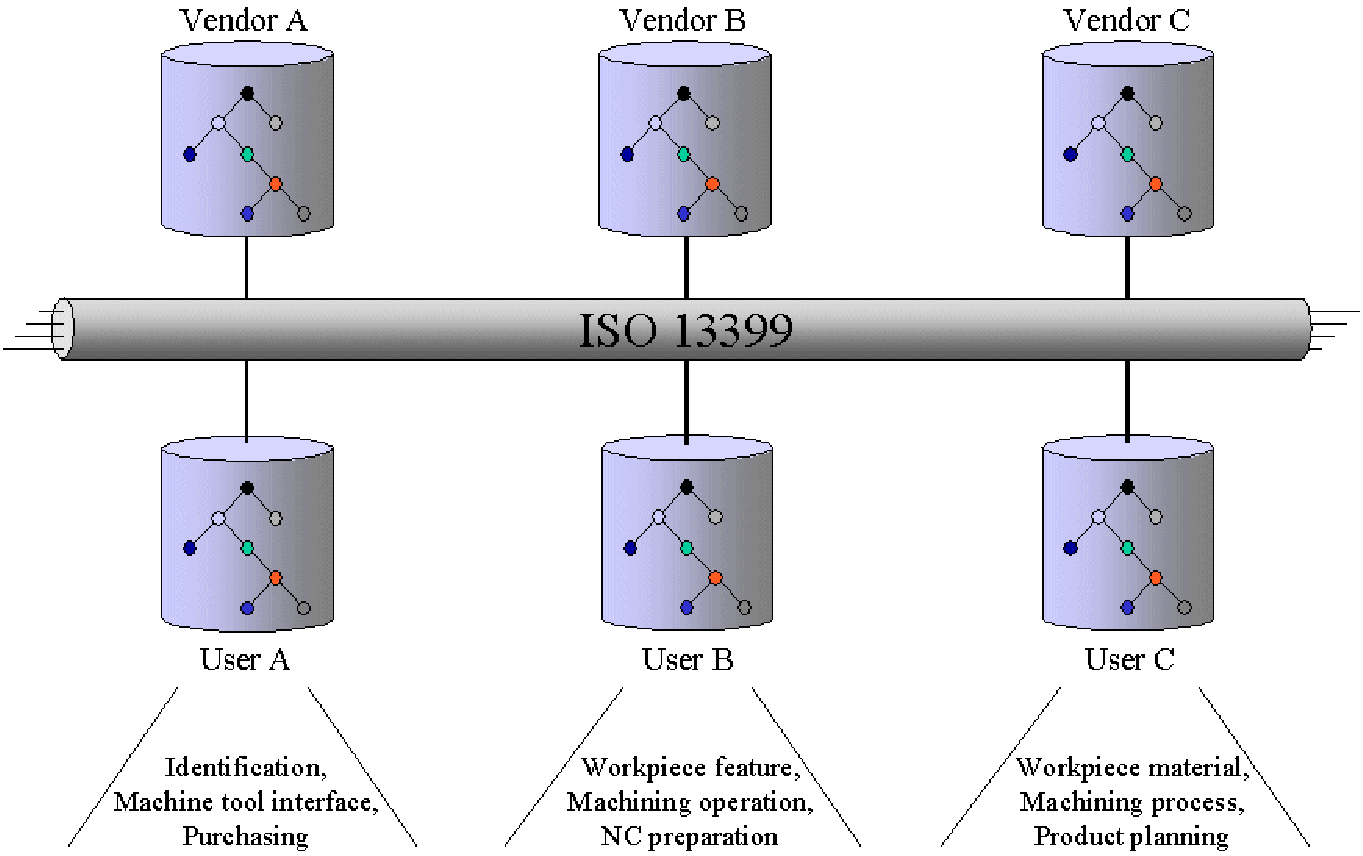

Figure 1 — Role of ISO 13399 in communication of cutting tool data and potential user viewpoints

While the use of ISO 13399 can facilitate the creation of an electronic catalogue, it does not set forth the rules or procedures for creating electronic catalogues. These are defined in ISO 13584.

The responsibility for providing different viewpoints lies outside the scope of ISO 13399. Viewpoints might be the material to be cut, the shape to be produced, the application for which the tool will be used or any other valid viewpoint.

NOTE Use of the modelled information for different purposes can be regarded as different viewpoints of the information. Although ISO 13399 does not provide the viewpoints, the information may be viewed from several alternative viewpoints (see Figure 1).

Different companies use different business models to determine their need for the communication of information about their products. For example, one cutting tool manufacturer could regrind its customers' tools while another could allow its customers to do the regrinding and provide the information to enable them to do so. Therefore, the two cutting tool manufacturers could have a different set of cutting tool properties to communicate using the information model and dictionaries provided by ISO 13399.

ISO 13399 defines only that information which could be communicated; it not specify what information must be communicated.

Cutting tool data representation and exchange —

Part 1: **Overview, fundamental principles and general information model**

1 Scope

This part of ISO 13399 covers the main categories of cutting tool data and the relationships between them. It provides a general information model of data representation and information exchange for these categories, as well as an overview of the principles of product data exchange used in ISO 13399 as a whole, a description of the other parts of ISO 13399 and a method for transferring cutting tool data.

The following is not covered by this part of ISO 13399:

- general data transfer and representations other than cutting tool data;
- data describing rules, guidelines and expert knowledge used to design and manufacture machined parts and cutting tools;
- data describing why a particular design or manufacturing decision was made.

Cutting tool data and exchange for specific items (cutting, tool, adaptive and assembly items) and the classification and definition of terms used in the description of cutting tools are defined in the other parts of ISO 13399, respectively, and referenced by them.

ISO 13399 establishes a means of communicating industrial data between different computer systems that is independent of any proprietary system.

2 Normative references

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

ISO 639-2, *Codes for the representation of names of languages — Part 2: Alpha-3 code*

ISO 3002-1, *Basic quantities in cutting and grinding — Part 1: Geometry of the active part of cutting tools — General terms, reference systems, tool and working angles, chip breakers*

ISO 3002-3, *Basic quantities in cutting and grinding — Part 3: Geometric and kinematic quantities in cutting*

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

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

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

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

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

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

ISO 10303-46, *Industrial automation systems and integration — Product data representation and exchange — Part 46: Integrated generic resources: Visual presentation*

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

ISO/DIS 10303-56, *Industrial automation systems and integration — Product data representation and exchange — Part 56: Integrated generic resource: State*

ISO 10303-214, *Industrial automation systems and integration — Product data representation and exchange — Part 214: Application protocol: Core data for automotive mechanical design processes*

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

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

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 13399-100 and the following apply.

3.1
cutting tool
device or assembly of items for removing material from a workpiece through a shearing action at the defined cutting edge or edges of the device

NOTE A cutting tool could be the assembly of one or more adaptive items, a tool item and several cutting items on a tool item. See Figure 3.

3.2
information model
formal model of a bounded set of facts, concepts, or instructions to meet a specified requirement

NOTE See Reference [3].

NOTE "An information model is a formal description of types of ideas, facts and processes which together form a model of a portion of interest of the real world and which provides an *explicit set of interpretation rules*. (If an information model is written in EXPRESS or any other computer-sensible representation, it has the additional quality of being computer processible.)"[9].

4 Fundamental concepts and assumptions

ISO 13399 does not standardize cutting tools but the representation of information about cutting tools. Any standard dealing with information representation must be designed with the intent to standardize the *description* of the items that the standard deals with and not to standardize the items themselves. This method is also a step towards having standards that are not made obsolete by developments within the area where the standard applies [10].

The intent of ISO 13399 is achieved by means of a computer-processable information model that defines the framework within which data values can be placed. The benefits of such an information model are that it can be interpreted by different computer applications and be generic enough to describe many different kinds of tools. The aim of ISO 13399 is to allow for the different tools and machining operations that can be expected in the future, while meeting the needs of current types of tool and their uses. It should also be possible to describe a tool independently of its use, since the combination of modern tools and machines may allow a tool to be used for many different kinds of cutting operation.

ISO 13399 specifies the representation of information on

- cutting tools,
- cutting items,
- tool items,
- adaptive items, and
- assembly items.

See Figure 2.

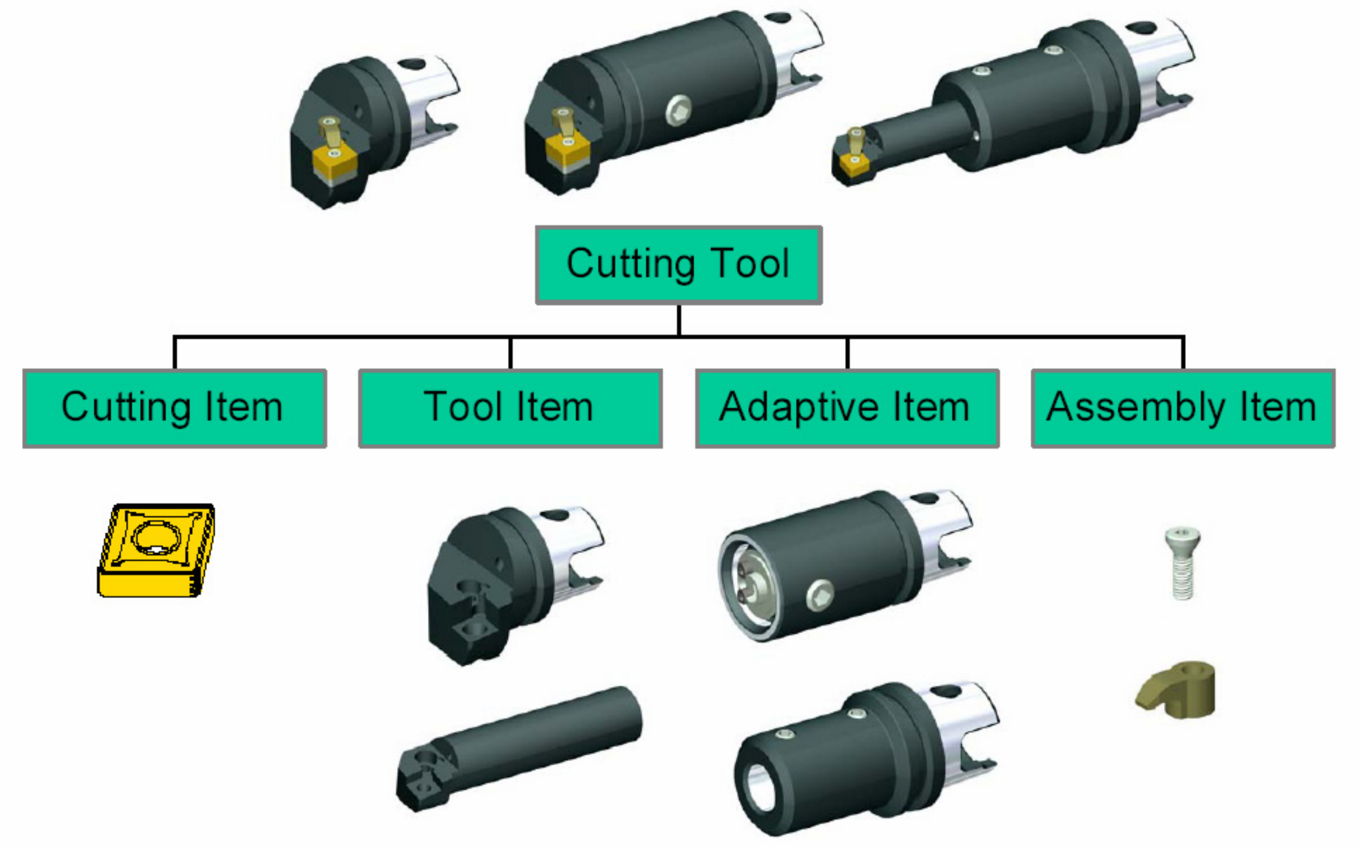

Figure 2 — Examples of the main groups

The definitions of these groups are given in their respective parts of ISO 13399. Different combinations of these groups are shown in Figure 3.

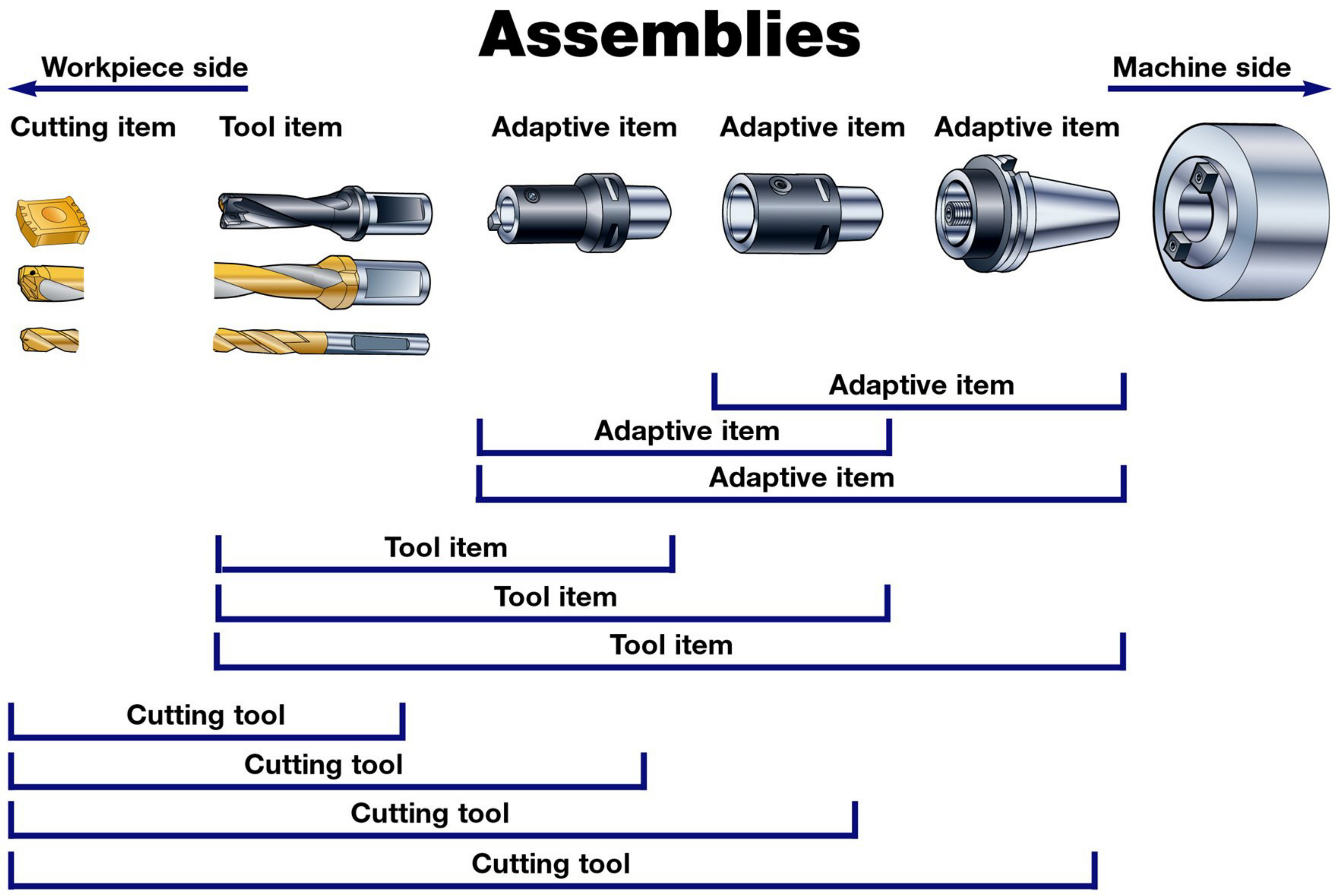

Figure 3 — Examples of different types of assemblies of items

The allowed relationships between these groups are shown in Table 1.

Table 1 — Matrix of Combinations

		Machine side		
		Tool item	Adaptive item	Assembly item
Workpiece side	Cutting item	X		X
	Tool item	X	X	X
	Adaptive item		X	X
	Assembly item	X	X	X

ISO 13399 does not deal with how to construct a cutting tool item, but provides an information representation for cutting tools. Tools may be classified according to any scheme, and viewed from many viewpoints.

5 Information requirements

5.1 Units of Functionality

5.1.1 classification

This unit of functionality provides the capability to classify items into specific categories.

EXAMPLE Examples are the classification as raw materials, parts, or tools, or as safety or in-process items.

Additionally items may be classified by their attributes.

EXAMPLE Examples for categories of attribute based classification are screws, bolts, nuts, shafts, or brackets.

The following application objects are used by the classification UoF:

- classification_association
- classification_association_relationship
- classification_attribute
- classification_system
- din4000_reference
- external_library_reference
- general_classification
- general_classification_hierarchy
- plib_class_reference
- plib_property_reference
- specific_item_classification
- specific_item_classification_hierarchy

5.1.2 dimension_tolerance

This unit of functionality specifies the representation of geometric dimensions and of tolerances limiting geometric dimensions. Tolerances for geometric dimensions are characterized by plus-minus-tolerances, limits and fits, dimension ranges, or dimension limits and describe the allowable deviation range.

The following application objects are used by the dimension_tolerance UoF:

- limits_and_fits
- plus_minus_bounds
- value_limitation

5.1.3 effectivity

This unit of functionality provides the capability to represent information concerning the validity of data. Implicit propagation of data specifying validity is not available.

The validity of data can be expressed by effectivities that specify time ranges within which data may be used.

The following application objects are used by the effectivity UoF:

- duration
- effectivity
- effectivity_assignment
- effectivity_relationship

5.1.4 external_reference_mechanism

This unit of functionality provides a reference mechanism to specify external documents that are associated with objects defined in this part of ISO 13399. These documents may contain data that are not in scope of this part of ISO 13399. This includes conventional non-digital data and digital data in other computer interpretable formats.

EXAMPLE Technical drawings on paper, standards documents, or hand-written documents, are examples of non-digital data referenced by the external reference mechanism.

EXAMPLE CAD data in a system's native format, text documents in SGML, or NC-data in ISO 6983 are examples of digital data referenced by the external reference mechanism.

The following application objects are used by the external_reference_mechanism UoF:

- cartesian_coordinate_space
- cartesian_coordinate_space_2d
- cartesian_coordinate_space_3d
- digital_document
- digital_file
- document
- document_assignment
- document_content_property
- document_creation_property
- document_file
- document_format_property
- document_location_property
- document_representation

- document_size_property
- document_type_property
- document_version
- document_version_relationship
- external_file_id_and_location
- external_geometric_model
- external_model
- external_picture
- geometric_model_relationship_with_transformation
- named_size
- physical_document
- physical_file
- rectangular_size
- transformation
- transformation_2d
- transformation_3d

5.1.5 individual_definition_structure

The following application objects are used by the individual_definition_structure UoF:

- physical_item_definition
- physical_item_structure_association

5.1.6 individual_management_data

The following application objects are used by the individual_management_data UoF:

- item_version_relationship
- location
- physical_item
- physical_item_definition
- physical_item_location_association
- physical_item_state_association
- physical_item_structure_association

- physical_item_version
- realized_item_association
- state

5.1.7 item_definition_structure

This unit of functionality provides the capability to represent relationships between items to build up various kinds of structures.

The structures are defined for item definitions, for item instances that are particular occurrences of item definitions or between item definitions and item instances.

Among these structures is a hierarchical assembly structure to represent the relationships between constituents and the assembly.

The following application objects are used by the item_definition_structure UoF:

- assembly_association
- assembly_definition
- item_definition_relationship
- item_instance
- item_structure_association
- mated_item_relationship
- mating_association
- mating_definition
- quantified_instance

5.1.8 item_property

This unit of functionality specifies various properties for a part, such as material, recyclability, mass, and general properties.

The information about recyclability is required to describe whether and how material or component parts may be reused. The costs, as planned or calculated for a part, as well as a quality measure property to keep track of the results of some processes, e.g., prototype building, may be specified. The mass is required to express the weight of a part. Shape related properties that are rotation and translation invariant such as the centre of mass and the moments of inertia may also be specified.

The general properties are required to define property data which are not covered by one of the other UoFs, and, due to their variety, cannot be specified all by explicit application objects.

EXAMPLE Noise development of an engine, maintenance intervals of parts, and heat or noise absorption of insulating parts between the engine and passenger compartment are examples for general properties.

The following application objects are used by the item_property UoF:

- coating

- coupling
- cutting_condition
- cutting_data_association
- grade
- grade_relationship
- item_characteristic_association
- item_property_association
- material_designation
- property
- property_relationship
- property_value_association
- property_value_representation
- property_value_representation_relationship
- substrate
- workpiece_feature

5.1.9 product_management_data

This unit of functionality provides the capability to represent product management information. It includes information about items that are either raw materials, parts, or tools, about versions and views of items, and about persons and dates.

Additionally, this unit of functionality provides the capability to represent descriptive information about objects in different languages and to assign alias identifiers to objects.

The following application objects are used by the product_management_data UoF:

- alias_identification
- application_context
- date_time
- item
- item_definition
- item_version
- item_version_relationship
- language

- multi_language_string
- numerical_value
- organization
- person
- person_in_organization
- person_organization_assignment
- property_value
- specific_item_classification
- specific_item_classification_hierarchy
- string_value
- string_with_language
- unit
- value_limit
- value_list
- value_range
- value_with_unit

5.2 Application objects

SCHEMA ISO13399_1;

5.2.1 default_language_string

EXPRESS specification:

```
TYPE default_language_string = STRING;
END_TYPE;
```

5.2.2 alias_select

EXPRESS specification:

```
TYPE alias_select = SELECT
  (item,
  item_version,
  item_definition,
  property,
  grade,
  classification_attribute,
  classification_system,
  document,
  document_representation,
  document_version,
  general_classification,
  item_instance,
  organization,
  document_type_property);
END_TYPE;
```

5.2.3 assigned_document_select

EXPRESS specification:

```
  TYPE assigned_document_select = SELECT
    (document_version,
    document);
  END_TYPE;
```

5.2.4 classification_source_select

EXPRESS specification:

```
  TYPE classification_source_select = SELECT
    (plib_class_reference,
    document,
    external_library_reference);
  END_TYPE;
```

5.2.5 classified_element_select

EXPRESS specification:

```
  TYPE classified_element_select = SELECT
    (item_version,
    item,
    item_definition,
    document,
    property,
    property_value_association,
    item_instance,
    document_file,
    document_version,
    document_representation,
    material_designation);
  END_TYPE;
```

5.2.6 documented_element_select

EXPRESS specification:

```
  TYPE documented_element_select = SELECT
    (item,
    item_version,
    classification_attribute,
    classification_system,
    item_definition,
    general_classification,
    item_structure_association,
    item_instance,
    organization,
    person,
    property,
    specific_item_classification,
    material_designation,
    mated_item_relationship,
    physical_item_structure_association);
  END_TYPE;
```

5.2.7 effective_element_select

EXPRESS specification:

```
  TYPE effective_element_select = SELECT
    (document,
    classification_system,
    document_file,
    document_representation,
    document_version,
    item,
    item_version,
    item_instance,
    property,
    property_value_association,
    material_designation,
    item_structure_association,
    realized_item_association,
    physical_item_state_association,
    physical_item_location_association,
    physical_item_structure_association);
  END_TYPE;
```

5.2.8 general_organizational_data_select

EXPRESS specification:

```
  TYPE general_organizational_data_select = SELECT
    (classification_association,
    classification_system,
    item_definition,
    document,
    document_file,
    document_representation,
    document_version,
    general_classification,
    item,
    item_structure_association,
    item_instance,
    item_version,
    item_version_relationship,
    person_in_organization,
    property,
    property_value_association,
    material_designation,
    physical_item_structure_association);
  END_TYPE;
```

5.2.9 item_characteristic_select

EXPRESS specification:

```
TYPE item_characteristic_select = SELECT
  (cutting_condition,
  coupling,
  workpiece_feature,
  material_designation,
  cutting_data_association,
  grade);
END_TYPE;
```

5.2.10 item_property_select

EXPRESS specification:

```
TYPE item_property_select = SELECT
  (item_definition,
  item_characteristic_select,
  document_file,
  document_representation,
  item_structure_association,
  item_instance,
  mated_item_relationship,
  physical_item_structure_association);
END_TYPE;
```

5.2.11 limitation_definition_select

EXPRESS specification:

```
TYPE limitation_definition_select = SELECT
  (limits_and_fits,
  plus_minus_bounds);
END_TYPE;
```

5.2.12 person_organization_select

EXPRESS specification:

```
TYPE person_organization_select = SELECT
  (organization,
  person_in_organization);
END_TYPE;
```

5.2.13 property_source_select

EXPRESS specification:

```
TYPE property_source_select = SELECT
  (plib_property_reference,
  external_library_reference,
  document);
END_TYPE;
```

5.2.14 string_select

EXPRESS specification:

```
  TYPE string_select = SELECT
    (default_language_string,
    multi_language_string);
  END_TYPE;
```

5.2.15 alias_identification

An alias_identification is a mechanism to associate an object with an additional identifier that is used to identify the object of interest in a different context, either in another organization, or in some other context. The scope of the alias_identification shall be specified either by the attribute 'alias_scope' or by the attribute 'description'.

EXAMPLE A document (e.g., a book) may have a unique document id (ISBN) and an alias_identification as inventory number in the context of the inventory of a company.

EXPRESS specification:

```
  ENTITY alias_identification;
    alias_id          :  STRING;
    alias_scope       :  OPTIONAL organization;
    alias_version_id  :  OPTIONAL STRING;
    description       :  OPTIONAL string_select;
    is_applied_to     :  alias_select;
  END_ENTITY;
```

5.2.15.1 alias_id

The alias_id specifies the identifier used in the context specified by the alias_scope, or by the description.

5.2.15.2 alias_scope

The alias_scope specifies the organization in which the alias_identification is valid.

The alias_scope need not be specified for a particular alias_identification.

5.2.15.3 alias_version_id

The alias_version_id specifies the version of the object as known in the context of the alias_identification.

NOTE An alias_version_id may be applied even if the object of interest does not have versions.

The alias_version_id need not be specified for a particular alias_identification.

5.2.15.4 description

The description specifies the type of the alias_identification.

EXAMPLE The description may be, e.g., 'inventory number'.

The description need not be specified for a particular alias_identification.

If present, there shall be exactly one object that defines the description for an alias_identification.

5.2.15.5 is_applied_to

The is_applied_to specifies the object that has an alias_identification.

5.2.16 application_context

An application_context is a shared universe of discourse.

EXPRESS specification:

```
ENTITY application_context;
  application_domain   :  STRING;
  description          :  OPTIONAL string_select;
END_ENTITY;
```

5.2.16.1 application_domain

The application_domain is the identification of the applications for which an object may be relevant.

Where applicable the following values shall be used:

'machining':	The object may be relevant for machining;
'manufacturing simulation':	The object may be relevant for manufacturing simulation.
'nc preparation':	The object may be relevant for nc preparation.
'process planning':	The object may be relevant for process planning.
'product planning':	The object may be relevant for product planning.
'purchasing':	The object might be relevant for purchasing;

5.2.16.2 description

The description specifies additional information about the application_context.

The description need not be specified for a particular application_context.

If present, there shall be exactly one object that defines the description for a application_context.

5.2.17 assembly_association

An assembly_association is the relation between an assembly_definition and an item_instance representing a constituent of the assembly. The assembly_definition and the item_definition that serves as 'definition' of the item_instance shall share at least one context.

NOTE The constituent may also be an assembly.

EXPRESS specification:

```
ENTITY assembly_association
  SUBTYPE OF(item_structure_association);
  SELF\item_structure_association.relating   :  assembly_definition;
END_ENTITY;
```

5.2.17.1 relating

The relating specifies the assembly_definition that has subordinate constituents.

5.2.18 assembly_definition

An assembly_definition is a definition of an item_version that contains other subordinate objects.

An assembly_definition is a type of item_definition.

EXPRESS specification:

```
ENTITY assembly_definition
  SUBTYPE OF(item_definition);
  assembly_type  :  OPTIONAL STRING;
END_ENTITY;
```

5.2.18.1 assembly_type

The assembly_type specifies the kind of the assembly_definition.

EXAMPLE 'functional assembly', 'manufacturing assembly', and 'design assembly' are examples of an assembly_type.

5.2.19 cartesian_coordinate_space

A cartesian_coordinate_space is a coordinate space in which geometric and annotation elements may be defined. It is either two-dimensional or three-dimensional. An origin for coordinate values is implicitly defined. The units applicable to the coordinate values of elements defined in the cartesian_coordinate_space are specified.

NOTE Cartesian transformation objects are used to specify the relationships between coordinate spaces.

Each cartesian_coordinate_space is a cartesian_coordinate_space_3d or a cartesian_coordinate_space_2d.

EXPRESS specification:

```
ENTITY cartesian_coordinate_space
    ABSTRACT SUPERTYPE OF (ONEOF(cartesian_coordinate_space_2d,
  cartesian_coordinate_space_3d));
      unit_of_values : OPTIONAL SET [2:?] OF unit;
  END_ENTITY;
```

5.2.19.1 unit_of_values

The unit_of_values specifies the various units in which any values are expressed. In the case where geometricelements are defined in the cartesian_coordinate_space, there shall be at least two units specified, the length unit and the plane angle unit. The same length unit is applied to each coordinate direction. Only one unit of a kind shall be specified.

NOTE If elements with different units are required, they have to be separated into different models with their own cartesian_coordinate_space.

EXAMPLE A length measure unit measured in inches and an angle measure unit measured in degrees are examples for two members of the attribute unit_of_values assigned to the same cartesian_coordinate_space.

The unit_of_values need not be specified for a particular cartesian_coordinate_space.

5.2.20 cartesian_coordinate_space_2d

A cartesian_coordinate_space_2d is a two-dimensional coordinate space. Any two-dimensional geometric and annotation element shall be defined in a cartesian_coordinate_space_2d.

A cartesian_coordinate_space_2d is a type of cartesian_coordinate_space.

EXPRESS specification:

```
ENTITY cartesian_coordinate_space_2d
  SUBTYPE OF(cartesian_coordinate_space);
END_ENTITY;
```

5.2.21 cartesian_coordinate_space_3d

A cartesian_coordinate_space_3d is a three-dimensional coordinate space. Any three-dimensional geometric data shall be defined in a cartesian_coordinate_space_3d.

A cartesian_coordinate_space_3d is a type of cartesian_coordinate_space.

EXPRESS specification:

```
ENTITY cartesian_coordinate_space_3d
  SUBTYPE OF(cartesian_coordinate_space);
END_ENTITY;
```

5.2.22 classification_system

A classification_system is the scheme used to define the categorization of an item.

EXAMPLE A dictionary collecting the conventional definitions of technical properties is an example of classification_system.

EXPRESS specification:

```
ENTITY classification_system;
  description    :  OPTIONAL string_select;
  id             :  STRING;
INVERSE
  allowed_classification  :  SET OF general_classification FOR
  used_classification_system;
END_ENTITY;
```

5.2.22.1 description

The description specifies additional information about the classification_system.

The description need not be specified for a particular classification_system.

If present, there shall be exactly one object that defines the description for a classification_system.

5.2.22.2 id

The id specifies the identifier of the classification_system.

5.2.23 classification_association

A classification_association associates a general_classification with an object.

EXPRESS specification:

```
ENTITY classification_association;
```

```
    associated_classification  :  general_classification;
    classified_element         :  classified_element_select;
    defintional                :  OPTIONAL BOOLEAN;
    role                       :  OPTIONAL STRING;
  END_ENTITY;
```

5.2.23.1 associated_classification

The associated_classification specifies the general_classification object that provides classification information.

5.2.23.2 classified_element

The classified_element specifies the object that is classified.

There shall be exactly one object that defines the classified_item for a classification_association.

5.2.23.3 defintional

The definitional specifies whether a general_classification serves as definition. A value of 'true' indicates that the general_classification is definitional. The 'associated_classification' does not take precedence over the descriptions of the 'classified_element' made using property_value objects.

The attribute definitional shall only be used, if the classified_element is of one of the types material_designation, item, item_version, or item_definition.

The definitional need not be specified for a particular classification_association.

5.2.23.4 role

The role specifies the relationship between the general_classification and the associated item.

The role need not be specified for a particular classification_association.

5.2.24 classification_association_relationship

A classification_association_relationship is a relationship between two classification_association obejcts.

EXPRESS specification:

```
  ENTITY classification_association_relationship;
    related           :  classification_association;
    relating          :  classification_association;
    relationship_type :  STRING;
  END_ENTITY;
```

5.2.24.1 related

Specifies the second general_classification that is part of the classification_relationship

5.2.24.2 relating

Specifies the first general_classification that is part of the classification_relationship

5.2.24.3 relationship_type

Specifies the type of relationship represented by the classification_relationship

Where applicable the following values shall be used:

'combination': specifies that the object being classified, by the general_classification objects that are a part of the classification_relationship, is classified as being a combination of the general_classification objects.
'alternative': specifies that the object being classified can be considered as being either one or the other of the participating general_classifications.

5.2.25 classification_attribute

A classification_attribute is a characteristic used to classify an object associated with the corresponding general_classification. The definition attribute of each 'allowed_value' shall refer to the property identified within 'attribute_definition'.

EXPRESS specification:

```
ENTITY classification_attribute;
  allowed_value              :  SET OF property_value_representation;
  associated_classification  :  general_classification;
  attribute_definition       :  property;
  description                :  OPTIONAL string_select;
  id                         :  STRING;
  name                       :  OPTIONAL string_select;
END_ENTITY;
```

5.2.25.1 allowed_value

The allowed_value specifies the set of property_value_representation objects that represent characteristic values of the classification_attribute.

5.2.25.2 associated_classification

The associated_classification specifies the general_classification the classification_attribute is a characteristic of.

5.2.25.3 attribute_definition

The attribute_definition specifies the property that characterizes the allowed values.

NOTE The specification of compound characteristics can be realized by using property_relationship with 'relation_type' set to 'peer' or 'decomposition'.

5.2.25.4 description

The description specifies additional information about the classification_attribute.

The description need not be specified for a particular classification_attribute.

If present, there shall be exactly one object that defines the description for a classification_attribute.

5.2.25.5 id

The id specifies the identifier of the classification_attribute that shall be unique within the scope of the associated general_classification.

EXAMPLE 'a' or 'b' for length or width attributes or 'r' for radius attributes are examples for identifiers of classification_attribute objects. The meaning of such ids is usually specified in external sources.

5.2.25.6 name

The name specifies the word or group of words by which the classification_attribute is referred to.

EXAMPLE 'length', 'width', or 'radius' are examples for the name of a classification_attribute.

The name need not be specified for a particular classification_attribute.

If present, there shall be exactly one object that defines the name for a classification_attribute.

5.2.26 coating

EXPRESS specification:

```
ENTITY coating;
      coating_name    : STRING;
      coating_process : STRING;
END_ENTITY;
```

5.2.27 coupling

EXPRESS specification:

```
ENTITY coupling;
  coupling_type  :  STRING;
  pieces         :  OPTIONAL numerical_value;
  side           :  STRING;
  size           :  numerical_value;
  style          :  STRING;
END_ENTITY;
```

5.2.27.1 coupling_type

Mount style may be one of:

Code	Description
0	Other clamping systems, not listed below
1	Without fixing hole [ISO 1832:1991]
2	With cylindrical fixing hole [ISO 1832:1991]
3	With partly cylindrical fixing hole, countersink on one side only [ISO 1832:1991]
4	With partly cylindrical fixing hole, countersinks on both sides [ISO 1832:1991]
5	With notch clamping
6	With convex prismatic cross section
7	With concave prismatic cross section

5.2.27.2 size

The size of the coupling.

EXAMPLE diameter of a through hole.

5.2.28 cutting_condition

Conditions relevant to the cutting process, e.g. finishing, roughing, interrupted cut, smooth cut. There may be more than one cutting_condition relevant for any item.

EXPRESS specification:

```
ENTITY cutting_condition;
  condition_name    :  STRING;
END_ENTITY;
```

5.2.29 cutting_data_association

Cutting data is information about the cutting conditions and performance of the cutting tool.

NOTE Certain cutting data only appear through the combination of different items, machine tools and workpieces.

EXPRESS specification:

```
ENTITY cutting_data_association;
  associated_material  :  material_designation;
  depth_of_cut         :  value_with_unit;
  feed                 :  value_with_unit;
  speed                :  value_with_unit;
END_ENTITY;
```

5.2.29.1 associated_item

Specifies one or more items associated with the cutting data.

5.2.29.2 associated_material

Specifies the workpiece_material associated with the cutting data.

5.2.29.3 depth_of_cut

Specifies the perpendicular distance between the original and final surfaces of the workpiece.

5.2.29.4 feed

Specifies the feed according to ISO 3002-3.

5.2.29.5 speed

Specifies the cutting speed according to ISO 3002-1

5.2.30 date_time

A date_time is the specification of a date and an optional time of day.

EXPRESS specification:

```
ENTITY date_time;
  date :  STRING;
  time :  OPTIONAL STRING;
END_ENTITY;
```

5.2.30.1 date

The date specifies the calendar time, defined according to the Gregorian calendar, conveying information about the year, the month, and the day in no specific order. The representation of a date shall be complete, i.e., millenium, century, and year-within-century data shall be included.

5.2.30.2 time

The time specifies a moment of occurrence measured by hour, minute, and second.

The time need not be specified for a particular date_time.

5.2.31 digital_document

A digital_document is a piece of product data that is archived in a digital format.

A digital_document is a type of document_representation.

EXPRESS specification:

```
ENTITY digital_document
  SUBTYPE OF(document_representation);
  file :  SET OF digital_file;
END_ENTITY;
```

5.2.31.1 file

The file specifies a computer interpretable realization of the digital_document.

5.2.32 digital_file

A digital_file contains computer interpretable data.

A digital_file is a type of document_file.

EXPRESS specification:

```
ENTITY digital_file
  SUBTYPE OF(document_file);
INVERSE
  associated_model_space  :  SET OF external_model FOR is_defined_as;
END_ENTITY;
```

5.2.33 din4000_reference

DIN 4000 reference is the information about the comparability of a property between the ISO 13399 standard and the DIN 4000 standard For the reference description the properties DIN 4000 - Part No and characteristics code no is required.

EXPRESS specification:

```
ENTITY din4000_reference
  SUBTYPE OF(external_library_reference);
  characteristics_code_no :  STRING;
  part_no                 :  STRING;
END_ENTITY;
```

5.2.34 document

A document is a logical document that serves as the identifier for a container for some product data.

EXAMPLE A document can either be any kind of drawing, solid model, or visible information about the item that is geometrically described with the properties of the ISO 13399.

EXPRESS specification:

```
ENTITY document;
  description    :  OPTIONAL string_select;
  document_id    :  STRING;
  name           :  string_select;
INVERSE
  associated_version   :  SET [1:?] OF document_version FOR
  associated_document;
END_ENTITY;
```

5.2.34.1 description

The description specifies additional information about the document.

The description need not be specified for a particular document.

If present, there shall be exactly one object that defines the description for a document.

5.2.34.2 document_id

The document_id specifies the identifier of the document.

5.2.34.3 name

The name specifies the word or group of words by which the document is referred to.

There shall be exactly one object that defines the name for a document.

5.2.35 document_assignment

A document_assignment is a mechanism to associate a document with an object, where the assigned document provides information about the object it is associated to.

EXPRESS specification:

```
ENTITY document_assignment;
  assigned_document :  assigned_document_select;
  is_assigned_to    :  documented_element_select;
  role              :  STRING;
END_ENTITY;
```

5.2.35.1 assigned_document

The assigned_document specifies a document, or a document_version that is used to provide information.

There shall be exactly one object that defines the assigned_document for a document_assignment.

5.2.35.2 is_assigned_to

The is_assigned_to specifies the object that information is provided for.

There shall be exactly one object that the document_assignment is_assigned_to.

5.2.35.3 role

The role specifies the meaning of the document_assignment.

Where applicable the following values shall be used:

'additional information': The assigned document provides information that is relevant for the associated object, but is not a description of the associated object itself;
'behavior': The assigned document specifies information about the behaviour of the associated object;
'description': The assigned document provides textual information for the associated object itself;
'informative': The assigned document may or may not be considered;
'mandatory': The associated object shall conform to the content of the assigned document;
'mathematical description': The assigned document specifies the associated object by providing the algorithmic specification of its behavior.

5.2.36 document_content_property

A document_content_property specifies characteristics precising the content of a document_file or of a document_representation. At least one of the optional attributes shall be specified for each instance of this object.

In the case where a document_content_property is referred by a document_representation, the characteristics apply to all individual document_file objects, whereas in the case where it is referred by a document_file, the characteristics apply on an individual basis.

EXPRESS specification:

```
ENTITY document_content_property;
  detail_level      :  OPTIONAL STRING;
  geometry_type     :  OPTIONAL STRING;
  languages         :  SET OF language;
  real_world_scale  :  OPTIONAL numerical_value;
END_ENTITY;
```

5.2.36.1 detail_level

The detail_level specifies the level of detail that the document_file or the document_representation provides.

Where applicable the following values shall be used:

'rough 3d shape': 3D shape model without edge rounds and fillets;
'rounded edges': 3D shape model with edge rounds and fillets.

The detail_level need not be specified for a particular document_content_property.

5.2.36.2 geometry_type

The geometry_type specifies the kind or kinds of geometry that an object contains.

Where applicable the following values shall be used:

'3D wireframe model': The document contains a 3D shape model in wireframe representation;
'2D shape': The document contains a 2D shape model or contours only;
'surface model': The document contains a 3D shape model in surface representation;
'closed volume': The document contains a 3D shape model in closed body topological surface representation;
'solid model': The document contains a 3D shape model in advanced boundary representation;
'solid and surface model': The document contains a 3D shape model in surface and advanced boundary representation;

'assembly': The document contains an assembly structure with reference to the assembled components and their transformation matrices;
'assembly with mating elements': The document contains an assembly structure including the mating components only, such as screws or rivets, with exact positioning information. This assembly representation is intended to be overlayed with the assembly structure for the main components;
'2D drawing': The document contains a technical drawing without 3D shape representation;
'drawing derived from 3D data': The document contains a technical drawing that has been derived from a 3D shape model;
'drawing related to 3D data': The document contains a technical drawing that visualizes a 3D shape model and possibly establishes associative links to the 3D shape model.

The geometry_type need not be specified for a particular document_content_property.

5.2.36.3 languages

The languages specifies which language or languages are used in the characterized objects.

EXAMPLE 'Japanese' and 'German' are examples for the fact that annotation on a drawing is provided in the language 'Japanese' or 'German' respectively.

5.2.36.4 real_world_scale

The real_world_scale specifies the scale that is used in the document_file or in the document_representation the document_content_property is referred by.

The real_world_scale need not be specified for a particular document_content_property.

5.2.37 document_creation_property

A document_creation_property specifies characteristics of document_file or of document_representation objects. It specifies the context of the creation of the object. At least one of the optional attributes shall be specified for each instance of this object.

In the case where a document_creation_property is referred by a document_representation the characteristics apply to all individual document_file objects, whereas in the case it is referred by a document_file, the characteristics apply on an individual basis.

EXPRESS specification:

```
ENTITY document_creation_property;
  creating_interface   :  OPTIONAL STRING;
  creating_system      :  STRING;
  operating_system     :  OPTIONAL STRING;
END_ENTITY;
```

5.2.37.1 creating_interface

The creating_interface specifies the computer application used to create the document_file or document_representation object.

EXAMPLE 'Postscript Printer Driver' is an example for a creating interface of a physical_document.

The creating_interface need not be specified for a particular document_creation_property.

5.2.37.2 creating_system

The creating_system specifies the computer application or the machine which is used to create the object that is characterized.

5.2.37.3 operating_system

The operating_system specifies the operating system that is used to execute the computer application that created the characterized object.

The operating_system need not be specified for a particular document_creation_property.

5.2.38 document_file

A document_file is one of potentially more files on a computer system or in actual stacks of paper that make up a document_representation.

EXPRESS specification:

```
ENTITY document_file
    ABSTRACT SUPERTYPE OF (ONEOF(digital_file, physical_file));
      content                : OPTIONAL document_content_property;
      creation               : OPTIONAL document_creation_property;
      document_file_type     : OPTIONAL document_type_property;
      external_id_and_location : SET OF external_file_id_and_location;
      file_format            : OPTIONAL document_format_property;
      file_id                : STRING;
      size                   : OPTIONAL document_size_property;
      version_id             : OPTIONAL STRING;
END_ENTITY;
```

5.2.38.1 content

The content characterizes the content of the document_file.

The content need not be specified for a particular document_file.

5.2.38.2 creation

The creation specifies further details of the context of the creation of the document_file.

The creation need not be specified for a particular document_file.

5.2.38.3 document_file_type

The document_file_type specifies the format of the document_file. It shall only be specified, if the document_file does not participate in a document.

The document_file_type need not be specified for a particular document_file.

5.2.38.4 external_id_and_location

The external_id_and_location specifies alternatives of the identifier and location of the document_file.

EXAMPLE A copy of a document may be found in another department with a different id.

5.2.38.5 file_format

The file_format specifies the characteristics of the document_file that specify the format of the object.

The file_format need not be specified for a particular document_file.

5.2.38.6 file_id

The file_id specifies the identifier which is used to locate the file either on a computer system or in a repository of paper documents.

5.2.38.7 size

The size specifies characteristics for the size of the document_file.

The size need not be specified for a particular document_file.

5.2.38.8 version_id

The version_id specifies the identification of the version that distinguishes one document_file object from other versions of document_file objects with the same file_id.

The version_id need not be specified for a particular document_file.

5.2.39 document_format_property

A document_format_property specifies characteristics of a document_file or of a document_representation that specify the format of the object. At least one of the optional attributes shall be specified for each instance of this object.

In the case where a document_format_property is referred by a document_representation, the characteristics apply to all individual document_file objects, whereas in the case it is referred by a document_file the characteristics apply on an individual basis.

EXPRESS specification:

```
ENTITY document_format_property;
  character_code    :  OPTIONAL STRING;
  data_format       :  OPTIONAL STRING;
  size_format       :  OPTIONAL rectangular_size;
END_ENTITY;
```

5.2.39.1 character_code

The character_code specifies the character code that is used in the characterized object.

Where applicable the following values shall be used:

'binary': The document contains data in binary format;
'IEC 61286': The coded character set used to encode the document data according to IEC 61286;
'ISO 646': The coded character set used to encode the document data according to ISO 646;

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

'ISO 3098-1': The coded character set used to encode the document data is according to ISO 3098-1;
'ISO 6937': The coded character set used to encode the document data is according to ISO/IEC 6937;
'ISO 8859-1': The coded character set used to encode the document data according to ISO 8859-1;

NOTE The character set in ISO 8859-1 is identical to the character set commonly known as LATIN-1.

'ISO 10646': The coded character set used to encode the document data according to ISO/IEC 10646.

The character_code need not be specified for a particular document_format_property.

5.2.39.2 data_format

The data_format specifies the convention that was used to structure the information in the characterized object.

Where applicable the following values shall be used:

'DXF': The document contains data in Drawing Exchange File format;
'IGES': The document contains data in Initial Graphics Exchange Specification format;
'ISO 10303-203': The document contains data in ISO 10303-203 format;
'ISO 10303-214': The document contains data in ISO 10303-214 format;
'TIFF CCITT GR4': The document contains data in TIFF CCITT GR4 format;
'VDAFS': The document contains data in VDAFS format;
'VOXEL': The document contains data in VOXEL format.

The data_format need not be specified for a particular document_format_property.

5.2.39.3 size_format

The size_format specifies the dimensions of a physical presentation of the object the size_format is provided for.

EXAMPLE 'ISO A0' is an example for the size_format of a drawing that is stored digitally.

The size_format need not be specified for a particular document_format_property.

5.2.40 document_location_property

A document_location_property specifies where a document_file or a document_representation can be found in a digital or physical data storage system.

In the case where a document_location_property is referred by a document_representation, the characteristics apply to all individual objects, whereas in the case it is referred by a document_file, the characteristics apply on an individual basis.

EXPRESS specification:

```
ENTITY document_location_property;
  location_name  :  STRING;
END_ENTITY;
```

5.2.40.1 location_name

The location_name specifies the location, where the object that refers to the document_location_property, can be found. 'C:\mpbs\programs' and '/usr/local/bin' are examples for a location_name.

NOTE Multiple paths may be specified for a single object, e.g., a database in the context of an electronic vault.

EXAMPLE The linking mechanism of many operating systems allows references to a single object from various places in the file system without copying its contents.

5.2.41 document_representation

A document_representation is one of potentially more alternative representations of a document.

EXAMPLE A logical document, which contains a shape model, may be represented in the different native formats of two 3D CAD systems.

Each document_representation is a digital_document or a physical_document.

EXPRESS specification:

```
ENTITY document_representation
    ABSTRACT SUPERTYPE OF (ONEOF(digital_document, physical_document));
      associated_document_version : document_version;
      common_location              : SET OF document_location_property;
      content                      : OPTIONAL document_content_property;
      creation                     : OPTIONAL document_creation_property;
      description                  : OPTIONAL string_select;
      id                           : STRING;
      representation_format        : OPTIONAL document_format_property;
      size                         : OPTIONAL document_size_property;
END_ENTITY;
```

5.2.41.1 associated_document_version

The associated_document_version specifies the version of the logical document that is being represented.

5.2.41.2 common_location

The common_location specifies the location of a document_representation, where all its constituents can be found.

NOTE Different common locations represent alternative representations of the same physical location.

5.2.41.3 content

The content specifies characteristics of the content of the document_representation.

The content need not be specified for a particular document_representation.

5.2.41.4 creation

The creation specifies further details of the creation of the document_representation.

The creation need not be specified for a particular document_representation.

5.2.41.5 description

The description specifies additional information about the document_representation.

The description need not be specified for a particular document_representation.

If present, there shall be exactly one object that defines the description for a document_representation.

5.2.41.6 id

The id specifies the identifier of the document_representation.

5.2.41.7 representation_format

The representation_format specifies the format of the document represented by document_representation.

The representation_format need not be specified for a particular document_representation.

5.2.41.8 size

The size specifies the size of the represented document.

The size need not be specified for a particular document_representation.

5.2.42 document_size_property

A document_size_property specifies the size of a document_file or of a document_representation object. At least one of the optional attributes shall be specified for each instance of this object.

In the case where a document_size_property is referred by a document_representation, the size information is the sum of the sizes of all individual objects that are collected by this object, whereas in the case it is referred by a document_file, the size information is the one of the individual objects that is referenced.

EXPRESS specification:

```
  ENTITY document_size_property;
    file_size  :  OPTIONAL value_with_unit;
    page_count :  OPTIONAL value_with_unit;
  END_ENTITY;
```

5.2.42.1 file_size

The file_size specifies the value_with_unit that represents the size of a digitally stored document. The file_size shall only be applied in cases where the document_size_property is referred by a digital_document or a document_file.

EXAMPLE '15021 Bytes' and 'less than 500 Bytes' are examples for a file_size.

The file_size need not be specified for a particular document_size_property.

5.2.42.2 page_count

The page_count specifies the number of pages of the application object the document_size_property is referred by. The page_count shall only be used in cases where the document_size_property is referred by a physical_file or a physical_document.

EXAMPLE '42 pages' and 'more than 1 page' are examples of a page_count.

The page_count need not be specified for a particular document_size_property.

5.2.43 document_type_property

A document_type_property specifies the kind of a document_file.

EXPRESS specification:

```
  ENTITY document_type_property;
    document_type_name         :  STRING;
    used_classification_system :  OPTIONAL classification_system;
  END_ENTITY;
```

5.2.43.1 document_type_name

The document_type_name specifies the word or the group of words that describe the kind of object the characteristics are provided for.

Where applicable the following values shall be used:

'geometry':	The document represents a shape model;
'NC data':	The document represents numerical control data;
'FE data':	The document represents finite element data;
'sample data':	The document represents measured data;
'process plan':	The document represents process planning data;
'check plan':	The document represents quality control planning data;
'drawing':	The document represents a technical drawing.

5.2.43.2 used_classification_system

The used_classification_system specifies the classification_system the document_type_name is defined in.

EXAMPLE A document_type_property may be used with 'specification type 1' and 'specification type 2' as values for the attribute 'document_type_name' due to a company wide definition. By inclusion of these types in a general_classification with name 'specification' and description 'specification of company XY' these types of documents are better classified and the meaning of 'specification type 1' and 'specification type 2' can be further detailed in an attached document to the general_classification or by a reference to a classification_system for document types with that organization.

The used_classification_system need not be specified for a particular document_type_property.

5.2.44 document_version

A document_version is a release of a document.

EXPRESS specification:

```
ENTITY document_version;
  associated_document  :  document;
  description          :  OPTIONAL string_select;
  id                   :  STRING;
END_ENTITY;
```

5.2.44.1 associated_document

The associated_document specifies the logical document the document_version is a version of.

5.2.44.2 description

The description specifies additional information about the document_version.

The description need not be specified for a particular document_version.

If present, there shall be exactly one object that defines the description for a document_version.

5.2.44.3 id

The id specifies the identifier of the document_version. The id shall be unique within the scope of the associated document.

5.2.45 document_version_relationship

A document_version_relationship is a relationship between two document_version objects.

EXPRESS specification:

```
ENTITY document_version_relationship;
  description    :  OPTIONAL string_select;
  related        :  document_version;
  relating       :  document_version;
  relation_type  :  STRING;
END_ENTITY;
```

5.2.45.1 description

The description specifies additional information about the document_version_relationship.

The description need not be specified for a particular document_version_relationship.

If present, there shall be exactly one object that defines the description for a document_version_relationship.

5.2.45.2 related

The related specifies the second of the two objects related by the document_version_relationship.

NOTE The semantics of this attribute are defined by the attribute relation_type.

5.2.45.3 relating

The relating specifies the first of the two objects related by the document_version_relationship.

NOTE The semantics of this attribute are defined by the attribute relation_type.

5.2.45.4 relation_type

The relation_type specifies the meaning of the relationship.

Where applicable the following values shall be used:

'derivation': The application object defines a deriving relationship where the related document_version is based on the relating document_version which is an earlier version of the same or of a different document;
'hierarchy': The application object defines a hierarchical relationship where the related document_version is a sub version of the relating document_version;

EXAMPLE 'Revision 1.1' and 'revision 1.2' are examples of sub-versions of 'version 1'.

'sequence': The application object defines a version sequence where the relating document_version is the preceding version and the related document_version is the following version.

Each document shall be referred to by at most one document_version_relationship as related and a 'relation_type' of 'sequence';

'supplied document': The application object defines a relationship between two document_version objects representing the same object in different organizational contexts.

NOTE The different organizational contexts can be represented by different general organizational data associations with the related document_version objects.

5.2.46 duration

A duration is the definition of a period of time.

EXPRESS specification:

```
ENTITY duration;
  time        :  STRING;
  time_unit   :  STRING;
END_ENTITY;
```

5.2.46.1 time

The time specifies the extend of the duration.

5.2.46.2 time_unit

The time_unit specifies the unit in which the time is specified.

5.2.47 effectivity

An effectivity is the identification of the valid use of an aspect of product data tracked by date or event.

NOTE An instance of effectivity may define a period of ineffectivity, i.e., of invalid use, if it is referred to by an instance of effectivity_assignment with 'effectivity_indication' set to 'false'.

Either 'start_definition' or 'end_definition' or both shall be specified for a particular effectivity. The only exception from this rule is an effectivity that 'inherits' its interval from another effectivity through an effectivity_relationship with relation_type 'inheritance'. If a start_definition and no end_definition is specified, the period is added to start_definition in order to get the actual end of the effectivity or ineffectivity period.

NOTE If period is not specified, the effectivity or ineffectivity period is endless. If no start_definition, but an end_definition is specified, the period shall also be specified and is subtracted from end_definition in order to get the actual start of the effectivity or ineffectivity period.

NOTE This standard does not allow to specify effectivity or ineffectivity intervals which are left-unbounded. If both start_definition and end_definition are specified, the period should not be specified because of the redundancy. If specified, it shall be equal to the measure of time between the start_definition and the end_definition of the effectivity or ineffectivity period.

EXPRESS specification:

```
ENTITY effectivity;
  concerned_organization  :  SET OF organization;
  description             :  OPTIONAL string_select;
  effectivity_context     :  OPTIONAL STRING;
  end_definition          :  OPTIONAL date_time;
  id                      :  OPTIONAL STRING;
  period                  :  OPTIONAL duration;
  start_definition        :  OPTIONAL date_time;
  version_id              :  OPTIONAL STRING;
END_ENTITY;
```

5.2.47.1 concerned_organization

The concerned_organization specifies the set of organization objects in which the effectivity is valid.

EXAMPLE The effectivity of the same item may be different in the various production sites of a manufacturer.

The case where the concerned_organization is an empty set means that the effectivity regards any organization that may consider the product data, the effectivity is applied to.

NOTE As effectivity is most of the time organization dependent, it is recommended always to populate the concerned_organization

5.2.47.2 description

The description specifies additional information about the effectivity.

The description need not be specified for a particular effectivity.

If present, there shall be exactly one object that defines the description for an effectivity.

5.2.47.3 effectivity_context

The effectivity_context specifies the life cycle stage for which the effectivity is valid.

The effectivity_context need not be specified for a particular effectivity.

5.2.47.4 end_definition

The end_definition specifies the end of the period. The bound specified by the end_definition is excluded from the interval of effectivity.

The end_definition need not be specified for a particular effectivity.

If present, there shall be exactly one object that defines the end_definition for an effectivity.

5.2.47.5 id

The id specifies the identifier of the effectivity.

The id need not be specified for a particular effectivity.

5.2.47.6 period

The period specifies the period of time in which the effectivity is defined, either starting at the point in time specified by 'start_definition' or ending at the point in time specified by 'end_definition'. period shall be specified with a positive value.

The period need not be specified for a particular effectivity.

5.2.47.7 start_definition

The start_definition specifies the start of the period. The bound specified by the start_definition is included in the interval of effectivity.

NOTE The meaning of this attribute is further detailed by the attribute role of effectivity_assignment.

Either start_definition or end_definition shall be specified for a particular effectivity.

The start_definition need not be specified for a particular effectivity.

If present, there shall be exactly one object that defines the start_definition for an effectivity.

5.2.47.8 version_id

The version_id specifies the identification of a particular version of the effectivity.

The version_id need not be specified for a particular effectivity.

5.2.48 effectivity_assignment

An effectivity_assignment associates an effectivity with the object whose effectivity is controlled by the associated effectivity. The association of an effectivity to product data does not imply any statement concerning the effectivity outside of the specified interval. The same applies in the absence of any assigned effectivity, i.e. no statement concerning the effectivity is implied.

EXPRESS specification:

```
ENTITY effectivity_assignment;
  assigned_effectivity    :  effectivity;
  effective_element       :  effective_element_select;
  effectivity_indication  :  BOOLEAN;
  role                    :  STRING;
END_ENTITY;
```

5.2.48.1 assigned_effectivity

The assigned_effectivity specifies the effectivity that is assigned.

5.2.48.2 effective_element

The effective_element specifies the object that has an effectivity assigned to it.

There shall be exactly one object that defines the effective_element for an effectivity_assignment.

5.2.48.3 effectivity_indication

The effectivity_indication specifies whether the assigned_effectivity defines a period of effectivity (value equal 'TRUE') or a period of ineffectivity (value equal 'FALSE') for the effective_element. In the first case, use of the effective_element is or was valid during the considered period.

NOTE This does not imply that use of the effectivity_element is mandatory.

In the second case, use of effective_element is or was not valid during the considered period.

5.2.48.4 role

The role specifies the relationship between the effectivity and the object that has an effectivity assigned to it.

Where applicable the following values shall be used:

'actual': The actual period during which the effectivity lasted;
'planned': The period associated with the effectivity defines a planned period of time during which the associated object is or was supposed to be effective;
'required': The associated object must be kept effective for this period.

5.2.49 effectivity_relationship

An effectivity_relationship is a relationship between two effectivity objects.

NOTE Sometimes the effectivity is not dependent on particular dates but on the effectivity of other items. In this case the dates are not instantiated and there is an effectivity_relationship to the reference effectivity.

EXPRESS specification:

```
ENTITY effectivity_relationship;
  description    :  OPTIONAL string_select;
  related        :  effectivity;
  relating       :  effectivity;
  relation_type  :  STRING;
END_ENTITY;
```

5.2.49.1 description

The description specifies additional information about the effectivity_relationship.

The description need not be specified for a particular effectivity_relationship.

If present, there shall be exactly one object that defines the description for an effectivity_relationship.

5.2.49.2 related

The related specifies the second of the two effectivity objects related by the effectivity_relationship.

NOTE The semantics of this attribute are defined by the attribute 'relation_type'.

5.2.49.3 relating

The relating specifies the first of the two effectivity objects related by the effectivity_relationship.

NOTE The semantics of this attribute are defined by the attribute 'relation_type'.

5.2.49.4 relation_type

The relation_type specifies the meaning of the relationship.

Where applicable the following values shall be used:

'constraint': The time period between the start and end definition of the related effectivity shall be within the time period of the relating effectivity;
'inheritance': The related effectivity shall not have a 'start_definition' and 'end_definition' specified but inherits the effectivity dates from the relating effectivity.

5.2.50 external_file_id_and_location

An external_file_id_and_location specifies the location of a file in an external storage system.

EXPRESS specification:

```
ENTITY external_file_id_and_location;
  external_id    :  OPTIONAL STRING;
  location       :  document_location_property;
END_ENTITY;
```

5.2.50.1 external_id

The external_id specifies the identifier of a document in an external storage system.

EXAMPLE An example for external_id is 'specification.txt'.

The external_id need not be specified for a particular external_file_id_and_location.

5.2.50.2 location

The location specifies the location of the document_file in the external storage system.

EXAMPLE Examples for the location are 'D:\project1\' and '/projects/project1/'.

EXAMPLE The combination of file id and its location taken from the previous examples are D:\project1\specification.txt' and '/projects/project1/specification.txt' respectively.

5.2.51 external_geometric_model

An external_geometric_model is the identification of a model that contains geometry in a 3D context only.

An external_geometric_model is a type of external_model.

EXPRESS specification:

```
ENTITY external_geometric_model
  SUBTYPE OF(external_model);
  model_extent   :  OPTIONAL STRING;
END_ENTITY;
```

5.2.51.1 model_extent

The model_extent specifies the radius of a sphere that contains all elements of the model and whose centre is at the origin of the cartesian_coordinate_space of the external_geometric_model. The model_extent is specified using a length unit.

The model_extent need not be specified for a particular external_geometric_model.

5.2.52 external_library_reference

An external_library_reference is a mechanism to refer to an entry in an external library other than ISO 13584.

EXPRESS specification:

```
ENTITY external_library_reference;
  description    :  OPTIONAL string_select;
  external_id    :  STRING;
  library_type   :  STRING;
END_ENTITY;
```

5.2.52.1 description

The description specifies additional information about the external_library_reference.

The description need not be specified for a particular external_library_reference.

If present, there shall be exactly one object that defines the description for an external_library_reference.

5.2.52.2 external_id

The external_id specifies the unique identifier of the referenced entry in the external library.

5.2.52.3 library_type

The library_type specifies the type of library that is used.

5.2.53 external_model

An external_model is the identification of a model that is described in a digital_file and by the cartesian_coordinate_space that is needed to further process the externally described information.

Each external_model is an external_picture or an external_geometric_model.

EXPRESS specification:

```
ENTITY external_model
    ABSTRACT SUPERTYPE OF (ONEOF(external_picture,
  external_geometric_model));
      description   : OPTIONAL string_select;
      is_defined_as : digital_file;
      is_defined_in : cartesian_coordinate_space;
      model_id      : STRING;
END_ENTITY;
```

5.2.53.1 description

The description specifies additional information about the external_model.

The description need not be specified for a particular external_model.

If present, there shall be exactly one object that defines the description for an external_model.

5.2.53.2 is_defined_as

The is_defined_as specifies the digital_file that contains the externally defined geometry information.

5.2.53.3 is_defined_in

The is_defined_in specifies the cartesian_coordinate_space that defines the context for the externally described geometry.

5.2.53.4 model_id

The model_id specifies the identifier of the external_model.

5.2.54 external_picture

An external_picture is the identification of a model that is described by a two dimensional image.

An external_picture is a type of external_model.

EXPRESS specification:

```
ENTITY external_picture
  SUBTYPE OF(external_model);
  SELF\external_model.is_defined_in :  cartesian_coordinate_space_2d;
END_ENTITY;
```

5.2.54.1 is_defined_in

The is_defined_in specifies the cartesian_coordinate_space_2d that defines the context for the externally defined geometry.

For picture formats having no explicit origin, e.g., raster format, the lower left corner is defined as the origin of the picture, the x-axis pointing to the right, the y-axis pointing to the top.

5.2.55 general_classification_hierarchy

A general_classification_hierarchy defines a hierarchical relationship between two instances of general_classification.

EXAMPLE With general_classification, part families such as screws can be classified. Additionally subclasses can be built and linked to the super-category screws, such as machine screw or self-tapping screw.

NOTE The general_classification_hierarchy is used to build up hierarchical structures of general_classification objects.

EXPRESS specification:

```
  ENTITY general_classification_hierarchy;
    sub_classification   :  general_classification;
    super_classification :  general_classification;
  END_ENTITY;
```

5.2.55.1 sub_classification

The sub_classification specifies the lower level of general_classification in a general_classification_hierarchy that is included in the super class.

5.2.55.2 super_classification

The super_classification specifies the higher level of general_classification in a general_classification_hierarchy that includes the sub class.

5.2.56 general_classification

A general_classification is a classification of an object which characterizes all objects of the same kind; such a classification is independent from the application of the classified object.

EXAMPLE A fastener with subclasses, such as fastener with 2 or 3 fixture holes, a bracket with subclasses, e.g., with 90 or 100 degrees bending angle, or screws with subclasses such as metal screws or machine screws are examples for general_classification.

EXAMPLE For facilitating an effective part retrieval for slotted screws, several general_classification objects may be defined to classify screws.

EXPRESS specification:

```
  ENTITY general_classification;
    classification_source      :  OPTIONAL classification_source_select;
    description                :  OPTIONAL string_select;
    id                         :  STRING;
    used_classification_system :  OPTIONAL classification_system;
    version_id                 :  OPTIONAL STRING;
  END_ENTITY;
```

5.2.56.1 classification_source

The classification_source specifies the external_library_reference or the plib_class_reference that contains the specification of the general_classification.

The classification_source need not be specified for a particular general_classification.

If present, there shall be exactly one object that defines the classification_source for a general_classification.

5.2.56.2 description

The description specifies additional information about the general_classification.

EXAMPLE 'fastener with 2 fixture holes' or 'bracket with 90 degree bending angle' are examples for description.

The description need not be specified for a particular general_classification.

If present, there shall be exactly one object that defines the description for a general_classification.

5.2.56.3 id

The id specifies the identifier of the general_classification.

5.2.56.4 used_classification_system

The used_classification_system specifies the classification_system that contains the information about the definition of the classification and how to interpret the name of the general_classification.

The used_classification_system need not be specified for a particular general_classification.

5.2.56.5 version_id

The version_id specifies the identification of a particular version of the general_classification.

The version_id need not be specified for a particular general_classification.

5.2.57 geometric_model_relationship_with_transformation

A geometric_model_relationship_with_transformation is a relationship between two model objects with the additional information about a geometric transformation. This transformation defines the location and orientation of the related model relative to the relating model.

EXPRESS specification:

```
  ENTITY geometric_model_relationship_with_transformation;
    description       :  OPTIONAL string_select;
    model_placement   :  transformation;
    related           :  external_model;
    relating          :  external_model;
    relation_type     :  STRING;
  END_ENTITY;
```

5.2.57.1 description

The description specifies additional information about the geometric_model_relationship_with_transformation.

The description need not be specified for a particular geometric_model_relationship_with_transformation.

If present, there shall be exactly one object that defines the description for a geometric_model_relationship_with_transformation.

5.2.57.2 model_placement

The model_placement specifies the geometric transformation that places and orients the related model relative to the relating model.

5.2.57.3 related

The related specifies the second of the two model objects related by the geometric_model_relationship_with_transformation.

NOTE The semantics of this attribute are defined by the attribute relation_type.

There shall be exactly one object that the geometric_model_relationship_with_transformation is related to.

5.2.57.4 relating

The relating specifies the first of the two model objects related by the geometric_model_relationship_with_transformation.

NOTE The semantics of this attribute are defined by the attribute relation_type.

There shall be exactly one object that the geometric_model_relationship_with_transformation is relating to.

5.2.57.5 relation_type

The relation_type specifies the meaning of the relationship.

EXAMPLE The relation_type may be, e.g., 'basis for' if the relating geometric_model provides a detailed representation of the shape of an object and if the related geometric_model provides an idealized representation of the shape of the same object.

5.2.58 grade

EXPRESS specification:

```
ENTITY grade;
      coating              : OPTIONAL coating;
      cutting_condition    : SET [1:?] OF cutting_condition;
      identifier           : STRING;
      standard_designation : OPTIONAL STRING;
      substrate            : substrate;
      workpiece_material   : SET [1:?] OF material_designation;
END_ENTITY;
```

5.2.59 grade_relationship

EXPRESS specification:

```
ENTITY grade_relationship;
      related       : grade;
      relating      : grade;
      relation_type : STRING;
END_ENTITY;
```

5.2.59.1 related

5.2.59.2 relating

5.2.59.3 relation_type

5.2.60 item

An item is either a single object or a unit in a group of objects. It collects the information that is common to all versions of the object. An item shall always be classified as 'cutting item', 'tool item', 'adaptive item', 'assembly item', or 'accessory item' using a specific_item_classification. Additionally, if an assembly_definition exists for at least one version of the item, the item shall be classified as being an 'assembly' using specific_item_classification.

NOTE An item may be either a single piece part or an assembly of arbitrary complexity.

EXAMPLE In the context of cutting tools, an item may be the cutting tool as a whole, the assembly of the tool body, an insert, a shim, a screw, an adaptor, or a solid drill.

EXPRESS specification:

```
ENTITY item;
  description :  OPTIONAL string_select;
  id   :  STRING;
  name :  string_select;
INVERSE
  associated_version   :  SET [1:?] OF item_version FOR associated_item;
  item_classification  :  SET [1:?] OF specific_item_classification FOR
  associated_item;
END_ENTITY;
```

5.2.60.1 description

The description specifies additional information about the item.

The description need not be specified for a particular item.

If present, there shall be exactly one object that defines the description for an item.

5.2.60.2 id

The id specifies the identifier of the item. For the id, an owner shall be specified by a person_organization_assignment with role 'id owner'. The id shall be unique within the scope of the organization that is specified by the person_organization_assignment with the role 'id owner'.

NOTE The scope of uniqueness is usually dependent on the form of implementation; it may be a physical file or a data base.

5.2.60.3 name

The name specifies the word or group of words used to refer to the item.

There shall be exactly one object that defines the name for an item.

5.2.60.4 associated_version

The associated_item specifies the item with which the item_version is associated.

5.2.61 item_characteristic_association

An item_charactersitic_association associates a characteristic to an item_definition.

```
ENTITY item_characteristic_association;
  associated_characteristic  :  item_characteristic_select;
  associated_item            :  item_definition;
END_ENTITY;
```

5.2.61.1 associated_characteristic

Specifies the associated characteristic that in turn characterizes the item_definition.

5.2.61.2 associated_item

Specifies the associated item_definition.

5.2.62 item_definition

An item_definition is a view of an item_version. This view is relevant for the requirements of one or more application domains and collects product data of the item_version.

NOTE The selection of data describing an item_version may be different for assembly purposes, shipping purposes or analysis purposes.

Each item_definition may be a mating_definition, an assembly_definition, or a physical_item_definition.

EXPRESS specification:

```
ENTITY item_definition
    SUPERTYPE OF (ONEOF(assembly_definition, mating_definition,
  physical_item_definition));
     additional_context      : SET OF application_context;
     associated_item_version : item_version;
     id                      : STRING;
     initial_context         : application_context;
     name                    : OPTIONAL string_select;
  END_ENTITY;
```

5.2.62.1 additional_context

The additional_context specifies the set of application_context objects in which this view of the item_version is also relevant. The additional_context shall not contain the application_context that is referenced as the 'initial_context'.

5.2.62.2 associated_item_version

The associated_item_version specifies the item_version for which the item_definition is a view.

5.2.62.3 id

The id specifies the identifier of the item_definition.

5.2.62.4 initial_context

The initial_context specifies the application_context in which this view of the item_version has been designed primarily.

5.2.62.5 name

The name specifies the word or group of words used to refer to the item_definition.

The name need not be specified for a particular item_definition.

If present, there shall be exactly one object that defines the name for a item_definition.

5.2.63 item_definition_relationship

An item_definition_relationship is a relationship between two item_definition objects whose meaning is defined by the attribute 'relation_type'.

EXPRESS specification:

```
  ENTITY item_definition_relationship
     description   : OPTIONAL string_select;
     related       : item_definition;
     relating      : item_definition;
     relation_type : STRING;
  END_ENTITY;
```

5.2.63.1 description

The description specifies additional information about the item_definition_relationship.

The description need not be specified for a particular item_definition_relationship.

If present, there shall be exactly one object that defines the description for a item_definition_relationship.

5.2.63.2 related

The related specifies the second of the item_definition objects that are part of the relationship.

5.2.63.3 relating

The relating specifies the first of the item_definition objects that are part of the relationship.

5.2.63.4 relation_type

The relation_type specifies the meaning of the relationship.

5.2.64 item_instance

An item_instance is the occurrence of an object in a product structure that is defined by an item_definition.

An item_instance shall be used, at least once, in an assembly_association.

EXAMPLE In the case of an indexable milling cutter, the item 'insert' is defined once. Its item_definition carries all the information necessary to define the insert (e.g., its dimensions) independent of its usage. Additionally, there are 3 item_instance objects for this insert since there are 3 equal inserts used in this milling cutter. Each of these instances may carry additional information such as placement or function.

EXPRESS specification:

```
  ENTITY item_instance;
    definition     :  item_definition;
```

```
     description    :  OPTIONAL string_select;
     id             :  STRING;
  END_ENTITY;
```

5.2.64.1 definition

The definition specifies the item_definition that serves as a definition for this particular occurrence.

There shall be exactly one object that defines the definition for an item_instance.

5.2.64.2 description

The description specifies additional information about the item_instance.

The description need not be specified for a particular item_instance.

If present, there shall be exactly one object that defines the description for a item_instance.

5.2.64.3 id

The id specifies the identifier of the item_instance.

The id for an item_instance must only be unique within the context of the defining item.

5.2.65 item_structure_association

An item_structure_association is a relationship between a item_definition and an item_instance.

EXPRESS specification:

```
   ENTITY item_structure_association
      ABSTRACT SUPERTYPE OF (ONEOF(mating_association,
    assembly_association));
        placement : OPTIONAL
    geometric_model_relationship_with_transformation;
        related   : item_instance;
        relating  : item_definition;
   END_ENTITY;
```

5.2.65.1 placement

The placement specifies the geometric_model_relationship_with_transformation that specify the transformation information which is used to locate and orient the constituent in the coordinate space of the assembly_definition.

The placement need not be specified for a particular item_structure_relationship.

If present, there shall be exactly one object that defines the placement for an item_structure_relationship.

5.2.65.2 related

The related specifies the item_instance that is part of the item_structure_association.

NOTE The semantics of this attribute are defined by the specializations of item_structure_association.

5.2.65.3 relating

The relating specifies the item_definition that is part of the item_structure_association.

NOTE The semantics of this attribute are defined by the specializations of item_structure_association.

5.2.66 item_version

An item_version is a version of an item and serves as the collector of the data characterizing a physically realizable object in various application contexts.

NOTE An item_version may be produced, consumed, used to produce other item_version objects, or offered to the market.

NOTE The collection of defining information may be incomplete, i.e., not all of the item_definition objects needed to define an item_version are associated with the item_version.

NOTE The set of item_version objects of an item represents the history of the item within a particular life cycle stage or over its complete life cycle.

NOTE An item_version does not have to be referenced by any item_definition.

NOTE In the case, where two item_version objects are related to each other by an item_version_relationship with a relation_type of value 'supplied part', the relating item_version is the version carrying the necessary information, e.g., shape and properties, and the related is the version either supplied by or supplied for another company.

EXPRESS specification:

```
ENTITY item_version;
  associated_item   :  item;
  description       :  OPTIONAL string_select;
  id                :  STRING;
END_ENTITY;
```

5.2.66.1 associated_item

The associated_item specifies the item with which the item_version is associated.

5.2.66.2 description

The description specifies additional information about the item_version.

The description need not be specified for a particular item_version.

If present, there shall be exactly one object that defines the description for an item_version.

5.2.66.3 id

The id specifies the identifier of the item_version. The id shall be unique within the scope of the associated item.

5.2.67 item_version_relationship

An item_version_relationship is a relationship between two item_version objects.

EXPRESS specification:

```
ENTITY item_version_relationship;
```

```
    description    :  OPTIONAL string_select;
    related        :  item_version;
    relating       :  item_version;
    relation_type  :  STRING;
  END_ENTITY;
```

5.2.67.1 description

The description specifies additional information about the item_version_relationship.

The description need not be specified for a particular item_version_relationship.

If present, there shall be exactly one object that defines the description for a item_version_relationship.

5.2.67.2 related

The related specifies the second of the two item_version objects related by the item_version_relationship.

NOTE The semantics of this attribute are defined by the attribute relation_type.

5.2.67.3 relating

The relating specifies the first of the two item_version objects related by the item_version_relationship.

NOTE The semantics of this attribute are defined by the attribute relation_type.

5.2.67.4 relation_type

The relation_type specifies the meaning of the relationship.

Where applicable the following values shall be used:

'derivation': The application object defines a deriving relationship where the related item_version is based on the relating item_version which is an earlier version of the same or of a different item;
'hierarchy': The application object defines a hierarchical relationship where the related item_version is a subordinate version of the relating item_version;

EXAMPLE 'Rev. 1.1' and 'rev. 1.2' are subordinates of 'version 1'.

'sequence': The application object defines a version sequence where the relating item_version is the preceding version of the related item_version that is the following version. For a given item_version there shall be at most one item_version_relationship of this relation_type referring to this item_version as 'relating' and at most one item_version_relationship of this relation_type referring as 'related';
'supplied item': The application object defines a relationship between two item_version objects representing the same object in different organizational contexts.

NOTE The different organizational contexts can be represented by different general organizational data associations.

NOTE The concept of alias_identification may also be used for item_version. However, item_version_relationship shall be used if the related item_version corresponds to an item at the supplier's site and if a distinction of additional information such as name or organizational information is needed.

EXAMPLE Usually, items that are provided by suppliers have at the supplier's site an id different from the one at the contractor's site: an item that is identified in a company by the identifier 'C425' and the version id 'V2.0' may correspond to the item that the supplier identifies as '2X45' with the version id 'V3.2'.

5.2.68 language

A language is a specification of the language in which information is given.

EXPRESS specification:

```
  ENTITY language;
    country_code   :  OPTIONAL STRING;
    language_code  :  STRING;
  END_ENTITY;
```

5.2.68.1 country_code

The country_code specifies the country, as addition to the language, according to ISO 3166-1.

EXAMPLE Possible values for country_code are, e.g., 'GBR' for the United Kingdom or 'USA' for the United States of America.

5.2.68.2 language_code

The language_code specifies the language of the text information in the Alpha-3 code specified in ISO 639-2.

EXAMPLE Possible values for language_code are, e.g., 'eng' for English, 'fra' for French, 'rus' for Russian, or 'deu' for German.

5.2.69 limits_and_fits

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

EXPRESS specification:

```
  ENTITY limits_and_fits;
    deviation      :  STRING;
    fitting_type   :  OPTIONAL STRING;
    grade          :  STRING;
  END_ENTITY;
```

5.2.69.1 deviation

The deviation specifies the enumeration of limits and fits class descriptors by characters.

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

5.2.69.2 fitting_type

The fitting_type specifies whether the tolerance declaration applies to a shaft or to a hole.

The fitting_type need not be specified for a particular limits_and_fits.

5.2.69.3 grade

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

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

5.2.70 location

EXPRESS specification:

```
ENTITY location;
  location_id    :  STRING;
  location_name  :  OPTIONAL STRING;
  location_type  :  STRING;
END_ENTITY;
```

5.2.71 mated_item_relationship

A mated_item_relationship is a relationship between two mating_association objects. This relationship specifies additional information about the mating of two particular items that go into a mating_definition. The two mating_association objects that are referenced by the mated_item_relationship shall refer to the same mating_definition.

EXPRESS specification:

```
ENTITY mated_item_relationship;
  mating_material   :  SET OF quantified_instance;
  related           :  mating_association;
  relating          :  mating_association;
END_ENTITY;
```

5.2.71.1 mating_material

The mating_material specifies the set of quantified_instance objects used as material for the mating.

5.2.71.2 related

The related specifies the second of the two mating_association objects related by the mated_item_relationship.

5.2.71.3 relating

The relating specifies the first of the two mating_association objects related by the mated_item_relationship.

5.2.72 material_designation

A material_designation is an identifier for the substance out of which an item is made.

EXPRESS specification:

```
ENTITY material_designation;
  material_name  :  STRING;
END_ENTITY;
```

5.2.72.1 material_name

The material_name specifies the word or group of words by which the material_designation is referred to.

5.2.73 mating_association

A mated_item_association is a relationship between a mating_definition and an item_instance that is used within the mating_definition.

A mating_association is a type of item_structure association.

EXPRESS specification:

```
ENTITY mating_association
  SUBTYPE OF(item_structure_association);
  SELF\item_structure_association.relating   :  mating_definition;
END_ENTITY;
```

5.2.73.1 relating

The relating specifies the mating_definition that is the result of the mating operation.

5.2.74 mating_definition

A mating_definition is a view of an item_version, defining the physical connection of two or more item_instance objects. It includes technical information about the kind of connection. This information is independent from the hierarchical assembly structure.

A mating_definition is a type of item_definition.

EXPRESS specification:

```
ENTITY mating_definition
  SUBTYPE OF(item_definition);
  mating_type    :  STRING;
INVERSE
  mated_items    :  SET [2:?] OF mating_association FOR relating;
END_ENTITY;
```

5.2.74.1 mating_type

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

Where applicable the following values shall be used:

'bolted joint': Attachment of two or more items that are screwed together using screws, bolt nuts, and washers;
'brazing': Attachment of two or more items established by soldering or brazing;
'clasping': Attachment of two or more items established by using a clamp at one common outer border;
'doweling': Attachment of two or more items using a pin that is fit into each of each part by pressing;
'flanging': Attachment of two or more items by crimping a projected length of one part over the edge of the other part;
'press fit': Attachment of two or more items obtained by forces resulting from the selected size tolerances;
'welding': Attachment of two or more items by welding together with one weld bead.

5.2.75 multi_language_string

A multi_language_string represents text information, expressed in one or more languages, that is associated with objects.

EXPRESS specification:

```
ENTITY multi_language_string;
  additional_language_string :  SET OF string_with_language;
  primary_language_string    :  string_with_language;
END_ENTITY;
```

5.2.75.1 additional_language_string

The additional_language_dependent_string specifies the string_with_language objects that represent the text information in a particular language.

5.2.75.2 primary_language_string

The primary_language_dependent_string specifies the string_with_language that represents the text information in the original language.

5.2.76 named_size

A named_size is the definition of the size of a document_file or of a document_representation where the size is specified by a standardized identifier.

A named_size is a type of rectangular_size.

EXPRESS specification:

```
ENTITY named_size
  SUBTYPE OF(rectangular_size);
  referenced_standard  :  OPTIONAL classification_system;
  size                 :  STRING;
END_ENTITY;
```

5.2.76.1 referenced_standard

The referenced_standard specifies a standard according to which the size is specified.

The referenced_standard need not be specified for a particular named_size.

5.2.76.2 size

The size specifies the size of the object. If the size differs from the dimensions specified by the inherited 'width' and 'height' attributes the size is overridden.

5.2.77 numerical_value

A numerical_value is a quantity expressed with a numerical value and a unit.

A numerical_value is a type of value_with_unit.

EXPRESS specification:

```
ENTITY numerical_value
  SUBTYPE OF(value_with_unit);
  value_component   :  STRING;
INVERSE
  limitation :  SET [0:1] OF value_limitation FOR limited_value;
END_ENTITY;
```

5.2.77.1 value_component

The value_component specifies the quantity of the numerical_value.

5.2.78 organization

An organization is a group of people involved in a particular business process.

EXPRESS specification:

```
ENTITY organization;
  delivery_address  :  OPTIONAL STRING;
  id                :  STRING;
  organization_name :  STRING;
  organization_type :  OPTIONAL STRING;
  postal_address    :  OPTIONAL STRING;
  visitor_address   :  OPTIONAL STRING;
END_ENTITY;
```

5.2.78.1 delivery_address

The delivery_address specifies the address where goods are delivered.

The delivery_address need not be specified for a particular organization.

5.2.78.2 id

The id specifies the identifier of the organization.

NOTE The assignment of this attribute is usually controlled by a registration authority. The registration authority may be a public organization that assigns identifiers to corporations, or it may be the parent corporation that assigns component identifiers to its components.

EXAMPLE The id may be the code assigned to the organization for a stock market listing, or it may be a department number.

5.2.78.3 organization_name

The organization_name specifies the word or group of words used to refer to the organization.

5.2.78.4 organization_type

The organization_type specifies the type of the organization.

Where applicable the following values shall be used:

'company':	The organization_type specifies that the organization is a company;
'department':	The organization_type specifies that the organization is a department;
'plant':	The organization_type specifies that the organization is a plant.

The organization_type need not be specified for a particular organization.

5.2.78.5 postal_address

The postal_address specifies the address where letter mail is delivered.

The postal_address need not be specified for a particular organization.

5.2.78.6 visitor_address

The visitor_address specifies the address where the organization receives visitors.

The visitor_address need not be specified for a particular organization.

5.2.79 person

A person is an individual human being who has some relationship to product data. The person shall always be identified in the context of one or more organizations.

EXPRESS specification:

```
ENTITY person;
  person_name :  STRING;
  preferred_business_address :  OPTIONAL STRING;
INVERSE
  associated_organization    :  SET [1:?] OF person_in_organization FOR
                                associated_person;
END_ENTITY;
```

5.2.79.1 person_name

The person_name specifies the word or group of words used to refer to the person.

NOTE The person_name includes the first, middle, and last names as well as titles, if applicable.

5.2.79.2 preferred_business_address

The preferred_business_address specifies the location of the office of the person.

The preferred_business_address need not be specified for a particular person.

5.2.80 person_in_organization

A person_in_organization is the specification of a person in the context of an organization.

EXPRESS specification:

```
ENTITY person_in_organization;
  associated_organization :  organization;
  associated_person       :  person;
  id                      :  OPTIONAL STRING;
  location                :  OPTIONAL STRING;
  role                    :  STRING;
END_ENTITY;
```

5.2.80.1 associated_organization

The associated_organization specifies the organization with which the person is associated.

5.2.80.2 associated_person

The associated_person specifies the person.

5.2.80.3 id

The id specifies an identifier of the person. The identifier shall be unique within the scope of the 'associated_organization'.

EXAMPLE The id may be a staff number or a user id in a computer system.

The id need not be specified for a particular person_in_organization.

5.2.80.4 location

The location specifies the relevant address of the person_in_organization.

The location need not be specified for a particular person_in_organization.

5.2.80.5 role

The role specifies the relationship between the person and the organization.

5.2.81 person_organization_assignment

A person_organization_assignment is an object that associates an organization or a person_in_organization with product data.

This assignment provides additional information for the associated object. The provision of such data through this assignment has an organizational character whereas some objects require the same kind of data mandatorily in order to be semantically complete. This assignment shall not be used to associate the corresponding organizational data with an object whose attributes are referencing the organizational data directly.

The person_organization_assignment shall not be used for the purpose of specifying a relationship between two person_in_organization objects, e.g., a succession in organizational responsibility.

EXPRESS specification:

```
ENTITY person_organization_assignment;
  assigned_person_organization  :  person_organization_select;
  description                   :  OPTIONAL string_select;
  is_applied_to                 :  SET [1:?] OF
                                   general_organizational_data_select;
  role                          :  STRING;
END_ENTITY;
```

5.2.81.1 assigned_person_organization

The assigned_person_organization specifies the concerned organization or the person_in_organization.

There shall be exactly one object that defines the assigned_person_organization for a person_organization_assignment.

5.2.81.2 description

The description specifies additional information about the person_organization_assignment.

The description need not be specified for a particular person_organization_assignment.

If present, there shall be exactly one object that defines the description for a person_organization_assignment.

5.2.81.3 is_applied_to

The is_applied_to specifies the object with which the person_organization_assignment is associated.

There shall be at least one object that the person_organization_assignment is_applied_to.

5.2.81.4 role

The role specifies the responsibility of the assigned person or organization with respect to the object that it is applied to.

Where applicable the following values shall be used:

'author': The referenced object has been created by the assigned person or organization. The author holds the copyright;
'classification officer': The assigned person or organization is formally responsible for the classification of the referenced object;
'creator': The referenced object has been created by the assigned person or organization;
'custodian': The assigned person or organization is responsible for the existence and integrity of the referenced object;
'customer': The assigned person or organization acts as a purchaser or consumer of the referenced object;
'design supplier': The assigned person or organization is the one who delivers the data describing the referenced object;

EXAMPLE The 'design supplier' may be a design by means of a 3D model or a drawing.

'editor': The assigned person or organization is responsible for making any changes to any attribute of the referenced object;
'id owner': The assigned person or organization is the one responsible for the designation of an identifier;
'location': The assigned organization is the place where the referenced object can be found or where it takes place;
'owner': The assigned person or organization owns the referenced object, and has final say over its disposition and any changes to it;
'supplier': The assigned person or organization is the one who delivers the actual (physical) object (e.g., a dealer);

5.2.82 physical_document

A physical_document is a piece of product data that is archived in a non-digital form.

EXAMPLE Paper plots of technical drawings, micro fiche, or paper documents such as calculations or test reports are examples for a physical_document.

A physical_document is a type of document_representation.

EXPRESS specification:

```
ENTITY physical_document
  SUBTYPE OF(document_representation);
  file :  SET OF physical_file;
END_ENTITY;
```

5.2.82.1 file

The file specifies the physical realization of the physical_document.

5.2.83 physical_file

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

A physical_file is a type of document_file.

EXPRESS specification:

```
ENTITY physical_file
  SUBTYPE OF(document_file);
END_ENTITY;
```

5.2.84 physical_item

EXPRESS specification:

```
ENTITY physical_item
  SUBTYPE OF(item);
INVERSE
  SELF\item.associated_version   :  SET [1:?] OF physical_item_version FOR
                                    associated_item;
END_ENTITY;
```

5.2.85 physical_item_location_association

EXPRESS specification:

```
ENTITY physical_item_location_association;
  located_item   :  physical_item_definition;
  location       :  location;
END_ENTITY;
```

5.2.86 physical_item_definition

EXPRESS specification:

```
ENTITY physical_item_definition
  SUBTYPE OF(item_definition);
  SELF\item_definition.associated_item_version   :  physical_item_version;
END_ENTITY;
```

5.2.87 physical_item_state_association

EXPRESS specification:

```
ENTITY physical_item_state_association;
  associated_physical_item   :  physical_item_definition;
  associated_state           :  state;
  role                       :  STRING;
END_ENTITY;
```

5.2.88 physical_item_structure_association

EXPRESS specification:

```
ENTITY physical_item_structure_association;
  related    :  physical_item_definition;
  relating   :  physical_item_definition;
END_ENTITY;
```

5.2.89 physical_item_version

EXPRESS specification:

```
ENTITY physical_item_version
  SUBTYPE OF(item_version);
  SELF\item_version.associated_item :  physical_item;
END_ENTITY;
```

5.2.90 plib_class_reference

A plib_class_reference designates a class in a library compliant to ISO 13584 (Parts Library).

EXPRESS specification:

```
ENTITY plib_class_reference;
  code          :  STRING;
  supplier_bsu  :  STRING;
  version       :  STRING;
END_ENTITY;
```

5.2.90.1 code

The code specifies the class in the PLIB library. The format of this code is defined in ISO 13584-42.

5.2.90.2 supplier_bsu

The supplier_bsu (basic semantic unit) specifies the supplier of the class in a PLIB library, in which the class is defined. The format of this specification is defined in ISO 13584-26.

5.2.90.3 version

The version specifies the identification of a particular version of a class in a PLIB library. The format of this version is defined in ISO 13584-42.

5.2.91 plib_property_reference

A plib_property_reference designates a property in a library compliant to ISO 13584.

EXPRESS specification:

```
ENTITY plib_property_reference;
  code        :  STRING;
  name_scope  :  plib_class_reference;
  version     :  STRING;
END_ENTITY;
```

5.2.91.1 code

The code specifies the property in the PLIB library. The format of this code is defined in ISO 13584-42.

5.2.91.2 name_scope

The name_scope specifies the plib_class_reference in which the property is visible.

5.2.91.3 version

The version specifies the identification of a particular version of a property in a PLIB library. The format of this version is defined in ISO 13584-42.

5.2.92 plus_minus_bounds

A plus_minus_bounds is the specification of the allowable deviation from a numerical value.

EXPRESS specification:

```
ENTITY plus_minus_bounds;
  lower_bound          :  STRING;
  significant_digits   :  OPTIONAL STRING;
  upper_bound          :  STRING;
  value_determination  :  OPTIONAL STRING;
END_ENTITY;
```

5.2.92.1 lower_bound

The lower_bound specifies the value of the tolerance that shall be subtracted from the exact value to establish the minimum allowed value.

5.2.92.2 significant_digits

The significant_digits specifies the number of decimal digits indicating the accuracy of the lower_bound and upper_bound values.

The significant_digits need not be specified for a particular plus_minus_bounds.

5.2.92.3 upper_bound

The upper_bound specifies the value of the tolerance that shall be added to the exact value to establish the maximum allowed value.

5.2.92.4 value_determination

The value_determination specifies information on how the plus_minus_bounds shall be interpreted.

Where applicable the following values shall be used:

'calculated':	The value has been calculated;
'designed':	The value represents a value intended by the design;
'estimated':	The value has been estimated;
'measured':	The value has been measured;
'required':	The value represents a requirement.

NOTE A plus_minus_bounds may be specified in the design stage of a product but it may also be documented as measured on a prototype.

The value_determination need not be specified for a particular plus_minus_bounds.

5.2.93 property

A property is the definition of a particular quality.

NOTE A property may reflect physics or arbitrary user defined measurements.

EXPRESS specification:

```
ENTITY property;
  allowed_unit       :  SET OF unit;
  description        :  OPTIONAL string_select;
  id                 :  STRING;
  property_source    :  OPTIONAL property_source_select;
  property_type      :  STRING;
  version_id         :  OPTIONAL STRING;
END_ENTITY;
```

5.2.93.1 allowed_unit

The allowed_unit specifies the unit or set of units that are accepted.

EXAMPLE A company may accept a mass specified in kilograms or tons, but not in grams or pounds.

5.2.93.2 description

The description specifies additional information about the property.

The description need not be specified for a particular property.

If present, there shall be exactly one object that defines the description for a property.

5.2.93.3 id

The id specifies the identifier of the property.

5.2.93.4 property_source

The property_source specifies the external_library_reference or plib_property_reference object that defines this kind of property.

The property_source need not be specified for a particular property.

If present, there shall be exactly one object that defines the property_source for a property.

5.2.93.5 property_type

The property_type specifies the kind of property the property defines.

Where applicable the following values shall be used:

'cost':	a property that specifies costs.
'dimension':	a property that describes a dimension of an object.
'mass':	a quantity of matter that an object consists of.
'shape':	a property that describes the shape of an object.
'specification':	a property that specifies a certain characteristic of an object.
'usage':	a property that describes the use of an object.

5.2.93.6 version_id

The version_id specifies the identification of a particular version of a property.

The version_id need not be specified for a particular property.

5.2.94 property_relationship

A property_relationship is a relationship between two property objects.

EXAMPLE A property may reflect physics or arbitrary user defined measurements.

EXPRESS specification:

```
  ENTITY property_relationship;
    description   :  OPTIONAL string_select;
    related       :  property;
    relating      :  property;
    relation_type :  STRING;
  END_ENTITY;
```

5.2.94.1 description

The description specifies additional information about the property_relationship.

The description need not be specified for a particular property_relationship.

If present, there shall be exactly one object that defines the description for a property_relationship.

5.2.94.2 related

The related specifies the second of the two property objects related by the property_relationship.

NOTE The semantics of this attribute are defined by the attribute relation_type.

5.2.94.3 relating

The relating specifies the first of the two property objects related by the property_relationship.

NOTE The semantics of this attribute are defined by the attribute relation_type.

5.2.94.4 relation_type

The relation_type specifies the meaning of the relationship.

Where applicable the following values shall be used:

'decomposition': The property_relationship defines a relationship where the related property is a member of a group of property objects that is established by the relating property;
'dependency': The related property is dependent upon the relating property;
'hierarchy': The application object defines a hierarchical relationship where the related property is on a lower level than the relating property;
'peer': The related property shall not be used without the relating property and vice versa;
'substitution': The property_relationship defines a relationship where the related property replaces the relating property;
'value domain': The property_relationship defines a relationship where the values assigned to the relating property shall be within the limits indicated by the values assigned to the related property.

5.2.95 property_value

A property_value is the numerical or textual value of a property.

Each property_value is a value_list, a string_value, or a value_with_unit.

EXPRESS specification:

```
ENTITY property_value
    ABSTRACT SUPERTYPE OF (ONEOF(value_with_unit, string_value,
  value_list));
      value_name : STRING;
END_ENTITY;
```

5.2.95.1 value_name

The value_name specifies the word or group of words by which the property_value is referred to.

EXAMPLE 'l1' or 'vol2' are examples for the value_name of a property_value.

5.2.96 property_value_association

A property_value_association is a mechanism to assign a property_value_representation to an object.

EXPRESS specification:

```
ENTITY property_value_association;
  definitional                 :  OPTIONAL BOOLEAN;
  described_element            :  item_property_select;
  describing_property_value    :  property_value_representation;
  description                  :  OPTIONAL string_select;
  validity_context             :  OPTIONAL organization;
END_ENTITY;
```

5.2.96.1 definitional

The definitional specifies whether the associated property_value_representation object may be used to distinguish the described_element from others of the same kind. A value of 'true' indicates that the associated property_value_representation distinguishes it from others.

The definitional is only meaningful, if the described_element is of type item_definition.

The definitional need not be specified for a particular property_value_association.

5.2.96.2 described_element

The described_element specifies the object that is characterized by the property_value.

There shall be exactly one object that defines the described_element for an property_value_association.

5.2.96.3 describing_property_value

The describing_property_value specifies the value that is being assigned.

5.2.96.4 description

The description specifies additional information about the property_value_association.

The description need not be specified for a particular property_value_association.

If present, there shall be exactly one object that defines the description for a property_value_association.

5.2.96.5 validity_context

The validity_context specifies the context in which a property_value_association is applicable.

EXAMPLE A property with property_type 'cost' is applicable in the context of supplier A and may be different if the item is supplied by a different supplier B.

The validity_context need not be specified for a particular property_value_association.

If present, there shall be exactly one object that defines the validity_context for a property_value_association.

5.2.97 property_value_representation

A property_value_representation is the representation of property.

EXPRESS specification:

```
ENTITY property_value_representation;
  definition           :  property;
  global_unit          :  OPTIONAL unit;
  qualifier            :  OPTIONAL STRING;
  specified_value      :  property_value;
  value_determination  :  OPTIONAL STRING;
END_ENTITY;
```

5.2.97.1 definition

The definition specifies the property that the property_value_representation characterizes.

5.2.97.2 global_unit

The global_unit specifies a unit that is valid for all property_value that are referenced as 'specified_value' by the property_value_representation.

The global_unit need not be specified for a particular property_value_representation.

5.2.97.3 qualifier

The qualifier specifies the kind of the property_value_representation.

The following values shall be used:

'nominal':	The value is the nominal value;
'specified':	The value is specified;
'typical':	The value is a typical value.

The qualifier need not be specified for a particular property_value_representation.

5.2.97.4 specified_value

The specified_value specifies the property_value that qualifies the property_value_representation by a value_with_unit, a string_value, or an arbitrary aggregate thereof.

5.2.97.5 value_determination

The value_determination specifies information on how the property_value_representation shall be interpreted.

Where applicable the following values shall be used:

'calculated':	The value has been calculated;
'designed':	The value represents a value intended by the design;
'estimated':	The value has been estimated;
'measured':	The value has been measured;
'required':	The value represents a requirement;
'set point':	The value is used as the initialization value.

The value_determination need not be specified for a particular property_value_representation.

5.2.98 property_value_representation_relationship

A property_value_representation_relationship is a relationship between two property_value_representation objects.

EXPRESS specification:

```
ENTITY property_value_representation_relationship;
  description    :  OPTIONAL string_select;
  related        :  property_value_representation;
  relating       :  property_value_representation;
  relation_type  :  STRING;
END_ENTITY;
```

5.2.98.1 description

The description specifies additional information about the property_value_representation_relationship.

The description need not be specified for a particular property_value_representation_relationship.

If present, there shall be exactly one object that defines the description for a property_value_representation_relationship.

5.2.98.2 related

The related specifies the second of the two objects related by the property_value_representation_relationship.

NOTE The semantics of this attribute is defined by the attribute relation_type.

5.2.98.3 relating

The relating specifies the first of the two objects related by the property_value_representation_relationship.

NOTE The semantics of this attribute is defined by the attribute relation_type.

5.2.98.4 relation_type

The relation_type specifies the meaning of the relationship.

EXAMPLE property_value_representation objects representing 'cycle time', 'feed time', and 'drill time' for a process may be related by a property_value_representation_relationship with relation_type 'dependency'.

Where applicable the following values shall be used:

'decomposition': The property_value_representation_relationship specifies a relationship where the related property_value_representation is one of potentially more sub values of the relating property_value_representation;

'dependency': The property_value_representation_relationship defines a deriving relationship where the related property_value_representation is based on the relating property_value_representation;
'equivalence': The property_value_representation_relationship defines a relationship where the related property_value_representation represents the same matter of fact as by the relating property_value_representation;

EXAMPLE The distance of 1 km and 1000 m is the same. 1 km is equivalent to 1000 m, but they have different values and units.

'substitution': The property_value_representation_relationship defines a relationship where the related property_value_representation replaces the relating property_value_representation.

5.2.99 quantified_instance

A quantified_instance is the identification of the quantified occurrence of an object that is defined as an item_definition.

EXPRESS specification:

```
ENTITY quantified_instance
  SUBTYPE OF(item_instance);
  quantity   :  numerical_value;
END_ENTITY;
```

5.2.99.1 quantity

The quantity specifies a numerical_value specifying the quantity of occurrences.

5.2.100 realized_item_association

EXPRESS specification:

```
ENTITY realized_item_association;
  physical_item :  physical_item;
  realized_item :  item;
END_ENTITY;
```

5.2.101 rectangular_size

A rectangular_size is the definition of the planar size of an object.

Each rectangular_size may be a named_size.

EXPRESS specification:

```
ENTITY rectangular_size;
  density :  OPTIONAL value_with_unit;
  height  :  value_with_unit;
  width:  value_with_unit;
END_ENTITY;
```

5.2.101.1 density

The density specifies the resolution of the object if it is a raster picture.

The density need not be specified for a particular rectangular_size.

5.2.101.2 height

The height specifies the size of the object in vertical direction.

5.2.101.3 width

The width specifies the size of the object in horizontal direction.

5.2.102 specific_item_classification

A specific_item_classification is a classification of an item with respect to specific criteria. The specific criteria are covered in the 'classification_name' attribute.

NOTE If an item requires classification by more than one criterion, several specific_item_classification objects are associated to the same item.

NOTE For the attribute 'classification_name' a set of predefined values is specified hereafter. If values other than the proposed ones are used, they should be of general classifying nature. This kind of classification ought not to be used to classify names or identifiers of objects, e.g., in order to classify part families; for this purpose general_classification ought to be used.

EXPRESS specification:

```
ENTITY specific_item_classification;
  associated_item      :  SET [1:?] OF item;
  classification_name  :  STRING;
  description          :  OPTIONAL string_select;
END_ENTITY;
```

5.2.102.1 associated_item

The associated_item specifies the Item with which a particular specific_item_classification is associated.

5.2.102.2 classification_name

The classification_name provides high level classification information.

Where applicable the following values shall be used:

'accessory item': The item is of type accesory item.
'adaptive item': The item is of type adaptive item.
'assembly': This type of classification shall be used for any item that has an assembly_definition provided for at least one of its versions, i.e., it is decomposed further;
'assembly item': The item is of type assembly item.
'cutting item': The item is of type cutting item.
'cutting tool': The item is a complete cutting tool.
'detail': This type of classification shall be used for any item that has no assembly_definition provided for any of its versions, i.e., it is not further decomposed;
'tool item': The item is of type tool item.

5.2.102.3 description

The description specifies additional information about the specific_item_classification.

The description need not be specified for a particular specific_item_classification.

If present, there shall be exactly one object that defines the description for a specific_item_classification.

5.2.103 specific_item_classification_hierarchy

A specific_item_classification_hierarchy is used to build up hierachical structures of specific_item_classification.

EXPRESS specification:

```
ENTITY specific_item_classification_hierarchy;
  sub_classification   :  specific_item_classification;
  super_classification :  specific_item_classification;
END_ENTITY;
```

5.2.103.1 sub_classification

The sub_classification specifies the lower level of specific_item_classification in a specific_item_classification_hierarchy that is included in the super class.

5.2.103.2 super_classification

The super_classification specifies the higher level of specific_item_classification in a specific_item_classification_hierarchy that includes the sub class.

5.2.104 state

EXPRESS specification:

```
ENTITY state;
  name :  STRING;
END_ENTITY;
```

5.2.105 string_value

A string_value represents a sequence of one or more alphanumeric characters.

A string_value is a type of property_value.

EXPRESS specification:

```
ENTITY string_value
  SUBTYPE OF(property_value);
  value_specification  :  string_select;
END_ENTITY;
```

5.2.105.1 value_specification

The description specifies additional information about the string_value.

There shall be exactly one object that defines the description for a string_value.

5.2.106 string_with_language

A string_with_language represents text information in a specific language together with an identification of the language used.

EXPRESS specification:

```
ENTITY string_with_language;
```

```
    contents                :  STRING;
    language_specification  :  language;
  INVERSE
    used_by :  SET [1:?] OF multi_language_string FOR
    primary_language_string;
  END_ENTITY;
```

5.2.106.1 contents

The contents is textual information stored in the language identified by the language attribute.

5.2.106.2 language_specification

The language_specification specifies the language in which the contents is given.

5.2.107 substrate

EXPRESS specification:

```
  ENTITY substrate;
    name :  STRING;
  END_ENTITY;
```

5.2.108 transformation

A transformation is a geometric transformation composed of translation and rotation. Scaling is not included.

Each transformation is a transformation_3d or a transformation_2d.

EXPRESS specification:

```
  ENTITY transformation
      ABSTRACT SUPERTYPE OF (ONEOF(transformation_2d, transformation_3d));
  END_ENTITY;
```

5.2.109 transformation_2d

A transformation_2d is the definition of a geometric transformation in 2D space.

A transformation_2d is a type of transformation.

EXPRESS specification:

```
  ENTITY transformation_2d
    SUBTYPE OF(transformation);
  END_ENTITY;
```

5.2.110 transformation_3d

A transformation_3d is the definition of a geometric transformation in 3D space.

A transformation_3d is a type of transformation.

EXPRESS specification:

```
  ENTITY transformation_3d
    SUBTYPE OF(transformation);
```

```
END_ENTITY;
```

5.2.111 unit

A unit is a quantity chosen as a standard in terms of which other quantities may be expressed.

The types of units supported by this standard are SI units as well as derived or conversion based units as defined in ISO 10303-41.

EXPRESS specification:

```
ENTITY unit;
  unit_name   :  STRING;
END_ENTITY;
```

5.2.111.1 unit_name

The unit_name specifies the term representing the kind of unit.

EXAMPLE 'gram', 'litre', or 'volt' are examples for the unit_name.

5.2.112 value_limit

A value_limit is a qualified numerical value representing either the lower limit or the upper limit of a particular physical characteristic.

EXAMPLE '30.5 max' and '5 min' are examples for a value_limit.

A value_limit is a type of value_with_unit.

EXPRESS specification:

```
ENTITY value_limit
  SUBTYPE OF(value_with_unit);
  limit             :  STRING;
  limit_qualifier   :  STRING;
END_ENTITY;
```

5.2.112.1 limit

The limit specifies the value of the limit.

5.2.112.2 limit_qualifier

The limit_qualifier specifies the kind of limit.

The following values shall be used:

'maximum': The specified limit is an upper limit;
'minimum': The specified limit is a lower limit.

5.2.113 value_limitation

A value_limitation is a mechanism to add information about limitations to a numerical_value.

EXPRESS specification:

```
ENTITY value_limitation;
  envelope       :  OPTIONAL BOOLEAN;
  is_defined_by  :  limitation_definition_select;
  limited_value  :  numerical_value;
END_ENTITY;
```

5.2.113.1 envelope

The envelope specifies whether a geometric constraint resulting from the limitation has to be fulfilled in itself or not. A value of 'true' indicates that the envelope of the perfect shape corresponding to the maximum material shall not be larger than the specified dimension and tolerance.

NOTE The concept of the envelope principle is defined in ISO 8015 [19].

NOTE A dimension where the envelope principle applies, appears on drawings with an 'E' enclosed in a circle following the dimension value.

The envelope need not be specified for a particular value_limitation.

5.2.113.2 is_defined_by

The is_defined_by specifies the limitation that is associated with the numerical_value.

There shall be exactly one object that the value_limitation is_defined_by.

5.2.113.3 limited_value

The limited_value specifies the numerical_value that is limited.

5.2.114 value_list

A value_list is an ordered collection of property_value objects.

EXAMPLE A property may be composed of different property values such as 'mass', 'speed', and 'age' which are altogether necessary in a given context. The value_list collects all of them in a given order, such that each is identifiable by its index in the list.

A value_list is a type of property_value.

EXPRESS specification:

```
ENTITY value_list
  SUBTYPE OF(property_value);
  values  :  LIST [1:?] OF property_value;
END_ENTITY;
```

5.2.114.1 values

The values specifies the ordered collection of property_value objects that together are provided as a property_value.

5.2.115 value_range

A value_range is a pair of numerical values representing the range in which the value shall lie.

A value_range is a type of value_with_unit.

EXPRESS specification:

```
ENTITY value_range
  SUBTYPE OF(value_with_unit);
  lower_limit :  STRING;
  upper_limit :  STRING;
END_ENTITY;
```

5.2.115.1 lower_limit

The lower_limit specifies the minimum acceptable value that is constrained by the value_range.

5.2.115.2 upper_limit

The upper_limit specifies the maximum acceptable value that is constrained by the value_range.

5.2.116 value_with_unit

A value_with_unit is either a single numerical measure, or a range of numerical measures with upper, lower, or upper and lower bounds.

A value_with_unit is a type of property_value.

Each value_with_unit is a value_range, a numerical_value, or a value_limit.

EXPRESS specification:

```
ENTITY value_with_unit
    ABSTRACT SUPERTYPE OF (ONEOF(value_limit, numerical_value,
  value_range))
    SUBTYPE OF(property_value);
      significant_digits : OPTIONAL STRING;
      unit_component     : OPTIONAL unit;
END_ENTITY;
```

5.2.116.1.1 significant_digits

The significant_digits specifies the number of decimal digits that are relevant for the use of the value_with_unit. If present, the numerical measure or range may be specified using more digits than the significant digits but shall not be specified using less digits.

The significant_digits need not be specified for a particular value_with_unit.

5.2.116.2 unit_component

The unit_component specifies the unit in which the value_with_unit is expressed.

The unit_component need not be specified for a particular value_with_unit.

5.2.117 workpiece_feature

A workpiece_feature is a characteristic of the workpiece to be machined by a cutting tool.

EXPRESS specification:

```
ENTITY workpiece_feature;
  description    :  OPTIONAL string_select;
```

```
    id             :  STRING;
    representation :  SET [1:?] OF external_model;
  END_ENTITY;
```

5.2.117.1 description

The description specifies additional information about the characteristic.

The description need not be specified for a particular characteristic.

If present, there shall be exactly one object that defines the description for a characteristic.

6 Application interpreted model

6.1 Mapping specification

In the following, "Application element" designates any entity data type defined in Clause 5, any of its explicit attributes and any subtype constraint. "AIM element" designates any entity data type defined in 6.2 or imported with a USE FROM statement, from another EXPRESS schema, any of its attributes and any subtype constraint defined in 6.2 or imported with a USE FROM statement.

This clause contains the mapping specification that defines how each UoF and application element of this part of ISO 13399 (see Clause 5) maps to one or more AIM elements (see 6.2).

The mapping for each application element is specified in a separate subclause below. The mapping specification of an attribute of an ARM entity is a subclause of the clause that contains the mapping specification of this entity. Each mapping specification subclause contains up to five elements.

Title: The clause title contains:

- the name of the considered ARM entity or subtype constraint, or
- the name of the considered ARM entity atribute when this attribute refers to a type that is not an entity data type or a SELECT type that contains or may contain entity data types, or
- a composite expression, <attribute name> to <referred type>, when this attribute refers to a type that is not an entity data type or a SELECT type that contains or may contain entity data types.

AIM element: This section contains, depending on the considered application element:

- the name of one or more AIM entity data types;
- the name of a AIM entity attribute, presented with the syntax <entity name>.<attribute name>, when the considered ARM attribute refers to a type that is not an entity data type or a SELECT type that contains or may contain entity data types;
- the term PATH, when the considered ARM entity attribute refers to an entity data type or to a SELECT type that contains or may contain entity data types;
- the term IDENTICAL MAPPING, when both application objects involved in an application assertion map to the same instance of a MIM entity data type;
- the syntax /SUPERTYPE(<supertype name>)/, when the considered ARM entity is mapped as its supertype;
- one or more constructs /SUBTYPE(<subtype name>)/, when the mapping of the considered ARM entity is the union of the mapping of its subtypes.

When the mapping of an application element involves more than one AIM element, each of these AIM elements is presented on a separate line in the mapping specification, enclosed between parentheses or brackets.

Source: This section contains:

- the ISO standard number and part number in which the AIM element is defined, for those AIM elements that are defined in a common resource document;
- the ISO standard number and part number of this part of ISO 13399, for those AIM elements that are defined in the AIM schema of this part.

This section is omitted when the keywords PATH or IDENTICAL MAPPING are used in the AIM element section.

Rules: This section contains the name of one or more global rules that apply to the population of the AIM entity data types listed in the AIM element section or in the reference path. When no rule applies, this section is omitted.

A reference to a global rule may be followed by a reference to the subclause in which the rule is defined.

Constraint: This section contains the name of one or more subtype constraints that apply to the population of the AIM entity data types listed in the AIM element section or in the reference path. When no subtype constraint applies, this section is omitted.

A reference to a subtype constraint may be followed by a reference to the subclause in which the subtype constraint is defined.

Reference path: This section contains:

- the reference path to its supertypes in the common resources, for each AIM element created within this part of ISO 13399;
- the specification of the relationships between AIM elements, when the mapping of an application element requires to relate instances of several AIM entity data types. In such a case, each line in the reference path documents the role of an AIM element relative to the referring AIM element or to the next referred AIM element.

For the expression of reference paths and of the constraints between AIM elements, the following notational conventions apply:

[] enclosed section constrains multiple AIM elements or sections of the reference path are required to satisfy an information requirement;

() enclosed section constrains multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;

{} enclosed section constrains the reference path to satisfy an information requirement;

<> enclosed section constrains at one or more required reference path;

|| enclosed section constrains the supertype entity;

-> the attribute, whose name precedes the -> symbol, references the entity or select type whose name follows the -> symbol;

<- the entity or select type, whose name precedes the <- symbol, is referenced by the entity attribute whose name follows the <- symbol;

[i] the attribute, whose name precedes the [i] symbol, is an aggregate; any element of that aggregate is referred to;

[n] the attribute, whose name precedes the [n] symbol, is an ordered aggregate; member n of that aggregate is referred to;

=> the entity, whose name precedes the => symbol, is a supertype of the entity whose name follows the => symbol;

<= the entity, whose name precedes the <= symbol, is a subtype of the entity whose name follows the <= symbol;

= the string, select, or enumeration type is constrained to a choice or value;

\ the reference path expression continues on the next line;

* one or more instances of the relationship entity data type may be assembled in a relationship tree structure. The path between the relationship entity and the related entities, is enclosed with braces;

-- the text following is a comment or introduces a clause reference;

*> the select or enumeration type, whose name precedes the *> symbol, is extended into the select or enumeration type whose name follows the *> symbol;

<* the select or enumeration type, whose name precedes the <* symbol, is an extension of the select or enumeration type whose name follows the <* symbol.

6.1.1 classification UoF

6.1.1.1 classification_association

#1: If classification_association.definitional is TRUE.

#2: If classification_association.definitional is FALSE.

#3: If classification_association.definitional is not instantiated.

AIM element: applied_classification_assignment

Source: ISO 10303-214

Rules: restrict_applied_classification_assignment_role

Reference path:

```
applied_classification_assignment <=
classification_assignment
{classification_assignment.role ->
classification_role
classification_role.name
#1: (classification_role.name = 'definitional')
#2: (classification_role.name = 'non-definitional')
#3: (classification_role.name = '')}
```

6.1.1.1.1 definitional

#1: If classification_association.definitional is TRUE.

#2: If classification_association.definitional is FALSE.

#3: If classification_association.definitional is not instantiated.

AIM element: classification_role.name

Source: ISO 10303-41

Rules: restrict_applied_classification_assignment_role,
dependent_instantiable_classification_role

Reference path:
```
applied_classification_assignment <=
classification_assignment
classification_assignment.role ->
classification_role
classification_role.name
{#1: (classification_role.name = 'definitional')
#2: (classification_role.name = 'non-definitional')
#3: (classification_role.name = '')}
```

6.1.1.1.2 role

AIM element: classification_role.description

Source: ISO 10303-41

Reference path:
```
applied_classification_assignment <=
classification_assignment
classification_assignment.role ->
classification_role
classification_role.description
```

6.1.1.1.3 classification_association to general_classification (as associated_classification)

AIM element: PATH

Reference path:
```
applied_classification_assignment <=
classification_assignment
classification_assignment.assigned_class ->
group =>
class
```

6.1.1.1.4 classification_association to document (as classified_element)

AIM element: PATH

Reference path:
```
applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = product
product
{product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}
```

6.1.1.1.5 classification_association to document_file (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = document_file
document_file

6.1.1.1.6 classification_association to document_representation (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = product_definition
product_definition

6.1.1.1.7 classification_association to document_version (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item = product_definition_formation
{product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.1.1.8 classification_association to item (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = product
product

6.1.1.1.9 classification_association to item_definition (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = product_definition
product_definition

6.1.1.1.10 classification_association to item_instance (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item

classification_item = product_definition_relationship
product_definition_relationship =>
product_definition_usage =>
assembly_component_usage

6.1.1.1.11 classification_association to item_version (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = product_definition_formation
product_definition_formation

6.1.1.1.12 classification_association to material_designation (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = material_designation
material_designation

6.1.1.1.13 classification_association to property (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = general_property
general_property

6.1.1.1.14 classification_association to property_value_association (as classified_element)

AIM element: PATH

Reference path: applied_classification_assignment
applied_classification_assignment.items[i] ->
classification_item
classification_item = property_definition
property_definition

6.1.1.2 classification_association_relationship

AIM element:

Source:

Rules:

Reference path:

6.1.1.2.1 classification_association_relationship to classification_association (as related)

AIM element:

Source:

Rules:

Reference path:

6.1.1.2.2 classification_association_relationship to classification_association (as relating)

AIM element:

Source:

Rules:

Reference path:

6.1.1.2.3 relationship_type

AIM element:

Source:

Rules:

Reference path:

6.1.1.3 classification_attribute

AIM element: property_definition

Source: ISO 10303-41

6.1.1.3.1 description

AIM element: property_definition.description

Source: ISO 10303-41

6.1.1.3.2 id

AIM element: property_definition.id

Source: ISO 10303-41

6.1.1.3.3 name

AIM element: property_definition.name

Source: ISO 10303-41

6.1.1.3.4 classification_attribute to property_value_representation (as allowed_value)

AIM element: PATH

Reference path: property_definition
{derived_property_select = property_definition
derived_property_select <-

general_property_association.derived_definition
general_property_association
general_property_association.base_definition ->
general_property}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation

6.1.1.3.5 classification_attribute to general_classification (as associated_classification)

AIM element: PATH

Reference path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
characterized_class

6.1.1.3.6 classification_attribute to property (as attribute_definition)

AIM element: PATH

Reference path: property_definition
derived_property_select = property_definition
derived_property_select <-
general_property_association.derived_definition
general_property_association
{general_property_association.name = 'attribute definition'}
general_property_association.base_definition ->
general_property

6.1.1.3.7 classification_attribute to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([property_definition.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_property_definition

Reference path: property_definition
#1: (attribute_language_item = property_definition
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = property_definition
multi_language_attribute_item <-

multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.1.3.8 classification_attribute to multi_language_string (as name)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string. D:\Work\Documents\Standards\STEP\10303-214\html_with_change_marks\htmls\clause5\aim_index\pxref.htm

AIM element: #1: ([property_definition.name]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_property_definition

Reference path: property_definition
#1: (attribute_language_item = property_definition
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'name'})
#2: (multi_language_attribute_item = property_definition
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'name'})

6.1.1.4 classification_system

AIM element: class_system

Source: ISO 10303-214

Reference path: class_system <=
group

6.1.1.4.1 description

AIM element: group.description

Source: ISO 10303-41

Reference path: class_system <=
group
group.description

6.1.1.4.2 id

AIM element: group.name

Source: ISO 10303-41

Reference path: class_system <=
group
group.name

6.1.1.4.3 classification_system to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([group.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_group

Reference path: class_system <=
group
#1: (attribute_language_item = group
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = group
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.1.5 din4000_reference

#1: If din4000_reference is referenced by a general_classification as the classification_source.

#2: If din4000_reference is referenced by a property as the property_source.

AIM element: #1: (externally_defined_class)
#2: (externally_defined_general_property)

Source: ISO 10303-214
ISO 10303-214

Reference path: #1: (externally_defined_class <=
[class <=
group]
[externally_defined_item])
#2: (externally_defined_general_property <=

[general_property]
[externally_defined_item])

6.1.1.5.1 characteristics_code_no

#1: If din4000_reference is referenced by a general_classification as the classification_source.

#2: If din4000_reference is referenced by a property as the property_source.

AIM element: externally_defined_item.item_id

Source: ISO 10303-41

Reference path: #1: (externally_defined_class <=)
#2: (externally_defined_general_property <=)
externally_defined_item
externally_defined_item.item_id
{externally_defined_item.item_id ->
source_item
source_item = identifier
identifier}

6.1.1.5.2 part_no

#1: If din4000_reference is referenced by a general_classification as the classification_source.

#2: If din4000_reference is referenced by a property as the property_source.

AIM element: external_source.source_id

Source: ISO 10303-41

Reference path: #1: (externally_defined_class <=)
#2: (externally_defined_general_property <=)
externally_defined_item
externally_defined_item.source ->
external_source
external_source.source_id
{external_source.source_id ->
source_item
source_item = identifier
identifier}

6.1.1.6 external_library_reference

#1: If external_library_reference is referenced by a general_classification as the classification_source.

#2: If external_library_reference is referenced by a property as the property_source.

AIM element: #1: (externally_defined_class)
#2: (externally_defined_general_property)

Source: ISO 10303-214
ISO 10303-214

Reference path: #1: (externally_defined_class <=
[class <=
group]

[externally_defined_item])
#2: (externally_defined_general_property <=
[general_property]
[externally_defined_item])

6.1.1.6.1 description

#1: If external_library_reference is referenced by a general_classification as the classification_source.

#2: If external_library_reference is referenced by a property as the property_source.

AIM element: external_source.description

Source: ISO 10303-41

Reference path: #1: (externally_defined_class <=)
#2: (externally_defined_general_property <=)
externally_defined_item
externally_defined_item.source ->
external_source
external_source.description

6.1.1.6.2 external_id

#1: If external_library_reference is referenced by a general_classification as the classification_source.

#2: If external_library_reference is referenced by a property as the property_source.

AIM element: externally_defined_item.item_id

Source: ISO 10303-41

Reference path: #1: (externally_defined_class <=)
#2: (externally_defined_general_property <=)
externally_defined_item
externally_defined_item.item_id
{externally_defined_item.item_id ->
source_item
source_item = identifier
identifier}

6.1.1.6.3 library_type

#1: If external_library_reference is referenced by a general_classification as the classification_source.

#2: If external_library_reference is referenced by a property as the property_source.

AIM element: external_source.source_id

Source: ISO 10303-41

Reference path: #1: (externally_defined_class <=)
#2: (externally_defined_general_property <=)
externally_defined_item
externally_defined_item.source ->
external_source
external_source.source_id
{external_source.source_id ->

source_item
source_item = identifier
identifier}

6.1.1.6.4 external_library_reference to multi_language_string (as description)

#1: If external_library_reference is referenced by a general_classification as the classification_source.

#2: If external_library_reference is referenced by a property as the property_source.

#3: For the primary_language_dependent_string.

#4: For any additional_language_dependent_string.

AIM element: #3: ([external_source.description]
[PATH])
#4: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_external_source

Reference path: #1: (externally_defined_class <=)
#2: (externally_defined_general_property <=)
externally_defined_item
externally_defined_item.source ->
external_source
#3: (attribute_language_item = external_source
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#4: (multi_language_attribute_item = external_source
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.1.7 general_classification

#1: If the general_classification is not referenced by an classification_attribute.

#2: If the general_classification is referenced by an classification_attribute.

AIM element: #1: (class)
#2: (characterized_class)

Source: ISO 10303-214
ISO 10303-214

Reference path: #1: (class <=)
#2: (characterized_class <=
class <=)
group

6.1.1.7.1 description

AIM element: group.description

Source: ISO 10303-41

Reference path: class <=
group
group.description

6.1.1.7.2 id

AIM element: group.name

Source: ISO 10303-41

Reference path: class <=
group
group.name

6.1.1.7.3 version_id

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Rules: dependent_instantiable_identification_role,
restrict_version_assignment_for_class

Reference path: class
identification_item = class
identification_item <-
applied_identification_assignment.items[i]
applied_identification_assignment <=
identification_assignment
{identification_assignment.role ->
identification_role
identification_role.name = 'version'}
identification_assignment.assigned_id

6.1.1.7.4 general_classification to external_library_reference (as classification_source)

AIM element: IDENTICAL MAPPING

Reference path: {class =>
externally_defined_class}

6.1.1.7.5 general_classification to plib_class_reference (as classification_source)

AIM element: IDENTICAL MAPPING

Reference path: {class =>
externally_defined_class <=
externally_defined_item
externally_defined_item.source ->
external_source =>
known_source <=

pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}

6.1.1.7.6 general_classification to multi_language_string (as description)

#1: If the general_classification is not referenced by an classification_attribute.

#2: If the general_classification is referenced by an classification_attribute.

#3: For the primary_language_dependent_string.

#4: For any additional_language_dependent_string.

AIM element: #3: ([group.description]
[PATH])
#4: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_group

Reference path: #2: (characterized_class <=
class <=)
#1: (class <=)
group
#3: (attribute_language_item = group
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#4: (multi_language_attribute_item = group
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.1.7.7 general_classification to classification_system (as used_classification_system)

#1: If the general_classification is not referenced by an classification_attribute.

#2: If the general_classification is referenced by an classification_attribute.

AIM element: PATH

Rules: restrict_applied_classification_assignment_role,
restrict_class_system_assignment_for_class

Reference path: #2: (characterized_class <=
class)
#1: (class)
classification_item = class
classification_item <-
applied_classification_assignment.items[i]
applied_classification_assignment <=
classification_assignment

{classification_assignment.role ->
classification_role
classification_role.name = 'class system membership'}
classification_assignment.assigned_class ->
group =>
class_system

6.1.1.8 general_classification_hierarchy

AIM element: group_relationship

Source: ISO 10303-41

Rules: restrict_group_relationship_for_general_classification_hierarchy

Reference path: {group_relationship
[group_relationship.name = 'class hierarchy']
[[group_relationship.related_group ->]
[group_relationship.relating_group ->]
group =>
class]}

6.1.1.8.1 general_classification_hierarchy to general_classification (as sub_classification)

AIM element: PATH

Reference path: group_relationship
group_relationship.related_group ->
group =>
class

6.1.1.8.2 general_classification_hierarchy to general_classification (as super_classification)

AIM element: PATH

Reference path: group_relationship
group_relationship.relating_group ->
group =>
class

6.1.1.9 plib_class_reference

AIM element: externally_defined_class

Source: ISO 10303-214

Rules: restrict_name_for_known_source

Reference path: externally_defined_class <=
[class <=
group]
[externally_defined_item
{externally_defined_item.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}]

6.1.1.9.1 code

AIM element: externally_defined_item.item_id

Source: ISO 10303-41

Reference path: externally_defined_class <=
externally_defined_item
{externally_defined_item
externally_defined_item.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}
externally_defined_item.item_id
{externally_defined_item.item_id ->
source_item
source_item = identifier
identifier}

6.1.1.9.2 supplier_bsu

AIM element: organization.id

Source: ISO 10303-41

Rules: dependent_instantiable_organization_role,
externally_defined_class_with_known_source_requirement

Reference path: externally_defined_class <=
{externally_defined_class <=
externally_defined_item
externally_defined_item.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}
class
organization_item = class
organization_item <-
applied_organization_assignment.items[i]
applied_organization_assignment <=
organization_assignment
{organization_assignment.role ->
organization_role
organization_role.name = 'class supplier'}
organization_assignment.assigned_organization ->
organization
organization.id

6.1.1.9.3 version

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Rules: plib_class_reference_requires_version

Reference path: externally_defined_class
{externally_defined_class <=
externally_defined_item
externally_defined_item.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}
external_identification_item = externally_defined_class
external_identification_item <-
applied_external_identification_assignment.items[i]
applied_external_identification_assignment <=
external_identification_assignment <=
{external_identification_assignment
external_identification_assignment.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}
identification_assignment
{identification_assignment.role ->
identification_role
identification_role.name = 'version'}
identification_assignment.assigned_id

6.1.1.10 plib_property_reference

AIM element: externally_defined_general_property

Source: ISO 10303-214

Reference path: externally_defined_general_property <=
[general_property]
[externally_defined_item]

6.1.1.10.1 code

AIM element: externally_defined_item.item_id

Source: ISO 10303-41

Reference path: externally_defined_general_property <=
externally_defined_item
externally_defined_item.item_id
{externally_defined_item.item_id ->
source_item
source_item = identifier
identifier}

6.1.1.10.2 version

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Rules: plib_property_reference_requires_version

Reference path: externally_defined_general_property
{externally_defined_general_property <=

externally_defined_item
externally_defined_item.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}
external_identification_item = externally_defined_general_property
external_identification_item <-
applied_external_identification_assignment.items[i]
applied_external_identification_assignment <=
external_identification_assignment <=
{external_identification_assignment
external_identification_assignment.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}
identification_assignment
{identification_assignment.role ->
identification_role
identification_role.name = 'version'}
identification_assignment.assigned_id

6.1.1.10.3 plib_property_reference to plib_class_reference (as name_scope)

AIM element: PATH

Source: ISO 10303-41

Rules: plib_property_reference_requires_name_scope,
restrict_externally_defined_item_relationship

Reference path: externally_defined_general_property <=
externally_defined_item <-
{externally_defined_item.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}
externally_defined_item_relationship.relating_item
externally_defined_item_relationship
{externally_defined_item_relationship.name = 'name scope'}
externally_defined_item_relationship.related_item ->
externally_defined_item =>
{externally_defined_item.source ->
external_source =>
known_source <=
pre_defined_item
pre_defined_item.name = 'ISO 13584 library'}
externally_defined_class

6.1.1.11 specific_item_classification

AIM element: product_related_product_category

Source: ISO 10303-41

6.1.1.11.1 classification_name

AIM element: product_category.name

Source: ISO 10303-41

Reference path: product_related_product_category <=
product_category
product_category.name
{(product_category.name)
(product_category.name = 'accessory item')
(product_category.name = 'adaptive item')
(product_category.name = 'assembly')
(product_category.name = 'assembly item')
(product_category.name = 'cutting item')
(product_category.name = 'cutting tool')
(product_category.name = 'detail')
(product_category.name = 'tool item')}

6.1.1.11.2 description

AIM element: product_category.description

Source: ISO 10303-41

Reference path: product_related_product_category <=
product_category
product_category.description

6.1.1.11.3 specific_item_classification to item (as associated_item)

AIM element: PATH

Reference path: product_related_product_category
product_related_product_category.products[i] ->
product

6.1.1.11.4 specific_item_classification to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_category.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_related_product_category

Reference path: product_related_product_category
#1: (attribute_language_item = product_related_product_category
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment

attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_related_product_category
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.1.12 specific_item_classification_hierarchy

AIM element: product_category_relationship

Source: ISO 10303-41

Reference path: {product_category_relationship
product_category_relationship.name = 'hierarchy'}

6.1.1.12.1 specific_item_classification_hierarchy to specific_item_classification (as sub_classification)

AIM element: PATH

Reference path: product_category_relationship
product_category_relationship.sub_category ->
product_category =>
product_related_product_category

6.1.1.12.2 specific_item_classification_hierarchy to specific_item_classification (as super_classification)

AIM element: PATH

Reference path: product_category_relationship
product_category_relationship.category ->
product_category =>
product_related_product_category

6.1.2 dimension_tolerance UoF

6.1.2.1 limits_and_fits

AIM element: limits_and_fits

Source: ISO 10303-47

6.1.2.1.1 deviation

AIM element: limits_and_fits.form_variance

Source: ISO 10303-47

6.1.2.1.2 fitting_type

AIM element: limits_and_fits.zone_variance

Source: ISO 10303-47

6.1.2.1.3 grade

AIM element: limits_and_fits.grade

Source: ISO 10303-47

6.1.2.2 plus_minus_bounds

AIM element: tolerance_value

Source: ISO 10303-47

Rules: dependent_instantiable_tolerance_value

6.1.2.2.1 lower_bound

AIM element: tolerance_value.lower_bound

Source: ISO 10303-47

6.1.2.2.2 significant_digits

AIM element: precision_qualifier.precision_value

Source: ISO 10303-45

Rules: dependent_instantiable_precision_qualifier

Reference path:

```
[tolerance_value
tolerance_value.lower_bound ->
measure_with_unit <-]
[tolerance_value
tolerance_value.upper_bound ->
measure_with_unit <-]
measure_qualification.qualified_measure
measure_qualification
measure_qualification.qualifiers[i] ->
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value
```

6.1.2.2.3 upper_bound

AIM element: tolerance_value.upper_bound

Source: ISO 10303-47

6.1.2.2.4 value_determination

AIM element: type_qualifier.name

Source: ISO 10303-45

Rules: dependent_instantiable_type_qualifier

Reference path:

```
tolerance_value
[tolerance_value.lower_bound ->]
```

[tolerance_value.upper_bound->]
measure_with_unit <-
measure_qualification.qualified_measure
measure_qualification
measure_qualification.qualifiers[i] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier
type_qualifier.name
{(type_qualifier.name)
(type_qualifier.name = 'required')
(type_qualifier.name = 'designed')
(type_qualifier.name = 'calculated')
(type_qualifier.name = 'measured')
(type_qualifier.name = 'estimated')}

6.1.2.3 value_limitation

AIM element: plus_minus_tolerance

Source: ISO 10303-47

6.1.2.3.1 envelope

#1: If value_limitation.envelope is TRUE.

#2: If value_limitation.envelope is FALSE.

AIM element: #1: (applied_classification_assignment)
#2: (IDENTICAL MAPPING)

Source: ISO 10303-214

Rules: dependent_instantiable_classification_role

Reference path: #1: (plus_minus_tolerance
classification_item = plus_minus_tolerance
classification_item <-
applied_classification_assignment.items[i]
applied_classification_assignment <=
classification_assignment
{[classification_assignment.role ->
classification_role
classification_role.name = 'further tolerance requirement']
[classification_assignment.assigned_class ->
group
{[group.name = 'E']
[group.description = 'dimensioning principle']}]})
#2: (plus_minus_tolerance)

6.1.2.3.2 value_limitation to limits_and_fits (as is_defined_by)

AIM element: PATH

Reference path: plus_minus_tolerance
plus_minus_tolerance.range ->
tolerance_method_definition
tolerance_method_definition = limits_and_fits
limits_and_fits

6.1.2.3.3 value_limitation to plus_minus_bounds (as is_defined_by)

AIM element: PATH

Reference path: plus_minus_tolerance
plus_minus_tolerance.range ->
tolerance_method_definition
tolerance_method_definition = tolerance_value
tolerance_value

6.1.2.3.4 value_limitation to numerical_value (as limited_value)

AIM element: PATH

Reference path: plus_minus_tolerance
plus_minus_tolerance.toleranced_dimension ->
dimensional_characteristic
dimensional_characteristic = dimensional_size
dimensional_size
dimensional_characteristic = dimensional_size
dimensional_characteristic <-
dimensional_characteristic_representation.dimension
dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation <=
shape_representation <=
representation
representation.items[i] ->
representation_item =>
measure_representation_item

6.1.3 effectivity UoF

6.1.3.1 duration

AIM element: time_measure_with_unit

Source: ISO 10303-41

Rules: dependent_instantiable_measure_with_unit

6.1.3.1.1 time

AIM element: time_measure

Source: ISO 10303-41

Reference path: time_measure_with_unit <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = time_measure
time_measure

6.1.3.1.2 time_unit

AIM element: time_unit

Source: ISO 10303-41

Rules: dependent_instantiable_named_unit

Reference path: time_measure_with_unit <=
measure_with_unit
measure_with_unit.unit_component ->
unit
unit = named_unit
named_unit =>
time_unit

6.1.3.2 effectivity

#1: If at least one of the attributes start_definition or end_definition is specified.

#2: If neither one of the attributes start_definition and end_definition are specified.

#3: If a start_definition is given and no period is specified.

#4: If no start_definition is given or a period is specified.

AIM element: effectivity

Source: ISO 10303-41

Reference path: {#1: (effectivity =>
#3: (dated_effectivity)
#4: (time_interval_based_effectivity))
#2: (|effectivity| <-
effectivity_relationship.related_effectivity
effectivity_relationship
effectivity_relationship.name = 'inheritance')}

6.1.3.2.1 description

AIM element: effectivity.description

Source: ISO 10303-41

6.1.3.2.2 effectivity_context

AIM element: effectivity.name

Source: ISO 10303-41

6.1.3.2.3 id

AIM element: effectivity.id

Source: ISO 10303-41

6.1.3.2.4 version_id

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Rules: dependent_instantiable_identification_role,
restrict_version_assignment_for_effectivity

Reference path: effectivity
identification_item = effectivity
identification_item <-
applied_identification_assignment.items[i]
applied_identification_assignment <=
identification_assignment
{identification_assignment.role ->
identification_role
identification_role.name = 'version'}
identification_assignment.assigned_id

6.1.3.2.5 effectivity to organization (as concerned_organization)

AIM element: PATH

Reference path: effectivity <-
applied_organization_assignment.items[i]
applied_organization_assignment <=
organization_assignment
{organization_assignment.role ->
organization_role
organization_role.name = 'concerned organization'}
organization_assignment.assigned_organization ->
organization

6.1.3.2.6 effectivity to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([effectivity.description]
[PATH])
#2: (PATH)

Rules: restrict_multi_language_for_effectivity

Reference path: effectivity
#1: (attribute_language_item = effectivity
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = effectivity
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.3.2.7 effectivity to date_time (as end_definition)

#1: If a start_definition is given and no period is specified.

#2: If no start_definition is given or a period is specified.

#3: If start_definition or end_definition reference a date_time with no time specified.

#4: If start_definition or end_definition reference a date_time with a time specified.

AIM element: PATH

Reference path: effectivity =>
#1: (dated_effectivity
dated_effectivity.effectivity_end_date ->
date_time_or_event_occurrence
date_time_or_event_occurrence = date_time_select
#3: (date_time_select = date
date =>
calendar_date)
#4: (date_time_select = date_and_time
date_and_time))
#2: (time_interval_based_effectivity
time_interval_based_effectivity.effectivity_period ->
time_interval =>
time_interval_with_bounds
time_interval_with_bounds.secondary_bound ->
date_time_or_event_occurrence
date_time_or_event_occurrence = date_time_select
#3: (date_time_select = date
date =>
calendar_date)
#4: (date_time_select = date_and_time
date_and_time))

6.1.3.2.8 effectivity to duration (as period)

AIM element: PATH

Reference path: effectivity =>
time_interval_based_effectivity
time_interval_based_effectivity.effectivity_period ->
time_interval =>
time_interval_with_bounds
time_interval_with_bounds.duration ->
time_measure_with_unit

6.1.3.2.9 effectivity to date_time (as start_definition)

#1: If a start_definition is given and no period is specified.

#2: If no start_definition is given or a period is specified.

#3: If start_definition or end_definition reference a date_time with no time specified.

#4: If start_definition or end_definition reference a date_time with a time specified.

AIM element: PATH

Reference path: effectivity =>
#1: (dated_effectivity
dated_effectivity.effectivity_start_date ->
date_time_or_event_occurrence

date_time_or_event_occurrence = date_time_select
#3: (date_time_select = date
date =>
calendar_date)
#4: (date_time_select = date_and_time
date_and_time))
#2: (time_interval_based_effectivity
time_interval_based_effectivity.effectivity_period ->
time_interval =>
time_interval_with_bounds
time_interval_with_bounds.primary_bound ->
date_time_or_event_occurrence
date_time_or_event_occurrence = date_time_select
#3: (date_time_select = date
date =>
calendar_date)
#4: (date_time_select = date_and_time
date_and_time))

6.1.3.3 effectivity_assignment

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: #1: (applied_effectivity_assignment)
#2: (applied_ineffectivity_assignment)

Source: ISO 10303-214
ISO 10303-214

Reference path: #1: (applied_effectivity_assignment <=
effectivity_assignment)
#2: (applied_ineffectivity_assignment <=
effectivity_assignment)

6.1.3.3.1 effectivity_indication

AIM element: IDENTICAL MAPPING

6.1.3.3.2 role

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: object_role.name

Source: ISO 10303-41

Rules: dependent_instantiable_object_role

Reference path: #1: (applied_effectivity_assignment <=)
#2: (applied_ineffectivity_assignment <=)
effectivity_assignment
effectivity_assignment.role ->
object_role
object_role.name
{(object_role.name)

```
                  (object_role.name = 'actual')
                  (object_role.name = 'planned')
                  (object_role.name = 'required')}
```

6.1.3.3.3 effectivity_assignment to effectivity (as assigned_effectivity)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

```
Reference path:   #1: (applied_effectivity_assignment <=)
                  #2: (applied_ineffectivity_assignment <=)
                  effectivity_assignment
                  effectivity_assignment.assigned_effectivity ->
                  effectivity
```

6.1.3.3.4 effectivity_assignment to classification_system (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

```
Reference path:   #1: (applied_effectivity_assignment
                  applied_effectivity_assignment.items[i] ->)
                  #2: (applied_ineffectivity_assignment
                  applied_ineffectivity_assignment.items[i] ->)
                  effectivity_item
                  effectivity_item = class_system
                  class_system
```

6.1.3.3.5 effectivity_assignment to document (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

```
Reference path:   #1: (applied_effectivity_assignment
                  applied_effectivity_assignment.items[i] ->)
                  #2: (applied_ineffectivity_assignment
                  applied_ineffectivity_assignment.items[i] ->)
                  effectivity_item
                  effectivity_item = product
                  product
                  {product <-
                  product_related_product_category.products[i]
                  product_related_product_category <=
                  product_category
                  product_category.name = 'document'}
```

6.1.3.3.6 effectivity_assignment to document_file (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = document_file
document_file

6.1.3.3.7 effectivity_assignment to document_representation (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = product_definition
product_definition

6.1.3.3.8 effectivity_assignment to document_version (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = product_definition_formation
product_definition_formation
{product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.3.3.9 effectivity_assignment to item (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = product
product

6.1.3.3.10 effectivity_assignment to item_instance (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = product_definition_relationship
product_definition_relationship =>
product_definition_usage =>
assembly_component_usage

6.1.3.3.11 effectivity_assignment to item_structure_association (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = product_definition_relationship
product_definition_relationship

6.1.3.3.12 effectivity_assignment to item_version (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = product_definition_formation
product_definition_formation

6.1.3.3.13 effectivity_assignment to material_designation (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = material_designation
material_designation

6.1.3.3.14 effectivity_assignment to physical_item_location_association (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = applied_location_assignment
applied_location_assignment

6.1.3.3.15 effectivity_assignment to physical_item_state_association (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = applied_state_type_assignment
applied_state_type_assignment

6.1.3.3.16 effectivity_assignment to physical_item_structure_association (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)

#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = product_definition_relationship
product_definition_relationship

6.1.3.3.17 property

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = general_property
general_property

6.1.3.3.18 effectivity_assignment to property_value_association (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = property_definition
property_definition

6.1.3.3.19 effectivity_assignment to realized_item_association (as effective_element)

#1: If effectivity_assignment.effectivity_indication is TRUE.

#2: If effectivity_assignment.effectivity_indication is FALSE.

AIM element: PATH

Reference path: #1: (applied_effectivity_assignment
applied_effectivity_assignment.items[i] ->)
#2: (applied_ineffectivity_assignment
applied_ineffectivity_assignment.items[i] ->)
effectivity_item
effectivity_item = product_definition_relationship
product_definition_relationship

6.1.3.4 effectivity_relationship

AIM element: effectivity_relationship

Source: ISO 10303-41

6.1.3.4.1 description

AIM element: effectivity_relationship.description

Source: ISO 10303-41

6.1.3.4.2 relation_type

AIM element: effectivity_relationship.name

Source: ISO 10303-41

Reference path: {effectivity_relationship
(effectivity_relationship.name)
(effectivity_relationship.name = 'constraint')
(effectivity_relationship.name = 'inheritance')}

6.1.3.4.3 effectivity_relationship to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([effectivity_relationship.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_effectivity_relationship

Reference path: effectivity_relationship
#1: (attribute_language_item = effectivity_relationship
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = effectivity_relationship
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.3.4.4 effectivity_relationship to effectivity (as related)

AIM element: PATH

Rules: restrict_effectivity_for_effectivity_relationship

Reference path: effectivity_relationship
effectivity_relationship.related_effectivity ->
effectivity

6.1.3.4.5 effectivity_relationship to effectivity (as relating)

AIM element: PATH

Rules: restrict_effectivity_for_effectivity_relationship

Reference path: effectivity_relationship
effectivity_relationship.relating_effectivity ->
effectivity

6.1.4 external_reference_mechanism UoF

6.1.4.1 cartesian_coordinate_space

AIM element: geometric_representation_context

Source: ISO 10303-42

6.1.4.1.1 cartesian_coordinate_space to unit (as unit_of_values)

#1: If the unit has a commonly used descriptor.

#2: If the unit descriptor is constructed of more than one unit name.

AIM element: global_unit_assigned_context.units

Source: ISO 10303-41

Reference path: geometric_representation_context <=
representation_context =>
global_unit_assigned_context
global_unit_assigned_context.units[i] ->
unit
#1: (unit = named_unit
named_unit)
#2: (unit = derived_unit
derived_unit)

6.1.4.2 cartesian_coordinate_space_2d

AIM element: geometric_representation_context

Source: ISO 10303-42

Reference path: {geometric_representation_context
geometric_representation_context.coordinate_space_dimension = 2}

6.1.4.3 cartesian_coordinate_space_3d

AIM element: geometric_representation_context

Source: ISO 10303-42

Reference path: {geometric_representation_context
geometric_representation_context.coordinate_space_dimension = 3}

6.1.4.4 digital_document

#1: If element digital_document.file is referencing no digital_file.

#2: If element digital_document.file is referencing one or more digital_file.

AIM element: #1: (product_definition)
#2: (product_definition_with_associated_documents)

Source: ISO 10303-41
ISO 10303-41

Reference path: {#1: (product_definition)
#2: (product_definition_with_associated_documents <=
product_definition)
product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'digital document definition'}

6.1.4.4.1 digital_document to digital_file (as file)

AIM element: PATH

Reference path: product_definition =>
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'digital document definition'}
product_definition_with_associated_documents
product_definition_with_associated_documents.documentation_ids[i] ->
document =>
{document.representation_types[i] ->
document_representation_type
document_representation_type.name = 'digital'}
document_file

6.1.4.5 digital_file

AIM element: document_file

Source: ISO 10303-214

Reference path: document_file <=
[document
{document.representation_types[1] ->
document_representation_type
document_representation_type.name = 'digital'}]
[characterized_object]

6.1.4.6 document

AIM element: product

Source: ISO 10303-41

Rules: product_requires_category,
product_requires_version,
restrict_product_category_for_product

Reference path: {product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.4.6.1 description

AIM element: product.description

Source: ISO 10303-41

6.1.4.6.2 document_id

AIM element: product.id

Source: ISO 10303-41

6.1.4.6.3 name

AIM element: product.name

Source: ISO 10303-41

6.1.4.6.4 document to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string. D:\Work\Documents\Standards\STEP\10303-214\html_with_change_marks\htmls\clause5\aim_index\pxref.htm

AIM element: #1: ([product.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product

Reference path: product
#1: (attribute_language_item = product
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.4.6.5 document to multi_language_string (as name)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product.name]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product

Reference path: product
#1: (attribute_language_item = product
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'name'})
#2: (multi_language_attribute_item = product
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'name'})

6.1.4.7 document_assignment

AIM element: applied_document_reference

Source: ISO 10303-214

Reference path: applied_document_reference <=
document_reference

6.1.4.7.1 role

AIM element: object_role.name

Source: ISO 10303-41

Rules: dependent_instantiable_object_role

Reference path: applied_document_reference <=
document_reference
document_reference.role ->
object_role
object_role.name
{(object_role.name)
(object_role.name = 'additional information')
(object_role.name = 'behavior')
(object_role.name = 'catalogue')
(object_role.name = 'description')
(object_role.name = 'informative')
(object_role.name = 'mandatory')

(object_role.name = 'manual')
(object_role.name = 'mathematical description')
(object_role.name = 'specification')}

6.1.4.7.2 document_assignment to document (as assigned_document)

AIM element: PATH

Reference path: applied_document_reference <=
document_reference
document_reference.assigned_document ->
document <-
{document.kind ->
document_type
document_type.product_data_type = 'configuration controlled document'}
document_product_association.relating_document
document_product_association
{document_product_association =>
document_product_equivalence}
document_product_association.related_product ->
product_or_formation_or_definition
product_or_formation_or_definition = product
product
{product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.4.7.3 document_assignment to document_version (as assigned_document)

AIM element: PATH

Reference path: applied_document_reference <=
document_reference
document_reference.assigned_document ->
document <-
{document.kind ->
document_type
document_type.product_data_type = 'configuration controlled document version'}
document_product_association.relating_document
document_product_association
{document_product_association =>
document_product_equivalence}
document_product_association.related_product ->
product_or_formation_or_definition
product_or_formation_or_definition = product_definition_formation
product_definition_formation
{product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.4.7.4 document_assignment to classification_attribute (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = property_definition
property_definition

6.1.4.7.5 document_assignment to classification_system (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = class_system
class_system

6.1.4.7.6 document_assignment to general_classification (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = class
class

6.1.4.7.7 document_assignment to item (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = product
product

6.1.4.7.8 document_assignment to item_definition (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = product_definition
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'}

6.1.4.7.9 document_assignment to item_instance (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = product_definition_relationship

product_definition_relationship =>
product_definition_usage =>
assembly_component_usage

6.1.4.7.10 document_assignment to item_structure_association (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = product_definition_relationship
product_definition_relationship

6.1.4.7.11 document_assignment to item_version (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = product_definition_formation
product_definition_formation

6.1.4.7.12 document_assignment to mated_item_relationship (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = product_definition_relationship
product_definition_relationship

6.1.4.7.13 document_assignment to material_designation (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = material_designation
material_designation

6.1.4.7.14 document_assignment to organization (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = organization
organization

6.1.4.7.15 document_assignment to person (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = person
person

6.1.4.7.16 document_assignment to physical_item_structure_association (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = product_definition_relationship
product_definition_relationship
{[product_definition_relationship.relating_product_definition ->
product_definition
product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence']
[product_definition_relationship.related_product_definition ->
product_definition
product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence']}

6.1.4.7.17 document_assignment to property (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = general_property
general_property

6.1.4.7.18 document_assignment to specific_item_classification (as is_assigned_to)

AIM element: PATH

Reference path: applied_document_reference
applied_document_reference.items[i] ->
document_reference_item
document_reference_item = product_related_product_category
product_related_product_category

6.1.4.8 document_content_property

AIM element: representation

Source: ISO 10303-43

Rules: restrict_representation_for_document_content_property,
restrict_representation_for_document_properties

Reference path: {representation <-
{[representation.name = 'document content']
[representation.context_of_items ->
representation_context
representation_context.context_type = 'document parameters']}
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'document property'}}

6.1.4.8.1 detail_level

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'detail level'}
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description)
(descriptive_representation_item.description = 'rough 3d shape')
(descriptive_representation_item.description = 'rounded edges')}

6.1.4.8.2 geometry_type

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'geometry type'}
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description)
(descriptive_representation_item.description = '3D wireframe model')
(descriptive_representation_item.description = '2D shape')
(descriptive_representation_item.description = 'surface model')
(descriptive_representation_item.description = 'closed volume')
(descriptive_representation_item.description = 'solid model')
(descriptive_representation_item.description = 'solid and surface model')
(descriptive_representation_item.description = 'assembly')
(descriptive_representation_item.description = 'assembly with mating elements')
(descriptive_representation_item.description = '2D drawing')
(descriptive_representation_item.description = 'drawing derived from 3D data')
(descriptive_representation_item.description = 'drawing related to 3D data')}

6.1.4.8.3 document_content_property to language (as languages)

AIM element: PATH

Reference path: representation
language_item = representation
language_item <-
language_assignment.items[i]
language_assignment <=
classification_assignment
{classification_assignment.role ->
classification_role
classification_role.name = 'language'}
classification_assignment.assigned_class ->
group =>
language

6.1.4.8.4 document_content_property to numerical_value (as real_world_scale)

#1: If the unit is not assigned globally.

#2: If the unit is assigned globally.

AIM element: #1: (measure_representation_item)
#2: (value_representation_item)

Source: ISO 10303-45
ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'real world scale'}
#1: (measure_representation_item
{measure_representation_item <=
measure_with_unit
[measure_with_unit.value_component ->
measure_value
measure_value = ratio_measure]
[measure_with_unit.unit_component ->
unit
unit = named_unit
named_unit =>
ratio_unit]})
#3: (value_representation_item)

6.1.4.9 document_creation_property

AIM element: representation

Source: ISO 10303-43

Rules: restrict_representation_for_document_creation_property,
restrict_representation_for_document_properties

Reference path: {representation <-
{[representation.name = 'document creation']
[representation.context_of_items ->
representation_context
representation_context.context_type = 'document parameters']}
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->

represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'document property'}}

6.1.4.9.1 creating_interface

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'creating interface'}
descriptive_representation_item
descriptive_representation_item.description

6.1.4.9.2 creating_system

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'creating system'}
descriptive_representation_item
descriptive_representation_item.description

6.1.4.9.3 operating_system

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'operating system'}
descriptive_representation_item
descriptive_representation_item.description

6.1.4.10 document_file

AIM element: document_file

Source: ISO 10303-214

Rules: restrict_properties_of_document_file

Reference path: {document_file <=
[document
{document.representation_types[i] ->
document_representation_type
(document_representation_type.name = 'digital')

(document_representation_type.name = 'physical')}]
[characterized_object]}

6.1.4.10.1 file_id

AIM element: document.id

Source: ISO 10303-41

Reference path: document_file <=
document
{document.representation_types[i] ->
document_representation_type
(document_representation_type.name = 'digital')
(document_representation_type.name = 'physical')}
document.id

6.1.4.10.2 version_id

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Rules: dependent_instantiable_identification_role,
restrict_version_assignment_for_document_file

Reference path: document_file
identification_item = document_file
identification_item <-
applied_identification_assignment.items[i]
applied_identification_assignment <=
identification_assignment
{identification_assignment.role ->
identification_role
identification_role.name = 'version'}
identification_assignment.assigned_id

6.1.4.10.3 document_file to document_content_property (as content)

AIM element: PATH

Reference path: document_file <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'document property'}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'document content'}

6.1.4.10.4 document_file to document_creation_property (as creation)

AIM element: PATH

Reference path:
```
document_file <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'document property'}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'document creation'}
```

6.1.4.10.5 document_file to document_type_property (as document_file_type)

AIM element: PATH

Reference path:
```
document_file <=
document
document.kind ->
document_type
```

6.1.4.10.6 document_file to external_file_id_and_location (as external_id_and_location)

AIM element: PATH

Reference path:
```
document_file
external_identification_item = document_file
external_identification_item <-
applied_external_identification_assignment.items[i]
applied_external_identification_assignment
```

6.1.4.10.7 document_file to document_format_property (as file_format)

AIM element: PATH

Reference path:
```
document_file <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'document property'}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'document format'}
```

6.1.4.10.8 document_file to document_size_property (as size)

AIM element: PATH

Reference path:
```
document_file <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'document property'}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'document size'}
```

6.1.4.11 document_format_property

AIM element: representation

Source: ISO 10303-43

Rules: restrict_representation_for_document_format_property, restrict_representation_for_document_properties

Reference path:
```
{representation <-
{[representation.name = 'document format']
[representation.context_of_items ->
representation_context
representation_context.context_type = 'document parameters']}
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'document property'}}
```

6.1.4.11.1 character_code

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path:
```
representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'character code'}
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description)
(descriptive_representation_item.description = 'binary')
(descriptive_representation_item.description = 'IEC 61286')
(descriptive_representation_item.description = 'ISO 646')
(descriptive_representation_item.description = 'ISO 6937')
```

(descriptive_representation_item.description = 'ISO 8859-1')
(descriptive_representation_item.description = 'ISO 10646')}

6.1.4.11.2 data_format

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'data format'}
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description)
(descriptive_representation_item.description = 'DXF')
(descriptive_representation_item.description = 'IGES')
(descriptive_representation_item.description = 'ISO 10303-203')
(descriptive_representation_item.description = 'ISO 10303-214')
(descriptive_representation_item.description = 'TIFF CCITT GR4')
(descriptive_representation_item.description = 'VDAFS')
(descriptive_representation_item.description = 'VOXEL')}

6.1.4.11.3 document_format_property to rectangular_size (as size_format)

AIM element: PATH

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'size format'}
geometric_representation_item =>
planar_extent

6.1.4.12 document_location_property

AIM element: external_source

Source: ISO 10303-41

6.1.4.12.1 location_name

AIM element: external_source.source_id

Source: ISO 10303-41

6.1.4.13 document_representation

AIM element: product_definition

Source: ISO 10303-41

Rules: restrict_properties_of_document_representation

Reference path: {product_definition
product_definition.frame_of_reference ->
product_definition_context <=

application_context_element
(application_context_element.name = 'digital document definition')
(application_context_element.name = 'physical document definition')}

6.1.4.13.1 description

AIM element: product_definition.description

Source: ISO 10303-41

6.1.4.13.2 id

AIM element: product_definition.id

Source: ISO 10303-41

6.1.4.13.3 document_representation to document_version (as associated_document_version)

AIM element: PATH

Reference path:
```
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
(application_context_element.name = 'digital document definition')
(application_context_element.name = 'physical document definition')}
product_definition.formation ->
product_definition_formation
{product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}
```

6.1.4.13.4 document_representation to document_location_property (as common_location)

AIM element: PATH

Reference path:
```
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
(application_context_element.name = 'digital document definition')
(application_context_element.name = 'physical document definition')}
external_identification_item = product_definition
external_identification_item <-
applied_external_identification_assignment.items[i]
applied_external_identification_assignment <=
external_identification_assignment
{external_identification_assignment <=
identification_assignment
{identification_assignment
identification_assignment.assigned_id = ''}
identification_assignment.role ->
identification_role
identification_role.name = 'common location'}
```

external_identification_assignment.source ->
external_source

6.1.4.13.5 document_representation to document_content_property (as content)

AIM element: PATH

Reference path: product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
(application_context_element.name = 'digital document definition')
(application_context_element.name = 'physical document definition')}
characterized_definition = product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'document property'}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'document content'}

6.1.4.13.6 document_representation to document_creation_property (as creation)

AIM element: PATH

Reference path: product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
(application_context_element.name = 'digital document definition')
(application_context_element.name = 'physical document definition')}
characterized_definition = product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'document property'}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'document creation'}

6.1.4.13.7 document_representation to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition

Reference path: product_definition
#1: (attribute_language_item = product_definition
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.4.13.8 document_representation to document_format_property (as representation_format)

AIM element: PATH

Reference path: product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
(application_context_element.name = 'digital document definition')
(application_context_element.name = 'physical document definition')}
characterized_definition = product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'document property'}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'document format'}

6.1.4.13.9 document_representation to document_size_property (as size)

AIM element: PATH

Reference path: product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
(application_context_element.name = 'digital document definition')
(application_context_element.name = 'physical document definition')}
characterized_definition = product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'document property'}
represented_definition = property_definition

represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'document size'}

6.1.4.14 document_size_property

AIM element: representation

Source: ISO 10303-43

Rules: restrict_representation_for_document_properties,
restrict_representation_for_document_size_property

Reference path: {representation <-
{[representation.name = 'document size']
[representation.context_of_items ->
representation_context
representation_context.context_type = 'document parameters']}
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'document property'}}

6.1.4.14.1 document_size_property to value_with_unit (as file_size)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a value_range.

#3: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

AIM element: PATH

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'file size'}
#1: (measure_representation_item)
#2: (compound_representation_item =>
value_range)
#3: (value_representation_item)

6.1.4.14.2 document_size_property to value_with_unit (as page_count)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a value_range.

#3: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

AIM element: PATH

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'page count'}
#1: (measure_representation_item <=
measure_with_unit
{measure_with_unit.value_component ->
measure_value
measure_value = count_measure})
#2: (compound_representation_item
{compound_representation_item =>
value_range}
compound_representation_item.item_element ->
compound_item_definition
compound_item_definition = set_representation_item
set_representation_item
set_representation_item[i] ->
representation_item =>
measure_representation_item <=
measure_with_unit
{measure_with_unit.value_component ->
measure_value
measure_value = count_measure})
#3: (value_representation_item)

6.1.4.15 document_type_property

AIM element: document_type

Source: ISO 10303-41

6.1.4.15.1 document_type_name

AIM element: document_type.product_data_type

Source: ISO 10303-41

Reference path: {document_type.product_data_type
(document_type.product_data_type)
(document_type.product_data_type = 'geometry')
(document_type.product_data_type = 'NC data')
(document_type.product_data_type = 'FE data')
(document_type.product_data_type = 'sample data')
(document_type.product_data_type = 'process plan')
(document_type.product_data_type = 'check plan')
(document_type.product_data_type = 'drawing')}

6.1.4.15.2 document_type_property to classification_system (as used_classification_system)

AIM element: PATH

Rules: dependent_instantiable_classification_role

Reference path: document_type
classification_item = document_type
classification_item <-
applied_classification_assignment.items[i]
applied_classification_assignment <=
classification_assignment

{classification_assignment.role ->
classification_role
classification_role.name = 'class system membership'}
classification_assignment.assigned_class ->
group =>
class_system

6.1.4.16 document_version

AIM element: product_definition_formation

Source: ISO 10303-41

Reference path: {product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.4.16.1 description

AIM element: product_definition_formation.description

Source: ISO 10303-41

6.1.4.16.2 id

AIM element: product_definition_formation.id

Source: ISO 10303-41

6.1.4.16.3 document_version to document (as associated_document)

AIM element: PATH

Reference path: product_definition_formation
product_definition_formation.of_product ->
product
{product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.4.16.4 document_version to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition_formation.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition_formation

Reference path: product_definition_formation
#1: (attribute_language_item = product_definition_formation
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition_formation
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.4.17 document_version_relationship

AIM element: product_definition_formation_relationship

Source: ISO 10303-41

6.1.4.17.1 description

AIM element: product_definition_formation_relationship.description

Source: ISO 10303-41

6.1.4.17.2 relation_type

AIM element: product_definition_formation_relationship.name

Source: ISO 10303-41

Reference path: {product_definition_formation_relationship
(product_definition_formation_relationship.name)
(product_definition_formation_relationship.name = 'derivation')
(product_definition_formation_relationship.name = 'hierarchy')
(product_definition_formation_relationship.name = 'sequence')
(product_definition_formation_relationship.name = 'supplied document')}

6.1.4.17.3 document_version_relationship to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition_formation_relationship.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition_formation_relationship

Reference path: product_definition_formation_relationship
#1: (attribute_language_item = product_definition_formation_relationship
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition_formation_relationship
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.4.17.4 document_version_relationship to document_version (as related)

AIM element: PATH

Reference path: product_definition_formation_relationship
product_definition_formation_relationship.related_product_definition_formation ->
product_definition_formation
{product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.4.17.5 document_version_relationship to document_version (as relating)

AIM element: PATH

Reference path: product_definition_formation_relationship
product_definition_formation_relationship.relating_product_definition_formation ->
product_definition_formation
{product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.4.18 external_file_id_and_location

AIM element: applied_external_identification_assignment

Source: ISO 10303-214

Reference path: applied_external_identification_assignment <=
external_identification_assignment
{external_identification_assignment <=
identification_assignment
identification_assignment.role ->
identification_role
identification_role.name = 'external document id and location'}

6.1.4.18.1 external_id

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Reference path: applied_external_identification_assignment <=
external_identification_assignment <=
identification_assignment
identification_assignment.assigned_id

6.1.4.18.2 external_file_id_and_location to document_location_property (as location)

AIM element: PATH

Reference path: applied_external_identification_assignment <=
external_identification_assignment
external_identification_assignment.source ->
external_source

6.1.4.19 external_geometric_model

AIM element: shape_representation

Source: ISO 10303-41

Reference path: {shape_representation <=
representation
representation.context_of_items ->
representation_context =>
{representation_context.context_type = 'external'}
geometric_representation_context}

6.1.4.19.1 model_extent

#1: If the unit is assigned globally.

#2: If the unit is not assigned globally.

AIM element: #1: (value_representation_item)
#2: (measure_representation_item)

Source: ISO 10303-43
ISO 10303-43

Rules: dependent_instantiable_measure_with_unit

Reference path: shape_representation <=
representation <-
representation_relationship.rep_1
representation_relationship
{representation_relationship.name = 'model extent association'}
representation_relationship.rep_2 ->
representation
{representation.name = 'model extent representation'}
representation.items[i] ->
representation_item =>
{representation_item.name = 'model extent value'}

#1: (value_representation_item)
#2: (measure_representation_item)

6.1.4.20 external_model

AIM element: representation

Source: ISO 10303-43

Reference path: {representation
[representation.context_of_items ->
representation_context
representation_context.context_type = 'external']
[representation.items[1] ->
(axis2_placement_2d)
(axis2_placement_3d)]}

6.1.4.20.1 description

AIM element: representation.description

Source: ISO 10303-43

6.1.4.20.2 model_id

AIM element: representation.name

Source: ISO 10303-43

6.1.4.20.3 external_model to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([representation.description]
[PATH])
#2: (PATH)

Source: ISO 10303-43

Rules: restrict_multi_language_for_representation

Reference path: representation
#1: (attribute_language_item = representation
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = representation
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.4.20.4 external_model to digital_file (as is_defined_as)

AIM element: PATH

Reference path:
```
representation <-
{representation.context_of_items ->
representation_context
representation_context.context_type = 'external'}
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'external definition'}
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
document_file
{document_file <=
document
document.representation_types[i] ->
document_representation_type
document_representation_type.name = 'digital'}
```

6.1.4.20.5 external_model to cartesian_coordinate_space (as is_defined_in)

AIM element: PATH

Reference path:
```
representation
representation.context_of_items ->
representation_context =>
{representation_context.context_type = 'external'}
geometric_representation_context
```

6.1.4.21 external_picture

AIM element: representation

Source: ISO 10303-43

Reference path:
```
{representation
[representation.context_of_items ->
representation_context =>
{representation_context.context_type = 'external'}
geometric_representation_context
geometric_representation_context.coordinate_space_dimension = 2]
[representation.items[1] ->
axis2_placement_2d]}
```

6.1.4.21.1 external_picture to cartesian_coordinate_space_2d (as is_defined_in)

AIM element: PATH

Reference path:
```
representation
representation.context_of_items ->
representation_context =>
{representation_context.context_type = 'external'}
```

geometric_representation_context
{geometric_representation_context.coordinate_space_dimension = 2}

6.1.4.22 geometric_model_relationship_with_transformation

#1: If referred to by an assembly_association.placement.

#2: If not referred to by an assembly_association.placement.

AIM element: #1: ([shape_representation_relationship]
[representation_relationship_with_transformation])
#2: (representation_relationship_with_transformation)

Source: ISO 10303-41
ISO 10303-43
ISO 10303-43

6.1.4.22.1 geometric_model_relationship_with_transformation to transformation (as model_placement)

#1: If the transformation is implicitely defined by two items.

#2: For a two-dimensional transformation.

#3: For a three-dimensional transformation.

#4: If the item is a placement.

#5: If the item is a transformation operator.

#6: If the transformation is explicitly defined.

AIM element: PATH

Reference path: representation_relationship_with_transformation
representation_relationship_with_transformation.transformation_operator ->
transformation
#1: (transformation = item_defined_transformation
item_defined_transformation
{#2: ([item_defined_transformation.transform_item_1 ->
representation_item =>
geometric_representation_item =>
placement =>
axis2_placement_2d]
[item_defined_transformation.transform_item_2 ->
representation_item =>
geometric_representation_item =>
#4: (placement =>
axis2_placement_2d)
#5: (cartesian_transformation_operator =>
{cartesian_transformation_operator.scl = 1.0}
cartesian_transformation_operator_2d)])
#3: ([item_defined_transformation.transform_item_1 ->
representation_item =>
geometric_representation_item =>
placement =>
axis2_placement_3d]
[item_defined_transformation.transform_item_2 ->
representation_item =>

geometric_representation_item =>
#4: (placement =>
axis2_placement_3d)
#5: (cartesian_transformation_operator =>
{cartesian_transformation_operator.scl = 1.0}
cartesian_transformation_operator_3d)])})
#6: (transformation = functionally_defined_transformation
functionally_defined_transformation
{functionally_defined_transformation =>
cartesian_transformation_operator =>
{cartesian_transformation_operator.scl = 1.0}
#2: (cartesian_transformation_operator_2d)
#3: (cartesian_transformation_operator_3d)})

6.1.4.23 named_size

#1: If no density is specified for the rectangular_size.

#2: If a density is specified for the rectangular_size.

AIM element: #1: ([planar_extent]
[descriptive_representation_item])
#2: ([planar_extent]
[descriptive_representation_item]
[measure_representation_item])

Source: ISO 10303-46
ISO 10303-46
ISO 10303-46
ISO 10303-46
ISO 10303-46

Reference path: {#1: ([planar_extent <=
geometric_representation_item <=]
[descriptive_representation_item <=])
#2: ([planar_extent <=
geometric_representation_item <=]
[descriptive_representation_item <=]
[measure_representation_item <=])
representation_item
representation_item.name = 'size format'}

6.1.4.23.1 size

#1: If no density is specified for the rectangular_size.

#2: If a density is specified for the rectangular_size.

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: {#1: ([planar_extent <=
geometric_representation_item <=]
[descriptive_representation_item <=])
#2: ([planar_extent <=
geometric_representation_item <=]
[descriptive_representation_item <=]
[measure_representation_item <=])

representation_item
representation_item.name = 'size format'}

6.1.4.23.2 named_size to classification_system (as referenced_standard)

#1: If no density is specified for the rectangular_size.

#2: If a density is specified for the rectangular_size.

AIM element: PATH

Rules: restrict_class_system_assignment_for_planar_extent

Reference path:

```
planar_extent
{#1: ([planar_extent <=
geometric_representation_item <=]
[descriptive_representation_item <=])
#2: ([planar_extent <=
geometric_representation_item <=]
[descriptive_representation_item <=]
[measure_representation_item <=])
representation_item
representation_item.name = 'size format'}
classification_item = planar_extent
classification_item <-
applied_classification_assignment.items[i]
applied_classification_assignment <=
classification_assignment
{classification_assignment.role ->
classification_role
classification_role.name = 'class system membership'}
classification_assignment.assigned_class ->
group =>
class_system
```

6.1.4.24 physical_document

AIM element: product_definition

Source: ISO 10303-41

Reference path:

```
{product_definition
product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical document definition'}
```

6.1.4.24.1 physical_document to physical_file (as file)

AIM element: PATH

Reference path:

```
product_definition =>
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical document definition'}
product_definition_with_associated_documents
product_definition_with_associated_documents.documentation_ids[i] ->
document =>
```

{document.representation_types[i] ->
document_representation_type
document_representation_type.name = 'physical'}
document_file

6.1.4.25 physical_file

AIM element: document_file

Source: ISO 10303-214

Reference path: document_file <=
[document
{document.representation_types[i] ->
document_representation_type
document_representation_type.name = 'physical'}]
[characterized_object]

6.1.4.26 rectangular_size

#1: If no density is specified for the rectangular_size.

#2: If a density is specified for the rectangular_size.

#3: If the rectangular_size is a named_size.

#4: If the rectangular_size is not a named_size.

AIM element: #1: (#3: ([planar_extent]
[descriptive_representation_item])
#4: (planar_extent))
#2: (#3: ([planar_extent]
[measure_representation_item]
[descriptive_representation_item])
#4: ([planar_extent]
[measure_representation_item]))

Source: ISO 10303-46
ISO 10303-46
ISO 10303-46
ISO 10303-46
ISO 10303-46
ISO 10303-46
ISO 10303-46
ISO 10303-46

Reference path: {#1: (#3: ([planar_extent <=
geometric_representation_item <=]
[descriptive_representation_item <=])
#4: (planar_extent <=
geometric_representation_item <=))
#2: (#3: ([planar_extent <=
geometric_representation_item <=]
[descriptive_representation_item <=]
[measure_representation_item <=])
#4: ([planar_extent <=
geometric_representation_item <=]
[measure_representation_item <=]))

representation_item
representation_item.name = 'size format'}

6.1.4.26.1 rectangular_size to value_with_unit (as density)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a value_range.

#3: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

AIM element: #1: (measure_representation_item)
#2: (value_range)
#3: (value_representation_item)

Source: ISO 10303-45
ISO 10303-45
ISO 10303-45

Reference path: #1: ({measure_representation_item <=
representation_item
representation_item.name = 'size format'})
#2: ({value_range <=
compound_representation_item <=
representation_item
representation_item.name = 'size format'})
#3: ({value_representation_item <=
representation_item
representation_item.name = 'size format'})

6.1.4.26.2 rectangular_size to value_with_unit (as height)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

#3: If the unit of the value range is not assigned globally.

#4: If the unit of the value range is assigned globally.

AIM element: PATH

Reference path: planar_extent
planar_extent.size_in_y ->
length_measure
measure_value = length_measure
measure_value <-
(measure_with_unit.value_component
measure_with_unit
#1: (measure_with_unit =>
measure_representation_item)
#3: (measure_with_unit =>
measure_representation_item <=
representation_item <-
set_representation_item[i]
set_representation_item
compound_item_definition = set_representation_item
compound_item_definition <-
compound_representation_item.item_element

compound_representation_item =>
value_range))
#4: (value_representation_item.value_component
value_representation_item <=
representation_item <-
set_representation_item[i]
set_representation_item
compound_item_definition = set_representation_item
compound_item_definition <-
compound_representation_item.item_element
compound_representation_item =>
value_range)
#2: (value_representation_item.value_component
value_representation_item)

6.1.4.26.3 rectangular_size to value_with_unit (as width)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

#3: If the unit of the value range is not assigned globally.

#4: If the unit of the value range is assigned globally.

AIM element: PATH

Reference path: planar_extent
planar_extent.size_in_x ->
length_measure
measure_value = length_measure
measure_value <-
(measure_with_unit.value_component
measure_with_unit
#1: (measure_with_unit =>
measure_representation_item)
#3: (measure_with_unit =>
measure_representation_item <=
representation_item <-
set_representation_item[i]
set_representation_item
compound_item_definition = set_representation_item
compound_item_definition <-
compound_representation_item.item_element
compound_representation_item =>
value_range))
#4: (value_representation_item.value_component
value_representation_item <=
representation_item <-
set_representation_item[i]
set_representation_item
compound_item_definition = set_representation_item
compound_item_definition <-
compound_representation_item.item_element
compound_representation_item =>
value_range)
#2: (value_representation_item.value_component
value_representation_item)

6.1.4.27 transformation

#1: If the transformation is used with a template instance.

#2: If the transformation is implicitly defined by two items.

#3: If the transformation is explicitly defined.

AIM element: #1: ([mapped_item.mapping_target]
[representation_map.mapping_origin])
#2: (item_defined_transformation)
#3: (functionally_defined_transformation)

Source: ISO 10303-43
ISO 10303-43
ISO 10303-43
ISO 10303-43

6.1.4.28 transformation_2d

#2: If the transformation is implicitly defined by two items.

#3: If the transformation is explicitly defined.

#4: If the item is a placement.

#5: If the item is a transformation operator.

AIM element: #2: (item_defined_transformation)
#3: (functionally_defined_transformation)

Source: ISO 10303-43
ISO 10303-43

Reference path: {#2: (item_defined_transformation
[item_defined_transformation.transform_item_1 ->
representation_item =>
geometric_representation_item =>
placement =>
axis2_placement_2d]
[item_defined_transformation.transform_item_2 ->
representation_item =>
geometric_representation_item =>
#4: (placement =>
axis2_placement_2d)
#5: (cartesian_transformation_operator =>
cartesian_transformation_operator_2d)])
#3: (functionally_defined_transformation =>
cartesian_transformation_operator =>
cartesian_transformation_operator_2d)}

6.1.4.29 transformation_3d

#2: If the transformation is implicitly defined by two items.

#3: If the transformation is explicitly defined.

#4: If the item is a placement.

#5: If the item is a transformation operator.

AIM element: #2: (item_defined_transformation)
#3: (functionally_defined_transformation)

Source: ISO 10303-43
ISO 10303-43

Reference path: {#2: (item_defined_transformation
[item_defined_transformation.transform_item_1 ->
representation_item =>
geometric_representation_item =>
placement =>
axis2_placement_3d]
[item_defined_transformation.transform_item_2 ->
representation_item =>
geometric_representation_item =>
#4: (placement =>
axis2_placement_3d)
#5: (cartesian_transformation_operator =>
cartesian_transformation_operator_3d)])
#3: (functionally_defined_transformation =>
cartesian_transformation_operator =>
cartesian_transformation_operator_3d)}

6.1.5 individual_definition_structure UoF

6.1.5.1 physical_item_definition

AIM element: product_definition

Source: ISO 10303-41

Rules: restrict_product_definition_context_for_product

Reference path: product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence'}

6.1.5.1.1 id

AIM element: product_definition.id

Source: ISO 10303-41

6.1.5.1.2 name

AIM element: product_definition.description

Source: ISO 10303-41

6.1.5.1.3 physical_item_definition to application_context (as additional_context)

AIM element: PATH

Reference path: product_definition <-
product_definition_context_association.definition
product_definition_context_association
{product_definition_context_association.role
product_definition_context_role.name = 'additional context'}
product_definition_context_association.frame_of_reference ->
product_definition_context

6.1.5.1.4 physical_item_definition to physical_item_version (as associated_item_version)

AIM element: PATH

Reference path: product_definition
product_definition.formation ->
product_definition_formation
{product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'physically realized product'}

6.1.5.1.5 physical_item_definition to application_context (as initial_context)

AIM element: PATH

Reference path: product_definition
product_definition.frame_of_reference ->
product_definition_context

6.1.5.1.6 physical_item_definition to multi_language_string (as name)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition

Reference path: product_definition
#1: (attribute_language_item = product_definition
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.5.2 physical_item_structure_association

AIM element: assembly_component_usage

Source: ISO 10303-44

Reference path: {assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
product_definition_relationship.name = 'physical occurrence usage'}

6.1.5.2.1 physical_item_structure_association to physical_item_definition (as related)

AIM element: PATH

Rules: restrict_product_definitions_for_product_definition_relationship

Reference path: assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
{product_definition_relationship.name = 'physical occurrence usage'}
product_definition_relationship.related_product_definition ->
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence'}

6.1.5.2.2 physical_item_structure_association to physical_item_definition (as relating)

AIM element: PATH

Rules: restrict_product_definitions_for_product_definition_relationship

Reference path: assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
{product_definition_relationship.name = 'physical occurrence usage'}
product_definition_relationship.relating_product_definition ->
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence'}

6.1.6 individual_management_data UoF

6.1.6.1 item_version_relationship

AIM element: product_definition_formation_relationship

Source: ISO 10303-41

6.1.6.1.1 description

AIM element: product_definition_formation_relationship.description

Source: ISO 10303-41

6.1.6.1.2 relation_type

AIM element: product_definition_formation_relationship.name

Source: ISO 10303-41

Reference path: product_definition_formation_relationship
{product_definition_formation_relationship.name
product_definition_formation_relationship.name = 'sequence'}

6.1.6.1.3 item_version_relationship to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string. D:\Work\Documents\Standards\STEP\10303-214\html_with_change_marks\htmls\clause5\aim_index\pxref.htm

AIM element: #1: ([product_definition_formation_relationship.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition_formation_relationship

Reference path: product_definition_formation_relationship
#1: (attribute_language_item = product_definition_formation_relationship
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition_formation_relationship
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.6.1.4 item_version_relationship to physical_item_version (as related)

AIM element: PATH

Reference path: product_definition_formation_relationship
product_definition_formation_relationship.related_product_definition_formation ->
product_definition_formation

6.1.6.1.5 item_version_relationship to physical_item_version (as relating)

AIM element: PATH

Reference path: product_definition_formation_relationship
product_definition_formation_relationship.relating_product_definition_formation ->
product_definition_formation

6.1.6.2 location

AIM element: location

Source: ISO 10303-41

6.1.6.2.1 location_id

AIM element: location.id

Source: ISO 10303-41

6.1.6.2.2 location_name

AIM element: location.name

Source: ISO 10303-41

6.1.6.2.3 location_type

AIM element: location.description

Source: ISO 10303-41

6.1.6.3 physical_item

AIM element: product

Source: ISO 10303-41

Rules: product_requires_category,
product_requires_id_owner,
product_requires_version,
restrict_product_category_for_product

Reference path: {product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name
product_category.name = 'physically realized product'}

6.1.6.3.1 description

AIM element: product.description

Source: ISO 10303-41

6.1.6.3.2 id

AIM element: product.id

Source: ISO 10303-41

6.1.6.3.3 name

AIM element: product.name

Source: ISO 10303-41

6.1.6.3.4 physical_item to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product

Reference path: product
#1: (attribute_language_item = product
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.6.3.5 physical_item to multi_language_string (as name)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product.name]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product

Reference path: product
#1: (attribute_language_item = product
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'name'})
#2: (multi_language_attribute_item = product
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=

attribute_value_assignment
attribute_value_assignment.attribute_name = 'name'})

6.1.6.4 physical_item_definition

AIM element: product_definition

Source: ISO 10303-41

Rules: restrict_product_definition_context_for_product

Reference path: product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence'}

6.1.6.4.1 id

AIM element: product_definition.id

Source: ISO 10303-41

6.1.6.4.2 name

AIM element: product_definition.description

Source: ISO 10303-41

6.1.6.4.3 physical_item_definition to application_context (as additional_context)

AIM element: PATH

Reference path: product_definition <-
product_definition_context_association.definition
product_definition_context_association
{product_definition_context_association.role
product_definition_context_role.name = 'additional context'}
product_definition_context_association.frame_of_reference ->
product_definition_context

6.1.6.4.4 physical_item_definition to physical_item_version (as associated_item_version)

AIM element: PATH

Reference path: product_definition
product_definition.formation ->
product_definition_formation
{product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'physically realized product'}

6.1.6.4.5 physical_item_definition to application_context (as initial_context)

AIM element: PATH

Reference path: product_definition
product_definition.frame_of_reference ->
product_definition_context

6.1.6.4.6 physical_item_definition to multi_language_string (as name)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition

Reference path: product_definition
#1: (attribute_language_item = product_definition
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.6.5 physical_item_location_association

AIM element: applied_location_assignment

Source: ISO 13399-1

Reference path: applied_location_assignment <=
location_assignment

6.1.6.5.1 role

AIM element: location_assignment.role

Source: ISO 10303-41

Reference path: applied_location_assignment <=
location_assignment
location_assignment.role

6.1.6.5.2 physical_item_location_assignment to physical_item_definition (as located_item)

AIM element: applied_location_assignment.located_product

Source: ISO 13399-1

Rules: restrict_product_defintion_context_for_applied_location_assignment

Reference path:
```
applied_location_assignment
applied_location_assignment.located_product ->
{product_definition
product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence'}
```

6.1.6.5.3 physical_item_location_assignment to location (as location)

AIM element: PATH

Reference path:
```
applied_location_assignment <=
location_assignment
location_assignment.location ->
location
```

6.1.6.6 physical_item_state_association

AIM element: applied_state_type_assignment

Source: ISO 10303-56

6.1.6.6.1 role

AIM element: state_type_role.name

Source: ISO 10303-56

Reference path:
```
applied_state_type_assignment
applied_state_type_assignment.role ->
state_type_role
state_type_role.name
```

6.1.6.6.2 physical_item_state_association to physical_item_definition (as associated_physical_item)

AIM element: PATH

Rules: restrict_product_defintion_context_for_applied_state_type_assignment

Reference path:
```
applied_state_type_assignment
applied_state_type_assignment.item_set[i] ->
state_type_of_item *>
applied_state_type_of_item
applied_state_type_of_item = product_definition
{product_definition
product_definition.frame_of_reference ->
product_definition_context <=
```

application_context_element
application_context_element.name = 'physical occurrence'}

6.1.6.6.3 physical_item_state_association to state (as associated_state)

AIM element: applied_state_assignment.assigned_state_type

Source: ISO 10303-56

6.1.6.7 physical_item_version

AIM element: product_definition_formation

Source: ISO 10303-41

6.1.6.7.1 description

AIM element: product_definition_formation.description

Source: ISO 10303-41

6.1.6.7.2 id

AIM element: product_definition_formation.id

Source: ISO 10303-41

6.1.6.7.3 physical_item_version to item (as associated_item)

AIM element: PATH

Reference path: product_definition_formation
product_definition_formation.of_product ->
product
{product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name
product_category.name = 'physically realized product'}

6.1.6.7.4 physical_item_version to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition_formation.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition_formation

Reference path: product_definition_formation
#1: (attribute_language_item = product_definition_formation

attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition_formation
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.6.8 realized_item_association

AIM element: product_definition_relationship

Source: ISO 10303-41

Rules: restrict_product_definitions_for_product_definition_relationship

Reference path: product_definition_relationship
{product_definition_relationship.name = 'physical realization' }

6.1.6.8.1 realized_item_association to physical_item (as physical_item)

AIM element: PATH

Reference path: product_definition_relationship
{product_definition_relationship.related_product_definition ->
product_definition
product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence'}

6.1.6.8.2 realized_item_association to item (as realized_item)

AIM element: PATH

Reference path: product_definition_relationship
{product_definition_relationship.relating_product_definition ->
product_definition
product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition' }

6.1.6.9 state

AIM element: state_type

Source: ISO 10303-56

6.1.6.9.1 name

AIM element: IDENTICAL MAPPING

6.1.7 item_definition_structure UoF

6.1.7.1 assembly_association

AIM element: assembly_component_usage

Source: ISO 10303-44

Rules: restrict_product_definitions_for_product_definition_relationship

6.1.7.1.1 assembly_association to geometric_model_relationship_with_transformation (as placement)

AIM element: PATH

Reference path: assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
[characterized_product_definition = product_definition_relationship
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
context_dependent_shape_representation.represented_product_relation
context_dependent_shape_representation
context_dependent_shape_representation.representation_relation ->
shape_representation_relationship <=]
[product_definition_relationship.relating_product_definition ->
product_definition
characterized_product_definition = product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition =>
product_definition_shape}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
{property_definition_representation =>
shape_definition_representation}
property_definition_representation.used_representation ->
representation <-
{representation =>
shape_representation}
representation_relationship.rep_2]
[product_definition_relationship.related_product_definition ->
product_definition
characterized_product_definition = product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition =>
product_definition_shape}
represented_definition = property_definition
represented_definition <-

property_definition_representation.definition
property_definition_representation
{property_definition_representation =>
shape_definition_representation}
property_definition_representation.used_representation ->
representation <-
{representation =>
shape_representation}
representation_relationship.rep_1]
representation_relationship =>
representation_relationship_with_transformation

6.1.7.1.2 assembly_association to assembly_definition (as relating)

AIM element: PATH

Reference path: assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
product_definition_relationship.relating_product_definition ->
product_definition
{product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'assembly definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}

6.1.7.2 assembly_definition

AIM element: product_definition

Source: ISO 10303-41

Rules: dependent_instantiable_product_definition_context_role

Reference path: {product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'assembly definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}

6.1.7.2.1 assembly_type

AIM element: product_definition.name

Source: ISO 10303-41

6.1.7.3 item_definition_relationship

AIM element: product_definition_relationship

Source: ISO 10303-41

6.1.7.3.1 item_definition_relationship to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition_relationship.description]
      [PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition_relationship

Reference path:
```
product_definition_relationship
#1: (attribute_language_item = product_definition_relationship
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition_relationship
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})
```

6.1.7.3.2 item_definition_relationship to item_definition (as related)

AIM element: PATH

Reference path:
```
product_definition_relationship
product_definition_relationship.related_product_definition ->
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'}
```

6.1.7.3.3 item_definition_relationship to item_definition (as relating)

AIM element: PATH

Reference path:
```
product_definition_relationship
product_definition_relationship.relating_product_definition ->
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'}
```

6.1.7.3.4 relation_type

AIM element: product_definition_relationship.name

Source: ISO 10303-41

6.1.7.4 item_instance

AIM element: assembly_component_usage

Source: ISO 10303-44

Rules: restrict_part_occurrence,
restrict_product_definition_context_for_product

6.1.7.4.1 description

AIM element: product_definition_relationship.description

Source: ISO 10303-41

Reference path: assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
product_definition_relationship.description

6.1.7.4.2 id

AIM element: product_definition_relationship.id

Source: ISO 10303-41

Reference path: assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
product_definition_relationship.id

6.1.7.4.3 item_instance to item_definition (as definition)

AIM element: PATH

Rules: restrict_product_definitions_for_product_definition_relationship

Reference path: assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
product_definition_relationship.related_product_definition ->
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'}

6.1.7.4.4 item_instance to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition_relationship.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition,
restrict_multi_language_for_product_definition_relationship

Reference path: assembly_component_usage <=
product_definition_usage <=
product_definition_relationship
#1: (attribute_language_item = product_definition_relationship
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition_relationship
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.7.5 item_structure_association

#1: If item_structure_association is an assembly_association.

AIM element: product_definition_relationship

Source: ISO 10303-41

Reference path: #1: {product_definition_relationship =>
product_definition_usage =>
assembly_component_usage}

6.1.7.5.1 item_structure_association to item_instance (as related)

#1: If item_structure_association is an assembly_association.

AIM element: #1: IDENTICAL MAPPING

Reference path: #1: {product_definition_relationship =>
product_definition_usage =>
assembly_component_usage}

6.1.7.5.2 item_structure_association to item_definition (as relating)

AIM element: PATH

Reference path: product_definition_relationship
product_definition_relationship.relating_product_definition ->
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=

application_context_element
application_context_element.name = 'part definition'}

6.1.7.6 mated_item_relationship

AIM element: product_definition_relationship

Source: ISO 10303-41

Reference path: {product_definition_relationship
product_definition_relationship.name = 'mating material'}

6.1.7.6.1 mated_item_relationship to quantified_instance (as mating_material)

AIM element: PATH

Reference path: product_definition_relationship
product_definition_relationship.related_product_definition ->
product_definition
{[product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part occurrence']
[name_attribute_select = product_definition
name_attribute_select <-
name_attribute.named_item
name_attribute
name_attribute.attribute_value = 'quantified instance']}

6.1.7.6.2 mated_item_relationship to mating_association (as related)

AIM element: PATH

Rules: dependent_instantiable_product_definition_context_role

Reference path: product_definition_relationship
product_definition_relationship.related_product_definition ->
product_definition
{product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}

6.1.7.6.3 mated_item_relationship to mating_association (as relating)

AIM element: PATH

Rules: dependent_instantiable_product_definition_context_role

Reference path: product_definition_relationship
product_definition_relationship.relating_product_definition ->
product_definition

{product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}

6.1.7.7 mating_association

AIM element: product_definition_relationship

Source: ISO 10303-41

Reference path: {product_definition_relationship
[product_definition_relationship.name = 'mating membership']
[product_definition_relationship.relating_product_definition ->
product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']]
[product_definition_relationship.related_product_definition ->
product_definition
product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}

6.1.7.7.1 mating_association to geometric_model_relationship_with_transformation (as placement)

AIM element: PATH

Rules: dependent_instantiable_product_definition_context_role

Reference path: product_definition_relationship
{product_definition_relationship.name = 'mating membership' }
[product_definition_relationship.relating_product_definition ->
product_definition
{product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element

```
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}
characterized_product_definition = product_definition
characterized_product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition =>
product_definition_shape}
property_definition
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
{property_definition_representation =>
shape_definition_representation}
property_definition_representation.used_representation ->
representation <-
representation_relationship.rep_2]
[product_definition_relationship.related_product_definition ->
product_definition
{product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}
characterized_product_definition = product_definition
characterized_product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition =>
product_definition_shape}
property_definition
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
{property_definition_representation =>
shape_definition_representation}
property_definition_representation.used_representation ->
representation <-
representation_relationship.rep_1]
representation_relationship =>
representation_relationship_with_transformation
```

6.1.7.7.2 mating_association to mating_definition (as relating)

AIM element: PATH

Rules: dependent_instantiable_product_definition_context_role

Reference path: product_definition_relationship.relating_product_definition ->
product_definition
{product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}

6.1.7.8 mating_definition

#1: If the mating_definition is the topmost node of a mating tree.

#2: If the mating_definition is a subnode of the mating tree.

AIM element: product_definition

Source: ISO 10303-41

Reference path: #1: ({product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']})
#2: (product_definition <-
{product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->
product_definition_context_role
product_definition_context_role.name = 'part definition type']}
product_definition_relationship.related_product_definition
product_definition_relationship
{[product_definition_relationship.name = 'mating membership']
[product_definition_relationship *]}
product_definition_relationship.relating_product_definition ->
product_definition
{product_definition <-
product_definition_context_association.definition
product_definition_context_association
[product_definition_context_association.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'mating definition']
[product_definition_context_association.role ->

product_definition_context_role
product_definition_context_role.name = 'part definition type']})

6.1.7.8.1 mating_type

AIM element: product_definition.name

Source: ISO 10303-41

Reference path: product_definition
(product_definition.name)
(product_definition.name = 'bolted joint')
(product_definition.name = 'brazing')
(product_definition.name = 'clasping')
(product_definition.name = 'doweling')
(product_definition.name = 'flanging')
(product_definition.name = 'press fit')
(product_definition.name = 'welding')}

6.1.7.9 quantified_instance

AIM element: quantified_assembly_component_usage

Source: ISO 10303-44

6.1.7.9.1 quantified_instance to numerical_value (as quantity)

AIM element: PATH

Rules: restrict_part_occurrence

Reference path: quantified_assembly_component_usage
quantified_assembly_component_usage.quantity ->
measure_with_unit

6.1.8 item_property UoF

6.1.8.1 coating

AIM element: product_defintion

Source: ISO 10303-41

Rules: coating_requires_product_definition,
restrict_product_definition_context_for_product

Reference path: {product_definition
[product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'material definition']
[product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=

product_category
product_category.name = 'coating']}

6.1.8.1.1 coating_name

AIM element: PATH

Reference path: product_definition <-
material_designation.definitions[i]
material_designation
material_designation.name

6.1.8.1.2 coating_system

AIM element: PATH

Reference path: product_definition <-
process_product_association.defined_product
process_product_association.process ->
product_definition_process
product_definition_process.identification

6.1.8.2 coupling

AIM element: representation

Source: ISO 13030-43

Rules: restrict_representation_for_coupling

Reference path: {representation <-
{[representation.name = 'coupling']
[representation.context_of_items ->
representation_context
representation_context.context_type = 'coupling parameters']}
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'coupling property'}}

6.1.8.2.1 coupling_type

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'coupling type'}
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description)
(descriptive_representation_item.description = '0')
(descriptive_representation_item.description = '1')

(descriptive_representation_item.description = '2')
(descriptive_representation_item.description = '3')
(descriptive_representation_item.description = '4')
(descriptive_representation_item.description = '5')
(descriptive_representation_item.description = '6')
(descriptive_representation_item.description = '7')}

6.1.8.2.2 side

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'side'}
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description)
(descriptive_representation_item.description = 'machine side')
(descriptive_representation_item.description = 'workpiece side')}

6.1.8.2.3 style

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'style'}
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description)
(descriptive_representation_item.description = '')
(descriptive_representation_item.description = '')}

6.1.8.2.4 coupling to numerical_value (as pieces)

#1: If the unit is not assigned globally.

#2: If the unit is assigned globally.

AIM element: #1: (measure_representation_item)
#2: (value_representation_item)

Source: ISO 10303-45
ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'coupling pieces'}
#1: (measure_representation_item
{measure_representation_item <=
measure_with_unit
[measure_with_unit.value_component ->

measure_value
measure_value = ratio_measure]
[measure_with_unit.unit_component ->
unit
unit = named_unit
named_unit =>
ratio_unit]})
#3: (value_representation_item)

6.1.8.2.5 coupling to numerical_value (as size)

#1: If the unit is not assigned globally.

#2: If the unit is assigned globally.

AIM element: #1: (measure_representation_item)
#2: (value_representation_item)

Source: ISO 10303-45
ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'coupling size'}
#1: (measure_representation_item
{measure_representation_item <=
measure_with_unit
[measure_with_unit.value_component ->
measure_value
measure_value = ratio_measure]
[measure_with_unit.unit_component ->
unit
unit = named_unit
named_unit =>
ratio_unit]})
#3: (value_representation_item)

6.1.8.3 cutting_condition

AIM element: representation

Source: ISO 13030-43

Rules: restrict_representation_for_cutting_condition

Reference path: {representation <-
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'cutting condition property'}
[representation.context_of_items ->
representation_context
representation_context.context_type = 'cutting condition parameters']
[representation.name = 'cutting condition']}

6.1.8.3.1 condition_name

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'condition name'}
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description)
(descriptive_representation_item.description = '')
(descriptive_representation_item.description = '')}

6.1.8.4 cutting_data_association

AIM element: representation

Source: ISO 13030-43

Rules: restrict_representation_for_cutting_data_association

Reference path: {representation <-
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'cutting data association'}
[representation.context_of_items ->
representation_context
representation_context.context_type = 'cutting data parameters']
[representation.name = 'cutting data association']}

6.1.8.4.1 cutting_data_association to material_designation (as associated_material)

AIM element: PATH

Reference path: representation <-
property_definition_representation.used_representation
property_definition_representation
{property_definition_representation.name = 'associated material'}
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
{property_definition.name = 'associated material'}
property_definition.definition ->
characterized_definition
characterized_definition = product_definition
product_definition <-
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'workpiece definition'}
material_designation.definitions[i]

material_designation
material_designation.name

6.1.8.4.2 cutting_data_association to value_with_unit (as depth_of_cut)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a value_range.

#3: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

AIM element: PATH

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'depth of cut'}
#1: (measure_representation_item)
#2: (compound_representation_item =>
value_range)
#3: (value_representation_item)

6.1.8.4.3 cutting_data_association to value_with_unit (as feed)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a value_range.

#3: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

AIM element: PATH

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'feed'}
#1: (measure_representation_item)
#2: (compound_representation_item =>
value_range)
#3: (value_representation_item)

6.1.8.4.4 cutting_data_association to value_with_unit (as speed)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a value_range.

#3: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

AIM element: PATH

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'speed'}
#1: (measure_representation_item)
#2: (compound_representation_item =>

value_range)
#3: (value_representation_item)

6.1.8.5 grade

AIM element: product_defintion

Source: ISO 10303-41

Rules: grade_requires_product_definition,
restrict_product_definition_context_for_product

Reference path: {product_definition
[product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'material definition']
[product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'grade']}

6.1.8.5.1 identifier

AIM element: product.id

Source: ISO 10303-41

Reference path: product_definition
product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product
product.id

6.1.8.5.2 standard_designation

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Rules: dependent_instantiable_identification_role

Reference path: product_definition
product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product
identification_item = product
identification_item <-
applied_identification_assignment.items[i]
applied_identification_assignment <=
identification_assignment
{identification_assignment.role ->
identification_role

identification_role.name = 'standard designation'}
identification_assignment.assigned_id

6.1.8.5.3 grade to coating (as coating)

AIM element: product_definition_relationship

Source: ISO 10303-41

Rules: restrict_product_definitions_for_product_definition_relationship

Reference path:
```
{product_definition_relationship
[product_definition_relationship.name = 'coating']
[product_definition_relationship.related_product ->
{product_definition
product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'coating'}]
[product_definition_relationship.relating_product ->
{product_definition
product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'grade'}]}
```

6.1.8.5.4 grade to cutting_condition (as cutting_condition)

AIM element: property_definition

Source: ISO 10303-41

Reference path:
```
{property_definition
[represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'cutting condition'}]
[property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition = product_definition
product_definition
{product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
```

product_category
product_category.name = 'grade'}]}

6.1.8.5.5 grade to substrate (as substrate)

AIM element: product_definition_relationship

Source: ISO 10303-41

Rules: restrict_product_definitions_for_product_definition_relationship

Reference path:
```
{product_definition_relationship
[product_definition_relationship.name = 'substrate']
[product_definition_relationship.related_product ->
{product_definition
product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'substrate'}]
[product_definition_relationship.relating_product ->
{product_definition
product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'grade'}]}
```

6.1.8.5.6 grade to material_designation (as workpiece_material)

AIM element: PATH

Reference path:
```
{product_definition_relationship
[product_definition_relationship.name = 'workpiece material']
[product_definition_relationship.related_product ->
{product_definition <-
material_designation.definitions[i]
material_designation
material_designation.name
[product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'workpiece']}]
[product_definition_relationship.relating_product ->
{product_definition
product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
```

product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'grade'}]}

6.1.8.6 grade_relationship

AIM element: product_definition_relationship

Source: ISO 10303-41

Rules: restrict_product_definitions_for_product_definition_relationship

6.1.8.6.1 relation_type

AIM element: product_definition_relationship.name

Source: ISO 10303-41

6.1.8.6.2 grade_relationship to grade (as related)

AIM element: product_definition_relationship.related_product_definition

Source: ISO 10303-41

Reference path: {product_definition_relationship.related_product_definition ->
{product_definition
[product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'material definition']
[product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'grade']}}

6.1.8.6.3 grade_relationship to grade (as relating)

AIM element: product_definition_relationship.relating_product_definition

Source: ISO 10303-41

Reference path: {product_definition_relationship.relating_product_definition ->
{product_definition
[product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'material definition']
[product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=

product_category
product_category.name = 'grade']}}

6.1.8.7 item_characteristic_association

6.1.8.7.1 item_characteristic_association to coupling (as associated_characteristic)

AIM element: property_defintion

Reference path: property_definition
{property_definition.name = 'coupling property' }
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'coupling'}

6.1.8.7.2 item_characteristic_association to cutting_condition (as associated_characteristic)

AIM element: property_defintion

Reference path: property_definition
{property_definition.name = 'cutting condition property' }
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'cutting condition'}

6.1.8.7.3 item_characteristic_association to cutting_data_association (as associated_characteristic)

AIM element: property_definition

Reference path: property_definition
{property_definition.name = 'cutting data association' }
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{representation.name = 'cutting data association'}

6.1.8.7.4 item_characteristic_association to grade (as associated_characteristic)

AIM element: product_definition_relationship

Source: ISO 10303-41

Reference path: {product_definition_relationship
[product_definition_relationship.name = 'item characteristic association']
[product_definition_relationship.related ->
product_definition
{product_definition.frame_of_reference ->

product_definition_context <=
application_context_element
application_context_element.name = 'material definition'}]}

6.1.8.7.5 item_characteristic_association to material_designation (as associated_characteristic)

AIM element: material_designation.definition

Source: ISO 10303-45

Reference path: material_designation
material_designation.definition ->
characterized_definition = characterized_product_definition
characterized_product_definition = product_definition
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'}

6.1.8.7.6 item_characteristic_association to workpiece_feature (as associated_characteristic)

AIM element: product_definition_relationship

Reference path: {product_definition_relationship
[product_definition_relationship.name = 'item characteristic association']
[product_definition_relationship.related ->
product_definition
product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'workpiece']}

6.1.8.7.7 item_characteristic_association to item_definition (as associated_item)

#1: If the associated_characteristic is a coupling, cutting_condition or a cutting_data_association

#2: If the associated_characteristic is a grade or a workpiece_feature

AIM element: #1: (property_definition.definition)
#2: (product_definition_relationship.relating)

Reference path: #1: (property_definition.definition ->
characterized_definition
characterized_definition = product_definition
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'})
#2: (product_definition_relationship.relating ->
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=

application_context_element
application_context_element.name = 'part definition'})

6.1.8.8 material_designation

AIM element: material_designation

Source: ISO 10303-45

6.1.8.8.1 material_name

AIM element: material_designation.name

Source: ISO 10303-45

6.1.8.9 property

AIM element: general_property

Source: ISO 10303-41

6.1.8.9.1 description

AIM element: general_property.description

Source: ISO 10303-41

6.1.8.9.2 id

AIM element: general_property.id

Source: ISO 10303-41

6.1.8.9.3 property_type

AIM element: general_property.name

Source: ISO 10303-41

Reference path: {(general_property.name)
(general_property.name = 'cost')
(general_property.name = 'dimension')
(general_property.name = 'mass')
(general_property.name = 'shape')
(general_property.name = 'specification')
(general_property.name = 'usage')}

6.1.8.9.4 version_id

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Reference path: general_property
identification_item = general_property
identification_item <-
applied_identification_assignment.items[i]

applied_identification_assignment <=
identification_assignment
{identification_assignment.role ->
identification_role
identification_role.name = 'version'}
identification_assignment.assigned_id

6.1.8.9.5 property to unit (as allowed_unit)

#1: If the unit has a commonly used descriptor.

#2: If the unit descriptor is constructed of more than one unit name.

AIM element: PATH

Reference path: general_property
represented_definition = general_property
represented_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
{[representation.name = 'allowed units']
[representation.items[1] ->
representation_item =>
descriptive_representation_item
descriptive_representation_item.description = 'allowed units']}
representation.context_of_items ->
representation_context =>
global_unit_assigned_context
global_unit_assigned_context.units[i] ->
unit
#1: (unit = named_unit
named_unit)
#2: (unit = derived_unit
derived_unit)

6.1.8.9.6 property to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([general_property.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_general_property

Reference path: general_property
#1: (attribute_language_item = general_property
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})

```
                    #2: (multi_language_attribute_item = general_property
                    multi_language_attribute_item <-
                    multi_language_attribute_assignment.items[i]
                    multi_language_attribute_assignment
                    {multi_language_attribute_assignment <=
                    attribute_value_assignment
                    attribute_value_assignment.attribute_name = 'description'})
```

6.1.8.9.7 property to external_library_reference (as property_source)

AIM element: IDENTICAL MAPPING

Reference path: {general_property =>
externally_defined_general_property}

6.1.8.9.8 property to plib_property_reference (as property_source)

AIM element: IDENTICAL MAPPING

Reference path: {general_property =>
externally_defined_general_property}

6.1.8.10 property_relationship

AIM element: general_property_relationship

Source: ISO 10303-41

6.1.8.10.1 description

AIM element: general_property_relationship.description

Source: ISO 10303-41

6.1.8.10.2 relation_type

AIM element: general_property_relationship.name

Source: ISO 10303-41

Reference path: {(general_property_relationship.name)
(general_property_relationship.name = 'dependency')
(general_property_relationship.name = 'hierarchy')}

6.1.8.10.3 property_relationship to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([general_property_relationship.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_general_property_relationship

Reference path: general_property_relationship
#1: (attribute_language_item = general_property_relationship
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = general_property_relationship
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.8.10.4 property_relationship to property (as related)

AIM element: PATH

Reference path: general_property_relationship
general_property_relationship.related_property ->
general_property

6.1.8.10.5 property_relationship to property (as relating)

AIM element: PATH

Reference path: general_property_relationship
general_property_relationship.relating_property ->
general_property

6.1.8.11 property_value_association

AIM element: property_definition

Source: ISO 10303-41

6.1.8.11.1 definitional

#1: If item_property_association.definitional is TRUE.

#2: If item_property_association.definitional is FALSE.

#3: If item_property_association.definitional is not instantiated.

AIM element: general_property_association.name

Source: ISO 10303-41

Reference path: property_definition
derived_property_select = property_definition
derived_property_select <-
general_property_association.derived_definition
general_property_association
general_property_association.name
{#1: (general_property_association.name = 'definitional')

#2: (general_property_association.name = 'non-definitional')
#3: (general_property_association.name = '')}

6.1.8.11.2 description

AIM element: property_definition.description

Source: ISO 10303-41

6.1.8.11.3 property_value_association to property_value_representation (as describing_property_value)

#1 If the described_element is not a material_designation.

#2 If the described_element is a material_designation

AIM element: PATH

Reference path: property_definition
represented_definition = property_definition
represented_definition <-
#1: (property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->)
#2: (material_property_representation.definition
material_property_representation
material_property_representation.used_representation ->)
representation

6.1.8.11.4 property_value_association to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([property_definition.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_property_definition

Reference path: property_definition
#1: (attribute_language_item = property_definition
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = property_definition
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.8.11.5 property_value_association to coupling (as described_element)

AIM element: property_definition_relationship

Source: ISO 10303-45

Reference path: property_definition <-
property_definition_relationship.relating_property_definition
{property_definition_relationship
[property_definition_relationship.name = 'coupling']
[property_definition_relationship.related_ property_definition ->
property_definition
{property_definition.name = 'coupling property'}]}

6.1.8.11.6 property_value_association to cutting_condition (as described_element)

AIM element: property_definition_relationship

Source: ISO 10303-45

Reference path: property_definition <-
property_definition_relationship.relating_property_definition
{property_definition_relationship
[property_definition_relationship.name = 'cutting condition']
[property_definition_relationship.related_ property_definition ->
property_definition
{property_definition.name = 'cutting condition'}]}

6.1.8.11.7 property_value_association to cutting_data_association (as described_element)

AIM element: property_definition_relationship

Source: ISO 10303-45

Reference path: property_definition <-
property_definition_relationship.relating_property_definition
{property_definition_relationship
[property_definition_relationship.name = 'cutting data association']
[property_definition_relationship.related_ property_definition ->
property_definition
{property_definition.name = 'cutting data association'}]}

6.1.8.11.8 property_value_association to document_file (as described_element)

AIM element: PATH

Reference path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
document_file

6.1.8.11.9 property_value_association to document_representation (as described_element)

AIM element: PATH

Reference path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition
characterized_product_definition = product_definition
product_definition

6.1.8.11.10 property_value_association to item_definition (as described_element)

AIM element: PATH

Reference path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition = product_definition
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'}

6.1.8.11.11 property_value_association to item_instance (as described_element)

AIM element: PATH

Reference path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition
characterized_product_definition = product_definition_relationship
product_definition_relationship =>
product_definition_usage =>
assembly_component_usage

6.1.8.11.12 property_value_association to item_structure_association (as described_element)

AIM element: PATH

Reference path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition = product_definition_relationship
product_definition_relationship

6.1.8.11.13 property_value_association to grade (as described_element)

AIM element: PATH

Reference path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition

characterized_product_definition = product_definition
product_definition

6.1.8.11.14 property_value_association to mated_item_relationship (as described_element)

AIM element: PATH

Reference path:
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition = product_definition_relationship
product_definition_relationship

6.1.8.11.15 property_value_association to material_designation (as described_element)

AIM element: PATH

Reference path:
property_definition <-
material_property_representation.definition
material_property_representation
characterized_material_property = material_property_representation
characterized_material_property <-
material_designation_characterization.property
material_designation_characterization
material_designation_characterization.designation ->
material_designation

6.1.8.11.16 property_value_association to organization (as validity_context)

AIM element: PATH

Reference path:
property_definition
organization_item = property_definition
organization_item <-
applied_organization_assignment.items[i]
applied_organization_assignment <=
organization_assignment
{organization_assignment.role ->
organization_role
organization_role.name = 'validity context'}
organization_assignment.assigned_organization ->
organization

6.1.8.11.17 property_value_association to physical_item_structure_association (as described_element)

AIM element: PATH

Reference path:
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition = product_definition_relationship
product_definition_relationship
{[product_definition_relationship.relating_product_definition ->
product_definition
product_definition.frame_of_reference ->

product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence']
[product_definition_relationship.related_product_definition ->
product_definition
product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence']}

6.1.8.11.18 property_value_association to workpiece_feature (as described_element)

AIM element: PATH

Reference path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_definition =>
general_feature

6.1.8.12 property_value_representation

AIM element: representation

Source: ISO 10303-43

6.1.8.12.1 qualifier

AIM element: type_qualifier.name

Source: ISO 10303-45

Rules: dependent_instantiable_type_qualifier

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'qualifier'}
qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier
type_qualifier.name
{(type_qualifier.name = 'nominal')
(type_qualifier.name = 'specified')
(type_qualifier.name = 'typical')}

6.1.8.12.2 value_determination

AIM element: type_qualifier.name

Source: ISO 10303-45

Rules: dependent_instantiable_type_qualifier

Reference path: representation
representation.items[i] ->
representation_item =>
{representation_item.name = 'value interpretation'}
qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier
type_qualifier.name
{(type_qualifier.name)
(type_qualifier.name = 'calculated')
(type_qualifier.name = 'designed')
(type_qualifier.name = 'estimated')
(type_qualifier.name = 'measured')
(type_qualifier.name = 'required')
(type_qualifier.name = 'set point')}

6.1.8.12.3 property_value_representation to property (as definition)

AIM element: PATH

Reference path: representation <-
property_definition_representation.used_representation
property_definition_representation
property_definition_representation.definition ->
represented_definition
represented_definition = property_definition
property_definition
derived_property_select = property_definition
derived_property_select <-
general_property_association.derived_definition
general_property_association
general_property_association.base_definition ->
general_property

6.1.8.12.4 property_value_representation to unit (as global_unit)

#1: If the unit has a commonly used descriptor.

#2: If the unit descriptor is constructed of more than one unit name.

AIM element: PATH

Reference path: representation
representation.context_of_items ->
representation_context =>
global_unit_assigned_context
global_unit_assigned_context.units[i] ->
unit
#1: (unit = named_unit
named_unit)
#2: (unit = derived_unit
derived_unit)

6.1.8.12.5 property_value_representation to property_value (as specified_value)

#1: If the specified_value is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the specified_value is a value_range.

#3: If the specified_value is a string_value.

#4: If no significant_digits are given for specified_value.

#5: If significant_digits are given for specified_value.

#6: If the specified_value is a numerical_value or a value_limit and the unit is specified globally.

#7: If the specified_value is a value_list.

AIM element: PATH

Reference path:
```
representation
representation.items[i] ->
representation_item =>
#3: (descriptive_representation_item
{descriptive_representation_item <=
representation_item =>
qualified_representation_item
qualified_representation_item.qualifiers[1] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier
type_qualifier.name
type_qualifier.name = 'string'})
#1: (#4: (measure_representation_item)
#5: ([measure_representation_item]
[qualified_representation_item]))
#2: (#4: (compound_representation_item =>
value_range)
#5: ([compound_representation_item =>
value_range]
[qualified_representation_item]))
#6: (#4: (value_representation_item)
#5: ([value_representation_item]
[qualified_representation_item]))
#7: (compound_representation_item)
```

6.1.8.13 property_value_representation_relationship

AIM element: representation_relationship

Source: ISO 10303-43

6.1.8.13.1 description

AIM element: representation_relationship.description

Source: ISO 10303-43

6.1.8.13.2 relation_type

AIM element: representation_relationship.name

Source: ISO 10303-43

6.1.8.13.3 property_value_representation_relationship to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([representation_relationship.description]
[PATH])
#2: (PATH)

Source: ISO 10303-43

Rules: restrict_multi_language_for_representation_relationship

Reference path: representation_relationship
#1: (attribute_language_item = representation_relationship
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = representation_relationship
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.8.13.4 property_value_representation_relationship to property_value_representation (as related)

AIM element: PATH

Reference path: representation_relationship
representation_relationship.rep_2 ->
representation

6.1.8.13.5 property_value_representation_relationship to property_value_representation (as relating)

AIM element: PATH

Reference path: representation_relationship
representation_relationship.rep_1 ->
representation

6.1.8.14 substrate

AIM element: product_defintion

Source: ISO 10303-41

Rules: substrate_requires_product_definition,
restrict_product_definition_context_for_product

Reference path: {product_definition
[product_definition.frame_of_reference ->
product_definition_context <=

application_context_element
application_context_element.name = 'material definition']
[product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'substrate']}

6.1.8.14.1 name

AIM element: PATH

Reference path: product_definition <-
material_designation.definitions[i]
material_designation
material_designation.name

6.1.8.15 workpiece_feature

AIM element: general_feature

Source: ISO 10303-214

Reference path: {general_feature <=
[shape_aspect
shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_product_definition
characterized_product_definition
characterized_product_definition = product_definition
product_definition product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'workpiece']
[feature_definition <=
characterized_object]}

6.1.8.15.1 description

AIM element: characterized_object.description

Source: ISO 10303-41

6.1.8.15.2 id

AIM element: characterized_object.name

Source: ISO 10303-41

6.1.8.15.3 workpiece_feature to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([characterized_object.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_general_feature

Reference path: general_feature
#1: (attribute_language_item = general_feature
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = general_feature
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.8.15.4 workpiece_feature to external_model (as representation)

AIM element: applied_document_reference

Source: ISO 10303-214

Reference path: {applied_document_reference <=
document_reference
[applied_document_reference.items[i] ->
document_reference_item
document_reference_item = general_feature
general_feature]
[document_reference.assigned_document ->
document_file <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition
{property_definition.name = 'external definition'}
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
property_definition_representation.used_representation ->
representation
{representation.context_of_items ->
representation_context
representation_context.context_type = 'external'}]}

6.1.9 product_management_data UoF

6.1.9.1 alias_identification

AIM element: applied_identification_assignment

Source: ISO 10303-214

Reference path: applied_identification_assignment <=
identification_assignment
{identification_assignment
identification_assignment.role ->
identification_role
identification_role.name = 'alias'}

6.1.9.1.1 alias_id

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Reference path: applied_identification_assignment <=
identification_assignment
identification_assignment.assigned_id

6.1.9.1.2 alias_version_id

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Rules: dependent_instantiable_identification_role,
restrict_version_assignment_for_applied_identification_assignment

Reference path: applied_identification_assignment
identification_item = applied_identification_assignment
identification_item <-
applied_identification_assignment.items[i]
applied_identification_assignment <=
identification_assignment
{identification_assignment.role ->
identification_role
identification_role.name = 'version'}
identification_assignment.assigned_id

6.1.9.1.3 description

AIM element: identification_role.description

Source: ISO 10303-41

Rules: dependent_instantiable_identification_role

Reference path: applied_identification_assignment <=
identification_assignment
identification_assignment.role ->
identification_role
identification_role.description

6.1.9.1.4 alias_identification to organization (as alias_scope)

AIM element: PATH

Reference path: applied_identification_assignment
organization_item = applied_identification_assignment
organization_item <-
applied_organization_assignment.items[i]
applied_organization_assignment <=
organization_assignment
{organization_assignment.role ->
organization_role
organization_role.name = 'alias scope'}
organization_assignment.assigned_organization ->
organization

6.1.9.1.5 alias_identification to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([identification_role.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_identification_role

Reference path: identification_role
#1: (attribute_language_item = identification_role
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = identification_role
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.9.1.6 alias_identification to classification_attribute (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = property_definition
property_definition

6.1.9.1.7 alias_identification to classification_system (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = class_system
class_system

6.1.9.1.8 alias_identification to document (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = product
product
{product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.9.1.9 alias_identification to document_representation (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = product_definition
product_definition

6.1.9.1.10 alias_identification to document_type_property (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = document_type
document_type

6.1.9.1.11 alias_identification to document_version (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = product_definition_formation
product_definition_formation
{product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.9.1.12 alias_identification to general_classification (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = class
class

6.1.9.1.13 alias_identification to grade (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = product_definition
product_definition
{product_definition
[product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'material definition']
[product_definition.formation ->
product_definition_formation
product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'grade']}

6.1.9.1.14 alias_identification to item (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = product
product

6.1.9.1.15 alias_identification to item_definition (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = product_definition
product_definition

6.1.9.1.16 alias_identification to item_instance (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = product_definition_relationship
product_definition_relationship =>
product_definition_usage =>
assembly_component_usage

6.1.9.1.17 alias_identification to item_version (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = product_definition_formation
product_definition_formation

6.1.9.1.18 alias_identification to organization (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = organization
organization

6.1.9.1.19 alias_identification to property (as is_applied_to)

AIM element: PATH

Reference path: applied_identification_assignment
applied_identification_assignment.items[i] ->
identification_item
identification_item = general_property
general_property

6.1.9.2 application_context

AIM element: product_definition_context

Source: ISO 10303-41

6.1.9.2.1 application_domain

AIM element: application_context.application

Source: ISO 10303-41

Reference path: product_definition_context <=
application_context_element
application_context_element.frame_of_reference ->
application_context
application_context.application
{(application_context.application)
(application_context.application = 'machining')
(application_context.application = 'manufacturing simulation')

(application_context.application = 'nc preparation')
(application_context.application = 'process planning')
(application_context.application = 'product planning')
(application_context.application = 'purchasing')}

6.1.9.2.2 description

AIM element: application_context.description

Source: ISO 10303-41

Reference path: product_definition_context <=
application_context_element
application_context_element.frame_of_reference ->
application_context
application_context.description

6.1.9.2.3 application_context to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([application_context.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_application_context

Reference path: product_definition_context <=
application_context_element
application_context_element.frame_of_reference ->
application_context
#1: (attribute_language_item = application_context
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = application_context
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.9.3 date_time

#1: If only a certain day is known.

#2: If a certain day and the time of day is known.

AIM element: #1: (calendar_date)
#2: (date_and_time)

Source: ISO 10303-41
ISO 10303-41

Rules: dependent_instantiable_date,
dependent_instantiable_date_and_time

6.1.9.3.1 date

#1: If only a certain day is known.

#2: If a certain day and the time of day is known.

AIM element: #1: (IDENTICAL MAPPING)
#2: (calendar_date)

Reference path: #2: (date_and_time
date_and_time.date_component ->
date =>
calendar_date)

6.1.9.3.2 time

AIM element: local_time

Source: ISO 10303-41

Reference path: date_and_time
date_and_time.time_component ->
local_time

6.1.9.4 item

AIM element: product

Source: ISO 10303-41

Rules: product_requires_category,
product_requires_id_owner,
product_requires_version,
restrict_product_category_for_product

Reference path: {product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name
(product_category.name = 'part')
(product_category.name = 'raw material')
(product_category.name = 'tool')}

6.1.9.4.1 description

AIM element: product.description

Source: ISO 10303-41

6.1.9.4.2 id

AIM element: product.id

Source: ISO 10303-41

6.1.9.4.3 name

AIM element: product.name

Source: ISO 10303-41

6.1.9.4.4 item to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product

Reference path: product
#1: (attribute_language_item = product
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.9.4.5 item to multi_language_string (as name)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product.name]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product

Reference path: product
#1: (attribute_language_item = product

attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'name'})
#2: (multi_language_attribute_item = product
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'name'})

6.1.9.5 item_definition

AIM element: product_definition

Source: ISO 10303-41

Rules: restrict_product_definition_context_for_product

Reference path: product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'}

6.1.9.5.1 id

AIM element: product_definition.id

Source: ISO 10303-41

6.1.9.5.2 name

AIM element: product_definition.description

Source: ISO 10303-41

6.1.9.5.3 item_definition to application_context (as additional_context)

AIM element: PATH

Reference path: product_definition <-
product_definition_context_association.definition
product_definition_context_association
{product_definition_context_association.role
product_definition_context_role.name = 'additional context'}
product_definition_context_association.frame_of_reference ->
product_definition_context

6.1.9.5.4 item_definition to item_version (as associated_item_version)

AIM element: PATH

Reference path: product_definition
product_definition.formation ->
product_definition_formation

6.1.9.5.5 item_definition to application_context (as initial_context)

AIM element: PATH

Reference path: product_definition
product_definition.frame_of_reference ->
product_definition_context

6.1.9.5.6 item_definition to multi_language_string (as name)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition

Reference path: product_definition
#1: (attribute_language_item = product_definition
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.9.6 item_version

AIM element: product_definition_formation

Source: ISO 10303-41

6.1.9.6.1 description

AIM element: product_definition_formation.description

Source: ISO 10303-41

6.1.9.6.2 id

AIM element: product_definition_formation.id

Source: ISO 10303-41

6.1.9.6.3 item_version to item (as associated_item)

AIM element: PATH

Reference path: product_definition_formation
product_definition_formation.of_product ->
product

6.1.9.6.4 item_version to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_definition_formation.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition_formation

Reference path: product_definition_formation
#1: (attribute_language_item = product_definition_formation
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition_formation
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.9.7 item_version_relationship

AIM element: product_definition_formation_relationship

Source: ISO 10303-41

6.1.9.7.1 description

AIM element: product_definition_formation_relationship.description

Source: ISO 10303-41

6.1.9.7.2 relation_type

AIM element: product_definition_formation_relationship.name

Source: ISO 10303-41

Reference path: {product_definition_formation_relationship
(product_definition_formation_relationship.name)
(product_definition_formation_relationship.name = 'derivation')
(product_definition_formation_relationship.name = 'hierarchy')
(product_definition_formation_relationship.name = 'sequence')
(product_definition_formation_relationship.name = 'supplied item')}

6.1.9.7.3 item_version_relationship to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string. D:\Work\Documents\Standards\STEP\10303-214\html_with_change_marks\htmls\clause5\aim_index\pxref.htm

AIM element: #1: ([product_definition_formation_relationship.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_definition_formation_relationship

Reference path: product_definition_formation_relationship
#1: (attribute_language_item = product_definition_formation_relationship
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_definition_formation_relationship
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.9.7.4 item_version_relationship to item_version (as related)

AIM element: PATH

Reference path: product_definition_formation_relationship
product_definition_formation_relationship.related_product_definition_formation ->
product_definition_formation

6.1.9.7.5 item_version_relationship to item_version (as relating)

AIM element: PATH

Reference path: product_definition_formation_relationship
product_definition_formation_relationship.relating_product_definition_formation ->
product_definition_formation

6.1.9.8 language

AIM element: language

Source: ISO 10303-214

Reference path: language <=
group

6.1.9.8.1 country_code

AIM element: group.description

Source: ISO 10303-41

Reference path: language <=
group
group.description

6.1.9.8.2 language_code

AIM element: group.name

Source: ISO 10303-41

Reference path: language <=
group
group.name

6.1.9.9 multi_language_string

#1: If additional_language_dependent_string is not populated.

#2: If additional_language_dependent_string is populated.

AIM element: #1: (attribute_language_assignment)
#2: ([attribute_language_assignment]
[multi_language_attribute_assignment])

Source: ISO 10303-214
ISO 10303-214
ISO 10303-214

Rules: dependent_instantiable_attribute_value_role

Reference path: #1: (attribute_language_assignment <=
attribute_classification_assignment
{attribute_classification_assignment.role ->
classification_role
classification_role.name = 'primary'})
#2: ([attribute_language_assignment <=
attribute_classification_assignment
{attribute_classification_assignment.role ->
classification_role
classification_role.name = 'primary'}]
[multi_language_attribute_assignment <=
attribute_value_assignment
{attribute_value_assignment.role ->
attribute_value_role
attribute_value_role.name = 'alternate language'}
attribute_language_item = attribute_value_assignment

attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment <=
attribute_classification_assignment
{[attribute_classification_assignment.role ->
classification_role
classification_role.name = 'translated']
[attribute_classification_assignment.attribute_name = 'attribute_value']}])

6.1.9.9.1 multi_language_string to string_with_language (as additional_language_dependent_string)

AIM element: IDENTICAL MAPPING

6.1.9.9.2 multi_language_string to string_with_language (as primary_language_dependent_string)

AIM element: IDENTICAL MAPPING

6.1.9.10 numerical_value

#1: If significant digits are given for the value_with_unit.

#2: If no significant digits are given for the value_with_unit.

#3: If the unit is not assigned globally.

#4: If the unit is assigned globally.

#5: For the quantity of a quantified_instamce.

AIM element: #3: (#1: ([measure_representation_item]
[qualified_representation_item])
#2: (measure_representation_item))
#4: (#1: ([value_representation_item]
[qualified_representation_item])
#2: (value_representation_item))
#5: ([measure_representation_item])

Source: ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-43
ISO 10303-45
ISO 10303-43
ISO 10303-45

Rules: dependent_instantiable_measure_with_unit

6.1.9.10.1 value_component

#1: If the unit is not assigned globally.

#2: If the unit is assigned globally

AIM element: #1: (measure_with_unit.value_component)
#2: (value_representation_item.value_component)

Source: ISO 10303-41
ISO 10303-43

Reference path:

```
#1: ({measure_representation_item <=
measure_with_unit
{(|measure_with_unit|)
(measure_with_unit =>
(amount_of_substance_measure_with_unit)
(area_measure_with_unit)
(celsius_temperature_measure_with_unit)
(electric_current_measure_with_unit)
(length_measure_with_unit)
(luminous_intensity_measure_with_unit)
(mass_measure_with_unit)
(plane_angle_measure_with_unit)
(ratio_measure_with_unit)
(solid_angle_measure_with_unit)
(thermodynamic_temperature_measure_with_unit)
(time_measure_with_unit)
(volume_measure_with_unit))}
measure_with_unit.value_component
{measure_with_unit.value_component ->
measure_value
(measure_value = area_measure
area_measure)
(measure_value = amount_of_substance_measure
amount_of_substance_measure)
(measure_value = celsius_temperature_measure
celsius_temperature_measure)
(measure_value = length_measure
length_measure)
(measure_value = electric_current_measure
electric_current_measure)
(measure_value = plane_angle_measure
plane_angle_measure)
(measure_value = ratio_measure
ratio_measure)
(measure_value = parameter_value
parameter_value)
(measure_value = descriptive_measure
descriptive_measure)
(measure_value = positive_plane_angle_measure
positive_plane_angle_measure)
(measure_value = count_measure
count_measure)
(measure_value = mass_measure
mass_measure)
(measure_value = time_measure
time_measure)
(measure_value = thermodynamic_temperature_measure
thermodynamic_temperature_measure)
(measure_value = luminous_intensity_measure
luminous_intensity_measure)
(measure_value = solid_angle_measure
solid_angle_measure)
(measure_value = volume_measure
volume_measure)
(measure_value = numeric_measure
numeric_measure)
(measure_value = context_dependent_measure
context_dependent_measure)
(measure_value = positive_length_measure
positive_length_measure)
(measure_value = positive_ratio_measure
positive_ratio_measure)}})
#2: (value_representation_item
value_representation_item.value_component)
```

6.1.9.11 organization

AIM element: organization

Source: ISO 10303-41

6.1.9.11.1 delivery_address

AIM element: organizational_address

Source: ISO 10303-41

Reference path:
```
organization <-
organizational_address.organizations[i]
organizational_address
{organizational_address.description = 'delivery address'}
```

6.1.9.11.2 id

AIM element: organization.id

Source: ISO 10303-41

6.1.9.11.3 organization_name

AIM element: organization.name

Source: ISO 10303-41

6.1.9.11.4 organization_type

AIM element: organization.description

Source: ISO 10303-41

6.1.9.11.5 postal_address

AIM element: organizational_address

Source: ISO 10303-41

Reference path:
```
organization <-
organizational_address.organizations[i]
organizational_address
{organizational_address.description = 'postal address'}
```

6.1.9.11.6 visitor_address

AIM element: organizational_address

Source: ISO 10303-41

Reference path: organization <-
organizational_address.organizations[i]
organizational_address
{organizational_address.description = 'visitor address'}

6.1.9.12 person

AIM element: person

Source: ISO 10303-41

Rules: person_requires_person_and_organization

Reference path: {person <-
person_and_organization.the_person
person_and_organization}

6.1.9.12.1 person_name

AIM element: <person.last_name>
<person.first_name>
<person.middle_names>

Source: ISO 10303-41
ISO 10303-41
ISO 10303-41

6.1.9.12.2 preferred_business_address

AIM element: personal_address

Source: ISO 10303-41

Reference path: person <-
personal_address.people[i]
personal_address

6.1.9.13 person_in_organization

AIM element: person_and_organization

Source: ISO 10303-41

6.1.9.13.1 id

AIM element: identification_assignment.assigned_id

Source: ISO 10303-41

Reference path: person_and_organization
identification_item = person_and_organization
identification_item <-
applied_identification_assignment.items[i]
applied_identification_assignment <=
identification_assignment
{identification_assignment.role ->

identification_role
identification_role.name = 'id'}
identification_assignment.assigned_id

6.1.9.13.2 location

AIM element: person_and_organization_address

Source: ISO 10303-214

Reference path: person_and_organization
[person_and_organization.the_person ->
person <-
personal_address.people[1]
personal_address =>]
[person_and_organization.the_organization ->
organization <-
organizational_address.organizations[1]
organizational_address =>]
person_and_organization_address
{person_and_organization_address <=
[organizational_address]
[personal_address]}

6.1.9.13.3 role

AIM element: person_and_organization.name

Source: ISO 10303-41

6.1.9.13.4 person_in_organization to organization (as associated_organization)

AIM element: PATH

Reference path: person_and_organization
person_and_organization.the_organization ->
organization

6.1.9.13.5 person_in_organization to person (as associated_person)

AIM element: PATH

Reference path: person_and_organization
person_and_organization.the_person ->
person

6.1.9.14 person_organization_assignment

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: #1: (applied_person_and_organization_assignment)
#2: (applied_organization_assignment)

Source: ISO 10303-214
ISO 10303-214

Reference path: #1: (applied_person_and_organization_assignment <=
person_and_organization_assignment)
#2: (applied_organization_assignment <=
organization_assignment)

6.1.9.14.1 description

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: #1: (person_and_organization_role.description)
#2: (organization_role.description)

Source: ISO 10303-41
ISO 10303-41

Rules: dependent_instantiable_organization_role,
dependent_instantiable_person_and_organization_role

Reference path: #1: (applied_person_and_organization_assignment <=
person_and_organization_assignment
person_and_organization_assignment.role ->
person_and_organization_role
person_and_organization_role.description)
#2: (applied_organization_assignment <=
organization_assignment
organization_assignment.role ->
organization_role
organization_role.description)

6.1.9.14.2 role

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: #1: (person_and_organization_role.description)
#2: (organization_role.description)

Source: ISO 10303-41
ISO 10303-41

Rules: dependent_instantiable_organization_role,
dependent_instantiable_person_and_organization_role

Reference path: #2: (applied_person_and_organization_assignment <=
person_and_organization_assignment
person_and_organization_assignment.role ->
person_and_organization_role
person_and_organization_role.name
{(person_and_organization_role.name)
(person_and_organization_role.name = 'author')
(person_and_organization_role.name = 'classification officer')
(person_and_organization_role.name = 'creator')
(person_and_organization_role.name = 'custodian')
(person_and_organization_role.name = 'customer')
(person_and_organization_role.name = 'design supplier')
(person_and_organization_role.name = 'editor')

```
(person_and_organization_role.name = 'editor rights')
(person_and_organization_role.name = 'id owner')
(person_and_organization_role.name = 'location')
(person_and_organization_role.name = 'owner')
(person_and_organization_role.name = 'supplier')})
#3: (applied_organization_assignment <=
organization_assignment
organization_assignment.role ->
organization_role
organization_role.name
{(organization_role.name)
(organization_role.name = 'author')
(organization_role.name = 'classification officer')
(organization_role.name = 'creator')
(organization_role.name = 'custodian')
(organization_role.name = 'customer')
(organization_role.name = 'design supplier')
(organization_role.name = 'editor')
(organization_role.name = 'id owner')
(organization_role.name = 'location')
(organization_role.name = 'owner')
(organization_role.name = 'supplier')})
```

6.1.9.14.3 person_organization_assignment to organization (as assigned_person_organization)

AIM element: PATH

Reference path:

```
applied_organization_assignment <=
organization_assignment
organization_assignment.assigned_organization ->
organization
```

6.1.9.14.4 person_organization_assignment to person_in_organization (as assigned_person_organization)

AIM element: PATH

Reference path:

```
applied_person_and_organization_assignment <=
person_and_organization_assignment
person_and_organization_assignment.assigned_person_and_organization ->
person_and_organization
```

6.1.9.14.5 person_organization_assignment to multi_language_string (as description)

#1: If used for a person as member of an organization.

#2: If used for an organization.

#3: For the primary_language_dependent_string.

#4: For any additional_language_dependent_string.

AIM element:

```
#1: (#3: ([person_and_organization_role.description]
[PATH])
#4: (PATH))
#2: (#3: ([organization_role.description]
[PATH])
#4: (PATH))
```

Source: ISO 10303-41

Rules: restrict_multi_language_for_organization_role,
restrict_multi_language_for_person_and_organization_role

Reference path:
```
#1: (applied_person_and_organization_assignment <=
person_and_organization_assignment
person_and_organization_assignment.role ->
person_and_organization_role
#3: (attribute_language_item = person_and_organization_role
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#4: (multi_language_attribute_item = person_and_organization_role
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'}))
#2: (applied_organization_assignment <=
organization_assignment
organization_assignment.role ->
organization_role
#3: (attribute_language_item = organization_role
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#4: (multi_language_attribute_item = organization_role
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'}))
```

6.1.9.14.6 person_organization_assignment to classification_association (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path:
```
#1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = applied_classification_assignment)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = applied_classification_assignment)
```

applied_classification_assignment <=
classification_assignment

6.1.9.14.7 person_organization_assignment to classification_system (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = class_system)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = class_system)
class_system

6.1.9.14.8 person_organization_assignment to document (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product)
product
{product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.9.14.9 person_organization_assignment to document_file (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = document_file)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item

organization_item = document_file)
document_file

6.1.9.14.10 person_organization_assignment to document_representation (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product_definition
product_definition)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product_definition
product_definition)

6.1.9.14.11 person_organization_assignment to document_version (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product_definition_formation)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product_definition_formation)
product_definition_formation
{product_definition_formation.of_product ->
product <-
product_related_product_category.products[i]
product_related_product_category <=
product_category
product_category.name = 'document'}

6.1.9.14.12 person_organization_assignment to general_classification (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = class)
#2: (applied_organization_assignment

applied_organization_assignment.items[i] ->
organization_item
organization_item = class)
class

6.1.9.14.13 person_organization_assignment to item (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product)
product

6.1.9.14.14 person_organization_assignment to item_definition (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product_definition)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product_definition)
product_definition
{product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'part definition'}

6.1.9.14.15 person_organization_assignment to item_structure_association (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product_definition_relationship)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->

organization_item
organization_item = product_definition_relationship)
product_definition_relationship

6.1.9.14.16 person_organization_assignment to item_instance (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product_definition_relationship)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product_definition_relationship)
product_definition_relationship =>
product_definition_usage =>
assembly_component_usage

6.1.9.14.17 person_organization_assignment to item_version (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product_definition_formation)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product_definition_formation)
product_definition_formation

6.1.9.14.18 person_organization_assignment to item_version_relationship (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product_definition_formation_relationship)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product_definition_formation_relationship)
product_definition_formation_relationship

6.1.9.14.19 person_organization_assignment to material_designation (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = material_designation)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = material_designation)
material_designation

6.1.9.14.20 person_organization_assignment to person_in_organization (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = person_and_organization)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = person_and_organization)
person_and_organization

6.1.9.14.21 person_organization_assignment to physical_item_structure_association (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = product_definition_relationship)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = product_definition_relationship)
product_definition_relationship

6.1.9.14.22 person_organization_assignment to property (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = general_property)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = general_property)
general_property

6.1.9.14.23 person_organization_assignment to property_value_association (as is_applied_to)

#1: If used for a person as member of an organization.

#2: If used for an organization.

AIM element: PATH

Reference path: #1: (applied_person_and_organization_assignment
applied_person_and_organization_assignment.items[i] ->
person_and_organization_item
person_and_organization_item = property_definition
property_definition)
#2: (applied_organization_assignment
applied_organization_assignment.items[i] ->
organization_item
organization_item = property_definition
property_definition)

6.1.9.15 property_value

#1: If property_value is a value_list.

#2: If property_value is a string_value.

#3: If property_value is a numerical_value or a value_limit and the unit is not assigned globally.

#4: If property_value is a value_range.

#5: If no significant_digits are given for the value_with_unit.

#6: If significant_digits are given for the value_with_unit.

#7: If the property_value is a numerical_value or a value_limit and the unit is specified globally.

AIM element: #1: (compound_representation_item)
#2: (descriptive_representation_item)
#3: (#5: (measure_representation_item)
#6: ([measure_representation_item]
[qualified_representation_item]))
#4: (#5: (value_range)
#6: ([value_range]
[qualified_representation_item]))
#7: (#5: (value_representation_item)

#6: ([value_representation_item]
[qualified_representation_item]))

Source:
ISO 10303-43
ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-214
ISO 10303-214
ISO 10303-45
ISO 10303-43
ISO 10303-43
ISO 10303-45

Reference path:
#2: (descriptive_representation_item
{descriptive_representation_item <=
representation_item =>
qualified_representation_item
qualified_representation_item.qualifiers[1] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier
type_qualifier.name
type_qualifier.name = 'string'})
#4: (#5: (value_range <=
compound_representation_item)
#6: ([value_range <=
compound_representation_item]
[qualified_representation_item]))

6.1.9.15.1 value_name

#1: If property_value is a value_list.

#2: If property_value is a string_value.

#3: If property_value is a numerical_value or a value_limit and the unit is not assigned globally.

#4: If property_value is a value_range.

#5: If the property_value is a numerical_value or a value_limit and the unit is specified globally.

AIM element: representation_item.name

Source: ISO 10303-43

Reference path:
#1: (compound_representation_item <=)
#2: (descriptive_representation_item <=)
#3: (measure_representation_item <=)
#4: (value_range <=
compound_representation_item <=)
#5: (value_representation_item <=)
representation_item
representation_item.name

6.1.9.16 specific_item_classification

AIM element: product_related_product_category

Source: ISO 10303-41

6.1.9.16.1 classification_name

AIM element: product_category.name

Source: ISO 10303-41

Reference path: product_related_product_category <=
product_category
product_category.name
{(product_category.name)
(product_category.name = 'accessory item')
(product_category.name = 'adaptive item')
(product_category.name = 'assembly')
(product_category.name = 'assembly item')
(product_category.name = 'cutting item')
(product_category.name = 'cutting tool')
(product_category.name = 'detail')
(product_category.name = 'part')
(product_category.name = 'tool item')}

6.1.9.16.2 description

AIM element: product_category.description

Source: ISO 10303-41

Reference path: product_related_product_category <=
product_category
product_category.description

6.1.9.16.3 specific_item_classification to item (as associated_item)

AIM element: PATH

Reference path: product_related_product_category
product_related_product_category.products[i] ->
product

6.1.9.16.4 specific_item_classification to multi_language_string (as description)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([product_category.description]
[PATH])
#2: (PATH)

Source: ISO 10303-41

Rules: restrict_multi_language_for_product_related_product_category

Reference path: product_related_product_category
#1: (attribute_language_item = product_related_product_category
attribute_language_item <-
attribute_language_assignment.items[i]

attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = product_related_product_category
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.9.17 specific_item_classification_hierarchy

AIM element: product_category_relationship

Source: ISO 10303-41

Reference path: {product_category_relationship
product_category_relationship.name = 'hierarchy'}

6.1.9.17.1 specific_item_classification_hierarchy to specific_item_classification (as sub_classification)

AIM element: PATH

Reference path: product_category_relationship
product_category_relationship.sub_category ->
product_category =>
product_related_product_category

6.1.9.17.2 specific_item_classification_hierarchy to specific_item_classification (as super_classification)

AIM element: PATH

Reference path: product_category_relationship
product_category_relationship.category ->
product_category =>
product_related_product_category

6.1.9.18 string_value

AIM element: descriptive_representation_item

Source: ISO 10303-45

Reference path: {descriptive_representation_item <=
representation_item =>
qualified_representation_item
qualified_representation_item.qualifiers[1] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier
type_qualifier.name
type_qualifier.name = 'string'}

6.1.9.18.1 value_specification

AIM element: descriptive_representation_item.description

Source: ISO 10303-45

6.1.9.18.2 string_value to multi_language_string (as value_specification)

#1: For the primary_language_dependent_string.

#2: For any additional_language_dependent_string.

AIM element: #1: ([descriptive_representation_item.description]
[PATH])
#2: (PATH)

Source: ISO 10303-45

Reference path: descriptive_representation_item
#1: (attribute_language_item = descriptive_representation_item
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment
{attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.attribute_name = 'description'})
#2: (multi_language_attribute_item = descriptive_representation_item
multi_language_attribute_item <-
multi_language_attribute_assignment.items[i]
multi_language_attribute_assignment
{multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_name = 'description'})

6.1.9.19 string_with_language

#1: If the string_with_language does not play the role of a additional_language_dependent_string.

#2: if the string_with_language play the role of an additional_language_dependent_string.

AIM element: #1: (attribute_language_assignment)
#2: ([attribute_language_assignment]
[multi_language_attribute_assignment])

Source: ISO 10303-214
ISO 10303-214
ISO 10303-214

Reference path: #1: (attribute_language_assignment <=
attribute_classification_assignment
{attribute_classification_assignment.role ->
classification_role
classification_role.name = 'primary'})
#2: ([attribute_language_assignment <=
attribute_classification_assignment
{attribute_classification_assignment.role ->
classification_role
classification_role.name = 'primary'}]
[multi_language_attribute_assignment <=

attribute_value_assignment
{attribute_value_assignment.role ->
attribute_value_role
attribute_value_role.name = 'alternate language'}
attribute_language_item = attribute_value_assignment
attribute_language_item <-
attribute_language_assignment.items[i]
attribute_language_assignment <=
attribute_classification_assignment
{[attribute_classification_assignment.role ->
classification_role
classification_role.name = 'translated']
[attribute_classification_assignment.attribute_name = 'attribute_value']}])

6.1.9.19.1 value

AIM element: #1: (IDENTICAL MAPPING)
#2: (attribute_value_assignment.attribute_value)

Reference path: #2: (multi_language_attribute_assignment <=
attribute_value_assignment
attribute_value_assignment.attribute_value)

6.1.9.19.2 string_with_language to language (as language_specification)

AIM element: PATH

Reference path: attribute_language_assignment <=
attribute_classification_assignment
attribute_classification_assignment.assigned_class ->
group =>
language

6.1.9.20 unit

#1: If the unit has a commonly used descriptor.

#2: If the unit descriptor is constructed of more than one unit name.

#3: If the unit is defined with respect to the system of units defined in ISO 10403-41.

#4: If the unit is defined with respect to another named_unit by means of a conversion factor.

#5: If the unit is dependent on the context of its usage.

#6: In order to express a distance.

#7: In order to express an amount of matter.

#8: In order to express the duration of periods.

#9: In order to express the movement of electrically charged particles.

#10: In order to express the degree of heat of a body.

#11: In order to express a quantity of substance in numbers of atoms.

#12: In order to express the brightness of a body.

#13: In order to express angles in planes.

#14: In order to express solid angles.

#15: In order to express the extent of a surface.

#16: In order to express the solid content of a body.

#17: In order to express the ratio between two quantities of the same kind.

#18: If the named unit is represented by an explicit expression.

#19: If the context dependent unit is represented by an explicit expression.

#20: If the derived unit is represented by an explicit expression.

AIM element: #1: (named_unit)
#2: (derived_unit)

Source: ISO 10303-41
ISO 10303-41

Rules: dependent_instantiable_derived_unit,
dependent_instantiable_named_unit

Reference path: {#1: (named_unit)
#18: (named_unit =>
named_unit_variable <=
variable_semantics)
#19: (named_unit =>
context_dependent_unit =>
expression_conversion_based_unit <=
variable_semantics)
(named_unit =>
<#3: (si_unit)
#4: (conversion_based_unit)
#5: (context_dependent_unit)>
<#6: (length_unit)
#7: (mass_unit)
#8: (time_unit)
#9: (electric_current_unit)
#10: (thermodynamic_temperature_unit)
#11: (amount_of_substance_unit)
#12: (luminous_intensity_unit)
#13: (plane_angle_unit)
#14: (solid_angle_unit)
#15: (area_unit)
#16: (volume_unit)
#17: (ratio_unit)>)
#2: (derived_unit)
#20: (derived_unit =>
derived_unit_variable <=
variable_semantics)}

6.1.9.20.1 unit_name

AIM element: #1: (#3: (si_unit.name)
#4: (conversion_based_unit.name)

#5: (context_dependent_unit.name))
#2: (derived_unit.name)

Source: ISO 10303-41
ISO 10303-41
ISO 10303-41
ISO 10303-41

Reference path: #1: (named_unit =>
#3: (si_unit
si_unit.name)
#4: (conversion_based_unit
conversion_based_unit.name)
#6: (context_dependent_unit
context_dependent_unit.name))
#2: (derived_unit
derived_unit.name)

6.1.9.21 value_limit

#1: If the unit is assigned globally.

#2: If the unit is not assigned globally.

AIM element: #2: ([qualified_representation_item]
[measure_representation_item])
#1: ([value_representation_item]
[qualified_representation_item])

Source: ISO 10303-45
ISO 10303-45
ISO 10303-43
ISO 10303-45

Rules: dependent_instantiable_measure_with_unit

6.1.9.21.1 limit

#1: If the unit is assigned globally.

#2: If the unit is not assigned globally.

AIM element: #1: (value_representation_item.value_component)
#2: (measure_with_unit.value_component)

Source: ISO 10303-43
ISO 10303-45

Reference path: #1: (value_representation_item
value_representation_item.value_component)
#2: ({measure_representation_item <=
measure_with_unit
{(|measure_with_unit|)
(measure_with_unit =>
(amount_of_substance_measure_with_unit)
(area_measure_with_unit)
(electric_current_measure_with_unit)
(length_measure_with_unit)
(luminous_intensity_measure_with_unit)

```
(mass_measure_with_unit)
(plane_angle_measure_with_unit)
(ratio_measure_with_unit)
(solid_angle_measure_with_unit)
(thermodynamic_temperature_measure_with_unit)
(time_measure_with_unit)
(volume_measure_with_unit))}
measure_with_unit.value_component
{measure_with_unit.value_component ->
measure_value
(measure_value = area_measure
area_measure)
(measure_value = amount_of_substance_measure
amount_of_substance_measure)
(measure_value = length_measure
length_measure)
(measure_value = electric_current_measure
electric_current_measure)
(measure_value = plane_angle_measure
plane_angle_measure)
(measure_value = ratio_measure
ratio_measure)
(measure_value = parameter_value
parameter_value)
(measure_value = descriptive_measure
descriptive_measure)
(measure_value = positive_plane_angle_measure
positive_plane_angle_measure)
(measure_value = count_measure
count_measure)
(measure_value = mass_measure
mass_measure)
(measure_value = time_measure
time_measure)
(measure_value = thermodynamic_temperature_measure
thermodynamic_temperature_measure)
(measure_value = luminous_intensity_measure
luminous_intensity_measure)
(measure_value = solid_angle_measure
solid_angle_measure)
(measure_value = volume_measure
volume_measure)
(measure_value = numeric_measure
numeric_measure)
(measure_value = context_dependent_measure
context_dependent_measure)
(measure_value = positive_length_measure
positive_length_measure)
(measure_value = positive_ratio_measure
positive_ratio_measure)}})
```

6.1.9.21.2 limit_qualifier

#1: If the limit is an upper bound.

#2: If the limit is a lower bound.

AIM element: type_qualifier.name

Source: ISO 10303-45

Rules: dependent_instantiable_type_qualifier

Reference path:
```
qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier = type_qualifier
type_qualifier
{#1: (type_qualifier.name = 'maximum')
#2: (type_qualifier.name = 'minimum')}
```

6.1.9.22 value_list

AIM element: compound_representation_item

Source: ISO 10303-43

6.1.9.22.1 value_list to property_value (as values)

#1: If values[i] is a value_list.

#2: If values[i] is a string_value.

#3: If values[i] is a numerical_value or a value_limit and the unit is not assigned globally.

#4: If values[i] is a value_range.

#5: If no significant_digits are given for values[i].

#6: If significant_digits are given for values[i].

#7: If values[i] is a numerical_value or a value_limit and the unit is specified globally.

AIM element: PATH

Reference path:
```
compound_representation_item
compound_representation_item.item_element ->
compound_item_definition
compound_item_definition = list_representation_item
list_representation_item[i] ->
representation_item =>
#1: (compound_representation_item)
#2: (descriptive_representation_item
{descriptive_representation_item <=
representation_item =>
qualified_representation_item
qualified_representation_item.qualifiers[1] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier
type_qualifier.name
type_qualifier.name = 'string'})
#3: (#5: (measure_representation_item)
#6: ([measure_representation_item]
[qualified_representation_item]))
#4: (#5: (compound_representation_item =>
value_range)
#6: ([compound_representation_item =>
value_range]
```

[qualified_representation_item]))
#7: (#5: (value_representation_item)
#6: ([value_representation_item]
[qualified_representation_item]))

6.1.9.23 value_range

#1: If no significant digits are given for the value_with_unit.

#2: If significant digits are given for the value_with_unit.

AIM element: #1: (value_range)
#2: ([value_range]
[qualified_representation_item])

Source: ISO 10303-214
ISO 10303-214
ISO 10303-45

Reference path: {value_range <=
compound_representation_item}

6.1.9.23.1 lower_limit

#1: If the unit is not assigned globally.

#2: If the unit is assigned globally.

AIM element: #1: (measure_with_unit.value_component)
#2: (value_representation_item.value_component)

Source: ISO 10303-41
ISO 10303-41

Rules: dependent_instantiable_measure_with_unit

Reference path: value_range <=
compound_representation_item
compound_representation_item.item_element ->
set_representation_item
#1: (set_representation_item[i] ->
representation_item =>
{representation_item.name = 'lower limit'}
measure_representation_item
measure_representation_item <=
measure_with_unit
{(|measure_with_unit|)
(measure_with_unit =>
(amount_of_substance_measure_with_unit)
(area_measure_with_unit)
(electric_current_measure_with_unit)
(length_measure_with_unit)
(luminous_intensity_measure_with_unit)
(mass_measure_with_unit)
(plane_angle_measure_with_unit)
(ratio_measure_with_unit)
(solid_angle_measure_with_unit)
(thermodynamic_temperature_measure_with_unit)
(time_measure_with_unit)

(volume_measure_with_unit))}
measure_with_unit.value_component
{measure_with_unit.value_component ->
measure_value
(measure_value = area_measure
area_measure)
(measure_value = amount_of_substance_measure
amount_of_substance_measure)
(measure_value = length_measure
length_measure)
(measure_value = electric_current_measure
electric_current_measure)
(measure_value = plane_angle_measure
plane_angle_measure)
(measure_value = ratio_measure
ratio_measure)
(measure_value = parameter_value
parameter_value)
(measure_value = descriptive_measure
descriptive_measure)
(measure_value = positive_plane_angle_measure
positive_plane_angle_measure)
(measure_value = count_measure
count_measure)
(measure_value = mass_measure
mass_measure)
(measure_value = time_measure
time_measure)
(measure_value = thermodynamic_temperature_measure
thermodynamic_temperature_measure)
(measure_value = luminous_intensity_measure
luminous_intensity_measure)
(measure_value = solid_angle_measure
solid_angle_measure)
(measure_value = volume_measure
volume_measure)
(measure_value = numeric_measure
numeric_measure)
(measure_value = context_dependent_measure
context_dependent_measure)
(measure_value = positive_length_measure
positive_length_measure)
(measure_value = positive_ratio_measure
positive_ratio_measure)}})
#2: (set_representation_item[i] ->
representation_item =>
{representation_item.name = 'lower limit'}
value_representation_item
value_representation_item.value_component)

6.1.9.23.2 upper_limit

#1: If the unit is not assigned globally.

#2: If the unit is assigned globally.

AIM element: #1: (measure_with_unit.value_component)
#2: (value_representation_item.value_component)

Source: ISO 10303-41
ISO 10303-41

Rules: dependent_instantiable_measure_with_unit

Reference path:
```
value_range <=
compound_representation_item
compound_representation_item.item_element ->
set_representation_item
#1: (set_representation_item[i] ->
representation_item =>
{representation_item.name = 'upper limit'}
measure_representation_item
measure_representation_item <=
measure_with_unit
{(|measure_with_unit|)
(measure_with_unit =>
(amount_of_substance_measure_with_unit)
(area_measure_with_unit)
(electric_current_measure_with_unit)
(length_measure_with_unit)
(luminous_intensity_measure_with_unit)
(mass_measure_with_unit)
(plane_angle_measure_with_unit)
(ratio_measure_with_unit)
(solid_angle_measure_with_unit)
(thermodynamic_temperature_measure_with_unit)
(time_measure_with_unit)
(volume_measure_with_unit))}
measure_with_unit.value_component
{measure_with_unit.value_component ->
measure_value
(measure_value = area_measure
area_measure)
(measure_value = amount_of_substance_measure
amount_of_substance_measure)
(measure_value = length_measure
length_measure)
(measure_value = electric_current_measure
electric_current_measure)
(measure_value = plane_angle_measure
plane_angle_measure)
(measure_value = ratio_measure
ratio_measure)
(measure_value = parameter_value
parameter_value)
(measure_value = descriptive_measure
descriptive_measure)
(measure_value = positive_plane_angle_measure
positive_plane_angle_measure)
(measure_value = count_measure
count_measure)
(measure_value = mass_measure
mass_measure)
(measure_value = time_measure
time_measure)
(measure_value = thermodynamic_temperature_measure
thermodynamic_temperature_measure)
(measure_value = luminous_intensity_measure
luminous_intensity_measure)
```

(measure_value = solid_angle_measure
solid_angle_measure)
(measure_value = volume_measure
volume_measure)
(measure_value = numeric_measure
numeric_measure)
(measure_value = context_dependent_measure
context_dependent_measure)
(measure_value = positive_length_measure
positive_length_measure)
(measure_value = positive_ratio_measure
positive_ratio_measure)})
#2: (set_representation_item[i] ->
representation_item =>
{representation_item.name = 'upper limit'}
value_representation_item
value_representation_item.value_component)

6.1.9.24 value_with_unit

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a value_range.

#3: If no significant digits are given for the value_with_unit.

#4: If significant digits are given for the value_with_unit.

#5: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.

AIM element: #1: (#3: (measure_representation_item)
#4: ([measure_representation_item]
[qualified_representation_item]))
#2: (#3: (value_range)
#4: ([value_range]
[qualified_representation_item]))
#5: (#3: (value_representation_item)
#4: ([value_representation_item]
[qualified_representation_item]))

Source: ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-45
ISO 10303-45

Reference path: #2: ({value_range <=
compound_representation_item})

6.1.9.24.1 significant_digits

AIM element: precision_qualifier.precision_value

Source: ISO 10303-45

Rules: dependent_instantiable_precision_qualifier

Reference path: qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value

6.1.9.24.2 value_with_unit to unit (as unit_component)

#1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.

#2: If the value_with_unit is a value_range.

#3: If the unit has a commonly used descriptor.

#4: If the unit descriptor is constructed of more than one unit name.

AIM element: PATH

Reference path: #1: (measure_representation_item <=)
#2: (compound_representation_item
compound_representation_item.item_element ->
compound_item_definition
compound_item_definition = set_representation_item
set_representation_item
set_representation_item[i] ->
representation_item =>
measure_representation_item <=)
measure_with_unit
measure_with_unit.unit_component ->
unit
#3: (unit = named_unit
named_unit)
#4: (unit = derived_unit
derived_unit)

6.2 AIM EXPRESS short listing

6.2.1 cutting_tool_schema types

6.2.1.1 applied_state_type_of_item

The applied_state_type_of_item type is an extension of the state_type_of_item type. It adds the data type product_definition to the list of alternate data types.

EXPRESS specification:

```
TYPE applied_state_type_of_item = SELECT BASED_ON state_type_of_item WITH;
  (product_definition);
END_TYPE;
```

6.2.1.2 attribute_language_item

An attribute_language_item type specifies those objects to which a language may be assigned through an attribute_language_assignment.

EXPRESS specification:

```
TYPE attribute_language_item = SELECT
    (property_definition,
     group,
     external_source,
     product_related_product_category,
     effectivity,
     effectivity_relationship,
     product,
     product_definition,
     product_definition_formation,
     product_definition_formation_relationship,
     representation,
     product_definition_relationship,
     general_property,
     general_property_relationship,
     representation_relationship,
     general_feature,
     identification_role,
     application_context,
     attribute_value_assignment,
     person_and_organization_role,
     organization_role,
     descriptive_representation_item);
END_TYPE;
```

6.2.1.3 classification_item

A classification_item type specifies those objects to which a class or a class_system may be assigned through an applied_classification_assignment.

EXPRESS specification:

```
TYPE classification_item = SELECT
    (product,
     document_file,
     product_definition,
     product_definition_formation,
     product_definition_relationship,
     material_designation,
     general_property,
     property_definition,
     class,
     plus_minus_tolerance,
     document_type,
     planar_extent);
END_TYPE;
```

6.2.1.4 document_reference_item

A document_reference_item type specifies those objects to which a document_reference may be assigned through an applied_document_reference.

EXPRESS specification:

```
TYPE document_reference_item = SELECT
    (property_definition,
     class_system,
     class,
     product,
     product_definition,
     product_definition_relationship,
     product_definition_formation,
     material_designation,
     organization,
     person,
     general_property,
     product_related_product_category,
     general_feature);
END_TYPE;
```

6.2.1.5 effectivity_item

An effectivity_item type specifies those objects to which an effectivity may be assigned through an applied_effectivity_assignment.

EXPRESS specification:

```
 TYPE effectivity_item = SELECT
     (class_system,
      product,
      product_definition,
      document_file,
      product_definition_formation,
      product_definition_relationship,
      material_designation,
      applied_location_assignment,
      general_property,
      property_definition,
      applied_state_type_assignment);
 END_TYPE;
```

6.2.1.6 external_identification_item

An external_identification_item type specifies those objects to which an external identifier and an external source may be assigned through an applied_external_identification_assignment.

EXPRESS specification:

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

6.2.1.7 identification_item

An identification_item type specifies those objects to which an identifier and its usage may be assigned through an applied_identification_assignment.

EXPRESS specification:

```
 TYPE identification_item = SELECT
```

```
       (document_file,
        product,
        general_property,
        applied_identification_assignment,
        property_definition,
        class_system,
        product_definition,
        document_type,
        class,
        product_definition_formation,
        product_definition_relationship,
        organization,
        person_and_organization,
        effectivity);
  END_TYPE;
```

6.2.1.8 language_item

A language_item type specifies those objects to which a classification of a language may be assigned through a language_assignment.

EXPRESS specification:

```
  TYPE language_item = SELECT
       (representation);
  END_TYPE;
```

6.2.1.9 location_assignment_item

A location_assignment_item type specifies those objects to which a location may be assigned through an applied_location_assignment.

EXPRESS specification:

```
  TYPE location_assignment_item = SELECT
       (product_definition);
  END_TYPE;
```

6.2.1.10 multi_language_attribute_item

A multi_language_attribute_item type specifies those objects to which a label in an alternate language for an existing attribute may be assigned through a multi_language_attribute_assignment.

EXPRESS specification:

```
  TYPE multi_language_attribute_item = SELECT
       (property_definition,
        group,
        external_source,
        product_related_product_category,
        effectivity,
        effectivity_relationship,
        product_definition,
        product,
        product_definition_formation,
        product_definition_formation_relationship,
        representation,
        product_definition_relationship,
        general_property,
        general_property_relationship,
        representation_relationship,
        general_feature,
        identification_role,
        application_context,
        person_and_organization_role,
        organization_role,
        descriptive_representation_item);
   END_TYPE;
```

6.2.1.11 organization_item

An organization_item type specifies those objects to which an organization may be assigned through an applied_organization_assignment.

EXPRESS specification:

```
  TYPE organization_item = SELECT
      (class,
       property_definition,
       applied_identification_assignment,
       applied_classification_assignment,
       class_system,
       product,
       document_file,
       product_definition,
       product_definition_formation,
       product_definition_relationship,
       product_definition_formation_relationship,
       material_designation,
       person_and_organization,
       general_property);
  END_TYPE;
```

6.2.1.12 person_and_organization_item

A person_and_organization_item type specifies those objects to which a person_and_organization may be assigned through an applied_person_and_organization_assignment.

EXPRESS specification:

```
  TYPE person_and_organization_item = SELECT
      (applied_classification_assignment,
       class_system,
       product,
       document_file,
       product_definition,
       product_definition_formation,
       class,
       product_definition_relationship,
       product_definition_formation_relationship,
       material_designation,
       person_and_organization,
       general_property,
       property_definition);
  END_TYPE;
```

6.2.2 cutting_tool_schema entities

6.2.2.1 applied_location_assignment

The applied_location_assignment entity allows the specification of a location for a product_definition of a physical product.

EXPRESS specification:

```
  ENTITY applied_location_assignment
    SUBTYPE OF(location_assignment);
      items : SET [1:?] OF location_assignment_item;
  END_ENTITY;
```

6.2.2.1.1 items

Specifies the product_definition of a physical product that is located.

6.2.2.2 general_feature

A general_feature is a type of feature_definition that provides a mechanism to categorize an area of interest in a shape and to associate additional information with it.

EXPRESS specification:

```
    ENTITY general_feature
      SUBTYPE OF(feature_definition, shape_aspect);
      WHERE
        WR1 :
              (SIZEOF(get_property_definition_representations(SELF)) <= 1)
    AND (SIZEOF(QUERY (pdr <* get_property_definition_representations(SELF)|
    ('CUTTING_TOOL_SCHEMA.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)))) <= 1);

        WR2 :
              SIZEOF(QUERY (sa <* get_shape_aspects(SELF)| (sa.description =
    'course of travel occurrence'))) = SIZEOF(QUERY (sa <*
    get_shape_aspects(SELF)| (sa.description = 'course of travel
    occurrence') AND (SIZEOF(QUERY (sar <* USEDIN(sa, 'CUTTING_TOOL_SCHEMA.'
    + 'SHAPE_ASPECT_RELATIONSHIP.' + 'RELATED_SHAPE_ASPECT')|
    ('CUTTING_TOOL_SCHEMA.' + 'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))
    AND (sar.name = 'course of travel'))) = 1)));
        WR3 :
              SIZEOF(QUERY (sa <* get_shape_aspects(SELF)| (sa.description =
    'course of travel occurrence'))) = SIZEOF(QUERY (sa <*
    get_shape_aspects(SELF)| (sa.description = 'course of travel
    occurrence') AND (SIZEOF(QUERY (sar <* USEDIN(sa, 'CUTTING_TOOL_SCHEMA.'
    + 'SHAPE_ASPECT_RELATIONSHIP.' + 'RELATED_SHAPE_ASPECT')|
    ('CUTTING_TOOL_SCHEMA.' + 'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))
    AND ('CUTTING_TOOL_SCHEMA.' + 'PATH_FEATURE_COMPONENT' IN
    TYPEOF(sar.relating_shape_aspect)))) = 1)));
        WR4 :
              SIZEOF(QUERY (sa <* get_shape_aspects(SELF)| (sa.description =
    'boundary occurrence'))) = SIZEOF(QUERY (sa <* get_shape_aspects(SELF)|
    (sa.description = 'boundary occurrence') AND (SIZEOF(QUERY (sar <*
    USEDIN(sa, 'CUTTING_TOOL_SCHEMA.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT')| ('CUTTING_TOOL_SCHEMA.' +
    'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (sar.description =
    'profile usage'))) = 1)));
        WR5 :
               (SIZEOF(QUERY (sa <* get_shape_aspects(SELF)|
    ('CUTTING_TOOL_SCHEMA.' + 'COMPOSITE_SHAPE_ASPECT' IN TYPEOF(sa)))) <=
    1) AND (SIZEOF(QUERY (sa <* get_shape_aspects(SELF)|
    (('CUTTING_TOOL_SCHEMA.' + 'COMPOSITE_SHAPE_ASPECT' IN TYPEOF(sa)) AND
    (sa.name = 'general compound feature')) AND (SIZEOF(QUERY (sar <*
    USEDIN(sa, 'CUTTING_TOOL_SCHEMA.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT')| ('CUTTING_TOOL_SCHEMA.' +
    'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
    ('CUTTING_TOOL_SCHEMA.' + 'INSTANCED_FEATURE' IN
    TYPEOF(sar.related_shape_aspect)))) >= 2))) <= 1);
    END_ENTITY;
```

6.2.3 cutting_tool_schema rules

6.2.3.1 coating_requires_product_definition

The coating_requires_product_definition rule ensures that each instance of product_definition_formation which has an associated of_product that is referenced by a product_related_product_category with a name of 'coating', is referenced as formation by exactly one instance of product_definition that references as its frame_of_reference a product_definition_context with a name of 'material definition'.

EXPRESS specification:

```
  RULE coating_requires_product_definition FOR
      (product_definition,
       product_definition_formation,
       product_related_product_category);
  WHERE
   WR1: SIZEOF ( QUERY ( pdf <* product_definition_formation | (
       SIZEOF ( QUERY ( prpc <* USEDIN ( pdf. of_product ,
       'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
       'PRODUCTS' ) | prpc. name = 'coating' )
       ) >0 ) AND ( SIZEOF ( QUERY ( pd <* USEDIN ( pdf ,
       'ISO13399_AIM_SHORT_FORM.'+'PRODUCT_DEFINITION.'+ 'FORMATION' ) |
       pd. frame_of_reference.name = 'material definition' ) ) <>1
       ) ) ) =0;
  END_RULE;
```

Argument definitions:

product_definition: the set of all instances of product_definition.

product_definition_formation: the set of all instances of product_definition_formation.

product_related_product_category: the set of all instances of product_related_product_category.

product_definition_formation: the set of all instances of product_definition_formation.

product_related_product_category: the set of all instances of product_related_product_category.

Formal propositions:

WR1: Each instance of product_definition_formation which has an associated of_product that is referenced by a product_related_product_category with a name of 'coating', is referenced as formation by exactly one instance of product_definition that references as its frame_of_reference a product_definition_context with a name of 'material definition'.

6.2.3.2 dependent_instantiable_attribute_value_role

The dependent_instantiable_attribute_value_role rule specifies that each instance of attribute_value_role is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_attribute_value_role FOR
    (attribute_value_role);
WHERE
 WR1: SIZEOF ( QUERY ( a <* attribute_value_role | NOT ( SIZEOF
     ( USEDIN ( a , '' ) ) >0 ) ) ) =0;
END_RULE;
```

Argument definitions:

attribute_value_role: the set of all instances of attribute_value_role.

Formal propositions:

WR1: For each instance of attribute_value_role, there shall be a reference to the attribute_value_role instance from an attribute of another entity.

6.2.3.3 dependent_instantiable_classification_role

The dependent_instantiable_classification_role rule specifies that each instance of classification_role is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_classification_role FOR
    (classification_role);
WHERE
 WR1: SIZEOF ( QUERY ( c <* classification_role | NOT ( SIZEOF (
     USEDIN ( c , '' ) ) >0 ) ) ) =0;
END_RULE;
```

Argument definitions:

classification_role: the set of all instances of classification_role.

Formal propositions:

WR1: For each instance of classification_role, there shall be a reference to the classification_role instance from an attribute of another entity.

6.2.3.4 dependent_instantiable_date

The dependent_instantiable_date rule specifies that each instance of date is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_date FOR
    (date);
```

```
  WHERE
   WR1: SIZEOF ( QUERY ( d <* date | NOT ( SIZEOF ( USEDIN ( d ,
        '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

date: the set of all instances of date.

Formal propositions:

WR1: For each instance of date, there shall be a reference to the date instance from an attribute of another entity.

6.2.3.5 dependent_instantiable_date_and_time

The dependent_instantiable_date_and_time rule specifies that each instance of date_and_time is dependent on the usage to define another entity.

EXPRESS specification:

```
  RULE dependent_instantiable_date_and_time FOR
      (date_and_time);
  WHERE
   WR1: SIZEOF ( QUERY ( d <* date_and_time | NOT ( SIZEOF (
       USEDIN ( d , '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

date_and_time: the set of all instances of date_and_time.

Formal propositions:

WR1: For each instance of date_and_time, there shall be a reference to the date_and_time instance from an attribute of another entity.

6.2.3.6 dependent_instantiable_derived_unit

The dependent_instantiable_derived_unit rule specifies that each instance of derived_unit is dependent on the usage to define another entity.

EXPRESS specification:

```
  RULE dependent_instantiable_derived_unit FOR
      (derived_unit);
  WHERE
   WR1: SIZEOF ( QUERY ( d <* derived_unit | NOT ( SIZEOF ( USEDIN
       ( d , '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

derived_unit: the set of all instances of derived_unit.

Formal propositions:

WR1: For each instance of derived_unit, there shall be a reference to the derived_unit instance from an attribute of another entity.

6.2.3.7 dependent_instantiable_identification_role

The dependent_instantiable_identification_role rule specifies that each instance of identification_role is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_identification_role FOR
    (identification_role);
WHERE
 WR1: SIZEOF ( QUERY ( i <* identification_role | NOT ( SIZEOF (
     USEDIN ( i , '' ) ) >0 ) ) ) =0;
END_RULE;
```

Argument definitions:

identification_role: the set of all instances of identification_role.

Formal propositions:

WR1: For each instance of identification_role, there shall be a reference to the identification_role instance from an attribute of another entity.

6.2.3.8 dependent_instantiable_measure_with_unit

The dependent_instantiable_measure_with_unit rule specifies that each instance of measure_with_unit is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_measure_with_unit FOR
    (measure_with_unit);
WHERE
 WR1: SIZEOF ( QUERY ( m <* measure_with_unit | NOT ( SIZEOF (
     USEDIN ( m , '' ) ) >0 ) ) ) =0;
END_RULE;
```

Argument definitions:

measure_with_unit: the set of all instances of measure_with_unit.

Formal propositions:

WR1: For each instance of measure_with_unit, there shall be a reference to the measure_with_unit instance from an attribute of another entity.

6.2.3.9 dependent_instantiable_named_unit

The dependent_instantiable_named_unit rule specifies that each instance of named_unit is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_named_unit FOR
    (named_unit);
```

```
  WHERE
   WR1: SIZEOF ( QUERY ( n <* named_unit | NOT ( SIZEOF ( USEDIN (
        n , '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

named_unit: the set of all instances of named_unit.

Formal propositions:

WR1: For each instance of named_unit, there shall be a reference to the named_unit instance from an attribute of another entity.

6.2.3.10 dependent_instantiable_object_role

The dependent_instantiable_object_role rule specifies that each instance of object_role is dependent on the usage to define another entity.

EXPRESS specification:

```
  RULE dependent_instantiable_object_role FOR
      (object_role);
  WHERE
   WR1: SIZEOF ( QUERY ( o <* object_role | NOT ( SIZEOF ( USEDIN
       ( o , '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

object_role: the set of all instances of object_role.

Formal propositions:

WR1: For each instance of object_role, there shall be a reference to the object_role instance from an attribute of another entity.

6.2.3.11 dependent_instantiable_organization_role

The dependent_instantiable_organization_role rule specifies that each instance of organization_role is dependent on the usage to define another entity.

EXPRESS specification:

```
  RULE dependent_instantiable_organization_role FOR
      (organization_role);
  WHERE
   WR1: SIZEOF ( QUERY ( o <* organization_role | NOT ( SIZEOF (
       USEDIN ( o , '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

organization_role: the set of all instances of organization_role.

Formal propositions:

WR1: For each instance of organization_role, there shall be a reference to the organization_role instance from an attribute of another entity.

6.2.3.12 dependent_instantiable_person_and_organization_role

The dependent_instantiable_person_and_organization_role rule specifies that each instance of person_and_organization_role is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_person_and_organization_role FOR
    (person_and_organization_role);
WHERE
 WR1: SIZEOF ( QUERY ( p <* person_and_organization_role | NOT (
     SIZEOF ( USEDIN ( p , '' ) ) >0 ) ) ) =0;
END_RULE;
```

Argument definitions:

person_and_organization_role: the set of all instances of person_and_organization_role.

Formal propositions:

WR1: For each instance of person_and_organization_role, there shall be a reference to the person_and_organization_role instance from an attribute of another entity.

6.2.3.13 dependent_instantiable_precision_qualifier

The dependent_instantiable_precision_qualifier rule specifies that each instance of precision_qualifier is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_precision_qualifier FOR
    (precision_qualifier);
WHERE
 WR1: SIZEOF ( QUERY ( p <* precision_qualifier | NOT ( SIZEOF (
     USEDIN ( p , '' ) ) >0 ) ) ) =0;
END_RULE;
```

Argument definitions:

precision_qualifier: the set of all instances of precision_qualifier.

Formal propositions:

WR1: For each instance of precision_qualifier, there shall be a reference to the precision_qualifier instance from an attribute of another entity.

6.2.3.14 dependent_instantiable_product_definition_context_role

The dependent_instantiable_product_definition_context_role rule specifies that each instance of product_definition_context_role is dependent on the usage to define another entity.

EXPRESS specification:

```
RULE dependent_instantiable_product_definition_context_role FOR
    (product_definition_context_role);
```

```
  WHERE
   WR1: SIZEOF ( QUERY ( p <* product_definition_context_role |
       NOT ( SIZEOF ( USEDIN ( p , '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

product_definition_context_role: the set of all instances of product_definition_context_role.

Formal propositions:

WR1: For each instance of product_definition_context_role, there shall be a reference to the product_definition_context_role instance from an attribute of another entity.

6.2.3.15 dependent_instantiable_tolerance_value

The dependent_instantiable_tolerance_value rule specifies that each instance of tolerance_value is dependent on the usage to define another entity.

EXPRESS specification:

```
  RULE dependent_instantiable_tolerance_value FOR
      (tolerance_value);
  WHERE
   WR1: SIZEOF ( QUERY ( t <* tolerance_value | NOT ( SIZEOF (
       USEDIN ( t , '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

tolerance_value: the set of all instances of tolerance_value.

Formal propositions:

WR1: For each instance of tolerance_value, there shall be a reference to the tolerance_value instance from an attribute of another entity.

6.2.3.16 dependent_instantiable_type_qualifier

The dependent_instantiable_type_qualifier rule specifies that each instance of type_qualifier is dependent on the usage to define another entity.

EXPRESS specification:

```
  RULE dependent_instantiable_type_qualifier FOR
      (type_qualifier);
  WHERE
   WR1: SIZEOF ( QUERY ( t <* type_qualifier | NOT ( SIZEOF (
       USEDIN ( t , '' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

type_qualifier: the set of all instances of type_qualifier.

Formal propositions:

WR1: For each instance of type_qualifier, there shall be a reference to the type_qualifier instance from an attribute of another entity.

6.2.3.17 externally_defined_class_with_known_source_requirement

The externally_defined_class_with_known_source_requirement rule specifies that each instance of externally_defined_class which references as the source a known_source is contained in the set of items of exactly one applied_organization_assignment which references as its role an organization_role with a name of 'class supplier'. This rule enforces the requirement for every plib_class_reference to have a supplier_bsu.

EXPRESS specification:

```
RULE externally_defined_class_with_known_source_requirement FOR
    (externally_defined_class);
WHERE
 WR1: SIZEOF ( QUERY ( edc <* externally_defined_class | (
     'CUTTING_TOOL_SCHEMA.'+ 'KNOWN_SOURCE' IN TYPEOF ( edc.
     source ) ) AND ( SIZEOF ( QUERY ( aoa <* USEDIN ( edc ,
     'CUTTING_TOOL_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS' )
     | aoa. role.name = 'class supplier' ) ) =0 ) ) ) =0;
END_RULE;
```

Argument definitions:

externally_defined_class: the set of all instances of externally_defined_class.

Formal propositions:

WR1: Each instance of externally_defined_class which references as the source a known_source shall be member in the set of items of an applied_organization_assignment. This organization_assignment shall reference as its role an organization_role with a name of 'class supplier'.

6.2.3.18 grade_requires_product_definition

The grade_requires_product_definition rule ensures that each instance of product_definition_formation which has an associated of_product that is referenced by a product_related_product_category with a name of 'grade', is referenced as formation by exactly one instance of product_definition that references as its frame_of_reference a product_definition_context with a name of 'material definition'.

EXPRESS specification:

```
RULE grade_requires_product_definition FOR
    (product_definition,
     product_definition_formation,
     product_related_product_category);
WHERE
 WR1: SIZEOF ( QUERY ( pdf <* product_definition_formation | (
     SIZEOF ( QUERY ( prpc <* USEDIN ( pdf. of_product ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
     'PRODUCTS' ) | prpc. name = 'grade' )
     ) >0 ) AND ( SIZEOF ( QUERY ( pd <* USEDIN ( pdf ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_DEFINITION.'+ 'FORMATION' ) |
     pd. frame_of_reference.name = 'material definition' ) ) <>1
     ) ) ) =0;
END_RULE;
```

Argument definitions:

product_definition: the set of all instances of product_definition.

product_definition_formation: the set of all instances of product_definition_formation.

product_related_product_category: the set of all instances of product_related_product_category.

Formal propositions:

WR1: Each instance of product_definition_formation which has an associated of_product that is referenced by a product_related_product_category with a name of 'grade', is referenced as formation by exactly one instance of product_definition that references as its frame_of_reference a product_definition_context with a name of 'material definition'.

6.2.3.19 person_requires_person_and_organization

The person_requires_person_and_organization rule ensures that each instance of person shall be referenced by at least one instance of person_and_organization as the_person. This rule enforces the requirement for each person to have at least one organization specified the person belongs to.

EXPRESS specification:

```
RULE person_requires_person_and_organization FOR
    (person);
WHERE
 WR1: SIZEOF ( QUERY ( p <* person |    SIZEOF ( USEDIN ( p ,
      'CUTTING_TOOL_SCHEMA.PERSON_AND_ORGANIZATION.THE_PERSON' )
      ) =0 ) ) =0;
END_RULE;
```

Argument definitions:

person: the set of all instances of person.

Formal propositions:

WR1: Each instance of person shall be referenced by at least one instance of person_and_organization as the_person.

6.2.3.20 plib_class_reference_requires_version

The plib_class_reference_requires_version rule specifies that each instance of externally_defined_class that has as its source a known_source is contained in the set of items of exactly one applied_external_identification_assignment which references as its role an identification_role with a name of 'version'. This rule enforces the requirement for every Plib_class_reference to have exactly one version.

EXPRESS specification:

```
RULE plib_class_reference_requires_version FOR
    (externally_defined_class);
WHERE
 WR1: SIZEOF ( QUERY ( edc <* externally_defined_class | (
      'CUTTING_TOOL_SCHEMA.'+ 'KNOWN_SOURCE' IN TYPEOF ( edc.
      source ) ) AND ( SIZEOF ( QUERY ( aei <* USEDIN ( edc ,
      'CUTTING_TOOL_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS'
      ) | aei. role.name ='version' ) ) <>1 ) ) ) =0;
 WR2: SIZEOF ( QUERY ( edc <* externally_defined_class | (
      'CUTTING_TOOL_SCHEMA.'+ 'KNOWN_SOURCE' IN TYPEOF ( edc.
      source ) ) AND ( SIZEOF ( QUERY ( aei <* USEDIN ( edc ,
      'CUTTING_TOOL_SCHEMA.APPLIED_IDENTIFICATION_ASSIGNMENT.ITEMS'
      ) | aei. role.name ='version' ) ) >0 ) ) ) =0;
  END_RULE;
```

Argument definitions:

externally_defined_class: the set of all instances of externally_defined_class.

Formal propositions:

WR1: Each instance of externally_defined_class that has as its source a known_source is contained in the set of items of exactly one applied_external_identification_assignment which references as its role an identification_role with a name of 'version'.

WR2: An instance of externally_defined_class that has as its source a known_source may not be contained in the set of items of an applied_identification_assignment which references as its role an identification_role with a name of 'version'.

6.2.3.21 plib_property_reference_requires_name_scope

The plib_property_reference_requires_name_scope rule specifies that each instance of externally_defined_general_property that has as its source a known_source is referenced as the relating_item by exactly one instance of externally_defined_item_relationship which has a name of 'name scope' and which references as the related_item an externally_defined_class that has as its source a known_source. This rule enforces the requirement for every plib_property_reference to have a name_scope.

EXPRESS specification:

```
  RULE plib_property_reference_requires_name_scope FOR
      (externally_defined_general_property);
  WHERE
   WR1: SIZEOF ( QUERY ( edgp <*
      externally_defined_general_property | (
      'CUTTING_TOOL_SCHEMA.'+ 'KNOWN_SOURCE' IN TYPEOF ( edgp.
      source ) ) AND ( SIZEOF ( QUERY ( edir <* USEDIN ( edgp ,
      'CUTTING_TOOL_SCHEMA.'+'EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.'+
      'RELATING_ITEM' ) | ( edir. name = 'name scope' ) AND (
      'CUTTING_TOOL_SCHEMA.'+ 'EXTERNALLY_DEFINED_CLASS' IN TYPEOF
      ( edir. related_item ) ) AND ( 'CUTTING_TOOL_SCHEMA.'+
      'KNOWN_SOURCE' IN TYPEOF ( edir. related_item.source ) ) )
      ) <>1 ) ) ) =0;
  END_RULE;
```

Argument definitions:

externally_defined_general_property: the set of all instances of externally_defined_general_property.

Formal propositions:

WR1: Each instance of externally_defined_general_property that has as its source a known_source is referenced as the relating_item by exactly one instance of externally_defined_item_relationship which has a name of 'name scope' and which references as the related_item an externally_defined_class that has as its source a known_source.

6.2.3.22 plib_property_reference_requires_version

The plib_property_reference_requires_version rule specifies that each instance of externally_defined_general_property that has as its source a known_source is contained in the set of items of exactly one applied_external_identification_assignment which references as its role an identification_role with a name of 'version'. This rule enforces the requirement for every plib_property_reference to have a version.

EXPRESS specification:

```
RULE plib_property_reference_requires_version FOR
    (externally_defined_general_property);
WHERE
 WR1: SIZEOF ( QUERY ( edgp <*
     externally_defined_general_property | (
     'CUTTING_TOOL_SCHEMA.'+ 'KNOWN_SOURCE' IN TYPEOF ( edgp.
     source ) ) AND ( SIZEOF ( QUERY ( edir <* USEDIN ( edgp ,
     'CUTTING_TOOL_SCHEMA.'+'APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.'+
     'ITEMS' ) | ( edir. role.name ='version' ) ) ) <>1 ) ) )
     =0;
END_RULE;
```

Argument definitions:

externally_defined_general_property: the set of all instances of externally_defined_general_property.

Formal propositions:

WR1: Each instance of externally_defined_general_property that has as its source a known_source is contained in the set of items of exactly one applied_external_identification_assignment which references as its role an identification_role with a name of 'version'.

6.2.3.23 product_requires_category

The product_requires_category rule specifies that each instance of product shall be referenced by at least one product_related_product_category.

EXPRESS specification:

```
RULE product_requires_category FOR
    (product,
     product_related_product_category);
WHERE
 WR1: SIZEOF ( QUERY ( p <* product | SIZEOF ( USEDIN ( p ,
     'CUTTING_TOOL_SCHEMA.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS'
     ) ) =0 ) ) =0;
END_RULE;
```

Argument definitions:

product: the set of all instances of product.

product_related_product_category: the set of all instances of product_related_product_category.

product_related_product_category: the set of all instances of product_related_product_category.

Formal propositions:

WR1: Each instance of product shall be referenced by at least one product_related_product_category.

6.2.3.24 product_requires_id_owner

The product_requires_id_owner rule specifies that each instance of product that is referenced by an product_related_product_category with a name of 'part', 'tool', or 'raw material' shall be either referenced by an instance of applied_person_and_organization_assignment or by an instance of applied_organization_assignment. This rule enforces the requirement that for every item an owner shall be specified for its id.

EXPRESS specification:

```
RULE product_requires_id_owner FOR
    (product);
WHERE
 WR1: SIZEOF ( QUERY ( prod <* product | ( SIZEOF ( QUERY ( prpc
     <* USEDIN ( prod ,
     'CUTTING_TOOL_SCHEMA.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS'
     ) | prpc. name IN ['part' , 'tool' , 'raw material' ] ) )
     >0 ) AND ( SIZEOF ( QUERY ( apoa <* USEDIN ( prod ,
     'CUTTING_TOOL_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS'
     ) | apoa\ person_and_organization_assignment.role. name<>
     'id owner' ) ) =1 ) ) ) =0; +
     (SIZEOF( QUERY (oa <* USEDIN ( prod,
     'CUTTING_TOOL_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS' ) |
     oa\organization_assignment.role.name = 'id owner' )) <> 1)
END_RULE;
```

Argument definitions:

product: the set of all instances of product.

Formal propositions:

WR1: Each instance of product that is referenced by a product_related_product_category with a name of 'part', 'tool', or 'raw material' is contained in the set of items of either exactly one applied_person_and_organization_assignment which references as its role an person_and_organization_role with a name of 'id owner' or exactly one applied_organization_assignment which references as its role an organization_role with a name of 'id owner'.

6.2.3.25 product_requires_version

The product_requires_version rule ensures that each instance of product is referenced by at least one instance of product_definition_formation. This rule enforces the requirement for every Item to have at least one Item version associated to it.

EXPRESS specification:

```
RULE product_requires_version FOR
    (product);
WHERE
 WR1: SIZEOF ( QUERY ( prod <* product | ( SIZEOF ( USEDIN (
     prod ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_DEFINITION_FORMATION.'+
     'OF_PRODUCT' ) ) =0 ) ) ) =0;
END_RULE;
```

Argument definitions:

product: the set of all instances of product.

Formal propositions:

WR1: For each instance of product there shall be one or more instances of product_definition_formation that refers as of_product attribute to this instance of product.

6.2.3.26 restrict_applied_classification_assignment_role

The restrict_applied_classification_assignment_role rule ensures the correct correlation between the assigned_class of the applied_classification_assignment and the name of the classification_role referenced as the role.

EXPRESS specification:

```
RULE restrict_applied_classification_assignment_role FOR
    (applied_classification_assignment);
WHERE
 WR1: SIZEOF ( QUERY ( aca <* applied_classification_assignment
     | ( 'CUTTING_TOOL_SCHEMA.'+ 'CLASS' IN TYPEOF ( aca.
     assigned_class ) ) AND NOT ( aca. role.name IN
     ['definitional' , 'non-definitional' , ''] ) ) ) ) =0;
 WR2: SIZEOF ( QUERY ( aca <* applied_classification_assignment
     | ( aca. role.name IN ['definitional' , 'non-definitional'
     , ''] ) AND NOT ( 'CUTTING_TOOL_SCHEMA.'+ 'CLASS' IN TYPEOF (
     aca. assigned_class ) ) ) ) =0;
 WR3: SIZEOF ( QUERY ( aca <* applied_classification_assignment
     | ( 'CUTTING_TOOL_SCHEMA.'+ 'CLASS_SYSTEM' IN TYPEOF ( aca.
     assigned_class ) ) AND ( aca. role.name<>
     'class system membership' ) ) ) =0;
 WR4: SIZEOF ( QUERY ( aca <* applied_classification_assignment
     | ( aca. role.name = 'class system membership' ) AND NOT (
     'CUTTING_TOOL_SCHEMA.'+ 'CLASS_SYSTEM' IN TYPEOF ( aca.
     assigned_class ) ) ) ) =0;
END_RULE;
```

Argument definitions:

applied_classification_assignment: the set of all instances of applied_classification_assignment.

Formal propositions:

WR1: If the assigned_class of an applied_classification_assignment is of type class, the classification_role referenced by the role attribute has a value of either 'definitional', 'non-definitional', or ''.

WR2: If an applied_classification_assignment references as its role a classification_role with a name of either 'definitional', 'non-definitional', or '', the assigned_class shall be of type class.

WR3: If the assigned_class of an applied_classification_assignment is of type class_system, the classification_role referenced by the role attribute has a value of 'class system membership'.

WR4: If an applied_classification_assignment references as its role a classification_role with a name of 'class system membership', the assigned_class shall be of type class_system.

6.2.3.27 restrict_class_system_assignment_for_class

The restrict_class_system_assignment_for_class rule specifies that each instance of class shall be referenced by at most one instance of applied_classification_assignment that specifies the classification system for the class.

EXPRESS specification:

```
  RULE restrict_class_system_assignment_for_class FOR
      (applied_classification_assignment,
       class);
  WHERE
   WR1: SIZEOF ( QUERY ( c <* class | SIZEOF ( QUERY ( aca <*
       USEDIN ( c ,
       'CUTTING_TOOL_SCHEMA.'+'CLASSIFICATION_ASSIGNMENT.'+
       'ASSIGNED_CLASS' ) | ( ( aca. role.name =
       'class system membership' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
       'APPLIED_CLASSIFICATION_ASSIGNMENT' IN TYPEOF ( aca ) ) ) )
       ) >1 ) ) =0;
  END_RULE;
```

Argument definitions:

applied_classification_assignment: the set of all instances of applied_classification_assignment.

class: the set of all instances of class.

Formal propositions:

WR1: For each instance of class there shall be only at most one instance of applied_classification_assignment with a role name of 'class system membership' that references that instance of class.

6.2.3.28 restrict_class_system_assignment_for_planar_extent

The restrict_class_system_assignment_for_planar_extent rule specifies that each instance of planar_extent with a name of 'size format' shall be referenced by at most one instance of applied_classification_assignment that specifies the referenced standard for the named_size.

EXPRESS specification:

```
  RULE restrict_class_system_assignment_for_planar_extent FOR
      (applied_classification_assignment,
       planar_extent);
  WHERE
   WR1: SIZEOF ( QUERY ( pe <* planar_extent | ( pe\
       representation_item.name = 'size format' ) AND ( SIZEOF (
       QUERY ( aca <* USEDIN ( pe ,
       'CUTTING_TOOL_SCHEMA.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS'
       ) | aca\ classification_assignment.role. name =
       'class system membership' ) ) >1 ) ) ) =0;
  END_RULE;
```

Argument definitions:

applied_classification_assignment: the set of all instances of applied_classification_assignment.

planar_extent: the set of all instances of planar_extent.

Formal propositions:

WR1: For each instance of planar_extent with a name of 'size format' there shall be at most one instance of applied_classification_assignment with a role name of 'class system membership'.

6.2.3.29 restrict_externally_defined_item_relationship

The restrict_externally_defined_item_relationship rule ensures the correct correlation between the name of an externally_defined_item_relationship and the related externally_defined_items.

EXPRESS specification:

```
RULE restrict_externally_defined_item_relationship FOR
    (externally_defined_item_relationship);
WHERE
 WR1: SIZEOF ( QUERY ( edir <*
     externally_defined_item_relationship | ( edir. name =
     'name scope' ) AND ( NOT ( 'CUTTING_TOOL_SCHEMA.'+
     'EXTERNALLY_DEFINED_GENERAL_PROPERTY' IN TYPEOF ( edir.
     relating_item ) ) OR NOT ( 'CUTTING_TOOL_SCHEMA.'+
     'KNOWN_SOURCE' IN TYPEOF ( edir. relating_item.source ) )
     OR NOT ( 'CUTTING_TOOL_SCHEMA.'+ 'EXTERNALLY_DEFINED_CLASS'
     IN TYPEOF ( edir. related_item ) ) OR NOT (
     'CUTTING_TOOL_SCHEMA.'+ 'KNOWN_SOURCE' IN TYPEOF ( edir.
     related_item.source ) ) ) ) ) =0;
END_RULE;
```

Argument definitions:

externally_defined_item_relationship: the set of all instances of externally_defined_item_relationship.

Formal propositions:

WR1: If an externally_defined_item_relationship has a name of 'name scope', the relating_item shall be of type externally_defined_general_property and shall reference as the source a known_source and the related_item shall be of type externally_defined_class and shall reference as its source a known_source.

6.2.3.30 restrict_group_relationship_for_general_classification_hierarchy

The restrict_group_relationship_for_general_classification_hierarchy rule ensures the correct correlation between the name of a group_relationship and the related groups of type class.

EXPRESS specification:

```
RULE restrict_group_relationship_for_general_classification_hierarchy FOR
    (class,
     group_relationship);
WHERE
 WR1: SIZEOF ( QUERY ( gr <* group_relationship | ( gr. name =
     'class hierarchy' ) AND ( NOT ( 'CUTTING_TOOL_SCHEMA.'+
     'CLASS' IN TYPEOF ( gr. related_group ) ) OR NOT (
     'CUTTING_TOOL_SCHEMA.'+ 'CLASS' IN TYPEOF ( gr.
     relating_group ) ) ) ) ) =0;
END_RULE;
```

Argument definitions:

class: the set of all instances of class.

group_relationship: the set of all instances of group_relationship.

Formal propositions:

WR1: If the group_relationship has a name of 'class hierarchy', the related_group and the relating_group shall be of type class.

6.2.3.31 restrict_multi_language_for_application_context

The restrict_multi_language_for_application_context rule specifies that each instance of application_context shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_application_context FOR
    (application_context);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* application_context | SIZEOF (
     QUERY ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* application_context | SIZEOF (
     QUERY ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* application_context | ( SIZEOF (
     QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

application_context: the set of all instances of application_context.

Formal propositions:

WR1: Each instance of application_context shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of application_context shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of application_context, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.32 restrict_multi_language_for_descriptive_representation_item

The restrict_multi_language_for_descriptive_representation_item rule specifies that each instance of descriptive_representation_item shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_descriptive_representation_item FOR
    (descriptive_representation_item);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* descriptive_representation_item |
     SIZEOF ( QUERY ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* descriptive_representation_item |
     SIZEOF ( QUERY ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* descriptive_representation_item |
     ( SIZEOF ( QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

descriptive_representation_item: the set of all instances of descriptive_representation_item.

Formal propositions:

WR1: Each instance of descriptive_representation_item shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of descriptive_representation_item shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of descriptive_representation_item, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.33 restrict_multi_language_for_effectivity

The restrict_multi_language_for_effectivity rule specifies that each instance of effectivity shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_effectivity FOR
    (effectivity);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* effectivity | SIZEOF ( QUERY (
     mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* effectivity | SIZEOF ( QUERY ( ala
     <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* effectivity | ( SIZEOF ( QUERY (
     mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

effectivity: the set of all instances of effectivity.

Formal propositions:

WR1: Each instance of effectivity shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of effectivity shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of effectivity, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.34 restrict_multi_language_for_effectivity_relationship

The restrict_multi_language_for_effectivity_relationship rule specifies that each instance of effectivity_relationship shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_effectivity_relationship FOR
    (effectivity_relationship);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* effectivity_relationship | SIZEOF
     ( QUERY ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* effectivity_relationship | SIZEOF
     ( QUERY ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* effectivity_relationship | (
     SIZEOF ( QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

effectivity_relationship: the set of all instances of effectivity_relationship.

Formal propositions:

WR1: Each instance of effectivity_relationship shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of effectivity_relationship shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of effectivity_relationship, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.35 restrict_multi_language_for_external_source

The restrict_multi_language_for_external_source rule specifies that each instance of external_source shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_external_source FOR
    (external_source);
WHERE
```

```
 WR1: SIZEOF ( QUERY ( ent <* external_source | SIZEOF ( QUERY (
     mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* external_source | SIZEOF ( QUERY (
     ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* external_source | ( SIZEOF ( QUERY
     ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

external_source: the set of all instances of external_source.

Formal propositions:

WR1: Each instance of external_source shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of external_source shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of external_source, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.36 restrict_multi_language_for_general_feature

The restrict_multi_language_for_general_feature rule specifies that each instance of general_feature shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_general_feature FOR
    (general_feature);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* general_feature | SIZEOF ( QUERY (
     mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
```

```
   WR2: SIZEOF ( QUERY ( ent <* general_feature | SIZEOF ( QUERY (
        ala <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
        ) =0;
   WR3: SIZEOF ( QUERY ( ent <* general_feature | ( SIZEOF ( QUERY
        ( mlaa1 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
        ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
        SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
        AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
        =0;
  END_RULE;
```

Argument definitions:

general_feature: the set of all instances of general_feature.

Formal propositions:

WR1: Each instance of general_feature shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of general_feature shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of general_feature, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.37 restrict_multi_language_for_general_property

The restrict_multi_language_for_general_property rule specifies that each instance of general_property shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
  RULE restrict_multi_language_for_general_property FOR
      (general_property);
  WHERE
   WR1: SIZEOF ( QUERY ( ent <* general_property | SIZEOF ( QUERY
        ( mlaa <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
        ) ) =0;
   WR2: SIZEOF ( QUERY ( ent <* general_property | SIZEOF ( QUERY
        ( ala <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
        ) =0;
```

```
 WR3: SIZEOF ( QUERY ( ent <* general_property | ( SIZEOF (
     QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

general_property: the set of all instances of general_property.

Formal propositions:

WR1: Each instance of general_property shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of general_property shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of general_property, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.38 restrict_multi_language_for_general_property_relationship

The restrict_multi_language_for_general_property_relationship rule specifies that each instance of general_property_relationship shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_general_property_relationship FOR
    (general_property_relationship);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* general_property_relationship |
     SIZEOF ( QUERY ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* general_property_relationship |
     SIZEOF ( QUERY ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* general_property_relationship | (
     SIZEOF ( QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

general_property_relationship: the set of all instances of general_property_relationship.

Formal propositions:

WR1: Each instance of general_property_relationship shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of general_property_relationship shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of general_property_relationship, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.39 restrict_multi_language_for_group

The restrict_multi_language_for_group rule specifies that each instance of group shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

EXPRESS specification:

```
  RULE restrict_multi_language_for_group FOR
      (group);
  WHERE
   WR1: SIZEOF ( QUERY ( ent <* group | SIZEOF ( QUERY ( mlaa <*
       USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
       'ITEMS' ) | ( mlaa. attribute_name<>'description' ) AND (
       mlaa. attribute_name<>'name' ) ) ) >0 ) ) =0;
   WR2: SIZEOF ( QUERY ( ent <* group | SIZEOF ( QUERY ( ala <*
       USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
       'ITEMS' ) | ( ala. attribute_name<>'description' ) AND (
       ala. attribute_name<>'name' ) ) ) >0 ) ) =0;
   WR3: SIZEOF ( QUERY ( ent <* group | ( SIZEOF ( QUERY ( mlaa1
       <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
       'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
       'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
       ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
       SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
       'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
       'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
       AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
       =0;
  END_RULE;
```

Argument definitions:

group: the set of all instances of group.

Formal propositions:

WR1: Each instance of group shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR2: Each instance of group shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR3: For each instance of group, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.40 restrict_multi_language_for_identification_role

The restrict_multi_language_for_identification_role rule specifies that each instance of identification_role shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
 RULE restrict_multi_language_for_identification_role FOR
     (identification_role);
 WHERE
  WR1: SIZEOF ( QUERY ( ent <* identification_role | SIZEOF (
      QUERY ( mlaa <* USEDIN ( ent ,
      'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
      'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
      ) ) =0;
  WR2: SIZEOF ( QUERY ( ent <* identification_role | SIZEOF (
      QUERY ( ala <* USEDIN ( ent ,
      'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
      'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
      ) =0;
  WR3: SIZEOF ( QUERY ( ent <* identification_role | ( SIZEOF (
      QUERY ( mlaa1 <* USEDIN ( ent ,
      'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
      'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
      'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
      'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
      ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
      SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
      'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
      'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
      'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
      'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
      AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
      =0;
 END_RULE;
```

Argument definitions:

identification_role: the set of all instances of identification_role.

Formal propositions:

WR1: Each instance of identification_role shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of identification_role shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of identification_role, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.41 restrict_multi_language_for_organization_role

The restrict_multi_language_for_organization_role rule specifies that each instance of organization_role shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_organization_role FOR
    (organization_role);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* organization_role | SIZEOF ( QUERY
     ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* organization_role | SIZEOF ( QUERY
     ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* organization_role | ( SIZEOF (
     QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

organization_role: the set of all instances of organization_role.

Formal propositions:

WR1: Each instance of organization_role shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of organization_role shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of organization_role, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.42 restrict_multi_language_for_person_and_organization_role

The restrict_multi_language_for_person_and_organization_role rule specifies that each instance of person_and_organization_role shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_person_and_organization_role FOR
    (person_and_organization_role);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* person_and_organization_role |
     SIZEOF ( QUERY ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* person_and_organization_role |
     SIZEOF ( QUERY ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* person_and_organization_role | (
     SIZEOF ( QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

person_and_organization_role: the set of all instances of person_and_organization_role.

Formal propositions:

WR1: Each instance of person_and_organization_role shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of person_and_organization_role shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of person_and_organization_role, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.43 restrict_multi_language_for_product

The restrict_multi_language_for_product rule specifies that each instance of product shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

EXPRESS specification:

```
RULE restrict_multi_language_for_product FOR
    (product);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* product | SIZEOF ( QUERY ( mlaa <*
     USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) AND (
     mlaa. attribute_name<>'name' ) ) ) >0 ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* product | SIZEOF ( QUERY ( ala <*
     USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) AND (
     ala. attribute_name<>'name' ) ) ) >0 ) ) =0;
 WR3: SIZEOF ( QUERY ( ent <* product | ( SIZEOF ( QUERY ( mlaa1
     <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

product: the set of all instances of product.

Formal propositions:

WR1: Each instance of product shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR2: Each instance of product shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR3: For each instance of product, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.44 restrict_multi_language_for_product_definition

The restrict_multi_language_for_product_definition rule specifies that each instance of product_definition shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

EXPRESS specification:

```
  RULE restrict_multi_language_for_product_definition FOR
      (product_definition);
  WHERE
   WR1: SIZEOF ( QUERY ( ent <* product_definition | SIZEOF (
       QUERY ( mlaa <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
       'ITEMS' ) | ( mlaa. attribute_name<>'description' ) AND (
       mlaa. attribute_name<>'name' ) ) ) >0 ) ) =0;
   WR2: SIZEOF ( QUERY ( ent <* product_definition | SIZEOF (
       QUERY ( ala <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
       'ITEMS' ) | ( ala. attribute_name<>'description' ) AND (
       ala. attribute_name<>'name' ) ) ) >0 ) ) =0;
   WR3: SIZEOF ( QUERY ( ent <* product_definition | ( SIZEOF (
       QUERY ( mlaa1 <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
       'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
       'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
       ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
       SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
       'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
       'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
       'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
       AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
       =0;
  END_RULE;
```

Argument definitions:

product_definition: the set of all instances of product_definition.

Formal propositions:

WR1: Each instance of product_definition shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR2: Each instance of product_definition shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR3: For each instance of product_definition, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.45 restrict_multi_language_for_product_definition_formation

The restrict_multi_language_for_product_definition_formation rule specifies that each instance of product_definition_formation shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_product_definition_formation FOR
    (product_definition_formation);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* product_definition_formation |
     SIZEOF ( QUERY ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* product_definition_formation |
     SIZEOF ( QUERY ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* product_definition_formation | (
     SIZEOF ( QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

product_definition_formation: the set of all instances of product_definition_formation.

Formal propositions:

WR1: Each instance of product_definition_formation shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of product_definition_formation shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of product_definition_formation, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.46 restrict_multi_language_for_product_definition_formation_relationship

The restrict_multi_language_for_product_definition_formation_relationship rule specifies that each instance of product_definition_formation_relationship shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_product_definition_formation_relationship
  FOR
```

```
        (product_definition_formation_relationship);
    WHERE
     WR1: SIZEOF ( QUERY ( ent <*
         product_definition_formation_relationship | SIZEOF ( QUERY
         ( mlaa <* USEDIN ( ent ,
         'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
         'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
         ) ) =0;
     WR2: SIZEOF ( QUERY ( ent <*
         product_definition_formation_relationship | SIZEOF ( QUERY
         ( ala <* USEDIN ( ent ,
         'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
         'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
         ) =0;
     WR3: SIZEOF ( QUERY ( ent <*
         product_definition_formation_relationship | ( SIZEOF (
         QUERY ( mlaa1 <* USEDIN ( ent ,
         'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
         'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
         'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
         'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
         ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
         SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
         'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
         'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
         'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
         'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
         AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
         =0;
    END_RULE;
```

Argument definitions:

product_definition_formation_relationship: the set of all instances of product_definition_formation_relationship.

Formal propositions:

WR1: Each instance of product_definition_formation_relationship shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of product_definition_formation_relationship shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of product_definition_formation_relationship, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.47 restrict_multi_language_for_product_definition_relationship

The restrict_multi_language_for_product_definition_relationship rule specifies that each instance of product_definition_relationship shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
    RULE restrict_multi_language_for_product_definition_relationship FOR
        (product_definition_relationship);
```

```
WHERE
 WR1: SIZEOF ( QUERY ( ent <* product_definition_relationship |
     SIZEOF ( QUERY ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* product_definition_relationship |
     SIZEOF ( QUERY ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* product_definition_relationship |
     ( SIZEOF ( QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

product_definition_relationship: the set of all instances of product_definition_relationship.

Formal propositions:

WR1: Each instance of product_definition_relationship shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of product_definition_relationship shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of product_definition_relationship, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.48 restrict_multi_language_for_product_related_product_category

The restrict_multi_language_for_product_related_product_category rule specifies that each instance of product_related_product_category shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
RULE restrict_multi_language_for_product_related_product_category FOR
    (product_related_product_category);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* product_related_product_category |
     SIZEOF ( QUERY ( mlaa <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
     ) ) =0;
 WR2: SIZEOF ( QUERY ( ent <* product_related_product_category |
     SIZEOF ( QUERY ( ala <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
     ) =0;
 WR3: SIZEOF ( QUERY ( ent <* product_related_product_category |
     ( SIZEOF ( QUERY ( mlaa1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
     'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
     ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
     SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
     'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
     'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
     AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
     =0;
END_RULE;
```

Argument definitions:

product_related_product_category: the set of all instances of product_related_product_category.

Formal propositions:

WR1: Each instance of product_related_product_category shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of product_related_product_category shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of product_related_product_category, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.49 restrict_multi_language_for_property_definition

The restrict_multi_language_for_property_definition rule specifies that each instance of property_definition shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

EXPRESS specification:

```
  RULE restrict_multi_language_for_property_definition FOR
      (property_definition);
  WHERE
     WR1: SIZEOF ( QUERY ( ent <* property_definition | SIZEOF (
        QUERY ( mlaa <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | ( mlaa. attribute_name<>'description' ) AND (
        mlaa. attribute_name<>'name' ) ) ) >0 ) ) =0;
     WR2: SIZEOF ( QUERY ( ent <* property_definition | SIZEOF (
        QUERY ( ala <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala. attribute_name<>'description' ) AND (
        ala. attribute_name<>'name' ) ) ) >0 ) ) =0;
    WR3: SIZEOF ( QUERY ( ent <* property_definition | ( SIZEOF (
        QUERY ( mlaa1 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
        ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
        SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
        AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
        =0;
  END_RULE;
```

Argument definitions:

property_definition: the set of all instances of property_definition.

Formal propositions:

WR1: Each instance of property_definition shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR2: Each instance of property_definition shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR3: For each instance of property_definition, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.50 restrict_multi_language_for_representation

The restrict_multi_language_for_representation rule specifies that each instance of representation shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

EXPRESS specification:

```
  RULE restrict_multi_language_for_representation FOR
      (representation);
  WHERE
    WR1: SIZEOF ( QUERY ( ent <* representation | SIZEOF ( QUERY (
        mlaa <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | ( mlaa. attribute_name<>'description' ) AND (
        mlaa. attribute_name<>'name' ) ) ) >0 ) ) =0;
    WR2: SIZEOF ( QUERY ( ent <* representation | SIZEOF ( QUERY (
        ala <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala. attribute_name<>'description' ) AND (
        ala. attribute_name<>'name' ) ) ) >0 ) ) =0;
    WR3: SIZEOF ( QUERY ( ent <* representation | ( SIZEOF ( QUERY
        ( mlaa1 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
        ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
        SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
        AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
        =0;
   END_RULE;
```

Argument definitions:

representation: the set of all instances of representation.

Formal propositions:

WR1: Each instance of representation shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR2: Each instance of representation shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of either 'description' or 'name'.

WR3: For each instance of representation, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.51 restrict_multi_language_for_representation_relationship

The restrict_multi_language_for_representation_relationship rule specifies that each instance of representation_relationship shall only be referenced by instances of multi_language_attribute_assignment or attribute_language_assignment that have an attribute attribute_name of 'description'.

EXPRESS specification:

```
   RULE restrict_multi_language_for_representation_relationship FOR
       (representation_relationship);
   WHERE
    WR1: SIZEOF ( QUERY ( ent <* representation_relationship |
        SIZEOF ( QUERY ( mlaa <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | ( mlaa. attribute_name<>'description' ) ) ) >0
        ) ) =0;
    WR2: SIZEOF ( QUERY ( ent <* representation_relationship |
        SIZEOF ( QUERY ( ala <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala. attribute_name<>'description' ) ) ) >0 )
        ) =0;
    WR3: SIZEOF ( QUERY ( ent <* representation_relationship | (
        SIZEOF ( QUERY ( mlaa1 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | SIZEOF ( QUERY ( mlaa2 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.'+
        'ITEMS' ) | ( mlaa1. attribute_name =mlaa2. attribute_name
        ) AND ( mlaa1. language =mlaa2. language ) ) ) >1 ) ) +
        SIZEOF ( QUERY ( ala1 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | SIZEOF ( QUERY ( ala2 <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'ATTRIBUTE_LANGUAGE_ASSIGNMENT.'+
        'ITEMS' ) | ( ala1. attribute_name =ala2. attribute_name )
        AND ( ala1. language =ala2. language ) ) ) >1 ) ) ) >0 ) )
        =0;
  END_RULE;
```

Argument definitions:

representation_relationship: the set of all instances of representation_relationship.

Formal propositions:

WR1: Each instance of representation_relationship shall be only in the set of items of those instances of multi_language_attribute_assignment that have an attribute attribute_name of 'description'.

WR2: Each instance of representation_relationship shall be only in the set of items of those instances of attribute_language_assignment that have an attribute attribute_name of 'description'.

WR3: For each instance of representation_relationship, there shall not be more than one multi_language_attribute_assignment or attribute_language_assignment with the same language and the same attribute_name.

6.2.3.52 restrict_name_for_known_source

The restrict_name_for_known_source rule ensures that each instance of known_source has a name of 'ISO 13584 library'.

EXPRESS specification:

```
  RULE restrict_name_for_known_source FOR
      (known_source);
  WHERE
   WR1: SIZEOF ( QUERY ( ks <* known_source | ( ks. name<>
        'ISO 13584 library' ) ) ) =0;
  END_RULE;
```

Argument definitions:

known_source: the set of all instances of known_source.

Formal propositions:

WR1: Each instance of known_source shall have a name of 'ISO 13584 library'.

6.2.3.53 restrict_part_occurrence

The restrict_part_occurrence rule specifies the restrictions that apply to product_definitions which reference as frame_of_reference a product_definition_context with a name of 'part occurrence'.

EXPRESS specification:

```
  RULE restrict_part_occurrence FOR
      (product_definition,
       product_definition_occurrence_relationship);
  WHERE
   WR1: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
       frame_of_reference.name = 'part occurrence' ) AND ( NOT (
       pd. name IN [ 'single instance' ,
       'quantified instance' ] ) ) ) ) =0;
   WR2: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
       frame_of_reference.name = 'part occurrence' ) AND ( SIZEOF
       ( QUERY ( cd <* USEDIN ( pd ,
       'CUTTING_TOOL_SCHEMA.'+'PRODUCT_DEFINITION_RELATIONSHIP.'+
       'RELATED_PRODUCT_DEFINITION' ) | ( 'CUTTING_TOOL_SCHEMA.'+
       'PRODUCT_DEFINITION_USAGE' IN TYPEOF ( cd ) ) ) ) =0 ) AND
       ( SIZEOF ( USEDIN ( pd ,
       'CUTTING_TOOL_SCHEMA.'+'PRODUCT_DEFINITION_OCCURRENCE_RELATIONSHIP.'+
       'OCCURRENCE' ) ) =0 ) ) ) =0;
   WR3: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
       frame_of_reference.name = 'part occurrence' ) AND ( pd.
       name = 'quantified instance' ) AND ( SIZEOF ( QUERY ( ppd
       <* USEDIN ( pd ,
       'CUTTING_TOOL_SCHEMA.'+'PROPERTY_DEFINITION.'+ 'DEFINITION' )
       | ( ppd. name = 'occurrence quantity' ) AND ( SIZEOF (
       QUERY ( pdr <* USEDIN ( ppd ,
       'CUTTING_TOOL_SCHEMA.'+'PROPERTY_DEFINITION_REPRESENTATION.'+
       'DEFINITION' ) | ( pdr. used_representation.name
       ='quantity' ) AND ( SIZEOF ( pdr. used_representation.items
       ) =1 ) AND ( SIZEOF ( QUERY ( i <* pdr.
       used_representation.items | ( 'CUTTING_TOOL_SCHEMA.'+
       'MEASURE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) AND (
       i.name = 'quantity measure' ) ) ) =1 ) ) ) =1 ) ) ) =0 ) )
       ) =0;
  END_RULE;
```

Argument definitions:

product_definition: the set of all instances of product_definition.

product_definition_occurrence_relationship: the set of all instances of product_definition_occurrence_relationship.

Formal propositions:

WR1: Each instance of product_definition which references as its frame_of_reference a product_definition_context with a name of 'part occurrence', shall either have a name of 'single instance' or 'quantified instance'.

WR2: Each instance of product_definition which references as its frame_of_reference a product_definition_context with a name of 'part occurrence' shall (at least) participate in a product_definition_usage relationship as the related_product_definition or in a product_definition_occurrence_relationship as occurrence.

WR3: Each instance of product_definition which references as its frame_of_reference a product_definition_context with a name of 'part occurrence' and which has a name of 'quantified instance', shall be referenced by a property_definition which has a name of 'occurrence quantity', and which is referenced as definition by a property_definition_representation which has as used_representation a representation which has a name of 'quantity' and which contains exactly one representation_item in its set of items. This representation_item is of type measure_representation_item and has a name of 'quantity measure'.

6.2.3.54 restrict_product_category_for_product

The restrict_product_category_for_product rule enforces the existence of a product_related_product_category with a name of either 'cutting item', 'tool item', 'adaptive item', 'accesory item', 'assembly item', 'document', 'grade', 'coating', 'physically realized product', or 'substrate' for each product.

EXPRESS specification:

```
RULE restrict_product_category_for_product FOR
    (product);
WHERE
 WR1: SIZEOF ( QUERY ( p <* product | SIZEOF ( QUERY ( prpc <*
     USEDIN ( p ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
     'PRODUCTS' ) | prpc. name IN ['cutting item' , 'tool item' ,
     'adaptive item' , 'accesory item' , 'assembly item' ,
     'document' , 'grade' , 'coating' , 'physically realized product' ,
     'substrate' ] ) ) =0 ) )
     =0;
 WR2: SIZEOF ( QUERY ( p <* product | SIZEOF ( QUERY ( prpc <*
     USEDIN ( p ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
     'PRODUCTS' ) | prpc. name IN ['cutting item' , 'tool item' ,
     'adaptive item' , 'accesory item' , 'assembly item' ,
     'document' , 'grade' , 'coating' , 'substrate' ]
     ) ) >1 ) ) =0;
 WR3: SIZEOF ( QUERY ( p <* product | SIZEOF ( QUERY ( prpc <*
     USEDIN ( p ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
     'PRODUCTS' ) | prpc. name IN [ 'physically realized product' ]
     ) ) >1 ) ) =0;
END_RULE;
```

Argument definitions:

product: the set of all instances of product.

Formal propositions:

WR1: Each instance of product shall be referenced by a product_related_product_category with a name of either 'cutting item', 'tool item', 'adaptive item', 'accesory item', 'assembly item', 'document', 'grade', 'coating', 'physically realized product', or 'substrate'.

WR2: Each instance of product shall be referenced by at most one product_related_product_category with a name of either 'cutting item', 'tool item', 'adaptive item', 'accesory item', 'assembly item', 'document', 'grade', 'coating', or 'substrate'.

WR3: Each instance of product shall be referenced by at most one product_related_product_category with a name of 'physically realized product'.

6.2.3.55 restrict_product_definition_context_for_applied_location_assignment

The restrict_product_definition_context_for_applied_location_assignment rule constrains that the name of the product_definition_context which is referenced as the frame_of_reference by a product_definition that is referenced as the located_product by an applied_location_assignment.

EXPRESS specification:

```
RULE restrict_product_definition_context_for_applied_location_assignment
  FOR
    (applied_location_assignment);
WHERE
 WR1: SIZEOF ( QUERY ( ala <* applied_location_assignment |
      ( ala.located_product.frame_of_reference. name<>
      'physical occurrence' ) ) ) =0;
END_RULE;
```

Argument definitions:

applied_location_assignment: the set of all instances of product_definition.

Formal propositions:

WR1: Each instance of applied_location_assignment shall reference as located_product a product_defintion that references a product_definition_context with a name of 'physical occurrence'.

6.2.3.56 restrict_product_definition_context_for_applied_state_type_assignment

The restrict_product_definition_context_for_applied_location_assignment rule constrains that the name of the product_definition_context which is referenced as the frame_of_reference by a product_definition that is referenced as the located_product by an applied_location_assignment.

EXPRESS specification:

```
RULE restrict_product_definition_context_for_applied_state_type_assignment
  FOR
    (applied_state_type_assignment);
WHERE
 WR1: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
      frame_of_reference.name <> 'physical occurrence' ) AND USEDIN ( pd ,
      'CUTTING_TOOL_SCHEMA.'+'APPLIED_STATE_TYPE_ASSIGNMENT.'+'ITEM_SET' )
      ) ) = 0;
END_RULE;
```

Argument definitions:

product_definition: the set of all instances of product_definition.

Formal propositions:

WR1: Each instance of applied_state_type _assignment shall reference as its item_set only product_defintions that reference a product_definition_context with a name of 'physical occurrence'.

6.2.3.57 restrict_product_definition_context_for_product

The restrict_product_definition_context_for_product rule ensures the correct correlation between the name of the product_definition_context which is referenced as the frame_of_reference by a product_definition and the product_related_product_category of the associated product.

EXPRESS specification:

```
RULE restrict_product_definition_context_for_product FOR
    (product_definition,
     product_related_product_category);
WHERE
 WR1: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
      frame_of_reference.name IN [ 'part definition' ,
     'part occurrence' ] ) AND ( SIZEOF ( QUERY ( prpc <* USEDIN
     ( pd. formation.of_product ,
     'CUTTING_TOOL_SCHEMA.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS'
     ) | prpc. name IN ['cutting item' , 'tool item' ,
     'adaptive item' , 'accesory item' , 'assembly item' ] ) ) =0
     ) ) ) =0;
 WR2: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
     frame_of_reference.name = 'physical occurrence' ) AND (
     SIZEOF ( QUERY ( prpc <* USEDIN ( pd. formation.of_product
     , 'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
     'PRODUCTS' ) | prpc. name = 'physically realized product' )
     ) =0 ) ) ) =0;
 WR3: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
     frame_of_reference.name IN[ 'physical document definition' ,
     'digital document definition' ] )
     AND ( SIZEOF ( QUERY ( prpc <* USEDIN ( pd.
     formation.of_product ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
     'PRODUCTS' ) | prpc. name ='document' ) ) =0 ) ) ) =0;
 WR4: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
     frame_of_reference.name = 'material definition' ) AND (
     SIZEOF ( QUERY ( prpc <* USEDIN ( pd. formation.of_product
     , 'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
     'PRODUCTS' ) | prpc. name IN [ 'grade' , 'substrate' , 'coating' ] )
     ) =0 ) ) ) =0;
END_RULE;
```

Argument definitions:

product_definition: the set of all instances of product_definition.

product_related_product_category: the set of all instances of product_related_product_category.

Formal propositions:

WR1: If the product_definition references as its frame_of_reference a product_definition_context with a name of 'part definition' or 'part occurrence', this product_definition shall reference as its formation a product_definition_formation which references as of_product a product which is in the set of products of a product_related_product_category with a name of 'cutting item', 'tool item', 'adaptive item'.

WR2: If the product_definition references as its frame_of_reference a product_definition_context with a name of 'physical occurrence', this product_definition shall reference as its formation a product_definition_formation which references as of_product a product which is in the set of products of a product_related_product_category with a name of 'physically realized product'.

WR3: If the product_definition references as its frame_of_reference a product_definition_context with a name of 'physical document definition' or 'digital document definition', this product_definition shall reference as its formation a product_definition_formation which references as of_product a product which is in the set of products of a product_related_product_category with a name of 'document'.

WR4: If the product_definition references as its frame_of_reference a product_definition_context with a name of 'material definition', this product_definition shall reference as its formation a product_definition_formation which references as of_product a product which is in the set of products of a product_related_product_category with a name of 'grade', 'substrate', or 'coating'.

6.2.3.58 restrict_product_definitions_for_product_definition_relationship

The restrict_product_definitions_for_product_definition_relationship rule ensures the correct correlation between the name or type of the product_definition_relationship and the product_definition_context of the related product_definitions.

EXPRESS specification:

```
RULE restrict_product_definitions_for_product_definition_relationship FOR
    (assembly_component_usage,
     product_definition_relationship);
WHERE
 WR1: SIZEOF ( QUERY ( pdr <* product_definition_relationship |
     ( 'CUTTING_TOOL_SCHEMA.'+ 'ASSEMBLY_COMPONENT_USAGE' IN
     TYPEOF ( pdr ) ) AND ( ( pdr.
     relating_product_definition.frame_of_reference. name<>
     'part definition' ) OR ( pdr.
     related_product_definition.frame_of_reference. name<>
     'part definition' ) ) AND ( ( pdr.
     relating_product_definition.frame_of_reference. name<>
     'physical occurrence' ) OR ( pdr.
     related_product_definition.frame_of_reference. name<>
     'physical occurrence' ) ) ) ) =0;
 WR2: SIZEOF ( QUERY ( pdr <* product_definition_relationship |
     ( pdr. name = 'physical realization' ) AND ( ( pdr.
     relating_product_definition.frame_of_reference. name<>
     'part definition' ) OR ( pdr.
     related_product_definition.frame_of_reference. name<>
     'physical occurrence' ) ) ) ) =0;
 WR3: SIZEOF ( QUERY ( pdr <* product_definition_relationship |
     ( pdr. name = 'physical occurrence usage' ) AND ( NOT (
     'CUTTING_TOOL_SCHEMA.'+ 'ASSEMBLY_COMPONENT_USAGE' IN TYPEOF
     ( pdr ) ) OR ( pdr.
     relating_product_definition.frame_of_reference. name<>
     'physical occurrence' ) OR ( pdr.
     related_product_definition.frame_of_reference. name<>
     'physical occurrence' ) ) ) ) =0;
 WR4: SIZEOF ( QUERY ( pdr <* product_definition_relationship |
     ( pdr. name = 'substrate' ) AND ( ( pdr.
     relating_product_definition.frame_of_reference. name<>
     'material definition' ) OR ( pdr.
     related_product_definition.frame_of_reference. name<>
     'material definition' ) ) ) ) =0;
 WR5: SIZEOF ( QUERY ( pdr <* product_definition_relationship |
     ( pdr. name = 'coating' ) AND ( ( pdr.
     relating_product_definition.frame_of_reference. name<>
     'material definition' ) OR ( pdr.
     related_product_definition.frame_of_reference. name<>
     'material definition' ) ) ) ) =0;
END_RULE;
```

Argument definitions:

assembly_component_usage: the set of all instances of assembly_component_usage.

product_definition_relationship: the set of all instances of product_definition_relationship.

Formal propositions:

WR1: If the product_definition_relationship is of type assembly_component_usage, the relating_product_definition and the related_product_definition shall either both reference as their frame_of_reference a product_definition_context with a name of 'part definition', or shall both reference as their frame_of_reference a product_definition_context with a name of 'physical occurrence'.

WR2: If the product_definition_relationship has a name of 'physical realization', the relating_product_definition shall reference as its frame_of_reference a product_definition_context with a name of 'part definition', and the related_product_definition shall reference as its frame_of_reference a product_definition_context with a name of 'physical occurrence'.

WR3: If the product_definition_relationship has a name of 'physical occurrence usage', it shall be of type assembly_component_usage and both the relating_product_definition and the related_product_definition shall reference as frame_of_reference a product_definition_context with a name of 'physical occurrence'.

WR4: If the product_definition_relationship has a name of 'substrate', the relating_product_definition shall reference as its frame_of_reference a product_definition_context with a name of 'material definition', and the related_product_definition shall reference as its frame_of_reference a product_definition_context with a name of 'material definition'.

WR5: If the product_definition_relationship has a name of 'coating', the relating_product_definition shall reference as its frame_of_reference a product_definition_context with a name of 'material definition', and the related_product_definition shall reference as its frame_of_reference a product_definition_context with a name of 'material definition'.

6.2.3.59 restrict_properties_of_document_file

The restrict_properties_of_document_file rule specifies the restrictions that apply for properties of digital_files.

EXPRESS specification:

```
RULE restrict_properties_of_document_file FOR
    (document_file);
WHERE
 WR1: SIZEOF ( QUERY ( df <* document_file | SIZEOF ( QUERY ( pd
     <* USEDIN ( df ,
     'CUTTING_TOOL_SCHEMA.'+'PROPERTY_DEFINITION.'+ 'DEFINITION' )
     | pd. name = 'document property' ) ) >1 ) ) =0;
 WR2: SIZEOF ( QUERY ( df <* document_file | SIZEOF ( QUERY ( rt
     <* df. representation_types | ( 'CUTTING_TOOL_SCHEMA.'+
     'DOCUMENT_REPRESENTATION_TYPE' IN TYPEOF ( rt ) ) AND ( rt.
     name IN ['digital' , 'physical'] ) ) ) =0 ) ) =0;
END_RULE;
```

Argument definitions:

document_file: the set of all instances of document_file.

Formal propositions:

WR1: Each instance of document_file shall be referenced as definition by at most one property_definition with a name of 'document property'.

WR2: Each instance of document_file shall reference as representation_types an instance of document_representation_type with a name of either 'digital', or 'physical'.

6.2.3.60 restrict_properties_of_document_representation

The restrict_properties_of_document_representation rule specifies the restrictions that apply for properties of document_representations.

EXPRESS specification:

```
RULE restrict_properties_of_document_representation FOR
    (product_definition);
WHERE
 WR1: SIZEOF ( QUERY ( pd <* product_definition | ( pd.
     frame_of_reference.name IN [ 'physical document definition'
     , 'digital document definition' ] ) AND (
     SIZEOF ( QUERY ( pd1 <* USEDIN ( pd ,
     'CUTTING_TOOL_SCHEMA.'+'PROPERTY_DEFINITION.'+ 'DEFINITION' )
     | pd1. name = 'document property' ) ) >1 ) ) ) =0;
END_RULE;
```

Argument definitions:

product_definition: the set of all instances of product_definition.

Formal propositions:

WR1: Each instance of product_definition that references as frame_of_reference a application_context_element with a name of either 'physical document definition' or 'digital document definition', shall be referenced as definition by at most one property_definition with a name of 'document property'.

6.2.3.61 restrict_representation_for_coupling

The restrict_representation_for_document_content_property rule specifies the restrictions that apply for representations with a name of 'coupling'.

EXPRESS specification:

```
RULE restrict_representation_for_coupling FOR
    (representation,
     representation_item);
WHERE
 WR1: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'coupling' ) AND ( ( SIZEOF ( r.items ) <4 ) OR (
     SIZEOF ( r.items ) >5 ) ) ) ) =0;
 WR2: SIZEOF ( QUERY ( ri <* representation_item | ( SIZEOF (
     QUERY ( r <* USEDIN ( ri ,
     'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS' ) | r.name =
     'coupling' ) ) >0 ) AND NOT ( ri. name IN [
     'coupling type' , 'pieces' , 'side' , 'size' , 'style' ] ) )
     ) =0;
 WR3: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'coupling' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'coupling type' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
     'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
     ) ) ) =0;
 WR4: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'coupling' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'pieces' ) AND ( SIZEOF (
     ['CUTTING_TOOL_SCHEMA.'+ 'MEASURE_REPRESENTATION_ITEM' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_REPRESENTATION_ITEM']* TYPEOF
     ( i ) ) =1 ) ) ) >1 ) ) ) =0;
 WR5: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'coupling' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'side' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
     'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
     ) ) ) =0;
 WR6: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'coupling' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'size' ) AND ( SIZEOF (
     ['CUTTING_TOOL_SCHEMA.'+ 'MEASURE_REPRESENTATION_ITEM' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_REPRESENTATION_ITEM']* TYPEOF
     ( i ) ) =1 ) ) ) >1 ) ) ) =0;
 WR7: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'coupling' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'style' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
     'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
     ) ) ) =0;
END_RULE;
```

Argument definitions:

representation: the set of all instances of representation.

representation_item: the set of all instances of representation_item.

Formal propositions:

WR1: For each instance of representation which has a name of 'coupling' there shall be at least four and at most five instances of representation_items in the set of items.

WR2: Each instance of representation_item that is referenced as items by an instance of representation with a name of 'coupling' shall have a name of either 'coupling type', 'pieces', 'side', 'size', or 'style'.

WR3: There shall be at most one instance of representation_item that have a name of 'coupling type' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'coupling'.

WR4: There shall be at most one instance of representation_item that have a name of 'pieces' and which is either of type measure_representation_item, or value_representation_item that is referenced as items by an instance of representation with a name of 'coupling'.

WR5: There shall be at most one instance of representation_item that have a name of 'side' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'coupling'.

WR6: There shall be at most one instance of representation_item that have a name of 'size' and which is either of type measure_representation_item, or value_representation_item that is referenced as items by an instance of representation with a name of 'coupling'.

WR7: There shall be at most one instance of representation_item that have a name of 'style' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'coupling'.

6.2.3.62 restrict_representation_for_cutting_condition

The restrict_representation_for_document_content_property rule specifies the restrictions that apply for representations with a name of 'cutting condition'.

EXPRESS specification:

```
RULE restrict_representation_for_cutting_condition FOR
    (representation,
     representation_item);
WHERE
 WR1: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'cutting condition' ) AND ( SIZEOF ( r.items ) <1 ) ) ) =0;
 WR2: SIZEOF ( QUERY ( ri <* representation_item | ( SIZEOF (
     QUERY ( r <* USEDIN ( ri ,
     'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS' ) | r.name =
     'cutting condition' ) ) >0 ) AND NOT ( ri. name IN [
     'condition name' ] ) )
     ) =0;
 WR3: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'cutting condition' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'condition name' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
     'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
     ) ) ) =0;
END_RULE;
```

Argument definitions:

representation: the set of all instances of representation.

representation_item: the set of all instances of representation_item.

Formal propositions:

WR1: For each instance of representation which has a name of 'cutting condition' there shall be at least one instance of representation_items in the set of items.

WR2: Each instance of representation_item that is referenced as items by an instance of representation with a name of 'cutting condition' shall have a name of 'condition name'.

WR3: There shall be at most one instance of representation_item that have a name of 'condition name' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'cutting condition'.

6.2.3.63 restrict_representation_for_cutting_data_association

The restrict_representation_for_document_content_property rule specifies the restrictions that apply for representations with a name of 'cutting data association'.

EXPRESS specification:

```
RULE restrict_representation_for_cutting_data_association FOR
    (representation,
     representation_item);
WHERE
 WR1: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'cutting data association' ) AND ( SIZEOF ( r.items ) <3 ) ) ) =0;
 WR2: SIZEOF ( QUERY ( ri <* representation_item | ( SIZEOF (
     QUERY ( r <* USEDIN ( ri ,
     'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS' ) | r.name =
     'cutting data association' ) ) >0 ) AND NOT ( ri. name IN [
     'depth of cut' , 'feed' , 'speed' ] ) )
     ) =0;
 WR3: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'cutting data association' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'depth of cut' ) AND ( SIZEOF (
     ['CUTTING_TOOL_SCHEMA.'+ 'MEASURE_REPRESENTATION_ITEM' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_RANGE' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_REPRESENTATION_ITEM']* TYPEOF
     ( i ) ) =1 ) ) ) >1 ) ) ) =0;
 WR4: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'cutting data association' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'feed' ) AND ( SIZEOF (
     ['CUTTING_TOOL_SCHEMA.'+ 'MEASURE_REPRESENTATION_ITEM' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_RANGE' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_REPRESENTATION_ITEM']* TYPEOF
     ( i ) ) =1 ) ) ) >1 ) ) ) =0;
 WR5: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'cutting data association' ) AND ( SIZEOF ( QUERY ( i <* r.items |
     ( i.name = 'speed' ) AND ( SIZEOF (
     ['CUTTING_TOOL_SCHEMA.'+ 'MEASURE_REPRESENTATION_ITEM' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_RANGE' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_REPRESENTATION_ITEM']* TYPEOF
     ( i ) ) =1 ) ) ) >1 ) ) ) =0;
END_RULE;
```

<u>**Argument definitions:**</u>

representation: the set of all instances of representation.

representation_item: the set of all instances of representation_item.

<u>**Formal propositions:**</u>

WR1: For each instance of representation which has a name of 'cutting data association' there shall be at least three instances of representation_items in the set of items.

WR2: Each instance of representation_item that is referenced as items by an instance of representation with a name of 'cutting data association' shall have a name of either 'depth of cut', 'feed', or 'speed'.

WR4: There shall be at most one instance of representation_item that have a name of 'depth of cut' and which is either of type measure_representation_item, value_range, or value_representation_item that is referenced as items by an instance of representation with a name of 'cutting data association'.

WR5: There shall be at most one instance of representation_item that have a name of 'feed' and which is either of type measure_representation_item, value_range, or value_representation_item that is referenced as items by an instance of representation with a name of 'cutting data association'.

WR6: There shall be at most one instance of representation_item that have a name of 'speed' and which is either of type measure_representation_item, value_range, or value_representation_item that is referenced as items by an instance of representation with a name of 'cutting data association'.

6.2.3.64 restrict_representation_for_document_content_property

The restrict_representation_for_document_content_property rule specifies the restrictions that apply for representations with a name of 'document content'.

<u>**EXPRESS specification:**</u>

```
  RULE restrict_representation_for_document_content_property FOR
      (representation,
       representation_item);
  WHERE
   WR1: SIZEOF ( QUERY ( r <* representation | ( r.name =
       'document content' ) AND ( ( SIZEOF ( r.items ) <1 ) OR (
       SIZEOF ( r.items ) >3 ) ) ) ) =0;
   WR2: SIZEOF ( QUERY ( ri <* representation_item | ( SIZEOF (
       QUERY ( r <* USEDIN ( ri ,
       'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS' ) | r.name =
       'document content' ) ) >0 ) AND NOT ( ri. name IN [
       'detail level' , 'geometry type' , 'real world scale' ] ) )
       ) =0;
   WR3: SIZEOF ( QUERY ( r <* representation | ( r.name =
       'document content' ) AND ( SIZEOF ( QUERY ( i <* r.items |
       ( i.name = 'detail level' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
       'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
       ) ) ) =0;
   WR4: SIZEOF ( QUERY ( r <* representation | ( r.name =
       'document content' ) AND ( SIZEOF ( QUERY ( i <* r.items |
       ( i.name = 'geometry type' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
       'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
       ) ) ) =0;
   WR5: SIZEOF ( QUERY ( r <* representation | ( r.name =
       'document content' ) AND ( SIZEOF ( QUERY ( i <* r.items |
       ( i.name = 'real world scale' ) AND ( SIZEOF (
       ['CUTTING_TOOL_SCHEMA.'+ 'MEASURE_REPRESENTATION_ITEM' ,
       'CUTTING_TOOL_SCHEMA.'+ 'VALUE_REPRESENTATION_ITEM']* TYPEOF
       ( i ) ) =1 ) ) ) >1 ) ) ) =0;
  END_RULE;
```

Argument definitions:

representation: the set of all instances of representation.

representation_item: the set of all instances of representation_item.

Formal propositions:

WR1: For each instance of representation which has a name of 'document content' there shall be at least one and at most three instances of representation_items in the set of items.

WR2: Each instance of representation_item that is referenced as items by an instance of representation with a name of 'document content' shall have a name of either 'detail level', 'geometry type', or 'real world scale'.

WR3: There shall be at most one instance of representation_item that have a name of 'detail level' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'document content'.

WR4: There shall be at most one instance of representation_item that have a name of 'geometry type' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'document content'.

WR5: There shall be at most one instance of representation_item that have a name of 'real world scale' and which is either of type measure_representation_item, or value_representation_item that is referenced as items by an instance of representation with a name of 'document content'.

6.2.3.65 restrict_representation_for_document_creation_property

The restrict_representation_for_document_creation_property rule specifies the restrictions that apply for representations with a name of 'document creation'.

EXPRESS specification:

```
  RULE restrict_representation_for_document_creation_property FOR
      (representation,
       representation_item);
  WHERE
   WR1: SIZEOF ( QUERY ( r <* representation | ( r.name =
       'document creation' ) AND ( ( SIZEOF ( r.items ) <2 ) OR (
       SIZEOF ( r.items ) >3 ) ) ) ) ) =0;
   WR2: SIZEOF ( QUERY ( ri <* representation_item | ( SIZEOF (
       QUERY ( r <* USEDIN ( ri ,
       'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS' ) | r.name =
       'document creation' ) ) >0 ) AND NOT ( ri. name IN [
       'creating interface' , 'creating system' ,
       'operating system' ] ) ) ) ) =0;
   WR3: SIZEOF ( QUERY ( r <* representation | ( r.name =
       'document creation' ) AND ( SIZEOF ( QUERY ( i <* r.items |
       ( i.name = 'creating interface' ) AND (
       'CUTTING_TOOL_SCHEMA.'+ 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF ( i ) ) ) ) >1 ) ) ) ) =0;
   WR4: SIZEOF ( QUERY ( r <* representation | ( r.name =
       'document creation' ) AND ( SIZEOF ( QUERY ( i <* r.items |
       ( i.name = 'creating system' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
       'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) <>1
       ) ) ) =0;
   WR5: SIZEOF ( QUERY ( r <* representation | ( r.name =
       'document creation' ) AND ( SIZEOF ( QUERY ( i <* r.items |
       ( i.name = 'operating system' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
       'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
       ) ) ) =0;
  END_RULE;
```

Argument definitions:

representation: the set of all instances of representation.

representation_item: the set of all instances of representation_item.

Formal propositions:

WR1: For each instance of representation which has a name of 'document creation' there shall be at least two one and at most three instances of representation_items in the set of items.

WR2: Each instance of representation_item that is referenced as items by an instance of representation with a name of 'document creation' shall have a name of either 'creating interface', 'creating system', or 'operating system'.

WR3: There shall be at most one instance of representation_item that have a name of 'creating interface' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'document creation'.

WR4: There shall be exactly one instance of representation_item that have a name of 'creating system' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'document creation'.

WR5: There shall be at most one instance of representation_item that have a name of 'operating system' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'document creation'.

6.2.3.66 restrict_representation_for_document_format_property

The restrict_representation_for_document_format_property rule specifies the restrictions that apply for representations with a name of 'document format'.

EXPRESS specification:

```
RULE restrict_representation_for_document_format_property FOR
    (representation,
     representation_item);
WHERE
 WR1: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'document format' ) AND ( ( SIZEOF ( r.items ) <1 ) OR (
     SIZEOF ( r.items ) >3 ) ) ) ) =0;
 WR2: SIZEOF ( QUERY ( ri <* representation_item | ( SIZEOF (
     QUERY ( r <* USEDIN ( ri ,
     'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS' ) | r.name =
     'document format' ) ) >0 ) AND NOT ( ri. name IN [
     'character code' , 'data format' ] ) ) ) =0;
 WR3: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'document format' ) AND ( SIZEOF ( QUERY ( i <* r.items | (
     i.name = 'character code' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
     'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
     ) ) ) =0;
 WR4: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'document format' ) AND ( SIZEOF ( QUERY ( i <* r.items | (
     i.name = 'data format' ) AND ( 'CUTTING_TOOL_SCHEMA.'+
     'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF ( i ) ) ) ) >1
     ) ) ) =0;
 WR5:  SIZEOF( QUERY( r <* representation |
              (r.name = 'document format') AND
              (SIZEOF( QUERY( i <* r.items |
                 (i.name = 'size format') AND
                 ('CUTTING_TOOL_SCHEMA.PLANAR_EXTENT' IN
                  TYPEOF(i))
                )) > 1)
             )) = 0;
END_RULE;
```

Argument definitions:

representation: the set of all instances of representation.

representation_item: the set of all instances of representation_item.

Formal propositions:

WR1: For each instance of representation which has a name of 'document format' there shall be at least one and at most three instances of representation_items in the set of items.

WR2: Each instance of representation_item that is referenced as items by an instance of representation with a name of 'document format' shall have a name of either 'character code', or 'data format'.

WR3: There shall be at most one instance of representation_item that have a name of 'character code' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'document format'.

WR4: There shall be at most one instance of representation_item that have a name of 'data format' and which is of type desriptive_representation_item that is referenced as items by an instance of representation with a name of 'document format'.

WR5: There shall be at most one instance of representation_item that have a name of 'size format' and which is of type planar_extent that is referenced as items by an instance of representation with a name of 'document format'.

6.2.3.67 restrict_representation_for_document_properties

The restrict_representation_for_document_properties rule specifies that each instance of property_definition with a name of 'document property' shall be represented by representations with a name of either 'document content', 'document creation', 'document format', or 'document size'. At the same time, this rules enforces that each instance of representation with a name of either 'document content', 'document creation', 'document format', or 'document size', shall be referenced by one instance of property_definition with a name of 'document property'.

EXPRESS specification:

```
RULE restrict_representation_for_document_properties FOR
     (property_definition,
      representation,
      representation_context);
WHERE
 WR1: SIZEOF ( QUERY ( pd <* property_definition | ( pd. name =
      'document property' ) AND ( SIZEOF ( QUERY ( pdr <* USEDIN
      ( pd ,

   'CUTTING_TOOL_SCHEMA.'+'PROPERTY_DEFINITION_REPRESENTATION.'+'DEFINITION
   '
      ) | NOT ( pdr. used_representation.name IN [
      'document content' , 'document creation' ,
      'document format' , 'document size' ] ) ) ) >0 ) ) ) =0;
 WR2: SIZEOF ( QUERY ( r <* representation | ( r.name IN [
      'document content' , 'document creation' ,
      'document format' , 'document size' ] ) AND ( SIZEOF (
      QUERY ( pdr <* USEDIN ( r ,
      'CUTTING_TOOL_SCHEMA.'+'PROPERTY_DEFINITION_REPRESENTATION.'
      +'USED_REPRESENTATION'
      ) | ( 'CUTTING_TOOL_SCHEMA.'+ 'PROPERTY_DEFINITION' IN TYPEOF
      ( pdr. definition ) ) AND ( pdr. definition.name =
      'document property' ) ) ) =0 ) ) ) =0;
 WR3: SIZEOF ( QUERY ( r <* representation | ( r.name IN [
      'document content' , 'document creation' ,
      'document format' , 'document size' ] ) AND (
      r.context_of_items. context_type<> 'document parameters' )
      ) ) =0;
 WR4: SIZEOF ( QUERY ( rc <* representation_context | ( rc.
      context_type = 'document parameters' ) AND ( SIZEOF ( QUERY
      ( r <* USEDIN ( rc ,
      'CUTTING_TOOL_SCHEMA.REPRESENTATION.CONTEXT_OF_ITEMS' ) | NOT
      ( r.name IN [ 'document content' , 'document creation' ,
      'document format' , 'document size' ] ) ) ) >0 ) ) ) =0;
END_RULE;
```

Argument definitions:

property_definition: the set of all instances of property_definition.

representation: the set of all instances of representation.

representation_context: the set of all instances of representation_context.

Formal propositions:

WR1: Each instance of property_definition with a name of 'document property' shall be referenced as definition by a property_definition_representation, which has as used_representation a representation with a name of either 'document content', 'document creation', 'document format', or 'document size'.

WR2: Each instance of representation with a name of either 'document content', 'document creation', 'document format', or 'document size' shall be referenced as used_representation by a property_definition_representation, which has as definition a property_definition with a name of 'document property'.

WR3: Each instance of representation with a name of either 'document content', 'document creation', 'document format', or 'document size' shall reference as context_of_items a representation_context, whith a context_type of 'document parameters'.

WR4: Each instance of representation_context with a context_type of 'document parameters' shall be referenced as context_of_items by a representation with a name of either 'document content', 'document creation', 'document format', or 'document size'.

6.2.3.68 restrict_representation_for_document_size_property

The restrict_representation_for_document_size_property rule specifies the restrictions that apply for representations with a name of 'document size'.

EXPRESS specification:

```
RULE restrict_representation_for_document_size_property FOR
    (representation,
     representation_item);
WHERE
 WR1: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'document size' ) AND ( ( SIZEOF ( r.items ) <1 ) OR (
     SIZEOF ( r.items ) >2 ) ) ) ) =0;
 WR2: SIZEOF ( QUERY ( ri <* representation_item | ( SIZEOF (
     QUERY ( r <* USEDIN ( ri ,
     'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS' ) | r.name =
     'document size' ) ) >0 ) AND NOT ( ri. name IN [
     'file size' , 'page count' ] ) ) ) ) =0;
 WR3: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'document size' ) AND ( SIZEOF ( QUERY ( i <* r.items | (
     i.name = 'file size' ) AND ( SIZEOF (
     ['CUTTING_TOOL_SCHEMA.'+ 'MEASURE_REPRESENTATION_ITEM' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_RANGE' , 'CUTTING_TOOL_SCHEMA.'+
     'VALUE_REPRESENTATION_ITEM']* TYPEOF ( i ) ) =1 ) ) ) >1 )
     ) ) =0;
 WR4: SIZEOF ( QUERY ( r <* representation | ( r.name =
     'document size' ) AND ( SIZEOF ( QUERY ( i <* r.items | (
     i.name = 'page count' ) AND ( SIZEOF (
     ['CUTTING_TOOL_SCHEMA.'+ 'MEASURE_REPRESENTATION_ITEM' ,
     'CUTTING_TOOL_SCHEMA.'+ 'VALUE_RANGE' , 'CUTTING_TOOL_SCHEMA.'+
     'VALUE_REPRESENTATION_ITEM']* TYPEOF ( i ) ) =1 ) ) ) >1 )
     ) ) =0;
END_RULE;
```

Argument definitions:

representation: the set of all instances of representation.

representation_item: the set of all instances of representation_item.

Formal propositions:

WR1: For each instance of representation which has a name of 'document size' there shall be at least one and at most two instances of representation_items in the set of items.

WR2: Each instance of representation_item that is referenced as items by an instance of representation with a name of 'document size' shall have a name of either 'file size', or 'page count'.

WR3: There shall be at most one instance of representation_item that have a name of 'file size' and which is either of type measure_representation_item, value_range, or value_representation_item that is referenced as items by an instance of representation with a name of 'document size'.

WR4: There shall be at most one instance of representation_item that have a name of 'page count' and which is either of type measure_representation_item, value_range, or value_representation_item that is referenced as items by an instance of representation with a name of 'document size'.

6.2.3.69 restrict_version_assignment_for_applied_identification_assignment

The restrict_version_assignment_for_applied_identification_assignment rule specifies that each instance of applied_identification_assignment shall be referenced by at most one instance of applied_identification_assignment that specifies the version of the applied_identification_assignment.

EXPRESS specification:

```
RULE restrict_version_assignment_for_applied_identification_assignment FOR
    (applied_identification_assignment);
WHERE
 WR1: SIZEOF ( QUERY ( ent <* applied_identification_assignment
     | SIZEOF ( QUERY ( ia <* USEDIN ( ent ,
      'CUTTING_TOOL_SCHEMA.'+'APPLIED_IDENTIFICATION_ASSIGNMENT.'+
      'ITEMS' ) | ia. role.name ='version' ) ) >1 ) ) =0;
END_RULE;
```

Argument definitions:

applied_identification_assignment: the set of all instances of applied_identification_assignment.

Formal propositions:

WR1: For each instance of applied_identification_assignment there shall be only at most one instance of applied_identification_assignment with a role name of 'version'.

6.2.3.70 restrict_version_assignment_for_class

The restrict_version_assignment_for_class rule specifies that each instance of class shall be referenced by at most one instance of applied_identification_assignment that specifies the version of the class.

EXPRESS specification:

```
RULE restrict_version_assignment_for_class FOR
    (class);
WHERE
```

```
   WR1: SIZEOF ( QUERY ( ent <* class | SIZEOF ( QUERY ( ia <*
        USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'APPLIED_IDENTIFICATION_ASSIGNMENT.'+
        'ITEMS' ) | ia. role.name ='version' ) ) >1 ) ) =0;
  END_RULE;
```

Argument definitions:

class: the set of all instances of class.

Formal propositions:

WR1: For each instance of class there shall be only at most one instance of applied_identification_assignment with a role name of 'version'.

6.2.3.71 restrict_version_assignment_for_document_file

The restrict_version_assignment_for_document_file rule specifies that each instance of document_file shall be referenced by at most one instance of applied_identification_assignment that specifies the version of the document_file.

EXPRESS specification:

```
  RULE restrict_version_assignment_for_document_file FOR
      (document_file);
  WHERE
   WR1: SIZEOF ( QUERY ( ent <* document_file | SIZEOF ( QUERY (
        ia <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'APPLIED_IDENTIFICATION_ASSIGNMENT.'+
        'ITEMS' ) | ia. role.name ='version' ) ) >1 ) ) =0;
  END_RULE;
```

Argument definitions:

document_file: the set of all instances of document_file.

Formal propositions:

WR1: For each instance of document_file there shall be at most one instance of applied_identification_assignment with a role name of 'version'.

6.2.3.72 restrict_version_assignment_for_effectivity

The restrict_version_assignment_for_effectivity rule specifies that each instance of effectivity shall be referenced by at most one instance of applied_identification_assignment that specifies the version of the effectivity.

EXPRESS specification:

```
  RULE restrict_version_assignment_for_effectivity FOR
      (effectivity);
  WHERE
   WR1: SIZEOF ( QUERY ( ent <* effectivity | SIZEOF ( QUERY ( ia
        <* USEDIN ( ent ,
        'CUTTING_TOOL_SCHEMA.'+'APPLIED_IDENTIFICATION_ASSIGNMENT.'+
        'ITEMS' ) | ia. role.name ='version' ) ) >1 ) ) =0;
  END_RULE;
```

Argument definitions:

effectivity: the set of all instances of effectivity.

Formal propositions:

WR1: For each instance of effectivity there shall be only at most one instance of applied_identification_assignment with a role name of 'version'.

6.2.3.73 substrate_requires_product_definition

The substrate_requires_product_definition rule ensures that each instance of product_definition_formation which has an associated of_product that is referenced by a product_related_product_category with a name of 'substrate', is referenced as formation by exactly one instance of product_definition that references as its frame_of_reference a product_definition_context with a name of 'material definition'.

EXPRESS specification:

```
RULE grade_requires_product_definition FOR
    (product_definition,
     product_definition_formation,
     product_related_product_category);
WHERE
 WR1: SIZEOF ( QUERY ( pdf <* product_definition_formation | (
     SIZEOF ( QUERY ( prpc <* USEDIN ( pdf. of_product ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_RELATED_PRODUCT_CATEGORY.'+
     'PRODUCTS' ) | prpc. name = 'substrate' )
     ) >0 ) AND ( SIZEOF ( QUERY ( pd <* USEDIN ( pdf ,
     'CUTTING_TOOL_SCHEMA.'+'PRODUCT_DEFINITION.'+ 'FORMATION' ) |
     pd. frame_of_reference.name = 'material definition' ) ) <>1
     ) ) ) =0;
END_RULE;
```

Argument definitions:

product_definition: the set of all instances of product_definition.

product_definition_formation: the set of all instances of product_definition_formation.

product_related_product_category: the set of all instances of product_related_product_category.

Formal propositions:

WR1: Each instance of product_definition_formation which has an associated of_product that is referenced by a product_related_product_category with a name of 'substrate', is referenced as formation by exactly one instance of product_definition that references as its frame_of_reference a product_definition_context with a name of 'material definition'.

Annex A
(normative)

AIM longform

```
SCHEMA CUTTING_TOOL_SCHEMA;

  CONSTANT
    DUMMY_GRI : GEOMETRIC_REPRESENTATION_ITEM := representation_item('') ||
  geometric_representation_item();
  END_CONSTANT;

  TYPE action_item = SELECT
    (applied_classification_assignment,
     applied_person_and_organization_assignment,
     class,
     class_system,
     document_file);
  END_TYPE;

  TYPE language_item = SELECT
    (representation);
  END_TYPE;

  TYPE label = STRING;
  END_TYPE;

  TYPE length_measure = REAL;
  END_TYPE;

  TYPE text = STRING;
  END_TYPE;

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

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

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

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

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

  TYPE mass_measure = REAL;
  END_TYPE;

  TYPE time_measure = REAL;
  END_TYPE;

  TYPE electric_current_measure = REAL;
  END_TYPE;

  TYPE thermodynamic_temperature_measure = REAL;
  END_TYPE;

  TYPE celsius_temperature_measure = REAL;
  END_TYPE;

  TYPE amount_of_substance_measure = REAL;
  END_TYPE;

  TYPE luminous_intensity_measure = REAL;
  END_TYPE;

  TYPE plane_angle_measure = REAL;
  END_TYPE;

  TYPE solid_angle_measure = REAL;
  END_TYPE;

  TYPE area_measure = REAL;
  END_TYPE;

  TYPE volume_measure = REAL;
  END_TYPE;

  TYPE ratio_measure = REAL;
  END_TYPE;

  TYPE parameter_value = REAL;
  END_TYPE;
```

```
TYPE numeric_measure = NUMBER;
END_TYPE;

TYPE context_dependent_measure = REAL;
END_TYPE;

TYPE descriptive_measure = STRING;
END_TYPE;

TYPE positive_length_measure = length_measure;
  WHERE
    WR1 :
          SELF > 0.00000;
END_TYPE;

TYPE positive_plane_angle_measure = plane_angle_measure;
  WHERE
    WR1 :
          SELF > 0.00000;
END_TYPE;

TYPE positive_ratio_measure = ratio_measure;
  WHERE
    WR1 :
          SELF > 0.00000;
END_TYPE;

TYPE count_measure = NUMBER;
END_TYPE;

TYPE unit = SELECT
   (named_unit,
    derived_unit);
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 value_qualifier = SELECT
   (precision_qualifier,
    type_qualifier);
 END_TYPE;

 TYPE identifier = STRING;
 END_TYPE;

 TYPE source_item = SELECT
   (identifier);
 END_TYPE;

 TYPE classification_item = SELECT
   (product,
    document_file,
    product_definition,
    product_definition_formation,
    product_definition_relationship,
    material_designation,
    general_property,
    property_definition,
    class,
    plus_minus_tolerance,
    document_type,
    planar_extent);
 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 shape_definition = SELECT
  (product_definition_shape,
   shape_aspect);
END_TYPE;

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

TYPE dimensional_characteristic = SELECT
  (dimensional_size);
END_TYPE;

TYPE person_and_organization_item = SELECT
  (applied_classification_assignment,
   class_system,
   product,
   document_file,
   product_definition,
   product_definition_formation,
   class,
   product_definition_relationship,
   product_definition_formation_relationship,
   material_designation,
   person_and_organization,
   general_property,
   property_definition);
END_TYPE;

TYPE action_request_item = SELECT
  (document_file);
END_TYPE;

TYPE ahead_or_behind = ENUMERATION OF
  (AHEAD,
   EXACT,
   BEHIND);
END_TYPE;

TYPE applied_state_type_of_item = SELECT
  (product_definition);
END_TYPE;

TYPE approval_item = SELECT
  (applied_classification_assignment,
   class,
   class_system,
   document_file);
END_TYPE;
```

```
TYPE attribute_language_item = SELECT
  (property_definition,
   group,
   external_source,
   product_related_product_category,
   effectivity,
   effectivity_relationship,
   product,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   representation,
   product_definition_relationship,
   general_property,
   general_property_relationship,
   representation_relationship,
   general_feature,
   identification_role,
   application_context,
   attribute_value_assignment,
   person_and_organization_role,
   organization_role,
   descriptive_representation_item);
END_TYPE;

TYPE date_time_or_event_occurrence = SELECT
  (date_time_select);
END_TYPE;

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

TYPE day_in_month_number = INTEGER;
  WHERE
    WR1 :
          (1 <= SELF) AND (SELF <= 31);
END_TYPE;

TYPE month_in_year_number = INTEGER;
  WHERE
    WR1 :
          (1 <= SELF) AND (SELF <= 12);
END_TYPE;

TYPE year_number = INTEGER;
END_TYPE;

TYPE hour_in_day = INTEGER;
  WHERE
    WR1 :
          (0 <= SELF) AND (SELF < 24);
END_TYPE;

TYPE minute_in_hour = INTEGER;
  WHERE
```

```
    WR1 :
          (0 <= SELF) AND (SELF <= 59);
END_TYPE;

TYPE second_in_minute = REAL;
  WHERE
    WR1 :
          (0 <= SELF) AND (SELF <= 60.0000);
END_TYPE;

TYPE transformation = SELECT
  (item_defined_transformation,
   functionally_defined_transformation);
END_TYPE;

TYPE multi_language_attribute_item = SELECT
  (property_definition,
   group,
   external_source,
   product_related_product_category,
   effectivity,
   effectivity_relationship,
   product_definition,
   product,
   product_definition_formation,
   product_definition_formation_relationship,
   representation,
   product_definition_relationship,
   general_property,
   general_property_relationship,
   representation_relationship,
   general_feature,
   identification_role,
   application_context,
   person_and_organization_role,
   organization_role,
   descriptive_representation_item);
END_TYPE;

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

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

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

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

TYPE date_and_time_item = SELECT
  (applied_classification_assignment,
   applied_organization_assignment,
   applied_person_and_organization_assignment,
   class,
   class_system,
   document_file);
END_TYPE;

TYPE organization_item = SELECT
  (class,
   property_definition,
   applied_identification_assignment,
   applied_classification_assignment,
   class_system,
   product,
   document_file,
   product_definition,
   product_definition_formation,
   product_definition_relationship,
   product_definition_formation_relationship,
   material_designation,
   person_and_organization,
   general_property);
END_TYPE;

TYPE external_identification_item = SELECT
  (document_file,
   product_definition,
   externally_defined_class,
   externally_defined_general_property);
END_TYPE;

TYPE identification_item = SELECT
  (document_file,
   product,
   general_property,
   applied_identification_assignment,
   property_definition,
   class_system,
   product_definition,
   document_type,
   class,
   product_definition_formation,
   product_definition_relationship,
   organization,
   person_and_organization,
   effectivity);
END_TYPE;

TYPE date_item = SELECT
  (applied_classification_assignment,
   applied_organization_assignment,
   applied_person_and_organization_assignment,
   class,
   class_system,
   document_file);
END_TYPE;

TYPE derived_property_select = SELECT
  (property_definition);
END_TYPE;

TYPE description_attribute_select = SELECT
  (application_context,
   context_dependent_shape_representation,
   effectivity,
   external_source,
   organization_role,
   person_and_organization_role,
   person_and_organization,
   property_definition_representation,
   representation);
END_TYPE;

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

TYPE document_reference_item = SELECT
  (property_definition,
   class_system,
   class,
   product,
   product_definition,
   product_definition_relationship,
   product_definition_formation,
   material_designation,
   organization,
   person,
   general_property,
   product_related_product_category,
   general_feature);
END_TYPE;

TYPE effectivity_item = SELECT
  (class_system,
   product,
   product_definition,
   document_file,
   product_definition_formation,
   product_definition_relationship,
   material_designation,
   applied_location_assignment,
   general_property,
   property_definition,
   applied_state_type_assignment);
END_TYPE;

TYPE location_assignment_item = SELECT
  (product_definition);
```

```
END_TYPE;

TYPE event_occurrence_item = SELECT
  (applied_classification_assignment,
   applied_person_and_organization_assignment,
   class,
   class_system,
   document_file);
END_TYPE;

TYPE founded_item_select = SELECT
  (representation_item);
END_TYPE;

TYPE geometric_set_select = SELECT
  (point);
END_TYPE;

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

TYPE invisible_item = SELECT
  (representation);
END_TYPE;

TYPE layered_item = SELECT
  (representation_item);
END_TYPE;

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

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

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

```
TYPE property_or_shape_select = SELECT
  (property_definition,
   shape_definition);
END_TYPE;

TYPE role_select = SELECT
  (effectivity_assignment);
END_TYPE;

TYPE security_classification_item = SELECT
  (class_system,
   document_file);
END_TYPE;

TYPE shape_tolerance_select = SELECT
  (plus_minus_tolerance);
END_TYPE;

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

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

TYPE time_interval_item = SELECT
  (applied_classification_assignment,
   applied_person_and_organization_assignment,
   class,
   class_system,
   document_file);
END_TYPE;

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

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

ENTITY applied_classification_assignment
  SUBTYPE OF(classification_assignment);
    items : SET [1:?] OF classification_item;
  WHERE
    WR1 :
          NOT ('CUTTING_TOOL_SCHEMA.' + 'CLASS' IN
TYPEOF(SELF.assigned_class)) OR (SIZEOF(QUERY (i <* SELF.items|
(SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'ACTION'), ('CUTTING_TOOL_SCHEMA.' +
'ACTION_DIRECTIVE'), ('CUTTING_TOOL_SCHEMA.' + 'ACTION_METHOD'),
('CUTTING_TOOL_SCHEMA.' + 'ACTION_PROPERTY'), ('CUTTING_TOOL_SCHEMA.' +
'APPLICATION_CONTEXT'), ('CUTTING_TOOL_SCHEMA.' + 'APPROVAL_STATUS'),
('CUTTING_TOOL_SCHEMA.' + 'ASSEMBLY_COMPONENT_USAGE'),
('CUTTING_TOOL_SCHEMA.' + 'CONFIGURATION_ITEM'), ('CUTTING_TOOL_SCHEMA.'
+ 'CONTRACT'), ('CUTTING_TOOL_SCHEMA.' + 'DOCUMENT_TYPE'),
('CUTTING_TOOL_SCHEMA.' + 'FEATURE_DEFINITION'), ('CUTTING_TOOL_SCHEMA.'
+ 'GENERAL_PROPERTY'), ('CUTTING_TOOL_SCHEMA.' +
'MATERIAL_DESIGNATION'), ('CUTTING_TOOL_SCHEMA.' +
'ORGANIZATIONAL_PROJECT'), ('CUTTING_TOOL_SCHEMA.' + 'PRODUCT'),
('CUTTING_TOOL_SCHEMA.' + 'PRODUCT_CONCEPT'), ('CUTTING_TOOL_SCHEMA.' +
'PRODUCT_CONCEPT_FEATURE_CATEGORY'), ('CUTTING_TOOL_SCHEMA.' +
'PRODUCT_DEFINITION'), ('CUTTING_TOOL_SCHEMA.' +
'PRODUCT_DEFINITION_FORMATION'), ('CUTTING_TOOL_SCHEMA.' +
'PROPERTY_DEFINITION'), ('CUTTING_TOOL_SCHEMA.' + 'RESOURCE_PROPERTY'),
('CUTTING_TOOL_SCHEMA.' + 'SECURITY_CLASSIFICATION_LEVEL'),
('CUTTING_TOOL_SCHEMA.' + 'SHAPE_ASPECT'), ('CUTTING_TOOL_SCHEMA.' +
'VERSIONED_ACTION_REQUEST') ] * TYPEOF(i)) <> 1))) = 0);
    WR2 :
          NOT ((('CUTTING_TOOL_SCHEMA.' + 'GROUP' IN
TYPEOF(SELF.assigned_class)) AND (SELF.assigned_class\group.name = 'E'))
AND (SELF.assigned_class\group.description = 'dimensioning principle'))
OR (SIZEOF(QUERY (i <* SELF.items| NOT ('CUTTING_TOOL_SCHEMA.' +
'PLUS_MINUS_TOLERANCE' IN TYPEOF(i)))) = 0);
    WR3 :
          NOT (('CUTTING_TOOL_SCHEMA.' + 'CLASS_SYSTEM' IN
TYPEOF(SELF.assigned_class)) AND (SELF.role.name = 'class system
membership')) OR (SIZEOF(QUERY (i <* SELF.items| (SIZEOF([
('CUTTING_TOOL_SCHEMA.' + 'APPROVAL_STATUS'), ('CUTTING_TOOL_SCHEMA.' +
'CLASS'), ('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'DOCUMENT_TYPE'), ('CUTTING_TOOL_SCHEMA.' +
'SECURITY_CLASSIFICATION_LEVEL'), ('CUTTING_TOOL_SCHEMA.' +
'APPROVAL_STATUS') ] * TYPEOF(i)) <> 1))) = 0);
END_ENTITY;

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

ENTITY language_assignment
  SUBTYPE OF(classification_assignment);
    items : SET [1:?] OF language_item;
  WHERE
    WR1 :
          'CUTTING_TOOL_SCHEMA.' + 'LANGUAGE' IN
TYPEOF(SELF.assigned_class);
    WR2 :
          SELF.role.name = 'language';
    WR3 :
          SIZEOF(SELF.items) = SIZEOF(QUERY (i <* SELF.items|
('CUTTING_TOOL_SCHEMA.' + 'REPRESENTATION' IN TYPEOF(i)) AND
(i\representation.name = 'document content')));
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, 'CUTTING_TOOL_SCHEMA.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
    WR2 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY;

ENTITY shape_representation
  SUBTYPE OF(representation);
END_ENTITY;

ENTITY shape_dimension_representation
  SUBTYPE OF(shape_representation);
  WHERE
    WR1 :
          SIZEOF(QUERY (temp <* SELF\representation.items| NOT
('CUTTING_TOOL_SCHEMA.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 > 0.00000))) = 0;
END_ENTITY;

ENTITY representation_item;
    name : label;
  WHERE
    WR1 :
          SIZEOF(using_representations(SELF)) > 0;
END_ENTITY;

ENTITY geometric_representation_item
  SUPERTYPE OF (ONEOF(point, direction, vector, placement,
cartesian_transformation_operator) ANDOR planar_extent)
  SUBTYPE OF(representation_item);
  DERIVE
    dim : dimension_count := dimension_of(SELF);
  WHERE
    WR1 :
          SIZEOF(QUERY (using_rep <* using_representations(SELF)| NOT
('CUTTING_TOOL_SCHEMA.GEOMETRIC_REPRESENTATION_CONTEXT' IN
TYPEOF(using_rep.context_of_items)))) = 0;
END_ENTITY;

ENTITY point
  SUBTYPE OF(geometric_representation_item);
END_ENTITY;

ENTITY cartesian_point
  SUBTYPE OF(point);
```

```
    coordinates : LIST [1:3] OF length_measure;
END_ENTITY;

ENTITY direction
  SUBTYPE OF(geometric_representation_item);
    direction_ratios : LIST [2:3] OF REAL;
  WHERE
    WR1 :
          SIZEOF(QUERY (tmp <* direction_ratios| (tmp <> 0.00000))) > 0;
END_ENTITY;

ENTITY vector
  SUBTYPE OF(geometric_representation_item);
    orientation : direction;
    magnitude   : length_measure;
  WHERE
    WR1 :
          magnitude >= 0.00000;
END_ENTITY;

ENTITY placement
  SUPERTYPE OF (ONEOF(axis2_placement_2d, axis2_placement_3d))
  SUBTYPE OF(geometric_representation_item);
    location : cartesian_point;
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 > 0.00000);
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, 1.00000);
  WHERE
    WR1 :
          scl > 0.00000;
END_ENTITY;

ENTITY functionally_defined_transformation;
    name        : label;
    description : OPTIONAL text;
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 planar_extent
  SUBTYPE OF(geometric_representation_item);
    size_in_x : length_measure;
    size_in_y : length_measure;
END_ENTITY;

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

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

```
ENTITY value_range
  SUBTYPE OF(compound_representation_item);
  WHERE
    WR1 :
          ('CUTTING_TOOL_SCHEMA.' + '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 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 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 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) ANDOR
measure_representation_item);
    value_component : measure_value;
    unit_component  : unit;
  WHERE
    WR1 :
          valid_units(SELF);
END_ENTITY;

ENTITY length_measure_with_unit
  SUBTYPE OF(measure_with_unit);
  WHERE
```

```
    WR1 :
          'CUTTING_TOOL_SCHEMA.LENGTH_UNIT' IN
TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;

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

ENTITY time_measure_with_unit
  SUBTYPE OF(measure_with_unit);
  WHERE
    WR1 :
          'CUTTING_TOOL_SCHEMA.TIME_UNIT' IN
TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;

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

ENTITY thermodynamic_temperature_measure_with_unit
  SUBTYPE OF(measure_with_unit);
  WHERE
    WR1 :
          'CUTTING_TOOL_SCHEMA.THERMODYNAMIC_TEMPERATURE_UNIT' IN
TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;

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

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

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

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

ENTITY solid_angle_measure_with_unit
  SUBTYPE OF(measure_with_unit);
  WHERE
    WR1 :
          'CUTTING_TOOL_SCHEMA.SOLID_ANGLE_UNIT' IN
TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;

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

ENTITY volume_measure_with_unit
  SUBTYPE OF(measure_with_unit);
  WHERE
    WR1 :
          'CUTTING_TOOL_SCHEMA.VOLUME_UNIT' IN
TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;

ENTITY ratio_measure_with_unit
  SUBTYPE OF(measure_with_unit);
  WHERE
    WR1 :
          'CUTTING_TOOL_SCHEMA.RATIO_UNIT' IN
TYPEOF(SELF\measure_with_unit.unit_component);
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) ANDOR
named_unit_variable);
    dimensions : dimensional_exponents;
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 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 conversion_based_unit
  SUBTYPE OF(named_unit);
    name              : label;
    conversion_factor : measure_with_unit;
END_ENTITY;

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

ENTITY expression_conversion_based_unit
  SUBTYPE OF(context_dependent_unit, variable_semantics);
  INVERSE
    associated_variable_environment : environment FOR semantics;
END_ENTITY;

ENTITY variable_semantics
  ABSTRACT SUPERTYPE;
END_ENTITY;

ENTITY derived_unit_variable
  SUBTYPE OF(derived_unit, variable_semantics);
  INVERSE
    associated_variable_environment : environment FOR semantics;
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 <> 1.00000);
    WR2 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY;

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

```
END_ENTITY;

ENTITY environment;
    syntactic_representation : generic_variable;
    semantics                : variable_semantics;
END_ENTITY;

ENTITY generic_variable
  ABSTRACT SUPERTYPE
  SUBTYPE OF(simple_generic_expression);
  INVERSE
    interpretation : environment FOR syntactic_representation;
END_ENTITY;

ENTITY simple_generic_expression
  ABSTRACT SUPERTYPE
  SUBTYPE OF(generic_expression);
END_ENTITY;

ENTITY generic_expression
  ABSTRACT SUPERTYPE OF (ONEOF(simple_generic_expression,
unary_generic_expression, binary_generic_expression,
multiple_arity_generic_expression));
  WHERE
    WR1 :
          is_acyclic(SELF);
END_ENTITY;

ENTITY unary_generic_expression
  ABSTRACT SUPERTYPE
  SUBTYPE OF(generic_expression);
    operand : generic_expression;
END_ENTITY;

ENTITY binary_generic_expression
  ABSTRACT SUPERTYPE
  SUBTYPE OF(generic_expression);
    operands : LIST [2:2] OF generic_expression;
END_ENTITY;

ENTITY multiple_arity_generic_expression
  ABSTRACT SUPERTYPE
  SUBTYPE OF(generic_expression);
    operands : LIST [2:?] OF generic_expression;
END_ENTITY;

ENTITY named_unit_variable
  SUBTYPE OF(named_unit, variable_semantics);
  INVERSE
    associated_variable_environment : environment FOR semantics;
END_ENTITY;

ENTITY length_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 1.00000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
```

```
 (SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
 0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
 0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
 0.00000);
END_ENTITY;

ENTITY mass_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
 (SELF\named_unit.dimensions.mass_exponent = 1.00000)) AND
 (SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
 0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
 0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
 0.00000);
END_ENTITY;

ENTITY time_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
 (SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.time_exponent = 1.00000)) AND
 (SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
 0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
 0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
 0.00000);
END_ENTITY;

ENTITY electric_current_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
 (SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.electric_current_exponent = 1.00000)) AND
 (SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
 0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
 0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
 0.00000);
END_ENTITY;

ENTITY thermodynamic_temperature_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
 (SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
 (SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
```

```
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
1.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
0.00000);
END_ENTITY;

ENTITY amount_of_substance_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
1.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
0.00000);
END_ENTITY;

ENTITY luminous_intensity_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
1.00000);
END_ENTITY;

ENTITY plane_angle_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
0.00000);
END_ENTITY;

ENTITY solid_angle_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
0.00000);
END_ENTITY;

ENTITY area_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 2.00000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
0.00000);
END_ENTITY;

ENTITY volume_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 3.00000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
0.00000);
END_ENTITY;

ENTITY ratio_unit
  SUBTYPE OF(named_unit);
  WHERE
    WR1 :
          ((((((SELF\named_unit.dimensions.length_exponent = 0.00000) AND
(SELF\named_unit.dimensions.mass_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.time_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.electric_current_exponent = 0.00000)) AND
(SELF\named_unit.dimensions.thermodynamic_temperature_exponent =
0.00000)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent =
0.00000)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent =
0.00000);
END_ENTITY;

ENTITY qualified_representation_item
  SUBTYPE OF(representation_item);
    qualifiers : SET [1:?] OF value_qualifier;
  WHERE
    WR1 :
          SIZEOF(QUERY (temp <* qualifiers|
('CUTTING_TOOL_SCHEMA.PRECISION_QUALIFIER' IN TYPEOF(temp)))) < 2;
END_ENTITY;

ENTITY precision_qualifier;
    precision_value : INTEGER;
END_ENTITY;
```

```
ENTITY type_qualifier;
    name : label;
END_ENTITY;

ENTITY value_representation_item
  SUBTYPE OF(representation_item);
    value_component : measure_value;
  WHERE
    WR1 :
          SIZEOF(QUERY (rep <* using_representations(SELF)| NOT
('CUTTING_TOOL_SCHEMA.GLOBAL_UNIT_ASSIGNED_CONTEXT' IN
TYPEOF(rep.context_of_items)))) = 0;
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 geometric_representation_context
  SUBTYPE OF(representation_context);
    coordinate_space_dimension : dimension_count;
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, 'CUTTING_TOOL_SCHEMA.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY;

ENTITY class
  SUBTYPE OF(group);
END_ENTITY;

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

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

ENTITY externally_defined_general_property
```

```
  SUBTYPE OF(general_property, externally_defined_item);
END_ENTITY;

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

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

ENTITY known_source
  SUBTYPE OF(external_source, pre_defined_item);
END_ENTITY;

ENTITY pre_defined_item;
    name : label;
END_ENTITY;

ENTITY characterized_class
  SUBTYPE OF(characterized_object, class);
END_ENTITY;

ENTITY characterized_object;
    name        : label;
    description : OPTIONAL text;
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;
    id                   : identifier;
    name                 : label;
    description          : OPTIONAL text;
    kind                 : document_type;
  INVERSE
    representation_types : SET OF document_representation_type FOR
represented_document;
END_ENTITY;
```

```
ENTITY document_type;
    product_data_type : label;
END_ENTITY;

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

ENTITY feature_definition
  SUBTYPE OF(characterized_object);
  WHERE
    WR1 :
          SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)))) <= 1;
    WR2 :
          (SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)))) = 0) OR (SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY (i <*
pdr.used_representation.items| (i.name = 'orientation') AND
('CUTTING_TOOL_SCHEMA.' + 'PLACEMENT' IN TYPEOF(i)))) <= 1))) = 1);
    WR3 :
          NOT ((SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'BARRING_HOLE'),
('CUTTING_TOOL_SCHEMA.' + 'BEAD'), ('CUTTING_TOOL_SCHEMA.' +
'HOLE_IN_PANEL'), ('CUTTING_TOOL_SCHEMA.' + 'FEATURE_IN_PANEL'),
('CUTTING_TOOL_SCHEMA.' + 'JOGGLE'), ('CUTTING_TOOL_SCHEMA.' +
'LOCATOR') ] * TYPEOF(SELF)) = 1) OR ('CUTTING_TOOL_SCHEMA.' +
'COMPOUND_FEATURE' IN TYPEOF(SELF)) AND (SIZEOF(QUERY (sa <*
get_shape_aspects(SELF)| (sa.name = 'compound feature in panel'))) = 1))
OR (SIZEOF(QUERY (pdr <* get_property_definition_representations(SELF)|
('CUTTING_TOOL_SCHEMA.' + 'DIRECTION_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)))) <= 1) AND ((SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.used_representation)))) =
0) OR (SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.used_representation)) AND
(pdr.used_representation.name = 'punch direction'))) = 1));
    WR4 :
          (SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)))) = 0) OR (SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| (('CUTTING_TOOL_SCHEMA.'
+ 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND ('CUTTING_TOOL_SCHEMA.' +
'GEOMETRIC_REPRESENTATION_CONTEXT' IN
TYPEOF(pdr.used_representation.context_of_items))) AND
(pdr.used_representation.context_of_items\geometric_representation_conte
xt.coordinate_space_dimension = 3))) = 1);
    WR5 :
          NOT ((SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'BOSS'),
('CUTTING_TOOL_SCHEMA.' + 'RIB'), ('CUTTING_TOOL_SCHEMA.' + 'SLOT'),
('CUTTING_TOOL_SCHEMA.' + 'ROUND_HOLE'), ('CUTTING_TOOL_SCHEMA.' +
'POCKET') ] * TYPEOF(SELF)) = 1) OR ('CUTTING_TOOL_SCHEMA.' +
'COMPOUND_FEATURE' IN TYPEOF(SELF)) AND (SIZEOF(QUERY (sa <*
get_shape_aspects(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'COMPOSITE_SHAPE_ASPECT' IN TYPEOF(sa)) AND (sa.name = 'compound feature
in solid'))) = 1)) OR (SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.used_representation)))) =
1) AND (SIZEOF(QUERY (pdr <*
get_property_definition_representations(SELF)| ('CUTTING_TOOL_SCHEMA.' +
'DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (i <* pdr.used_representation.items| (i.name =
'orientation') AND ('CUTTING_TOOL_SCHEMA.' + 'PLACEMENT' IN TYPEOF(i))))
= 1))) = 1);
    WR6 :
          SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'BEAD'),
('CUTTING_TOOL_SCHEMA.' + 'BARRING_HOLE'), ('CUTTING_TOOL_SCHEMA.' +
'BOSS'), ('CUTTING_TOOL_SCHEMA.' + 'COMPOUND_FEATURE'),
('CUTTING_TOOL_SCHEMA.' + 'EXTERNALLY_DEFINED_FEATURE_DEFINITION'),
('CUTTING_TOOL_SCHEMA.' + 'FEATURE_IN_PANEL'), ('CUTTING_TOOL_SCHEMA.' +
'GENERAL_FEATURE'), ('CUTTING_TOOL_SCHEMA.' + 'HOLE_IN_PANEL'),
('CUTTING_TOOL_SCHEMA.' + 'JOGGLE'), ('CUTTING_TOOL_SCHEMA.' +
'LOCATOR'), ('CUTTING_TOOL_SCHEMA.' + 'POCKET'), ('CUTTING_TOOL_SCHEMA.'
+ 'RIB'), ('CUTTING_TOOL_SCHEMA.' + 'REPLICATE_FEATURE'),
('CUTTING_TOOL_SCHEMA.' + 'ROUND_HOLE'), ('CUTTING_TOOL_SCHEMA.' +
'SLOT'), ('CUTTING_TOOL_SCHEMA.' + 'THREAD') ] * TYPEOF(SELF)) <= 1;
END_ENTITY;

ENTITY general_feature
  SUBTYPE OF(feature_definition);
  WHERE
    WR1 :
          (SIZEOF(get_property_definition_representations(SELF)) <= 1)
AND (SIZEOF(QUERY (pdr <* get_property_definition_representations(SELF)|
('CUTTING_TOOL_SCHEMA.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)))) <= 1);

    WR2 :
          SIZEOF(QUERY (sa <* get_shape_aspects(SELF)| (sa.description =
'course of travel occurrence'))) = SIZEOF(QUERY (sa <*
get_shape_aspects(SELF)| (sa.description = 'course of travel
occurrence') AND (SIZEOF(QUERY (sar <* USEDIN(sa, 'CUTTING_TOOL_SCHEMA.'
+ 'SHAPE_ASPECT_RELATIONSHIP.' + 'RELATED_SHAPE_ASPECT')|
('CUTTING_TOOL_SCHEMA.' + 'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))
AND (sar.name = 'course of travel'))) = 1)));
    WR3 :
          SIZEOF(QUERY (sa <* get_shape_aspects(SELF)| (sa.description =
'course of travel occurrence'))) = SIZEOF(QUERY (sa <*
get_shape_aspects(SELF)| (sa.description = 'course of travel
occurrence') AND (SIZEOF(QUERY (sar <* USEDIN(sa, 'CUTTING_TOOL_SCHEMA.'
+ 'SHAPE_ASPECT_RELATIONSHIP.' + 'RELATED_SHAPE_ASPECT')|
('CUTTING_TOOL_SCHEMA.' + 'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))
AND ('CUTTING_TOOL_SCHEMA.' + 'PATH_FEATURE_COMPONENT' IN
TYPEOF(sar.relating_shape_aspect)))) = 1)));
    WR4 :
          SIZEOF(QUERY (sa <* get_shape_aspects(SELF)| (sa.description =
'boundary occurrence'))) = SIZEOF(QUERY (sa <* get_shape_aspects(SELF)|
(sa.description = 'boundary occurrence') AND (SIZEOF(QUERY (sar <*
USEDIN(sa, 'CUTTING_TOOL_SCHEMA.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT')| ('CUTTING_TOOL_SCHEMA.' +
'SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (sar.description =
'profile usage'))) = 1)));
    WR5 :
          (SIZEOF(QUERY (sa <* get_shape_aspects(SELF)|
('CUTTING_TOOL_SCHEMA.' + 'COMPOSITE_SHAPE_ASPECT' IN TYPEOF(sa)))) <=
1) AND (SIZEOF(QUERY (sa <* get_shape_aspects(SELF)|
(('CUTTING_TOOL_SCHEMA.' + 'COMPOSITE_SHAPE_ASPECT' IN TYPEOF(sa)) AND
(sa.name = 'general compound feature')) AND (SIZEOF(QUERY (sar <*
USEDIN(sa, 'CUTTING_TOOL_SCHEMA.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT')| ('CUTTING_TOOL_SCHEMA.' +
'FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
('CUTTING_TOOL_SCHEMA.' + 'INSTANCED_FEATURE' IN
TYPEOF(sar.related_shape_aspect)))) >= 2))) <= 1);
END_ENTITY;

ENTITY class_system
  SUBTYPE OF(group);
END_ENTITY;

ENTITY language
  SUBTYPE OF(group);
  WHERE
    WR1 :
          (SIZEOF(QUERY (ca <* USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'CLASSIFICATION_ASSIGNMENT.' + 'ASSIGNED_CLASS')|
('CUTTING_TOOL_SCHEMA.' + 'LANGUAGE_ASSIGNMENT' IN TYPEOF(ca)))) > 0) OR
(SIZEOF(QUERY (aca <* USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_CLASSIFICATION_ASSIGNMENT.' + 'ASSIGNED_CLASS')|
('CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT' IN
TYPEOF(aca)))) > 0);
END_ENTITY;

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

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

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

ENTITY application_context_element
  SUPERTYPE OF (ONEOF(product_context, product_definition_context));
    name               : label;
    frame_of_reference : application_context;
```

```
END_ENTITY;

ENTITY product_definition_context
  SUBTYPE OF(application_context_element);
    life_cycle_stage : label;
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, 'CUTTING_TOOL_SCHEMA.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    WR2 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
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, 'CUTTING_TOOL_SCHEMA.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY;

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

ENTITY product_definition_formation;
    id          : identifier;
    description : OPTIONAL text;
    of_product  : product;
  UNIQUE
    UR1 : id, of_product;
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_usage
  SUBTYPE OF(product_definition_relationship);
  UNIQUE
    UR1 : id, relating_product_definition, related_product_definition;
  WHERE
    WR1 :
          acyclic_product_definition_relationship(SELF, [
SELF\product_definition_relationship.related_product_definition ],
'CUTTING_TOOL_SCHEMA.PRODUCT_DEFINITION_USAGE');
END_ENTITY;

ENTITY assembly_component_usage
  SUBTYPE OF(product_definition_usage);
    reference_designator : OPTIONAL identifier;
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 material_designation;
    name        : label;
    definitions : SET [1:?] OF characterized_definition;
END_ENTITY;

ENTITY product_definition_shape
  SUBTYPE OF(property_definition);
  UNIQUE
    UR1 : definition;
  WHERE
    WR1 :
          SIZEOF([
'CUTTING_TOOL_SCHEMA.CHARACTERIZED_PRODUCT_DEFINITION',
'CUTTING_TOOL_SCHEMA.CHARACTERIZED_OBJECT' ] *
TYPEOF(SELF\property_definition.definition)) > 0;
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, 'CUTTING_TOOL_SCHEMA.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
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, 'CUTTING_TOOL_SCHEMA.' +
'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY;

ENTITY plus_minus_tolerance;
    range                : tolerance_method_definition;
    toleranced_dimension : dimensional_characteristic;
  UNIQUE
    UR1 : toleranced_dimension;
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 limits_and_fits;
    form_variance : label;
    zone_variance : label;
    grade         : label;
    source        : text;
END_ENTITY;

ENTITY dimensional_size;
    applies_to : shape_aspect;
    name       : label;
  WHERE
    WR1 :
          applies_to.product_definitional = TRUE;
END_ENTITY;

ENTITY applied_person_and_organization_assignment
  SUBTYPE OF(person_and_organization_assignment);
    items : SET [1:?] OF person_and_organization_item;
  WHERE
    WR1 :
          NOT (SELF.role.name = 'signing for contract') OR
item_correlation(SELF.items, [ 'APPLIED_ORGANIZATION_ASSIGNMENT' ]);
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;
    the_person       : person;
    the_organization : organization;
  DERIVE
    name             : label := get_name_value(SELF);
    description      : text := get_description_value(SELF);
  WHERE
    WR1 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
    WR2 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
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 organization;
    id          : OPTIONAL identifier;
    name        : label;
    description : OPTIONAL text;
END_ENTITY;

ENTITY person_and_organization_role;
    name        : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    WR1 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
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_related_product_category
  SUBTYPE OF(product_category);
    products : SET [1:?] OF product;
END_ENTITY;

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

ENTITY effectivity
  SUPERTYPE OF (ONEOF(dated_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, 'CUTTING_TOOL_SCHEMA.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
    WR2 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'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 date;
    year_component : year_number;
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 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 coordinated_universal_time_offset;
    hour_offset          : INTEGER;
    minute_offset        : OPTIONAL INTEGER;
    sense                : ahead_or_behind;
    DERIVE
      actual_minute_offset : INTEGER := NVL(minute_offset, 0);
    WHERE
      WR1 :
            (0 <= hour_offset) AND (hour_offset < 24);
      WR2 :
            (0 <= actual_minute_offset) AND (actual_minute_offset <= 59);
      WR3 :
            NOT (((hour_offset <> 0) OR (actual_minute_offset <> 0)) AND
 (sense = exact));
  END_ENTITY;

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

  ENTITY time_interval_based_effectivity
    SUBTYPE OF(effectivity);
      effectivity_period : time_interval;
  END_ENTITY;

  ENTITY time_interval;
      id          : identifier;
      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 effectivity_relationship;
      name                  : label;
      description           : OPTIONAL text;
      related_effectivity   : effectivity;
      relating_effectivity  : effectivity;
  END_ENTITY;

  ENTITY general_property_relationship;
      name               : label;
      description        : OPTIONAL text;
      relating_property  : general_property;
      related_property   : general_property;
  END_ENTITY;

  ENTITY representation_relationship;
      name        : label;
      description : OPTIONAL text;
      rep_1       : representation;
      rep_2       : representation;
  END_ENTITY;

  ENTITY shape_representation_relationship
    SUBTYPE OF(representation_relationship);
    WHERE
      WR1 :
            'CUTTING_TOOL_SCHEMA.SHAPE_REPRESENTATION' IN
  TYPEOF(SELF\representation_relationship.rep_1) +
  TYPEOF(SELF\representation_relationship.rep_2);
  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 item_defined_transformation;
      name             : label;
      description      : OPTIONAL text;
      transform_item_1 : representation_item;
      transform_item_2 : representation_item;
  END_ENTITY;

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

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

  ENTITY multi_language_attribute_assignment
    SUBTYPE OF(attribute_value_assignment);
      items    : SET [1:?] OF multi_language_attribute_item;
    DERIVE
      language : label := get_multi_language(SELF);
    WHERE
      WR1 :
            SELF\attribute_value_assignment.role.name = 'alternate
  language';
      WR2 :
            (SIZEOF(USEDIN(SELF,
  'CUTTING_TOOL_SCHEMA.ATTRIBUTE_LANGUAGE_ASSIGNMENT.ITEMS')) = 1) AND
  (SIZEOF(QUERY (ala <* USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
  'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name =
  'attribute_value'))) = 1);

  END_ENTITY;
```

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

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

ENTITY applied_organization_assignment
  SUBTYPE OF(organization_assignment);
    items : SET [1:?] OF organization_item;
  WHERE
    WR1 :
          NOT (SELF.role.name = 'organization in contract') OR
 item_correlation(SELF.items, [ 'CONTRACT' ]);
    WR2 :
          NOT (SELF.role.name = 'signing for contract') OR
 item_correlation(SELF.items, [ 'APPLIED_ORGANIZATION_ASSIGNMENT' ]);
    WR3 :
          NOT (SELF.role.name = 'validity context') OR
 item_correlation(SELF.items, [ 'ACTION_PROPERTY', 'RESOURCE_PROPERTY',
 'PROPERTY_DEFINITION' ]);
    WR4 :
          NOT (SELF.role.name = 'alias scope') OR
 item_correlation(SELF.items, [ 'APPLIED_IDENTIFICATION_ASSIGNMENT' ]);
END_ENTITY;

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

ENTITY applied_identification_assignment
  SUBTYPE OF(identification_assignment);
    items : SET [1:?] OF identification_item;
  WHERE
    WR1 :
          NOT (SELF.role.name = 'version') OR
 item_correlation(SELF.items, [ 'ACTION', 'ACTION_DIRECTIVE',
 'SHAPE_REPRESENTATION', 'EFFECTIVITY', 'PRODUCT_CONCEPT',
 'CONFIGURATION_ITEM', 'PRODUCT_DEFINITION', 'PRODUCT_CONCEPT_FEATURE',
 'DOCUMENT_FILE', 'CLASS', 'APPLIED_IDENTIFICATION_ASSIGNMENT',
 'DRAUGHTING_MODEL',
 'MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION',
 'PRESENTATION_AREA' ]);
    WR2 :
          NOT (SELF.role.name = 'lot context') OR
 item_correlation(SELF.items, [ 'PRODUCT_DEFINITION' ]);
    WR3 :
          NOT (SELF.role.name = 'property change id context') OR
item_correlation(SELF.items, [ 'PROPERTY_DEFINITION_RELATIONSHIP',
'SHAPE_ASPECT_RELATIONSHIP', 'ACTION_PROPERTY' ]);
    WR4 :
          NOT (SELF.role.name = 'size id') OR
item_correlation(SELF.items, [ 'DIMENSIONAL_SIZE' ]);
    WR5 :
          NOT (SELF.role.name = 'model change id') OR
item_correlation(SELF.items, [ 'PROPERTY_DEFINITION', 'ACTION_PROPERTY'
]);
    WR6 :
          NOT (SELF.role.name = 'alias') OR item_correlation(SELF.items,
[ 'APPLICATION_CONTEXT', 'APPROVAL_STATUS', 'ASSEMBLY_COMPONENT_USAGE',
'CLASS', 'CLASS_SYSTEM', 'DOCUMENT_TYPE', 'DRAUGHTING_MODEL',
'GENERAL_PROPERTY',
'MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION',
'ORGANIZATION', 'PRODUCT', 'PRODUCT_CONCEPT', 'PRODUCT_CONCEPT_FEATURE',
'PRODUCT_CONCEPT_FEATURE_CATEGORY', 'PRODUCT_DEFINITION',
'PRODUCT_DEFINITION_FORMATION', 'PROPERTY_DEFINITION',
'SECURITY_CLASSIFICATION_LEVEL', 'SHAPE_REPRESENTATION' ]);
END_ENTITY;

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

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

ENTITY applied_external_identification_assignment
  SUBTYPE OF(external_identification_assignment);
    items : SET [1:?] OF external_identification_item;
  WHERE
    WR1 :
          NOT (SELF.role.name = 'alternative document id and location')
OR item_correlation(SELF.items, [ 'DOCUMENT_FILE' ]);
    WR2 :
          NOT (SELF.role.name = 'version') OR
item_correlation(SELF.items, [ 'EXTERNALLY_DEFINED_CLASS',
'EXTERNALLY_DEFINED_GENERAL_PROPERTY' ]);
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 :
          'CUTTING_TOOL_SCHEMA.PRODUCT_DEFINITION_RELATIONSHIP' IN
TYPEOF(SELF.represented_product_relation.definition);
    WR2 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    WR3 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'NAME_ATTRIBUTE.NAMED_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, 'CUTTING_TOOL_SCHEMA.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    WR2 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY;

ENTITY shape_definition_representation
  SUBTYPE OF(property_definition_representation);
  WHERE
    WR1 :
          ('CUTTING_TOOL_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(SELF.definition)) OR ('CUTTING_TOOL_SCHEMA.SHAPE_DEFINITION' IN
TYPEOF(SELF.definition.definition));
    WR2 :
          'CUTTING_TOOL_SCHEMA.SHAPE_REPRESENTATION' IN
TYPEOF(SELF.used_representation);
END_ENTITY;

ENTITY applied_location_assignment
  SUBTYPE OF(location_assignment);
    items : SET [1:?] OF location_assignment_item;
END_ENTITY;

ENTITY location_assignment
  ABSTRACT SUPERTYPE;
    id                : identifier;
    name              : label;
    description       : OPTIONAL text;
    assigned_location : location;
    role              : location_role;
END_ENTITY;

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

ENTITY location_role;
    id          : identifier;
    name        : label;
    description : OPTIONAL text;
```

```
END_ENTITY;

ENTITY applied_state_type_assignment;
    assigned_state_type : state_type;
    item_set            : SET [1:?] OF applied_state_type_of_item;
    role                : state_type_role;
END_ENTITY;

ENTITY state_type;
    name        : STRING;
    description : OPTIONAL STRING;
END_ENTITY;

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

ENTITY person_and_organization_address
  SUBTYPE OF(organizational_address, personal_address);
  WHERE
    WR1 :
          SIZEOF(SELF\personal_address.people) = 1;
    WR2 :
          SIZEOF(SELF\organizational_address.organizations) = 1;
    WR3 :
          SIZEOF(QUERY (pao <* USEDIN(SELF.people[1],
 'CUTTING_TOOL_SCHEMA.PERSON_AND_ORGANIZATION.THE_PERSON')|
 (pao.the_organization :=:
 SELF\organizational_address.organizations[1]))) = 1;
END_ENTITY;

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

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

ENTITY applied_effectivity_assignment
  SUBTYPE OF(effectivity_assignment);
    items : SET [1:?] OF effectivity_item;
  WHERE
    WR1 :
          SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'LOT_EFFECTIVITY'),
 ('CUTTING_TOOL_SCHEMA.' + 'SERIAL_NUMBERED_EFFECTIVITY'),
 ('CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION_EFFECTIVITY') ] *
 TYPEOF(SELF.assigned_effectivity)) = 0;
END_ENTITY;

ENTITY applied_ineffectivity_assignment
  SUBTYPE OF(effectivity_assignment);
    items : SET [1:?] OF effectivity_item;
  WHERE
    WR1 :
          SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'LOT_EFFECTIVITY'),
 ('CUTTING_TOOL_SCHEMA.' + 'SERIAL_NUMBERED_EFFECTIVITY'),
 ('CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION_EFFECTIVITY') ] *
 TYPEOF(SELF.assigned_effectivity)) = 0;
END_ENTITY;

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

ENTITY applied_document_reference
  SUBTYPE OF(document_reference);
    items : SET [1:?] OF document_reference_item;
  WHERE
    WR1 :
          NOT (SELF.role.name = 'general tolerance definition') OR
 item_correlation(SELF.items, [ 'REPRESENTATION' ]);
    WR2 :
          (SELF.role.name = 'general tolerance definition') OR
item_correlation(SELF.items, [ 'ACTION', 'ACTION_DIRECTIVE',
'ACTION_METHOD', 'ACTION_RELATIONSHIP', 'APPLIED_ACTION_ASSIGNMENT',
'APPROVAL', 'CERTIFICATION', 'CLASS', 'CLASS_SYSTEM',
'CONFIGURATION_DESIGN', 'CONFIGURATION_ITEM', 'CONTRACT',
'FEATURE_DEFINITION', 'GENERAL_PROPERTY', 'MATERIAL_DESIGNATION',
'ORGANIZATION', 'ORGANIZATIONAL_PROJECT', 'PERSON', 'PRODUCT_CONCEPT',
'PRODUCT_CONCEPT_FEATURE', 'PRODUCT_CONCEPT_FEATURE_CATEGORY',
'PRODUCT_DEFINITION', 'PRODUCT_DEFINITION_FORMATION',
'PRODUCT_DEFINITION_FORMATION_RELATIONSHIP',
'PRODUCT_DEFINITION_RELATIONSHIP', 'PRODUCT_DEFINITION_SUBSTITUTE',
'PRODUCT_RELATED_PRODUCT_CATEGORY', 'PROPERTY_DEFINITION',
'REPRESENTATION', 'RESOURCE_REQUIREMENT_TYPE',
'SECURITY_CLASSIFICATION', 'SHAPE_ASPECT', 'SHAPE_ASPECT_RELATIONSHIP',
'VERSIONED_ACTION_REQUEST' ]);
END_ENTITY;

ENTITY document_reference
  ABSTRACT SUPERTYPE;
    assigned_document : document;
    source            : label;
  DERIVE
    role              : object_role := get_role(SELF);
  WHERE
    WR1 :
          SIZEOF(USEDIN(SELF, 'CUTTING_TOOL_SCHEMA.' +
'ROLE_ASSOCIATION.' + 'ITEM_WITH_ROLE')) <= 1;
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);
    items    : SET [1:?] OF attribute_language_item;
  DERIVE
    language : label :=
SELF\attribute_classification_assignment.assigned_class.name;
  WHERE
    WR1 :
          SELF\attribute_classification_assignment.role.name IN [
'primary', 'translated' ];
    WR2 :
          'CUTTING_TOOL_SCHEMA.' + 'LANGUAGE' IN
TYPEOF(SELF\attribute_classification_assignment.assigned_class);
END_ENTITY;

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

ENTITY dimensional_characteristic_representation;
    dimension      : dimensional_characteristic;
    representation : shape_dimension_representation;
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.name = 'equivalence';
    WR2 :
          NOT ('CUTTING_TOOL_SCHEMA.' + 'PRODUCT' IN
TYPEOF(SELF.related_product)) OR
(SELF.relating_document.kind.product_data_type = 'configuration
controlled document') AND (SIZEOF(QUERY (prpc <*
USEDIN(SELF.related_product, 'CUTTING_TOOL_SCHEMA.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS')| (prpc.name =
'document'))) = 1);
    WR3 :
          NOT ('CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION_FORMATION' IN
TYPEOF(SELF.related_product)) OR
(SELF.relating_document.kind.product_data_type = 'configuration
controlled document version') AND (SIZEOF(QUERY (prpc <*
USEDIN(SELF.related_product\product_definition_formation.of_product,
'CUTTING_TOOL_SCHEMA.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')|
(prpc.name = 'document'))) = 1);
    WR4 :
          NOT ('CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION' IN
TYPEOF(SELF.related_product)) OR
(SELF.relating_document.kind.product_data_type = 'configuration
controlled document definition') AND (SIZEOF(QUERY (prpc <*
USEDIN(SELF.related_product\product_definition.formation.of_product,
'CUTTING_TOOL_SCHEMA.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')|
(prpc.name = 'document'))) = 1);
END_ENTITY;

ENTITY externally_defined_item_relationship;
    name          : label;
    description   : OPTIONAL text;
    relating_item : externally_defined_item;
    related_item  : externally_defined_item;
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, 'CUTTING_TOOL_SCHEMA.' +
'GENERAL_PROPERTY_ASSOCIATION.' + 'DERIVED_DEFINITION')) = 1;
    WR2 :
          derived_definition.name = base_definition.name;
END_ENTITY;

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

ENTITY id_attribute;
    attribute_value : identifier;
    identified_item : id_attribute_select;
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|
('CUTTING_TOOL_SCHEMA.PRECISION_QUALIFIER' IN TYPEOF(temp)))) < 2;
END_ENTITY;

ENTITY name_attribute;
    attribute_value : label;
    named_item      : name_attribute_select;
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_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_occurrence_relationship;
    name             : label;
    description      : OPTIONAL text;
    occurrence       : product_definition;
    occurrence_usage : assembly_component_usage;
  WHERE
    WR1 :
          occurrence_usage.relating_product_definition :<>: occurrence;
    WR2 :
          occurrence_usage.related_product_definition :<>: occurrence;
    WR3 :
          occurrence.formation :=:
occurrence_usage.related_product_definition.formation;
END_ENTITY;

ENTITY role_association;
    role           : object_role;
    item_with_role : role_select;
END_ENTITY;

RULE coating_requires_product_definition FOR
  (product_definition,
   product_definition_formation,
   product_related_product_category);
  WHERE
    WR1 :
          SIZEOF(QUERY (pdf <* product_definition_formation|
(SIZEOF(QUERY (prpc <* USEDIN(pdf.of_product, 'CUTTING_TOOL_SCHEMA.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS')| (prpc.name =
'coating'))) > 0) AND (SIZEOF(QUERY (pd <* USEDIN(pdf,
'CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION.' + 'FORMATION')|
(pd.frame_of_reference.name = 'material definition'))) <> 1))) = 0;
END_RULE;

RULE compatible_dimension FOR
  (cartesian_point,
   direction,
   representation_context,
   geometric_representation_context);
  WHERE
    WR1 :
          SIZEOF(QUERY (x <* cartesian_point| (SIZEOF(QUERY (y <*
geometric_representation_context| item_in_context(x, y) AND
(HIINDEX(x.coordinates) <> y.coordinate_space_dimension))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (x <* direction| (SIZEOF(QUERY (y <*
geometric_representation_context| item_in_context(x, y) AND
(HIINDEX(x.direction_ratios) <> y.coordinate_space_dimension))) > 0))) =
0;
END_RULE;

RULE dependent_instantiable_attribute_value_role FOR
  (attribute_value_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (a <* attribute_value_role| NOT (SIZEOF(USEDIN(a,
'')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_classification_role FOR
  (classification_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (c <* classification_role| NOT (SIZEOF(USEDIN(c,
'')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_date FOR
  (date);
  WHERE
    WR1 :
          SIZEOF(QUERY (d <* date| NOT (SIZEOF(USEDIN(d, '')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_date_and_time FOR
  (date_and_time);
  WHERE
    WR1 :
          SIZEOF(QUERY (d <* date_and_time| NOT (SIZEOF(USEDIN(d, '')) >
0))) = 0;
END_RULE;

RULE dependent_instantiable_derived_unit FOR
  (derived_unit);
  WHERE
    WR1 :
          SIZEOF(QUERY (d <* derived_unit| NOT (SIZEOF(USEDIN(d, '')) >
0))) = 0;
END_RULE;

RULE dependent_instantiable_identification_role FOR
  (identification_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (i <* identification_role| NOT (SIZEOF(USEDIN(i,
'')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_measure_with_unit FOR
  (measure_with_unit);
  WHERE
    WR1 :
          SIZEOF(QUERY (m <* measure_with_unit| NOT (SIZEOF(USEDIN(m,
'')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_named_unit FOR
  (named_unit);
  WHERE
    WR1 :
          SIZEOF(QUERY (n <* named_unit| NOT (SIZEOF(USEDIN(n, '')) >
0))) = 0;
END_RULE;

RULE dependent_instantiable_object_role FOR
  (object_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (o <* object_role| NOT (SIZEOF(USEDIN(o, '')) >
0))) = 0;
END_RULE;
```

```
RULE dependent_instantiable_organization_role FOR
  (organization_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (o <* organization_role| NOT (SIZEOF(USEDIN(o,
'')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_person_and_organization_role FOR
  (person_and_organization_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (p <* person_and_organization_role| NOT
(SIZEOF(USEDIN(p, '')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_precision_qualifier FOR
  (precision_qualifier);
  WHERE
    WR1 :
          SIZEOF(QUERY (p <* precision_qualifier| NOT (SIZEOF(USEDIN(p,
'')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_product_definition_context_role FOR
  (product_definition_context_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (p <* product_definition_context_role| NOT
(SIZEOF(USEDIN(p, '')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_tolerance_value FOR
  (tolerance_value);
  WHERE
    WR1 :
          SIZEOF(QUERY (t <* tolerance_value| NOT (SIZEOF(USEDIN(t, ''))
> 0))) = 0;
END_RULE;

RULE dependent_instantiable_type_qualifier FOR
  (type_qualifier);
  WHERE
    WR1 :
          SIZEOF(QUERY (t <* type_qualifier| NOT (SIZEOF(USEDIN(t, '')) >
0))) = 0;
END_RULE;

RULE externally_defined_class_with_known_source_requirement FOR
  (externally_defined_class);
  WHERE
    WR1 :
          SIZEOF(QUERY (edc <* externally_defined_class|
('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE' IN TYPEOF(edc.source)) AND
(SIZEOF(QUERY (aoa <* USEDIN(edc,
'CUTTING_TOOL_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
(aoa.role.name = 'class supplier'))) = 0))) = 0;
```

```
END_RULE;

RULE grade_requires_product_definition FOR
  (product_definition,
   product_definition_formation,
   product_related_product_category);
  WHERE
    WR1 :
          SIZEOF(QUERY (pdf <* product_definition_formation|
 (SIZEOF(QUERY (prpc <* USEDIN(pdf.of_product, 'CUTTING_TOOL_SCHEMA.' +
 'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS')| (prpc.name =
 'grade'))) > 0) AND (SIZEOF(QUERY (pd <* USEDIN(pdf,
 'CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION.' + 'FORMATION')|
 (pd.frame_of_reference.name = 'material definition'))) <> 1))) = 0;
END_RULE;

RULE person_requires_person_and_organization FOR
  (person);
  WHERE
    WR1 :
          SIZEOF(QUERY (p <* person| (SIZEOF(USEDIN(p,
 'CUTTING_TOOL_SCHEMA.PERSON_AND_ORGANIZATION.THE_PERSON')) = 0))) = 0;
END_RULE;

RULE plib_class_reference_requires_version FOR
  (externally_defined_class);
  WHERE
    WR1 :
          SIZEOF(QUERY (edc <* externally_defined_class|
 ('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE' IN TYPEOF(edc.source)) AND
 (SIZEOF(QUERY (aei <* USEDIN(edc,
 'CUTTING_TOOL_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS')|
 (aei.role.name = 'version'))) <> 1))) = 0;
    WR2 :
          SIZEOF(QUERY (edc <* externally_defined_class|
 ('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE' IN TYPEOF(edc.source)) AND
 (SIZEOF(QUERY (aei <* USEDIN(edc,
 'CUTTING_TOOL_SCHEMA.APPLIED_IDENTIFICATION_ASSIGNMENT.ITEMS')|
 (aei.role.name = 'version'))) > 0))) = 0;
END_RULE;

RULE plib_property_reference_requires_name_scope FOR
  (externally_defined_general_property);
  WHERE
    WR1 :
          SIZEOF(QUERY (edgp <* externally_defined_general_property|
 ('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE' IN TYPEOF(edgp.source)) AND
 (SIZEOF(QUERY (edir <* USEDIN(edgp, 'CUTTING_TOOL_SCHEMA.' +
 'EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.' + 'RELATING_ITEM')| ((edir.name
 = 'name scope') AND ('CUTTING_TOOL_SCHEMA.' + 'EXTERNALLY_DEFINED_CLASS'
 IN TYPEOF(edir.related_item))) AND ('CUTTING_TOOL_SCHEMA.' +
 'KNOWN_SOURCE' IN TYPEOF(edir.related_item.source)))) <> 1))) = 0;
END_RULE;

RULE plib_property_reference_requires_version FOR
  (externally_defined_general_property);
  WHERE
    WR1 :
          SIZEOF(QUERY (edgp <* externally_defined_general_property|
('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE' IN TYPEOF(edgp.source)) AND
(SIZEOF(QUERY (edir <* USEDIN(edgp, 'CUTTING_TOOL_SCHEMA.' +
'APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.' + 'ITEMS')|
(edir.role.name = 'version'))) <> 1))) = 0;
END_RULE;

RULE product_requires_category FOR
  (product,
   product_related_product_category);
  WHERE
    WR1 :
          SIZEOF(QUERY (p <* product| (SIZEOF(USEDIN(p,
'CUTTING_TOOL_SCHEMA.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')) = 0)))
= 0;
END_RULE;

RULE product_requires_id_owner FOR
  (product);
  WHERE
    WR1 :
          SIZEOF(QUERY (prod <* product| (SIZEOF(QUERY (prpc <*
USEDIN(prod,
'CUTTING_TOOL_SCHEMA.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')|
(prpc.name IN [ 'part', 'tool', 'raw material' ]))) > 0) AND
(SIZEOF(QUERY (apoa <* USEDIN(prod,
'CUTTING_TOOL_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS')|
(apoa\person_and_organization_assignment.role.name <> 'id owner'))) +
(SIZEOF(QUERY (oa <* USEDIN(prod,
'CUTTING_TOOL_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
(oa\organization_assignment.role.name = 'id owner'))) <> 1)))) = 0;
END_RULE;

RULE product_requires_version FOR
  (product);
  WHERE
    WR1 :
          SIZEOF(QUERY (prod <* product| (SIZEOF(USEDIN(prod,
'CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION_FORMATION.' +
'OF_PRODUCT')) = 0))) = 0;
END_RULE;

RULE restrict_applied_classification_assignment_role FOR
  (applied_classification_assignment);
  WHERE
    WR1 :
          SIZEOF(QUERY (aca <* applied_classification_assignment|
('CUTTING_TOOL_SCHEMA.' + 'CLASS' IN TYPEOF(aca.assigned_class)) AND NOT
(aca.role.name IN [ 'definitional', 'non-definitional', '' ]))) = 0;
    WR2 :
          SIZEOF(QUERY (aca <* applied_classification_assignment|
(aca.role.name IN [ 'definitional', 'non-definitional', '' ]) AND NOT
('CUTTING_TOOL_SCHEMA.' + 'CLASS' IN TYPEOF(aca.assigned_class)))) = 0;
    WR3 :
          SIZEOF(QUERY (aca <* applied_classification_assignment|
('CUTTING_TOOL_SCHEMA.' + 'CLASS_SYSTEM' IN TYPEOF(aca.assigned_class))
AND (aca.role.name <> 'class system membership'))) = 0;
    WR4 :
          SIZEOF(QUERY (aca <* applied_classification_assignment|
 (aca.role.name = 'class system membership') AND NOT
 ('CUTTING_TOOL_SCHEMA.' + 'CLASS_SYSTEM' IN
 TYPEOF(aca.assigned_class)))) = 0;
END_RULE;

RULE restrict_class_system_assignment_for_class FOR
   (applied_classification_assignment,
    class);
  WHERE
    WR1 :
          SIZEOF(QUERY (c <* class| (SIZEOF(QUERY (aca <* USEDIN(c,
 'CUTTING_TOOL_SCHEMA.' + 'CLASSIFICATION_ASSIGNMENT.' +
 'ASSIGNED_CLASS')| (aca.role.name = 'class system membership') AND
 ('CUTTING_TOOL_SCHEMA.' + 'APPLIED_CLASSIFICATION_ASSIGNMENT' IN
 TYPEOF(aca)))) > 1))) = 0;
END_RULE;

RULE restrict_class_system_assignment_for_planar_extent FOR
   (applied_classification_assignment,
    planar_extent);
  WHERE
    WR1 :
          SIZEOF(QUERY (pe <* planar_extent| (pe\representation_item.name
 = 'size format') AND (SIZEOF(QUERY (aca <* USEDIN(pe,
 'CUTTING_TOOL_SCHEMA.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')|
 (aca\classification_assignment.role.name = 'class system membership')))
 > 1))) = 0;
END_RULE;

RULE restrict_externally_defined_item_relationship FOR
   (externally_defined_item_relationship);
  WHERE
    WR1 :
          SIZEOF(QUERY (edir <* externally_defined_item_relationship|
 (edir.name = 'name scope') AND (((NOT ('CUTTING_TOOL_SCHEMA.' +
 'EXTERNALLY_DEFINED_GENERAL_PROPERTY' IN TYPEOF(edir.relating_item)) OR
 NOT ('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE' IN
 TYPEOF(edir.relating_item.source))) OR NOT ('CUTTING_TOOL_SCHEMA.' +
 'EXTERNALLY_DEFINED_CLASS' IN TYPEOF(edir.related_item))) OR NOT
 ('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE' IN
 TYPEOF(edir.related_item.source))))) = 0;
END_RULE;

RULE restrict_group_relationship_for_general_classification_hierarchy FOR
   (group_relationship,
    class);
  WHERE
    WR1 :
          SIZEOF(QUERY (gr <* group_relationship| (gr.name = 'class
 hierarchy') AND (NOT ('CUTTING_TOOL_SCHEMA.' + 'CLASS' IN
 TYPEOF(gr.related_group)) OR NOT ('CUTTING_TOOL_SCHEMA.' + 'CLASS' IN
 TYPEOF(gr.relating_group))))) = 0;
END_RULE;

RULE restrict_multi_language_for_application_context FOR
   (application_context);
  WHERE
```

```
    WR1 :
          SIZEOF(QUERY (ent <* application_context| (SIZEOF(QUERY (mlaa
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* application_context| (SIZEOF(QUERY (ala <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala.attribute_name <> 'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* application_context| (SIZEOF(QUERY (mlaa1
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_descriptive_representation_item FOR
  (descriptive_representation_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* descriptive_representation_item|
(SIZEOF(QUERY (mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* descriptive_representation_item|
(SIZEOF(QUERY (ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* descriptive_representation_item|
(SIZEOF(QUERY (mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_effectivity FOR
  (effectivity);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* effectivity| (SIZEOF(QUERY (mlaa <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
```

```
    WR2 :
          SIZEOF(QUERY (ent <* effectivity| (SIZEOF(QUERY (ala <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala.attribute_name <> 'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* effectivity| (SIZEOF(QUERY (mlaa1 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_effectivity_relationship FOR
  (effectivity_relationship);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* effectivity_relationship| (SIZEOF(QUERY
(mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* effectivity_relationship| (SIZEOF(QUERY
(ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* effectivity_relationship| (SIZEOF(QUERY
(mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_external_source FOR
  (external_source);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* external_source| (SIZEOF(QUERY (mlaa <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* external_source| (SIZEOF(QUERY (ala <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala.attribute_name <> 'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* external_source| (SIZEOF(QUERY (mlaa1 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_general_feature FOR
  (general_feature);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* general_feature| (SIZEOF(QUERY (mlaa <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* general_feature| (SIZEOF(QUERY (ala <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala.attribute_name <> 'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* general_feature| (SIZEOF(QUERY (mlaa1 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_general_property FOR
  (general_property);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* general_property| (SIZEOF(QUERY (mlaa <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* general_property| (SIZEOF(QUERY (ala <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala.attribute_name <> 'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* general_property| (SIZEOF(QUERY (mlaa1 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
 SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
 'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
 = ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_general_property_relationship FOR
  (general_property_relationship);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* general_property_relationship|
 (SIZEOF(QUERY (mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
 <> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* general_property_relationship|
 (SIZEOF(QUERY (ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
 'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* general_property_relationship|
 (SIZEOF(QUERY (mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
 = mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
 SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
 'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
 = ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_group FOR
  (group);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* group| (SIZEOF(QUERY (mlaa <* USEDIN(ent,
 'CUTTING_TOOL_SCHEMA.' + 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' +
 'ITEMS')| (mlaa.attribute_name <> 'description') AND
 (mlaa.attribute_name <> 'name'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* group| (SIZEOF(QUERY (ala <* USEDIN(ent,
 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')|
 (ala.attribute_name <> 'description') AND (ala.attribute_name <>
 'name'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* group| (SIZEOF(QUERY (mlaa1 <* USEDIN(ent,
 'CUTTING_TOOL_SCHEMA.' + 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' +
 'ITEMS')| (SIZEOF(QUERY (mlaa2 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
 = mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
 SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
 'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
 = ala2.language))) > 1))) > 0))) = 0;
```

```
END_RULE;

RULE restrict_multi_language_for_identification_role FOR
  (identification_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* identification_role| (SIZEOF(QUERY (mlaa
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* identification_role| (SIZEOF(QUERY (ala <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala.attribute_name <> 'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* identification_role| (SIZEOF(QUERY (mlaa1
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_organization_role FOR
  (organization_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* organization_role| (SIZEOF(QUERY (mlaa <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* organization_role| (SIZEOF(QUERY (ala <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala.attribute_name <> 'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* organization_role| (SIZEOF(QUERY (mlaa1 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_person_and_organization_role FOR
  (person_and_organization_role);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* person_and_organization_role|
(SIZEOF(QUERY (mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* person_and_organization_role|
(SIZEOF(QUERY (ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* person_and_organization_role|
(SIZEOF(QUERY (mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_product FOR
  (product);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* product| (SIZEOF(QUERY (mlaa <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description') AND (mlaa.attribute_name <> 'name'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* product| (SIZEOF(QUERY (ala <* USEDIN(ent,
'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')|
(ala.attribute_name <> 'description') AND (ala.attribute_name <>
'name'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* product| (SIZEOF(QUERY (mlaa1 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_product_definition FOR
  (product_definition);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* product_definition| (SIZEOF(QUERY (mlaa <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description') AND (mlaa.attribute_name <> 'name'))) > 0))) = 0;
```

```
    WR2 :
          SIZEOF(QUERY (ent <* product_definition| (SIZEOF(QUERY (ala <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala.attribute_name <> 'description') AND (ala.attribute_name
<> 'name'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* product_definition| (SIZEOF(QUERY (mlaa1
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_product_definition_formation FOR
  (product_definition_formation);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* product_definition_formation|
(SIZEOF(QUERY (mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* product_definition_formation|
(SIZEOF(QUERY (ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* product_definition_formation|
(SIZEOF(QUERY (mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE
restrict_multi_language_for_product_definition_formation_relationship
FOR
  (product_definition_formation_relationship);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* product_definition_formation_relationship|
(SIZEOF(QUERY (mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* product_definition_formation_relationship|
(SIZEOF(QUERY (ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* product_definition_formation_relationship|
(SIZEOF(QUERY (mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_product_definition_relationship FOR
  (product_definition_relationship);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* product_definition_relationship|
(SIZEOF(QUERY (mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* product_definition_relationship|
(SIZEOF(QUERY (ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* product_definition_relationship|
(SIZEOF(QUERY (mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_product_related_product_category FOR
  (product_related_product_category);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* product_related_product_category|
(SIZEOF(QUERY (mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* product_related_product_category|
(SIZEOF(QUERY (ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
'description'))) > 0))) = 0;
```

```
    WR3 :
          SIZEOF(QUERY (ent <* product_related_product_category|
 (SIZEOF(QUERY (mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
 = mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
 SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
 'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
 = ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_property_definition FOR
  (property_definition);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* property_definition| (SIZEOF(QUERY (mlaa
 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
 <> 'description') AND (mlaa.attribute_name <> 'name'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* property_definition| (SIZEOF(QUERY (ala <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
 'ITEMS')| (ala.attribute_name <> 'description') AND (ala.attribute_name
 <> 'name'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* property_definition| (SIZEOF(QUERY (mlaa1
 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
 = mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
 SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
 'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
 = ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_representation FOR
  (representation);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* representation| (SIZEOF(QUERY (mlaa <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
 <> 'description') AND (mlaa.attribute_name <> 'name'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* representation| (SIZEOF(QUERY (ala <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
 'ITEMS')| (ala.attribute_name <> 'description') AND (ala.attribute_name
 <> 'name'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* representation| (SIZEOF(QUERY (mlaa1 <*
 USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
 'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_multi_language_for_representation_relationship FOR
   (representation_relationship);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* representation_relationship| (SIZEOF(QUERY
(mlaa <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa.attribute_name
<> 'description'))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (ent <* representation_relationship| (SIZEOF(QUERY
(ala <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (ala.attribute_name <>
'description'))) > 0))) = 0;
    WR3 :
          SIZEOF(QUERY (ent <* representation_relationship| (SIZEOF(QUERY
(mlaa1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (mlaa2
<* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.' + 'ITEMS')| (mlaa1.attribute_name
= mlaa2.attribute_name) AND (mlaa1.language = mlaa2.language))) > 1))) +
SIZEOF(QUERY (ala1 <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' + 'ITEMS')| (SIZEOF(QUERY (ala2 <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.' +
'ITEMS')| (ala1.attribute_name = ala2.attribute_name) AND (ala1.language
= ala2.language))) > 1))) > 0))) = 0;
END_RULE;

RULE restrict_name_for_known_source FOR
   (known_source);
  WHERE
    WR1 :
          SIZEOF(QUERY (ks <* known_source| (ks.name <> 'ISO 13584
library'))) = 0;
END_RULE;

RULE restrict_part_occurrence FOR
   (product_definition,
    product_definition_occurrence_relationship);
  WHERE
    WR1 :
          SIZEOF(QUERY (pd <* product_definition|
(pd.frame_of_reference.name = 'part occurrence') AND NOT (pd.name IN [
'single instance', 'quantified instance' ]))) = 0;
    WR2 :
          SIZEOF(QUERY (pd <* product_definition|
((pd.frame_of_reference.name = 'part occurrence') AND (SIZEOF(QUERY (cd
<* USEDIN(pd, 'CUTTING_TOOL_SCHEMA.' +
'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION')|
('CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION_USAGE' IN TYPEOF(cd)))) =
0)) AND (SIZEOF(USEDIN(pd, 'CUTTING_TOOL_SCHEMA.' +
'PRODUCT_DEFINITION_OCCURRENCE_RELATIONSHIP.' + 'OCCURRENCE')) = 0))) =
0;
    WR3 :
          SIZEOF(QUERY (pd <* product_definition|
((pd.frame_of_reference.name = 'part occurrence') AND (pd.name =
'quantified instance')) AND (SIZEOF(QUERY (ppd <* USEDIN(pd,
'CUTTING_TOOL_SCHEMA.' + 'PROPERTY_DEFINITION.' + 'DEFINITION')|
(ppd.name = 'occurrence quantity') AND (SIZEOF(QUERY (pdr <* USEDIN(ppd,
'CUTTING_TOOL_SCHEMA.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
'DEFINITION')| ((pdr.used_representation.name = 'quantity') AND
(SIZEOF(pdr.used_representation.items) = 1)) AND (SIZEOF(QUERY (i <*
pdr.used_representation.items| ('CUTTING_TOOL_SCHEMA.' +
'MEASURE_REPRESENTATION_ITEM' IN TYPEOF(i)) AND (i.name = 'quantity
measure'))) = 1))) = 1))) = 0))) = 0;
END_RULE;

RULE restrict_product_category_for_product FOR
  (product);
  WHERE
    WR1 :
          SIZEOF(QUERY (p <* product| (SIZEOF(QUERY (prpc <* USEDIN(p,
'CUTTING_TOOL_SCHEMA.' + 'PRODUCT_RELATED_PRODUCT_CATEGORY.' +
'PRODUCTS')| (prpc.name IN [ 'cutting item', 'tool item', 'adaptive
item', 'accesory item', 'assembly item', 'document', 'grade', 'coating',
'physically realized product', 'substrate' ]))) = 0))) = 0;
    WR2 :
          SIZEOF(QUERY (p <* product| (SIZEOF(QUERY (prpc <* USEDIN(p,
'CUTTING_TOOL_SCHEMA.' + 'PRODUCT_RELATED_PRODUCT_CATEGORY.' +
'PRODUCTS')| (prpc.name IN [ 'cutting item', 'tool item', 'adaptive
item', 'accesory item', 'assembly item', 'document', 'grade', 'coating',
'substrate' ]))) > 1))) = 0;
    WR3 :
          SIZEOF(QUERY (p <* product| (SIZEOF(QUERY (prpc <* USEDIN(p,
'CUTTING_TOOL_SCHEMA.' + 'PRODUCT_RELATED_PRODUCT_CATEGORY.' +
'PRODUCTS')| (prpc.name IN [ 'physically realized product' ]))) > 1))) =
0;
END_RULE;

RULE restrict_product_definition_context_for_applied_location_assignment
FOR
  (applied_location_assignment);
  WHERE
    WR1 :
          SIZEOF(QUERY (ala <* applied_location_assignment|
(ala.items.frame_of_reference.name <> 'physical occurrence'))) = 0;
END_RULE;

RULE
restrict_product_definition_context_for_applied_state_type_assignment
FOR
  (applied_state_type_assignment);
  WHERE
    WR1 :
          SIZEOF(QUERY (pd <* product_definition|
(pd.frame_of_reference.name <> 'physical occurrence') AND USEDIN(pd,
('CUTTING_TOOL_SCHEMA.' + 'APPLIED_STATE_TYPE_ASSIGNMENT.' +
'ITEM_SET')))) = 0;
END_RULE;

RULE restrict_product_definition_context_for_product FOR
  (product_definition,
   product_related_product_category);
  WHERE
    WR1 :
          SIZEOF(QUERY (pd <* product_definition|
(pd.frame_of_reference.name IN [ 'part definition', 'part occurrence' ])
AND (SIZEOF(QUERY (prpc <* USEDIN(pd.formation.of_product,
'CUTTING_TOOL_SCHEMA.PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS')|
(prpc.name IN [ 'part', 'raw material', 'tool', 'cutting item', 'tool
item', 'adaptive item', 'accesory item', 'assembly item' ]))) = 0))) =
0;
    WR2 :
          SIZEOF(QUERY (pd <* product_definition|
(pd.frame_of_reference.name = 'physical occurrence') AND (SIZEOF(QUERY
(prpc <* USEDIN(pd.formation.of_product, 'CUTTING_TOOL_SCHEMA.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS')| (prpc.name =
'physically realized product'))) = 0))) = 0;
    WR3 :
          SIZEOF(QUERY (pd <* product_definition|
(pd.frame_of_reference.name IN [ 'physical document definition',
'digital document definition' ]) AND (SIZEOF(QUERY (prpc <*
USEDIN(pd.formation.of_product, 'CUTTING_TOOL_SCHEMA.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS')| (prpc.name =
'document'))) = 0))) = 0;
    WR4 :
          SIZEOF(QUERY (pd <* product_definition|
(pd.frame_of_reference.name = 'material definition') AND (SIZEOF(QUERY
(prpc <* USEDIN(pd.formation.of_product, 'CUTTING_TOOL_SCHEMA.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS')| (prpc.name IN [
'grade', 'substrate', 'coating' ]))) = 0))) = 0;
END_RULE;

RULE restrict_product_definitions_for_product_definition_relationship FOR
  (assembly_component_usage,
   product_definition_relationship);
  WHERE
    WR1 :
          SIZEOF(QUERY (pdr <* product_definition_relationship|
(('CUTTING_TOOL_SCHEMA.' + 'ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr))
AND ((pdr.relating_product_definition.frame_of_reference.name <> 'part
definition') OR (pdr.related_product_definition.frame_of_reference.name
<> 'part definition'))) AND
((pdr.relating_product_definition.frame_of_reference.name <> 'physical
occurrence') OR (pdr.related_product_definition.frame_of_reference.name
<> 'physical occurrence')))) = 0;
    WR2 :
          SIZEOF(QUERY (pdr <* product_definition_relationship| (pdr.name
= 'physical realization') AND
((pdr.relating_product_definition.frame_of_reference.name <> 'part
definition') OR (pdr.related_product_definition.frame_of_reference.name
<> 'physical occurrence')))) = 0;
    WR3 :
          SIZEOF(QUERY (pdr <* product_definition_relationship| (pdr.name
= 'physical occurrence usage') AND ((NOT ('CUTTING_TOOL_SCHEMA.' +
'ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr)) OR
(pdr.relating_product_definition.frame_of_reference.name <> 'physical
occurrence')) OR (pdr.related_product_definition.frame_of_reference.name
<> 'physical occurrence')))) = 0;
    WR4 :
          SIZEOF(QUERY (pdr <* product_definition_relationship| (pdr.name
= 'substrate') AND
((pdr.relating_product_definition.frame_of_reference.name <> 'material
definition') OR (pdr.related_product_definition.frame_of_reference.name
<> 'material definition')))) = 0;
    WR5 :
          SIZEOF(QUERY (pdr <* product_definition_relationship| (pdr.name
= 'coating') AND
((pdr.relating_product_definition.frame_of_reference.name <> 'material
definition') OR (pdr.related_product_definition.frame_of_reference.name
<> 'material definition')))) = 0;
END_RULE;

RULE restrict_properties_of_document_file FOR
  (document_file);
  WHERE
    WR1 :
          SIZEOF(QUERY (df <* document_file| (SIZEOF(QUERY (pd <*
USEDIN(df, 'CUTTING_TOOL_SCHEMA.' + 'PROPERTY_DEFINITION.' +
'DEFINITION')| (pd.name = 'document property'))) > 1))) = 0;
    WR2 :
          SIZEOF(QUERY (df <* document_file| (SIZEOF(QUERY (rt <*
df.representation_types| ('CUTTING_TOOL_SCHEMA.' +
'DOCUMENT_REPRESENTATION_TYPE' IN TYPEOF(rt)) AND (rt.name IN [
'digital', 'physical' ]))) = 0))) = 0;
END_RULE;

RULE restrict_properties_of_document_representation FOR
  (product_definition);
  WHERE
    WR1 :
          SIZEOF(QUERY (pd <* product_definition|
(pd.frame_of_reference.name IN [ 'physical document definition',
'digital document definition', 'physical model occurrence' ]) AND
(SIZEOF(QUERY (pd1 <* USEDIN(pd, 'CUTTING_TOOL_SCHEMA.' +
'PROPERTY_DEFINITION.' + 'DEFINITION')| (pd1.name = 'document
property'))) > 1))) = 0;
END_RULE;

RULE restrict_representation_for_coupling FOR
  (representation,
   representation_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (r <* representation| (r.name = 'coupling') AND
((SIZEOF(r.items) < 4) OR (SIZEOF(r.items) > 5)))) = 0;
    WR2 :
          SIZEOF(QUERY (ri <* representation_item| (SIZEOF(QUERY (r <*
USEDIN(ri, 'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS')| (r.name =
'coupling'))) > 0) AND NOT (ri.name IN [ 'coupling type', 'pieces',
'side', 'size', 'style' ]))) = 0;
```

```
    WR3 :
          SIZEOF(QUERY (r <* representation| (r.name = 'coupling') AND
(SIZEOF(QUERY (i <* r.items| (i.name = 'coupling type') AND
('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) > 1))) = 0;
    WR4 :
          SIZEOF(QUERY (r <* representation| (r.name = 'coupling') AND
(SIZEOF(QUERY (i <* r.items| (i.name = 'pieces') AND (SIZEOF([
('CUTTING_TOOL_SCHEMA.' + 'MEASURE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'VALUE_REPRESENTATION_ITEM') ] * TYPEOF(i)) =
1))) > 1))) = 0;
    WR5 :
          SIZEOF(QUERY (r <* representation| (r.name = 'coupling') AND
(SIZEOF(QUERY (i <* r.items| (i.name = 'side') AND
('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) > 1))) = 0;
    WR6 :
          SIZEOF(QUERY (r <* representation| (r.name = 'coupling') AND
(SIZEOF(QUERY (i <* r.items| (i.name = 'size') AND (SIZEOF([
('CUTTING_TOOL_SCHEMA.' + 'MEASURE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'VALUE_REPRESENTATION_ITEM') ] * TYPEOF(i)) =
1))) > 1))) = 0;
    WR7 :
          SIZEOF(QUERY (r <* representation| (r.name = 'coupling') AND
(SIZEOF(QUERY (i <* r.items| (i.name = 'style') AND
('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) > 1))) = 0;
END_RULE;

RULE restrict_representation_for_cutting_condition FOR
  (representation,
   representation_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (r <* representation| (r.name = 'cutting
condition') AND (SIZEOF(r.items) = 1))) = 0;
    WR2 :
          SIZEOF(QUERY (ri <* representation_item| (SIZEOF(QUERY (r <*
USEDIN(ri, 'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS')| (r.name =
'cutting condition'))) > 0) AND NOT (ri.name IN [ 'condition name' ])))
= 0;
    WR3 :
          SIZEOF(QUERY (r <* representation| (r.name = 'cutting
condition') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'condition name')
AND ('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) > 1))) = 0;
END_RULE;

RULE restrict_representation_for_cutting_data_association FOR
  (representation,
   representation_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (r <* representation| (r.name = 'cutting data
association') AND (SIZEOF(r.items) = 3))) = 0;
    WR2 :
          SIZEOF(QUERY (ri <* representation_item| (SIZEOF(QUERY (r <*
USEDIN(ri, 'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS')| (r.name =
'cutting data association'))) > 0) AND NOT (ri.name IN [ 'depth of cut',
'feed', 'speed' ]))) = 0;
    WR3 :
          SIZEOF(QUERY (r <* representation| (r.name = 'cutting data
association') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'depth of cut')
AND (SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'MEASURE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'VALUE_RANGE'), ('CUTTING_TOOL_SCHEMA.' +
'VALUE_REPRESENTATION_ITEM') ] * TYPEOF(i)) = 1))) > 1))) = 0;
    WR4 :
          SIZEOF(QUERY (r <* representation| (r.name = 'cutting data
association') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'feed') AND
(SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'MEASURE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'VALUE_RANGE'), ('CUTTING_TOOL_SCHEMA.' +
'VALUE_REPRESENTATION_ITEM') ] * TYPEOF(i)) = 1))) > 1))) = 0;
    WR5 :
          SIZEOF(QUERY (r <* representation| (r.name = 'cutting data
association') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'speed') AND
(SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'MEASURE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'VALUE_RANGE'), ('CUTTING_TOOL_SCHEMA.' +
'VALUE_REPRESENTATION_ITEM') ] * TYPEOF(i)) = 1))) > 1))) = 0;
END_RULE;

RULE restrict_representation_for_document_content_property FOR
  (representation,
   representation_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document
content') AND ((SIZEOF(r.items) < 1) OR (SIZEOF(r.items) > 3)))) = 0;
    WR2 :
          SIZEOF(QUERY (ri <* representation_item| (SIZEOF(QUERY (r <*
USEDIN(ri, 'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS')| (r.name =
'document content'))) > 0) AND NOT (ri.name IN [ 'detail level',
'geometry type', 'real world scale' ]))) = 0;
    WR3 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document
content') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'detail level') AND
('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) > 1))) = 0;
    WR4 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document
content') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'geometry type')
AND ('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) > 1))) = 0;
    WR5 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document
content') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'real world scale')
AND (SIZEOF([ ('CUTTING_TOOL_SCHEMA.' + 'MEASURE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'VALUE_REPRESENTATION_ITEM') ] * TYPEOF(i)) =
1))) > 1))) = 0;
END_RULE;

RULE restrict_representation_for_document_creation_property FOR
  (representation,
   representation_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document
creation') AND ((SIZEOF(r.items) < 2) OR (SIZEOF(r.items) > 3)))) = 0;
    WR2 :
          SIZEOF(QUERY (ri <* representation_item| (SIZEOF(QUERY (r <*
USEDIN(ri, 'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS')| (r.name =
'document creation'))) > 0) AND NOT (ri.name IN [ 'creating interface',
'creating system', 'operating system' ]))) = 0;
    WR3 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document
creation') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'creating
interface') AND ('CUTTING_TOOL_SCHEMA.' +
'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(i)))) > 1))) = 0;
    WR4 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document
creation') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'creating system')
AND ('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) <> 1))) = 0;
    WR5 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document
creation') AND (SIZEOF(QUERY (i <* r.items| (i.name = 'operating
system') AND ('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(i)))) > 1))) = 0;
END_RULE;

RULE restrict_representation_for_document_format_property FOR
  (representation,
   representation_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document format')
AND ((SIZEOF(r.items) < 1) OR (SIZEOF(r.items) > 3)))) = 0;
    WR2 :
          SIZEOF(QUERY (ri <* representation_item| (SIZEOF(QUERY (r <*
USEDIN(ri, 'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS')| (r.name =
'document format'))) > 0) AND NOT (ri.name IN [ 'character code', 'data
format' ]))) = 0;
    WR3 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document format')
AND (SIZEOF(QUERY (i <* r.items| (i.name = 'character code') AND
('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) > 1))) = 0;
    WR4 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document format')
AND (SIZEOF(QUERY (i <* r.items| (i.name = 'data format') AND
('CUTTING_TOOL_SCHEMA.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(i)))) > 1))) = 0;
    WR5 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document format')
AND (SIZEOF(QUERY (i <* r.items| (i.name = 'size format') AND
('CUTTING_TOOL_SCHEMA.PLANAR_EXTENT' IN TYPEOF(i)))) > 1))) = 0;
END_RULE;

RULE restrict_representation_for_document_properties FOR
  (representation,
   property_definition,
   representation_context);
  WHERE
    WR1 :
          SIZEOF(QUERY (pd <* property_definition| (pd.name = 'document
property') AND (SIZEOF(QUERY (pdr <* USEDIN(pd, 'CUTTING_TOOL_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.' + 'DEFINITION')| NOT
(pdr.used_representation.name IN [ 'document content', 'document
creation', 'document format', 'document size' ]))) > 0))) = 0;
    WR2 :
          SIZEOF(QUERY (r <* representation| (r.name IN [ 'document
content', 'document creation', 'document format', 'document size' ]) AND
(SIZEOF(QUERY (pdr <* USEDIN(r, 'CUTTING_TOOL_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.' + 'USED_REPRESENTATION')|
('CUTTING_TOOL_SCHEMA.' + 'PROPERTY_DEFINITION' IN
TYPEOF(pdr.definition)) AND (pdr.definition.name = 'document
property'))) = 0))) = 0;
    WR3 :
          SIZEOF(QUERY (r <* representation| (r.name IN [ 'document
content', 'document creation', 'document format', 'document size' ]) AND
(r.context_of_items.context_type <> 'document parameters'))) = 0;
    WR4 :
          SIZEOF(QUERY (rc <* representation_context| (rc.context_type =
'document parameters') AND (SIZEOF(QUERY (r <* USEDIN(rc,
'CUTTING_TOOL_SCHEMA.REPRESENTATION.CONTEXT_OF_ITEMS')| NOT (r.name IN [
'document content', 'document creation', 'document format', 'document
size' ]))) > 0))) = 0;
END_RULE;

RULE restrict_representation_for_document_size_property FOR
  (representation,
   representation_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document size')
AND ((SIZEOF(r.items) < 1) OR (SIZEOF(r.items) > 2)))) = 0;
    WR2 :
          SIZEOF(QUERY (ri <* representation_item| (SIZEOF(QUERY (r <*
USEDIN(ri, 'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS')| (r.name =
'document size'))) > 0) AND NOT (ri.name IN [ 'file size', 'page count'
]))) = 0;
    WR3 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document size')
AND (SIZEOF(QUERY (i <* r.items| (i.name = 'file size') AND (SIZEOF([
('CUTTING_TOOL_SCHEMA.' + 'MEASURE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'VALUE_RANGE'), ('CUTTING_TOOL_SCHEMA.' +
'VALUE_REPRESENTATION_ITEM') ] * TYPEOF(i)) = 1))) > 1))) = 0;
    WR4 :
          SIZEOF(QUERY (r <* representation| (r.name = 'document size')
AND (SIZEOF(QUERY (i <* r.items| (i.name = 'page count') AND (SIZEOF([
('CUTTING_TOOL_SCHEMA.' + 'MEASURE_REPRESENTATION_ITEM'),
('CUTTING_TOOL_SCHEMA.' + 'VALUE_RANGE'), ('CUTTING_TOOL_SCHEMA.' +
'VALUE_REPRESENTATION_ITEM') ] * TYPEOF(i)) = 1))) > 1))) = 0;
END_RULE;

RULE restrict_version_assignment_for_applied_identification_assignment
FOR
  (applied_identification_assignment);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* applied_identification_assignment|
(SIZEOF(QUERY (ia <* USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'APPLIED_IDENTIFICATION_ASSIGNMENT.' + 'ITEMS')| (ia.role.name =
'version'))) > 1))) = 0;
END_RULE;

RULE restrict_version_assignment_for_class FOR
  (class);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* class| (SIZEOF(QUERY (ia <* USEDIN(ent,
'CUTTING_TOOL_SCHEMA.' + 'APPLIED_IDENTIFICATION_ASSIGNMENT.' +
'ITEMS')| (ia.role.name = 'version'))) > 1))) = 0;
END_RULE;

RULE restrict_version_assignment_for_document_file FOR
  (document_file);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* document_file| (SIZEOF(QUERY (ia <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'APPLIED_IDENTIFICATION_ASSIGNMENT.' + 'ITEMS')| (ia.role.name =
'version'))) > 1))) = 0;
END_RULE;

RULE restrict_version_assignment_for_effectivity FOR
  (effectivity);
  WHERE
    WR1 :
          SIZEOF(QUERY (ent <* effectivity| (SIZEOF(QUERY (ia <*
USEDIN(ent, 'CUTTING_TOOL_SCHEMA.' +
'APPLIED_IDENTIFICATION_ASSIGNMENT.' + 'ITEMS')| (ia.role.name =
'version'))) > 1))) = 0;
END_RULE;

RULE substrate_requires_product_definition FOR
  (product_definition,
   product_definition_formation,
   product_related_product_category);
  WHERE
    WR1 :
          SIZEOF(QUERY (pdf <* product_definition_formation|
(SIZEOF(QUERY (prpc <* USEDIN(pdf.of_product, 'CUTTING_TOOL_SCHEMA.' +
'PRODUCT_RELATED_PRODUCT_CATEGORY.' + 'PRODUCTS')| (prpc.name =
'substrate'))) > 0) AND (SIZEOF(QUERY (pd <* USEDIN(pdf,
'CUTTING_TOOL_SCHEMA.' + 'PRODUCT_DEFINITION.' + 'FORMATION')|
(pd.frame_of_reference.name = 'material definition'))) <> 1))) = 0;
END_RULE;

RULE subtype_exclusiveness_characterized_object FOR
  (characterized_object);
  WHERE
    WR1 :
          SIZEOF(QUERY (co <* characterized_object| NOT
type_check_function(co, [ ('CUTTING_TOOL_SCHEMA.' +
'CHARACTERIZED_CLASS'), ('CUTTING_TOOL_SCHEMA.' + 'DOCUMENT_FILE'),
('CUTTING_TOOL_SCHEMA.' + 'FEATURE_COMPONENT_DEFINITION'),
('CUTTING_TOOL_SCHEMA.' + 'FEATURE_DEFINITION'), ('CUTTING_TOOL_SCHEMA.'
+ 'PRODUCT_CLASS'), ('CUTTING_TOOL_SCHEMA.' + 'PRODUCT_IDENTIFICATION')
], 3))) = 0;
```

```
END_RULE;

RULE subtype_exclusiveness_document_reference FOR
  (document_reference);
  WHERE
    WR1 :
          SIZEOF(QUERY (dr <* document_reference| NOT
type_check_function(dr, [ ('CUTTING_TOOL_SCHEMA.' +
'APPLIED_DOCUMENT_REFERENCE'), ('CUTTING_TOOL_SCHEMA.' +
'DRAUGHTING_SPECIFICATION_REFERENCE') ], 3))) = 0;
END_RULE;

RULE subtype_exclusiveness_feature_definition FOR
  (feature_definition);
  WHERE
    WR1 :
          SIZEOF(QUERY (fd <* feature_definition| NOT
type_check_function(fd, [ ('CUTTING_TOOL_SCHEMA.' + 'BARRING_HOLE'),
('CUTTING_TOOL_SCHEMA.' + 'BEAD'), ('CUTTING_TOOL_SCHEMA.' + 'BOSS'),
('CUTTING_TOOL_SCHEMA.' + 'COMPOUND_FEATURE'), ('CUTTING_TOOL_SCHEMA.' +
'EXTERNALLY_DEFINED_FEATURE_DEFINITION'), ('CUTTING_TOOL_SCHEMA.' +
'FEATURE_IN_PANEL'), ('CUTTING_TOOL_SCHEMA.' + 'GENERAL_FEATURE'),
('CUTTING_TOOL_SCHEMA.' + 'HOLE_IN_PANEL'), ('CUTTING_TOOL_SCHEMA.' +
'JOGGLE'), ('CUTTING_TOOL_SCHEMA.' + 'LOCATOR'), ('CUTTING_TOOL_SCHEMA.'
+ 'POCKET'), ('CUTTING_TOOL_SCHEMA.' + 'RIB'), ('CUTTING_TOOL_SCHEMA.' +
'ROUND_HOLE'), ('CUTTING_TOOL_SCHEMA.' + 'SLOT'),
('CUTTING_TOOL_SCHEMA.' + 'THREAD') ], 3))) = 0;
END_RULE;

RULE subtype_exclusiveness_pre_defined_item FOR
  (pre_defined_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (pdi <* pre_defined_item| NOT
type_check_function(pdi, [ ('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE'),
('CUTTING_TOOL_SCHEMA.' + 'PRE_DEFINED_COLOUR'), ('CUTTING_TOOL_SCHEMA.'
+ 'PRE_DEFINED_CURVE_FONT'), ('CUTTING_TOOL_SCHEMA.' +
'PRE_DEFINED_MARKER'), ('CUTTING_TOOL_SCHEMA.' +
'PRE_DEFINED_PRESENTATION_STYLE'), ('CUTTING_TOOL_SCHEMA.' +
'PRE_DEFINED_SYMBOL'), ('CUTTING_TOOL_SCHEMA.' +
'PRE_DEFINED_TEXT_FONT') ], 3))) = 0;
END_RULE;

RULE subtype_mandatory_pre_defined_item FOR
  (pre_defined_item);
  WHERE
    WR1 :
          SIZEOF(QUERY (pdi <* pre_defined_item| NOT
type_check_function(pdi, [ ('CUTTING_TOOL_SCHEMA.' + 'KNOWN_SOURCE'),
('CUTTING_TOOL_SCHEMA.' + 'PRE_DEFINED_COLOUR'), ('CUTTING_TOOL_SCHEMA.'
+ 'PRE_DEFINED_CURVE_FONT'), ('CUTTING_TOOL_SCHEMA.' +
'PRE_DEFINED_MARKER'), ('CUTTING_TOOL_SCHEMA.' +
'PRE_DEFINED_PRESENTATION_STYLE'), ('CUTTING_TOOL_SCHEMA.' +
'PRE_DEFINED_SYMBOL'), ('CUTTING_TOOL_SCHEMA.' +
'PRE_DEFINED_TEXT_FONT') ], 0))) = 0;
END_RULE;

FUNCTION acyclic
```

```
(* ***********************************
Functions in the schema CUTTING_TOOL_SCHEMA
*********************************** *)(arg1 : generic_expression;
       arg2 : SET OF generic_expression ) : BOOLEAN;
   LOCAL
      result : BOOLEAN;
   END_LOCAL;
      IF 'CUTTING_TOOL_SCHEMA.SIMPLE_GENERIC_EXPRESSION' IN TYPEOF(arg1)
THEN
         RETURN (TRUE);
      END_IF;
      IF arg1 IN arg2 THEN
         RETURN (FALSE);
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.UNARY_GENERIC_EXPRESSION' IN TYPEOF(arg1)
THEN
         RETURN (acyclic(arg1\unary_generic_expression.operand, arg2 + [
arg1 ]));
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.BINARY_GENERIC_EXPRESSION' IN TYPEOF(arg1)
THEN
         RETURN (acyclic(arg1\binary_generic_expression.operands[1],
(arg2 + [ arg1 ])) AND
acyclic(arg1\binary_generic_expression.operands[2], (arg2 + [ arg1 ])));
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.MULTIPLE_ARITY_GENERIC_EXPRESSION' IN
TYPEOF(arg1) THEN
         result := TRUE;
         REPEAT i := 1 TO
SIZEOF(arg1\multiple_arity_generic_expression.operands);
            result := result AND
acyclic(arg1\multiple_arity_generic_expression.operands[i], (arg2 + [
arg1 ]));
         END_REPEAT;
         RETURN (result);
      END_IF;
END_FUNCTION;

FUNCTION acyclic_mapped_representation
(parent_set : SET OF representation;
       children_set : SET OF representation_item ) : BOOLEAN;
   LOCAL
      x : SET OF representation_item;
      y : SET OF representation_item;
   END_LOCAL;
      x := QUERY (z <* children_set| 'CUTTING_TOOL_SCHEMA.MAPPED_ITEM' IN
TYPEOF(z));
      IF SIZEOF(x) > 0 THEN
         REPEAT i := 1 TO HIINDEX(x);
            IF x[i]\mapped_item.mapping_source.mapped_representation IN
parent_set THEN
               RETURN (FALSE);
            END_IF;
            IF NOT acyclic_mapped_representation((parent_set +
x[i]\mapped_item.mapping_source.mapped_representation),
x[i]\mapped_item.mapping_source.mapped_representation.items) THEN
               RETURN (FALSE);
            END_IF;
         END_REPEAT;
      END_IF;
      x := children_set - x;
      IF SIZEOF(x) > 0 THEN
         REPEAT i := 1 TO HIINDEX(x);
            y := QUERY (z <* bag_to_set(USEDIN(x[i], ''))|
'CUTTING_TOOL_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z));
            IF NOT acyclic_mapped_representation(parent_set, y) THEN
               RETURN (FALSE);
            END_IF;
         END_REPEAT;
      END_IF;
      RETURN (TRUE);
END_FUNCTION;

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, 'CUTTING_TOOL_SCHEMA.' +
'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,
'CUTTING_TOOL_SCHEMA.' + '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 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 : direction;
       axis2 : direction;
       axis3 : direction ) : LIST [2:3] OF direction;
   LOCAL
      u : LIST [2:3] OF direction;
      factor : REAL;
      d1 : direction;
      d2 : direction;
   END_LOCAL;
      IF dim = 3 THEN
         d1 := NVL(normalise(axis3), dummy_gri || direction([ 0.00000,
0.00000, 1.00000 ]));
         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 < 0.00000 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([ 1.00000, 0.00000 ]),
dummy_gri || direction([ 0.00000, 1.00000 ]) ];
            END_IF;
         END_IF;
      END_IF;
      RETURN (u);
END_FUNCTION;

FUNCTION build_2axes
(ref_direction : direction ) : LIST [2:2] OF direction;
   LOCAL
      d : direction := NVL(normalise(ref_direction), dummy_gri ||
direction([ 1.00000, 0.00000 ]));
   END_LOCAL;
      RETURN ([ d, orthogonal_complement(d) ]);
END_FUNCTION;

FUNCTION build_axes
(axis : direction;
       ref_direction : direction ) : LIST [3:3] OF direction;
   LOCAL
      d1 : direction;
      d2 : direction;
   END_LOCAL;
      d1 := NVL(normalise(axis), dummy_gri || direction([ 0.00000,
0.00000, 1.00000 ]));
      d2 := first_proj_axis(d1, ref_direction);
      RETURN ([ d2, normalise(cross_product(d1, d2)).orientation, d1 ]);
END_FUNCTION;

FUNCTION cross_product
(arg1 : direction;
       arg2 : direction ) : vector;
   LOCAL
      mag : REAL;
      res : direction;
      v1 : LIST [3:3] OF REAL;
      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 := 0.00000;
            REPEAT i := 1 TO 3;
               mag := mag + res.direction_ratios[i] *
res.direction_ratios[i];
            END_REPEAT;
            IF mag > 0.00000 THEN
               result := dummy_gri || vector(res, sqrt(mag));
            ELSE
               result := dummy_gri || vector(arg1, 0.00000);
            END_IF;
            RETURN (result);
         END;
      END_IF;
```

```
END_FUNCTION;

FUNCTION derive_dimensional_exponents
(x : unit ) : dimensional_exponents;
   LOCAL
      result : dimensional_exponents := dimensional_exponents(0.00000,
0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000);
   END_LOCAL;
      IF 'CUTTING_TOOL_SCHEMA.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;
      IF 'CUTTING_TOOL_SCHEMA.CARTESIAN_POINT' IN TYPEOF(item) THEN
         dim := SIZEOF(item\cartesian_point.coordinates);
         RETURN (dim);
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.DIRECTION' IN TYPEOF(item) THEN
         dim := SIZEOF(item\direction.direction_ratios);
         RETURN (dim);
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.VECTOR' IN TYPEOF(item) THEN
         dim :=
SIZEOF(item\vector.orientation\direction.direction_ratios);
         RETURN (dim);
      END_IF;
      x := using_representations(item);
      y := x[1].context_of_items;
      dim :=
y\geometric_representation_context.coordinate_space_dimension;
      RETURN (dim);
END_FUNCTION;

FUNCTION dimensions_for_si_unit
(n : si_unit_name ) : dimensional_exponents;
      CASE n OF
         metre :
               RETURN (dimensional_exponents(1.00000, 0.00000, 0.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         gram :
               RETURN (dimensional_exponents(0.00000, 1.00000, 0.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         second :
               RETURN (dimensional_exponents(0.00000, 0.00000, 1.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         ampere :
               RETURN (dimensional_exponents(0.00000, 0.00000, 0.00000,
1.00000, 0.00000, 0.00000, 0.00000));
         kelvin :
               RETURN (dimensional_exponents(0.00000, 0.00000, 0.00000,
0.00000, 1.00000, 0.00000, 0.00000));
         mole :
               RETURN (dimensional_exponents(0.00000, 0.00000, 0.00000,
0.00000, 0.00000, 1.00000, 0.00000));
         candela :
               RETURN (dimensional_exponents(0.00000, 0.00000, 0.00000,
0.00000, 0.00000, 0.00000, 1.00000));
         radian :
               RETURN (dimensional_exponents(0.00000, 0.00000, 0.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         steradian :
               RETURN (dimensional_exponents(0.00000, 0.00000, 0.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         hertz :
               RETURN (dimensional_exponents(0.00000, 0.00000, -1.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         newton :
               RETURN (dimensional_exponents(1.00000, 1.00000, -2.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         pascal :
               RETURN (dimensional_exponents(-1.00000, 1.00000, -2.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         joule :
               RETURN (dimensional_exponents(2.00000, 1.00000, -2.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         watt :
               RETURN (dimensional_exponents(2.00000, 1.00000, -3.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         coulomb :
               RETURN (dimensional_exponents(0.00000, 0.00000, 1.00000,
1.00000, 0.00000, 0.00000, 0.00000));
         volt :
               RETURN (dimensional_exponents(2.00000, 1.00000, -3.00000,
-1.00000, 0.00000, 0.00000, 0.00000));
         farad :
               RETURN (dimensional_exponents(-2.00000, -1.00000, 4.00000,
1.00000, 0.00000, 0.00000, 0.00000));
         ohm :
               RETURN (dimensional_exponents(2.00000, 1.00000, -3.00000,
-2.00000, 0.00000, 0.00000, 0.00000));
         siemens :
               RETURN (dimensional_exponents(-2.00000, -1.00000, 3.00000,
2.00000, 0.00000, 0.00000, 0.00000));
         weber :
               RETURN (dimensional_exponents(2.00000, 1.00000, -2.00000,
-1.00000, 0.00000, 0.00000, 0.00000));
         tesla :
               RETURN (dimensional_exponents(0.00000, 1.00000, -2.00000,
-1.00000, 0.00000, 0.00000, 0.00000));
         henry :
               RETURN (dimensional_exponents(2.00000, 1.00000, -2.00000,
-2.00000, 0.00000, 0.00000, 0.00000));
         degree_Celsius :
               RETURN (dimensional_exponents(0.00000, 0.00000, 0.00000,
0.00000, 1.00000, 0.00000, 0.00000));
         lumen :
               RETURN (dimensional_exponents(0.00000, 0.00000, 0.00000,
0.00000, 0.00000, 0.00000, 1.00000));
         lux :
               RETURN (dimensional_exponents(-2.00000, 0.00000, 0.00000,
0.00000, 0.00000, 0.00000, 1.00000));
         becquerel :
               RETURN (dimensional_exponents(0.00000, 0.00000, -1.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         gray :
               RETURN (dimensional_exponents(2.00000, 0.00000, -2.00000,
0.00000, 0.00000, 0.00000, 0.00000));
         sievert :
               RETURN (dimensional_exponents(2.00000, 0.00000, -2.00000,
0.00000, 0.00000, 0.00000, 0.00000));
      OTHERWISE :
            RETURN (?);
      END_CASE;
END_FUNCTION;

FUNCTION dot_product
(arg1 : direction;
       arg2 : direction ) : REAL;
   LOCAL
      scalar : REAL;
      vec1 : direction;
      vec2 : direction;
      ndim : INTEGER;
   END_LOCAL;
      IF NOT EXISTS(arg1) OR NOT EXISTS(arg2) THEN
         scalar := ?;
      ELSE
         IF arg1.dim <> arg2.dim THEN
            scalar := ?;
         ELSE
            BEGIN
               vec1 := normalise(arg1);
               vec2 := normalise(arg2);
```

```
              ndim := arg1.dim;
              scalar := 0.00000;
              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 first_proj_axis
(z_axis : direction;
      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 <> [ 1.00000, 0.00000, 0.00000 ] THEN
               v := dummy_gri || direction([ 1.00000, 0.00000, 0.00000
]);
            ELSE
               v := dummy_gri || direction([ 0.00000, 1.00000, 0.00000
]);
            END_IF;
         ELSE
            IF arg.dim <> 3 THEN
               RETURN (?);
            END_IF;
            IF cross_product(arg, z).magnitude = 0.00000 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 get_description_value
(obj : description_attribute_select ) : text;
   LOCAL
      description_bag : BAG OF description_attribute := USEDIN(obj,
'CUTTING_TOOL_SCHEMA.' + 'DESCRIPTION_ATTRIBUTE.' + 'DESCRIBED_ITEM');
   END_LOCAL;
      IF SIZEOF(description_bag) = 1 THEN
         RETURN (description_bag[1].attribute_value);
      ELSE
         RETURN (?);
      END_IF;
END_FUNCTION;

FUNCTION get_id_value
(obj : id_attribute_select ) : identifier;
   LOCAL
      id_bag : BAG OF id_attribute := USEDIN(obj, 'CUTTING_TOOL_SCHEMA.'
+ 'ID_ATTRIBUTE.' + 'IDENTIFIED_ITEM');
   END_LOCAL;
      IF SIZEOF(id_bag) = 1 THEN
         RETURN (id_bag[1].attribute_value);
      ELSE
         RETURN (?);
      END_IF;
END_FUNCTION;

FUNCTION get_multi_language
(x : attribute_value_assignment ) : label;
   LOCAL
      alas : BAG OF attribute_language_assignment := USEDIN(x,
'CUTTING_TOOL_SCHEMA.' + 'ATTRIBUTE_LANGUAGE_ASSIGNMENT.ITEMS');
   END_LOCAL;
      IF SIZEOF(alas) > 0 THEN
         RETURN (alas[1].language);
      END_IF;
      RETURN (?);
END_FUNCTION;

FUNCTION get_name_value
(obj : name_attribute_select ) : label;
   LOCAL
      name_bag : BAG OF name_attribute := USEDIN(obj,
'CUTTING_TOOL_SCHEMA.' + 'NAME_ATTRIBUTE.' + 'NAMED_ITEM');
   END_LOCAL;
      IF SIZEOF(name_bag) = 1 THEN
         RETURN (name_bag[1].attribute_value);
      ELSE
         RETURN (?);
      END_IF;
END_FUNCTION;

FUNCTION get_property_definition_representations
(c_def_instance : characterized_definition ) : 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(c_def_instance,
'CUTTING_TOOL_SCHEMA.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 + bag_to_set(USEDIN(pd_set[i],
'CUTTING_TOOL_SCHEMA.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
'DEFINITION'));
      END_REPEAT;
      RETURN (pdr_set);
END_FUNCTION;

FUNCTION get_role
(obj : role_select ) : object_role;
   LOCAL
      role_bag : BAG OF role_association := USEDIN(obj,
'CUTTING_TOOL_SCHEMA.' + 'ROLE_ASSOCIATION.' + 'ITEM_WITH_ROLE');
   END_LOCAL;
      IF SIZEOF(role_bag) = 1 THEN
         RETURN (role_bag[1].role);
      ELSE
         RETURN (?);
      END_IF;
END_FUNCTION;

FUNCTION get_shape_aspects
(c_def_instance : characterized_definition ) : SET OF shape_aspect;
   LOCAL
      pd_set : SET OF product_definition_shape := [];
      pdr_set : SET OF shape_aspect := [];
   END_LOCAL;
      pd_set := bag_to_set(QUERY (pd <* USEDIN(c_def_instance,
'CUTTING_TOOL_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
'CUTTING_TOOL_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
      IF SIZEOF(pd_set) < 1 THEN
         RETURN (pdr_set);
      END_IF;
      REPEAT i := 1 TO HIINDEX(pd_set);
         pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i],
'CUTTING_TOOL_SCHEMA.SHAPE_ASPECT.OF_SHAPE'));
      END_REPEAT;
      RETURN (pdr_set);
END_FUNCTION;

FUNCTION is_acyclic
(arg : generic_expression ) : BOOLEAN;
      RETURN (acyclic(arg, []));
END_FUNCTION;

FUNCTION item_correlation
(items : SET OF GENERIC;
       c_items : SET OF STRING ) : LOGICAL;
   LOCAL
      c_types : SET OF STRING := [];
      c_hit : INTEGER := 0;
   END_LOCAL;
      REPEAT i := 1 TO HIINDEX(c_items);
         c_types := c_types + [ 'CUTTING_TOOL_SCHEMA.' + c_items[i] ];
      END_REPEAT;
      REPEAT i := 1 TO HIINDEX(items);
         IF SIZEOF(c_types * TYPEOF(items[i])) = 1 THEN
            c_hit := c_hit + 1;
         END_IF;
      END_REPEAT;
      IF SIZEOF(items) = c_hit THEN
         RETURN (TRUE);
      ELSE
         RETURN (FALSE);
      END_IF;
END_FUNCTION;

FUNCTION item_in_context
(item : representation_item;
       cntxt : representation_context ) : BOOLEAN;
   LOCAL
      y : BAG OF representation_item;
   END_LOCAL;
      IF SIZEOF(USEDIN(item, 'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS')
* cntxt.representations_in_context) > 0 THEN
         RETURN (TRUE);
      ELSE
         y := QUERY (z <* USEDIN(item, '')|
'CUTTING_TOOL_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z));
         IF SIZEOF(y) > 0 THEN
            REPEAT i := 1 TO HIINDEX(y);
               IF item_in_context(y[i], cntxt) THEN
                  RETURN (TRUE);
               END_IF;
            END_REPEAT;
         END_IF;
      END_IF;
      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 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 := ?;
      ELSE
         ndim := arg.dim;
         IF 'CUTTING_TOOL_SCHEMA.VECTOR' IN TYPEOF(arg) THEN
            BEGIN
               v := dummy_gri ||
direction(arg.orientation.direction_ratios);
```

```
               IF arg.magnitude = 0.00000 THEN
                  RETURN (?);
               ELSE
                  vec := dummy_gri || vector(v, 1.00000);
               END_IF;
            END;
         ELSE
            v := dummy_gri || direction(arg.direction_ratios);
         END_IF;
         mag := 0.00000;
         REPEAT i := 1 TO ndim;
            mag := mag + v.direction_ratios[i] * v.direction_ratios[i];
         END_REPEAT;
         IF mag > 0.00000 THEN
            mag := sqrt(mag);
            REPEAT i := 1 TO ndim;
               v.direction_ratios[i] := v.direction_ratios[i] / mag;
            END_REPEAT;
            IF 'CUTTING_TOOL_SCHEMA.VECTOR' IN TYPEOF(arg) THEN
               vec.orientation := v;
               result := vec;
            ELSE
               result := v;
            END_IF;
         ELSE
            RETURN (?);
         END_IF;
      END_IF;
      RETURN (result);
END_FUNCTION;

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 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 'CUTTING_TOOL_SCHEMA.VECTOR' IN TYPEOF(vec) THEN
```

```
            v := dummy_gri ||
direction(vec.orientation.direction_ratios);
            mag := scalar * vec.magnitude;
         ELSE
            v := dummy_gri || direction(vec.direction_ratios);
            mag := scalar;
         END_IF;
         IF mag < 0.00000 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 : direction;
       x_axis : direction;
       arg : direction ) : direction;
   LOCAL
      y_axis : vector;
      v : direction;
      temp : vector;
   END_LOCAL;
      IF NOT EXISTS(arg) THEN
         v := dummy_gri || direction([ 0.00000, 1.00000, 0.00000 ]);
      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 type_check_function
(the_type : GENERIC;
       sub_names : SET OF STRING;
       criterion : INTEGER ) : LOGICAL;
      IF (NOT EXISTS(the_type) OR NOT ((0 <= criterion) AND (criterion <=
3))) OR (SIZEOF(sub_names) = 0) THEN
         RETURN (UNKNOWN);
      ELSE
         CASE criterion OF
            0 :
                  RETURN (SIZEOF(sub_names * TYPEOF(the_type)) > 0);
            1 :
                  RETURN (SIZEOF(sub_names * TYPEOF(the_type)) = 0);
            2 :
                  RETURN (SIZEOF(sub_names * TYPEOF(the_type)) = 1);
            3 :
                  RETURN (SIZEOF(sub_names * TYPEOF(the_type)) <= 1);
         END_CASE;
      END_IF;
END_FUNCTION;

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;
      next_items := QUERY (z <* bag_to_set(USEDIN(item, ''))|
('CUTTING_TOOL_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z)) OR
('CUTTING_TOOL_SCHEMA.FOUNDED_ITEM' IN TYPEOF(z)));
      IF SIZEOF(next_items) > 0 THEN
         REPEAT i := 1 TO HIINDEX(next_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 (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;
      results := [];
      result_bag := USEDIN(item,
'CUTTING_TOOL_SCHEMA.REPRESENTATION.ITEMS');
      IF SIZEOF(result_bag) > 0 THEN
         REPEAT i := 1 TO HIINDEX(result_bag);
            results := results + result_bag[i];
         END_REPEAT;
      END_IF;
      intermediate_items := using_items(item, []);
      IF SIZEOF(intermediate_items) > 0 THEN
         REPEAT i := 1 TO HIINDEX(intermediate_items);
            result_bag := USEDIN(intermediate_items[i],
'CUTTING_TOOL_SCHEMA.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 (results);
END_FUNCTION;
```

```
FUNCTION valid_calendar_date
(date : calendar_date ) : LOGICAL;
      CASE date.month_component OF
         1 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 31));
         2 :
               BEGIN
                  IF leap_year(date.year_component) THEN
                     RETURN ((1 <= date.day_component) AND
(date.day_component <= 29));
                  ELSE
                     RETURN ((1 <= date.day_component) AND
(date.day_component <= 28));
                  END_IF;
               END;
         3 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 31));
         4 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 30));
         5 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 31));
         6 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 30));
         7 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 31));
         8 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 31));
         9 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 30));
         10 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 31));
         11 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 30));
         12 :
               RETURN ((1 <= date.day_component) AND (date.day_component
<= 31));
      END_CASE;
      RETURN (FALSE);
END_FUNCTION;

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 'CUTTING_TOOL_SCHEMA.LENGTH_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(1.00000, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.MASS_MEASURE' IN TYPEOF(m.value_component)
THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 1.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.TIME_MEASURE' IN TYPEOF(m.value_component)
THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 1.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.ELECTRIC_CURRENT_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 0.00000, 1.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.THERMODYNAMIC_TEMPERATURE_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 0.00000, 0.00000, 1.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.CELSIUS_TEMPERATURE_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 0.00000, 0.00000, 1.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.AMOUNT_OF_SUBSTANCE_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 0.00000, 0.00000, 0.00000,
1.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.LUMINOUS_INTENSITY_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 1.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.PLANE_ANGLE_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.SOLID_ANGLE_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.AREA_MEASURE' IN TYPEOF(m.value_component)
THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(2.00000, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.VOLUME_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(3.00000, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.RATIO_MEASURE' IN TYPEOF(m.value_component)
THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(0.00000, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.POSITIVE_LENGTH_MEASURE' IN
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <>
dimensional_exponents(1.00000, 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      IF 'CUTTING_TOOL_SCHEMA.POSITIVE_PLANE_ANGLE_MEASURE' IN 
TYPEOF(m.value_component) THEN
         IF derive_dimensional_exponents(m.unit_component) <> 
dimensional_exponents(0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 
0.00000, 0.00000) THEN
            RETURN (FALSE);
         END_IF;
      END_IF;
      RETURN (TRUE);
END_FUNCTION;

FUNCTION value_range_wr1
(agg : AGGREGATE OF representation_item ) : BOOLEAN;
      BEGIN
         IF (SIZEOF(agg) = 2) AND ((SIZEOF(QUERY (i1 <* agg|
('CUTTING_TOOL_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(i1)))) = 2) 
OR (SIZEOF(QUERY (i2 <* agg|
('CUTTING_TOOL_SCHEMA.VALUE_REPRESENTATION_ITEM' IN TYPEOF(i2)))) = 2)) 
THEN
            RETURN (TRUE);
         ELSE
            RETURN (FALSE);
         END_IF;
      END;
END_FUNCTION;

FUNCTION value_range_wr2
(agg : AGGREGATE OF representation_item ) : 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 : AGGREGATE OF representation_item ) : BOOLEAN;
      BEGIN
         IF (SIZEOF(QUERY (i <* agg|
('CUTTING_TOOL_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(i)))) <> 2) 
OR (SIZEOF(QUERY (i1 <* agg|
('CUTTING_TOOL_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(i1)) AND 
(SIZEOF(QUERY (i2 <* agg| (i1\measure_with_unit.unit_component :=: 
i2\measure_with_unit.unit_component) OR 
(i1\measure_with_unit.unit_component :=: 
i2\measure_with_unit.unit_component))) = 2))) = 2) THEN
            RETURN (TRUE);
         ELSE
            RETURN (FALSE);
         END_IF;
      END;
END_FUNCTION;

FUNCTION vector_difference
(arg1 : vector_or_direction;
         arg2 : vector_or_direction ) : vector;
     LOCAL
        result : vector;
        res : direction;
        vec1 : direction;
        vec2 : direction;
        mag : REAL;
        mag1 : REAL;
        mag2 : REAL;
        ndim : INTEGER;
     END_LOCAL;
        IF (NOT EXISTS(arg1) OR NOT EXISTS(arg2)) OR (arg1.dim <> arg2.dim)
  THEN
           RETURN (?);
        ELSE
           BEGIN
              IF 'CUTTING_TOOL_SCHEMA.VECTOR' IN TYPEOF(arg1) THEN
                 mag1 := arg1.magnitude;
                 vec1 := arg1.orientation;
              ELSE
                 mag1 := 1.00000;
                 vec1 := arg1;
              END_IF;
              IF 'CUTTING_TOOL_SCHEMA.VECTOR' IN TYPEOF(arg2) THEN
                 mag2 := arg2.magnitude;
                 vec2 := arg2.orientation;
              ELSE
                 mag2 := 1.00000;
                 vec2 := arg2;
              END_IF;
              vec1 := normalise(vec1);
              vec2 := normalise(vec2);
              ndim := SIZEOF(vec1.direction_ratios);
              mag := 0.00000;
              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 > 0.00000 THEN
                 result := dummy_gri || vector(res, sqrt(mag));
              ELSE
                 result := dummy_gri || vector(vec1, 0.00000);
              END_IF;
           END;
        END_IF;
        RETURN (result);
  END_FUNCTION;

END_SCHEMA;
```

Annex B
(normative)

AIM short names

This table provides the short names of entities specified in the AIM of this part of ISO 13399. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

APPLIED_CLASSIFICATION_ASSIGNMENT	APCLAS
CLASSIFICATION_ASSIGNMENT	CLSASS
LANGUAGE_ASSIGNMENT	LNGASS
REPRESENTATION	RPRSNT
SHAPE_REPRESENTATION	SHPRPR
SHAPE_DIMENSION_REPRESENTATION	SHDMRP
REPRESENTATION_ITEM	RPRITM
GEOMETRIC_REPRESENTATION_ITEM	GMRPIT
POINT	POINT
CARTESIAN_POINT	CRTPNT
DIRECTION	DRCTN
VECTOR	VECTOR
PLACEMENT	PLCMNT
AXIS2_PLACEMENT_2D	A2PL2D
AXIS2_PLACEMENT_3D	A2PL3D
CARTESIAN_TRANSFORMATION_OPERATOR	CRTROP
FUNCTIONALLY_DEFINED_TRANSFORMATION	FNDFTR
CARTESIAN_TRANSFORMATION_OPERATOR_2D	CTO2
CARTESIAN_TRANSFORMATION_OPERATOR_3D	CTO3
PLANAR_EXTENT	PLNEXT
DESCRIPTIVE_REPRESENTATION_ITEM	DSRPIT
COMPOUND_REPRESENTATION_ITEM	CMRPIT
VALUE_RANGE	VLRNG
MAPPED_ITEM	MPPITM
REPRESENTATION_MAP	RPRMP
MEASURE_REPRESENTATION_ITEM	MSRPIT
MEASURE_WITH_UNIT	MSWTUN
LENGTH_MEASURE_WITH_UNIT	LMWU
MASS_MEASURE_WITH_UNIT	MMWU
TIME_MEASURE_WITH_UNIT	TMWU
ELECTRIC_CURRENT_MEASURE_WITH_UNIT	ECMWU
THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT	TTMWU
CELSIUS_TEMPERATURE_MEASURE_WITH_UNIT	CTMWU
AMOUNT_OF_SUBSTANCE_MEASURE_WITH_UNIT	AOSMWU
LUMINOUS_INTENSITY_MEASURE_WITH_UNIT	LIMWU
PLANE_ANGLE_MEASURE_WITH_UNIT	PAMWU
SOLID_ANGLE_MEASURE_WITH_UNIT	SAMWU
AREA_MEASURE_WITH_UNIT	AMWU
VOLUME_MEASURE_WITH_UNIT	VMWU
RATIO_MEASURE_WITH_UNIT	RMWU
NAMED_UNIT	NMDUNT
SI_UNIT	SUNT
DIMENSIONAL_EXPONENTS	DMNEXP
CONVERSION_BASED_UNIT	CNBSUN
CONTEXT_DEPENDENT_UNIT	CNDPUN
EXPRESSION_CONVERSION_BASED_UNIT	ECBU
VARIABLE_SEMANTICS	VRBSMN
DERIVED_UNIT_VARIABLE	DRUNVR
DERIVED_UNIT	DRVUNT

DERIVED_UNIT_ELEMENT	DRUNEL
ENVIRONMENT	ENVRNM
GENERIC_VARIABLE	GNRVRB
SIMPLE_GENERIC_EXPRESSION	SMGNEX
GENERIC_EXPRESSION	GNREXP
UNARY_GENERIC_EXPRESSION	UNGNEX
BINARY_GENERIC_EXPRESSION	BNGNEX
MULTIPLE_ARITY_GENERIC_EXPRESSION	MAGE
NAMED_UNIT_VARIABLE	NMUNVR
LENGTH_UNIT	LNGUNT
MASS_UNIT	MSSUNT
TIME_UNIT	TMUNT
ELECTRIC_CURRENT_UNIT	ELCRUN
THERMODYNAMIC_TEMPERATURE_UNIT	THTMUN
AMOUNT_OF_SUBSTANCE_UNIT	AOSU
LUMINOUS_INTENSITY_UNIT	LMINUN
PLANE_ANGLE_UNIT	PLANUN
SOLID_ANGLE_UNIT	SLANUN
AREA_UNIT	ARUNT
VOLUME_UNIT	VLMUNT
RATIO_UNIT	RTUNT
QUALIFIED_REPRESENTATION_ITEM	QLRPIT
PRECISION_QUALIFIER	PRCQLF
TYPE_QUALIFIER	TYPQLF
VALUE_REPRESENTATION_ITEM	VLRPIT
REPRESENTATION_CONTEXT	RPRCNT
GEOMETRIC_REPRESENTATION_CONTEXT	GMRPCN
GLOBAL_UNIT_ASSIGNED_CONTEXT	GUAC
GROUP	GROUP
CLASS	CLASS
EXTERNALLY_DEFINED_CLASS	EXD0
EXTERNALLY_DEFINED_ITEM	EXDFIT
EXTERNALLY_DEFINED_GENERAL_PROPERTY	EDGP
GENERAL_PROPERTY	GNRPRP
EXTERNAL_SOURCE	EXTSRC
KNOWN_SOURCE	KNWSRC
PRE_DEFINED_ITEM	PRDFIT
CHARACTERIZED_CLASS	CHRCLS
CHARACTERIZED_OBJECT	CHROBJ
DOCUMENT_FILE	DCMFL
DOCUMENT	DCMNT
DOCUMENT_TYPE	DCMTYP
DOCUMENT_REPRESENTATION_TYPE	DCRPTY
FEATURE_DEFINITION	FTRDFN
GENERAL_FEATURE	GNRFTR
CLASS_SYSTEM	CLS1
LANGUAGE	LNGG
CLASSIFICATION_ROLE	CLSRL
PRODUCT	PRDCT
PRODUCT_CONTEXT	PRDCNT
APPLICATION_CONTEXT_ELEMENT	APCNEL
PRODUCT_DEFINITION_CONTEXT	PRDFCN
APPLICATION_CONTEXT	APPCNT
PRODUCT_DEFINITION	PRDDFN
PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS	PDWAD
PRODUCT_DEFINITION_FORMATION	PRDFFR
PRODUCT_DEFINITION_RELATIONSHIP	PRDFRL
PRODUCT_DEFINITION_USAGE	PRDFUS
ASSEMBLY_COMPONENT_USAGE	ASCMUS
QUANTIFIED_ASSEMBLY_COMPONENT_USAGE	QACU
MATERIAL_DESIGNATION	MTRDSG

PRODUCT_DEFINITION_SHAPE	PRDFSH
PROPERTY_DEFINITION	PRPDFN
SHAPE_ASPECT	SHPASP
PLUS_MINUS_TOLERANCE	PLMNTL
TOLERANCE_VALUE	TLRVL
LIMITS_AND_FITS	LMANFT
DIMENSIONAL_SIZE	DMNSZ
APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT	APAOA
PERSON_AND_ORGANIZATION_ASSIGNMENT	PAOA
PERSON_AND_ORGANIZATION	PRANOR
PERSON	PERSON
ORGANIZATION	ORGNZT
PERSON_AND_ORGANIZATION_ROLE	PAOR
PRODUCT_DEFINITION_FORMATION_RELATIONSHIP	PDFR
PRODUCT_RELATED_PRODUCT_CATEGORY	PRPC
PRODUCT_CATEGORY	PRDCTG
EFFECTIVITY	EFFCTV
DATED_EFFECTIVITY	DTDEFF
DATE	DATE
CALENDAR_DATE	CLNDT
LOCAL_TIME	LCLTM
COORDINATED_UNIVERSAL_TIME_OFFSET	CUTO
DATE_AND_TIME	DTANTM
TIME_INTERVAL_BASED_EFFECTIVITY	TIBE
TIME_INTERVAL	TMINT
TIME_INTERVAL_WITH_BOUNDS	TIWB
EFFECTIVITY_RELATIONSHIP	EFFRLT
GENERAL_PROPERTY_RELATIONSHIP	GNPRRL
REPRESENTATION_RELATIONSHIP	RPRRLT
SHAPE_REPRESENTATION_RELATIONSHIP	SHRPRL
REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION	RRWT
ITEM_DEFINED_TRANSFORMATION	ITDFTR
IDENTIFICATION_ROLE	IDNRL
ATTRIBUTE_VALUE_ASSIGNMENT	ATVLAS
MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT	MLAA
ORGANIZATION_ROLE	ORGRL
ATTRIBUTE_VALUE_ROLE	ATVLRL
APPLIED_ORGANIZATION_ASSIGNMENT	APORAS
ORGANIZATION_ASSIGNMENT	ORGASS
APPLIED_IDENTIFICATION_ASSIGNMENT	APIDAS
IDENTIFICATION_ASSIGNMENT	IDNASS
EXTERNAL_IDENTIFICATION_ASSIGNMENT	EXIDAS
APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT	AEIA
CONTEXT_DEPENDENT_SHAPE_REPRESENTATION	CDSR
PROPERTY_DEFINITION_REPRESENTATION	PRDFRP
SHAPE_DEFINITION_REPRESENTATION	SHDFRP
APPLIED_LOCATION_ASSIGNMENT	APLCAS
LOCATION_ASSIGNMENT	LCTASS
LOCATION	LCTN
LOCATION_ROLE	LCTRL
APPLIED_STATE_TYPE_ASSIGNMENT	ASTA
STATE_TYPE	STTTYP
STATE_TYPE_ROLE	STT0
ADDRESS	ADDRSS
ORGANIZATIONAL_ADDRESS	ORGADD
PERSON_AND_ORGANIZATION_ADDRESS	PAO0
PERSONAL_ADDRESS	PRSADD
EFFECTIVITY_ASSIGNMENT	EFFASS
APPLIED_EFFECTIVITY_ASSIGNMENT	APEFAS
APPLIED_INEFFECTIVITY_ASSIGNMENT	APINAS
OBJECT_ROLE	OBJRL

APPLIED_DOCUMENT_REFERENCE	APDCRF
DOCUMENT_REFERENCE	DCMRFR
ATTRIBUTE_CLASSIFICATION_ASSIGNMENT	ATCLAS
ATTRIBUTE_LANGUAGE_ASSIGNMENT	ATLNAS
DESCRIPTION_ATTRIBUTE	DSCATT
DIMENSIONAL_CHARACTERISTIC_REPRESENTATION	DMCHRP
DOCUMENT_PRODUCT_ASSOCIATION	DCP1
DOCUMENT_PRODUCT_EQUIVALENCE	DCPREQ
EXTERNALLY_DEFINED_ITEM_RELATIONSHIP	EDIR
GENERAL_PROPERTY_ASSOCIATION	GNPRAS
GROUP_RELATIONSHIP	GRPRLT
ID_ATTRIBUTE	IDATT
MEASURE_QUALIFICATION	MSRQLF
NAME_ATTRIBUTE	NMATT
PRODUCT_CATEGORY_RELATIONSHIP	PRCTRL
PRODUCT_DEFINITION_CONTEXT_ASSOCIATION	PDCA
PRODUCT_DEFINITION_CONTEXT_ROLE	PDCR
PRODUCT_DEFINITION_OCCURRENCE_RELATIONSHIP	PDOR
ROLE_ASSOCIATION	RLASS

Annex C
(normative)

ARM EXPRESS listing

```
SCHEMA CUTTING_TOOL_SCHEMA;

  TYPE string_select = SELECT
    (default_language_string,
     multi_language_string);
  END_TYPE;

  TYPE default_language_string = STRING;
  END_TYPE;

  TYPE limitation_definition_select = SELECT
    (limits_and_fits,
     plus_minus_bounds);
  END_TYPE;

  TYPE classification_source_select = SELECT
    (plib_class_reference,
     document,
     external_library_reference);
  END_TYPE;

  TYPE property_source_select = SELECT
    (plib_property_reference,
     external_library_reference,
     document);
  END_TYPE;

  TYPE classified_element_select = SELECT
    (item_version,
     item,
     item_definition,
     document,
     property,
     property_value_association,
     item_instance,
     document_file,
     document_version,
     document_representation,
     material_designation);
  END_TYPE;

  TYPE item_property_select = SELECT
    (item_definition,
     item_characteristic_select,
     document_file,
     document_representation,
     item_structure_association,
     item_instance,
     mated_item_relationship,
     physical_item_structure_association);
```

```
END_TYPE;

TYPE item_characteristic_select = SELECT
  (cutting_condition,
   coupling,
   workpiece_feature,
   material_designation,
   cutting_data_association,
   grade);
END_TYPE;

TYPE alias_select = SELECT
  (item,
   item_version,
   item_definition,
   property,
   grade,
   classification_attribute,
   classification_system,
   document,
   document_representation,
   document_version,
   general_classification,
   item_instance,
   organization,
   document_type_property);
END_TYPE;

TYPE documented_element_select = SELECT
  (item,
   item_version,
   classification_attribute,
   classification_system,
   item_definition,
   general_classification,
   item_structure_association,
   item_instance,
   organization,
   person,
   property,
   specific_item_classification,
   material_designation,
   mated_item_relationship,
   physical_item_structure_association);
END_TYPE;

TYPE assigned_document_select = SELECT
  (document_version,
   document);
END_TYPE;

TYPE effective_element_select = SELECT
  (document,
   classification_system,
   document_file,
   document_representation,
   document_version,
   item,
   item_version,
   item_instance,
   property,
   property_value_association,
   material_designation,
   item_structure_association,
   realized_item_association,
   physical_item_state_association,
   physical_item_location_association,
   physical_item_structure_association);
END_TYPE;

TYPE person_organization_select = SELECT
  (organization,
   person_in_organization);
END_TYPE;

TYPE general_organizational_data_select = SELECT
  (classification_association,
   classification_system,
   item_definition,
   document,
   document_file,
   document_representation,
   document_version,
   general_classification,
   item,
   item_structure_association,
   item_instance,
   item_version,
   item_version_relationship,
   person_in_organization,
   property,
   property_value_association,
   material_designation,
   physical_item_structure_association);
END_TYPE;

ENTITY item_version;
    associated_item : item;
    description     : OPTIONAL string_select;
    id              : STRING;
END_ENTITY;

ENTITY physical_item_version
  SUBTYPE OF(item_version);
    SELF\item_version.associated_item : physical_item;
END_ENTITY;

ENTITY item;
    description         : OPTIONAL string_select;
    id                  : STRING;
    name                : string_select;
  INVERSE
    associated_version  : SET [1:?] OF item_version FOR associated_item;
    item_classification : SET [1:?] OF specific_item_classification FOR
associated_item;
END_ENTITY;
```

```
ENTITY physical_item
  SUBTYPE OF(item);
  INVERSE
    SELF\item.associated_version : SET [1:?] OF physical_item_version FOR
associated_item;
END_ENTITY;

ENTITY multi_language_string;
    additional_language_string : SET OF string_with_language;
    primary_language_string    : string_with_language;
END_ENTITY;

ENTITY string_with_language;
    contents               : STRING;
    language_specification : language;
  INVERSE
    used_by                : SET [1:?] OF multi_language_string FOR
primary_language_string;
END_ENTITY;

ENTITY language;
    country_code  : OPTIONAL STRING;
    language_code : STRING;
END_ENTITY;

ENTITY specific_item_classification;
    associated_item     : SET [1:?] OF item;
    classification_name : STRING;
    description         : OPTIONAL string_select;
END_ENTITY;

ENTITY item_definition
  SUPERTYPE OF (ONEOF(assembly_definition, mating_definition,
physical_item_definition));
    additional_context      : SET OF application_context;
    associated_item_version : item_version;
    id                      : STRING;
    initial_context         : application_context;
    name                    : OPTIONAL string_select;
END_ENTITY;

ENTITY assembly_definition
  SUBTYPE OF(item_definition);
    assembly_type : OPTIONAL STRING;
END_ENTITY;

ENTITY mating_definition
  SUBTYPE OF(item_definition);
    mating_type : STRING;
  INVERSE
    mated_items : SET [2:?] OF mating_association FOR relating;
END_ENTITY;

ENTITY item_definition_relationship
    description   : OPTIONAL string_select;
    related       : item_definition;
    relating      : item_definition;
```

```
     relation_type   : STRING;
END_ENTITY;

ENTITY mating_association
  SUBTYPE OF(item_structure_association);
    SELF\item_structure_association.relating : mating_definition;
END_ENTITY;

ENTITY item_structure_association
  ABSTRACT SUPERTYPE OF (ONEOF(mating_association,
assembly_association));
    placement : OPTIONAL
geometric_model_relationship_with_transformation;
    related   : item_instance;
    relating  : item_definition;
END_ENTITY;

ENTITY assembly_association
  SUBTYPE OF(item_structure_association);
    SELF\item_structure_association.relating : assembly_definition;
END_ENTITY;

ENTITY geometric_model_relationship_with_transformation;
    description     : OPTIONAL string_select;
    model_placement : transformation;
    related         : external_model;
    relating        : external_model;
    relation_type   : STRING;
END_ENTITY;

ENTITY transformation
  ABSTRACT SUPERTYPE OF (ONEOF(transformation_2d, transformation_3d));
END_ENTITY;

ENTITY transformation_2d
  SUBTYPE OF(transformation);
END_ENTITY;

ENTITY transformation_3d
  SUBTYPE OF(transformation);
END_ENTITY;

ENTITY external_model
  ABSTRACT SUPERTYPE OF (ONEOF(external_picture,
external_geometric_model));
    description   : OPTIONAL string_select;
    is_defined_as : digital_file;
    is_defined_in : cartesian_coordinate_space;
    model_id      : STRING;
END_ENTITY;

ENTITY external_picture
  SUBTYPE OF(external_model);
    SELF\external_model.is_defined_in : cartesian_coordinate_space_2d;
END_ENTITY;

ENTITY cartesian_coordinate_space
```

```
  ABSTRACT SUPERTYPE OF (ONEOF(cartesian_coordinate_space_2d,
cartesian_coordinate_space_3d));
    unit_of_values : OPTIONAL SET [2:?] OF unit;
END_ENTITY;

ENTITY cartesian_coordinate_space_2d
  SUBTYPE OF(cartesian_coordinate_space);
END_ENTITY;

ENTITY cartesian_coordinate_space_3d
  SUBTYPE OF(cartesian_coordinate_space);
END_ENTITY;

ENTITY unit;
    unit_name : STRING;
END_ENTITY;

ENTITY external_geometric_model
  SUBTYPE OF(external_model);
    model_extent : OPTIONAL STRING;
END_ENTITY;

ENTITY digital_file
  SUBTYPE OF(document_file);
  INVERSE
    associated_model_space : SET OF external_model FOR is_defined_as;
END_ENTITY;

ENTITY document_file
  ABSTRACT SUPERTYPE OF (ONEOF(digital_file, physical_file));
    content                   : OPTIONAL document_content_property;
    creation                  : OPTIONAL document_creation_property;
    document_file_type        : OPTIONAL document_type_property;
    external_id_and_location : SET OF external_file_id_and_location;
    file_format               : OPTIONAL document_format_property;
    file_id                   : STRING;
    size                      : OPTIONAL document_size_property;
    version_id                : OPTIONAL STRING;
END_ENTITY;

ENTITY physical_file
  SUBTYPE OF(document_file);
END_ENTITY;

ENTITY document_content_property;
    detail_level     : OPTIONAL STRING;
    geometry_type    : OPTIONAL STRING;
    languages        : SET OF language;
    real_world_scale : OPTIONAL numerical_value;
END_ENTITY;

ENTITY numerical_value
  SUBTYPE OF(value_with_unit);
    value_component : STRING;
  INVERSE
    limitation      : SET [0:1] OF value_limitation FOR limited_value;
END_ENTITY;
```

```
ENTITY value_with_unit
  ABSTRACT SUPERTYPE OF (ONEOF(value_limit, numerical_value,
value_range))
  SUBTYPE OF(property_value);
    significant_digits : OPTIONAL STRING;
    unit_component     : OPTIONAL unit;
END_ENTITY;

ENTITY property_value
  ABSTRACT SUPERTYPE OF (ONEOF(value_with_unit, string_value,
value_list));
    value_name : STRING;
END_ENTITY;

ENTITY string_value
  SUBTYPE OF(property_value);
    value_specification : string_select;
END_ENTITY;

ENTITY value_list
  SUBTYPE OF(property_value);
    values : LIST [1:?] OF property_value;
END_ENTITY;

ENTITY value_limit
  SUBTYPE OF(value_with_unit);
    limit           : STRING;
    limit_qualifier : STRING;
END_ENTITY;

ENTITY value_range
  SUBTYPE OF(value_with_unit);
    lower_limit : STRING;
    upper_limit : STRING;
END_ENTITY;

ENTITY value_limitation;
    envelope      : OPTIONAL BOOLEAN;
    is_defined_by : limitation_definition_select;
    limited_value : numerical_value;
END_ENTITY;

ENTITY limits_and_fits;
    deviation    : STRING;
    fitting_type : OPTIONAL STRING;
    grade        : STRING;
END_ENTITY;

ENTITY plus_minus_bounds;
    lower_bound         : STRING;
    significant_digits  : OPTIONAL STRING;
    upper_bound         : STRING;
    value_determination : OPTIONAL STRING;
END_ENTITY;

ENTITY document_creation_property;
    creating_interface : OPTIONAL STRING;
    creating_system    : STRING;
    operating_system   : OPTIONAL STRING;
END_ENTITY;

ENTITY document_type_property;
    document_type_name         : STRING;
    used_classification_system : OPTIONAL classification_system;
END_ENTITY;

ENTITY classification_system;
    description            : OPTIONAL string_select;
    id                     : STRING;
  INVERSE
    allowed_classification : SET OF general_classification FOR
 used_classification_system;
END_ENTITY;

ENTITY general_classification;
    classification_source      : OPTIONAL classification_source_select;
    description                : OPTIONAL string_select;
    id                         : STRING;
    used_classification_system : OPTIONAL classification_system;
    version_id                 : OPTIONAL STRING;
END_ENTITY;

ENTITY plib_class_reference;
    code         : STRING;
    supplier_bsu : STRING;
    version      : STRING;
END_ENTITY;

ENTITY document;
    description        : OPTIONAL string_select;
    document_id        : STRING;
    name               : string_select;
  INVERSE
    associated_version : SET [1:?] OF document_version FOR
 associated_document;
END_ENTITY;

ENTITY document_version;
    associated_document : document;
    description         : OPTIONAL string_select;
    id                  : STRING;
END_ENTITY;

ENTITY external_library_reference;
    description  : OPTIONAL string_select;
    external_id  : STRING;
    library_type : STRING;
END_ENTITY;

ENTITY din4000_reference
  SUBTYPE OF(external_library_reference);
    characteristics_code_no : STRING;
    part_no                 : STRING;
END_ENTITY;

ENTITY external_file_id_and_location;
```

```
        external_id : OPTIONAL STRING;
        location    : document_location_property;
    END_ENTITY;

    ENTITY document_location_property;
        location_name : STRING;
    END_ENTITY;

    ENTITY document_format_property;
        character_code : OPTIONAL STRING;
        data_format    : OPTIONAL STRING;
        size_format    : OPTIONAL rectangular_size;
    END_ENTITY;

    ENTITY rectangular_size;
        density : OPTIONAL value_with_unit;
        height  : value_with_unit;
        width   : value_with_unit;
    END_ENTITY;

    ENTITY named_size
      SUBTYPE OF(rectangular_size);
        referenced_standard : OPTIONAL classification_system;
        size                : STRING;
    END_ENTITY;

    ENTITY document_size_property;
        file_size  : OPTIONAL value_with_unit;
        page_count : OPTIONAL value_with_unit;
    END_ENTITY;

    ENTITY item_instance;
        definition  : item_definition;
        description : OPTIONAL string_select;
        id          : STRING;
    END_ENTITY;

    ENTITY quantified_instance
      SUBTYPE OF(item_instance);
        quantity : numerical_value;
    END_ENTITY;

    ENTITY physical_item_definition
      SUBTYPE OF(item_definition);
        SELF\item_definition.associated_item_version : physical_item_version;
    END_ENTITY;

    ENTITY application_context;
        application_domain : STRING;
        description        : OPTIONAL string_select;
    END_ENTITY;

    ENTITY organization;
        delivery_address  : OPTIONAL STRING;
        id                : STRING;
        organization_name : STRING;
        organization_type : OPTIONAL STRING;
        postal_address    : OPTIONAL STRING;
        visitor_address   : OPTIONAL STRING;
    END_ENTITY;

    ENTITY workpiece_feature;
        description    : OPTIONAL string_select;
        id             : STRING;
        representation : SET [1:?] OF external_model;
    END_ENTITY;

    ENTITY material_designation;
        material_name : STRING;
    END_ENTITY;

    ENTITY coupling;
        coupling_type : STRING;
        pieces        : OPTIONAL numerical_value;
        side          : STRING;
        size          : numerical_value;
        style         : STRING;
    END_ENTITY;

    ENTITY cutting_condition;
        condition_name : STRING;
    END_ENTITY;

    ENTITY property;
        allowed_unit    : SET OF unit;
        description     : OPTIONAL string_select;
        id              : STRING;
        property_source : OPTIONAL property_source_select;
        property_type   : STRING;
        version_id      : OPTIONAL STRING;
    END_ENTITY;

    ENTITY plib_property_reference;
        code       : STRING;
        name_scope : plib_class_reference;
        version    : STRING;
    END_ENTITY;

    ENTITY cutting_data_association;
        associated_material : material_designation;
        depth_of_cut        : value_with_unit;
        feed                : value_with_unit;
        speed               : value_with_unit;
    END_ENTITY;

    ENTITY classification_association;
        associated_classification : general_classification;
        classified_element        : classified_element_select;
        defintional               : OPTIONAL BOOLEAN;
        role                      : OPTIONAL STRING;
    END_ENTITY;

    ENTITY property_value_association;
        definitional              : OPTIONAL BOOLEAN;
        described_element         : item_property_select;
        describing_property_value : property_value_representation;
        description                : OPTIONAL string_select;
        validity_context           : OPTIONAL organization;
    END_ENTITY;

    ENTITY grade;
        coating              : OPTIONAL coating;
        cutting_condition    : SET [1:?] OF cutting_condition;
        identifier           : STRING;
        standard_designation : OPTIONAL STRING;
        substrate            : substrate;
        workpiece_material   : SET [1:?] OF material_designation;
    END_ENTITY;

    ENTITY coating;
        coating_name    : STRING;
        coating_process : STRING;
    END_ENTITY;

    ENTITY substrate;
        name : STRING;
    END_ENTITY;

    ENTITY document_representation
      ABSTRACT SUPERTYPE OF (ONEOF(digital_document, physical_document));
        associated_document_version : document_version;
        common_location             : SET OF document_location_property;
        content                     : OPTIONAL document_content_property;
        creation                    : OPTIONAL document_creation_property;
        description                 : OPTIONAL string_select;
        id                          : STRING;
        representation_format       : OPTIONAL document_format_property;
        size                        : OPTIONAL document_size_property;
    END_ENTITY;

    ENTITY digital_document
      SUBTYPE OF(document_representation);
        file : SET OF digital_file;
    END_ENTITY;

    ENTITY physical_document
      SUBTYPE OF(document_representation);
        file : SET OF physical_file;
    END_ENTITY;

    ENTITY mated_item_relationship;
        mating_material : SET OF quantified_instance;
        related         : mating_association;
        relating        : mating_association;
    END_ENTITY;

    ENTITY physical_item_structure_association;
        related  : physical_item_definition;
        relating : physical_item_definition;
    END_ENTITY;

    ENTITY property_value_representation;
        definition          : property;
        global_unit         : OPTIONAL unit;
        qualifier           : OPTIONAL STRING;
        specified_value     : property_value;
        value_determination : OPTIONAL STRING;
    END_ENTITY;

    ENTITY classification_attribute;
        allowed_value             : SET OF property_value_representation;
        associated_classification : general_classification;
        attribute_definition      : property;
        description               : OPTIONAL string_select;
        id                        : STRING;
        name                      : OPTIONAL string_select;
    END_ENTITY;

    ENTITY general_classification_hierarchy;
        sub_classification   : general_classification;
        super_classification : general_classification;
    END_ENTITY;

    ENTITY item_version_relationship;
        description   : OPTIONAL string_select;
        related       : item_version;
        relating      : item_version;
        relation_type : STRING;
    END_ENTITY;

    ENTITY specific_item_classification_hierarchy;
        sub_classification   : specific_item_classification;
        super_classification : specific_item_classification;
    END_ENTITY;

    ENTITY alias_identification;
        alias_id         : STRING;
        alias_scope      : OPTIONAL organization;
        alias_version_id : OPTIONAL STRING;
        description      : OPTIONAL string_select;
        is_applied_to    : alias_select;
    END_ENTITY;

    ENTITY property_relationship;
        description   : OPTIONAL string_select;
        related       : property;
        relating      : property;
        relation_type : STRING;
    END_ENTITY;

    ENTITY item_characteristic_association;
        associated_characteristic : item_characteristic_select;
        associated_item           : item_definition;
    END_ENTITY;

    ENTITY property_value_representation_relationship;
        description   : OPTIONAL string_select;
        related       : property_value_representation;
        relating      : property_value_representation;
        relation_type : STRING;
    END_ENTITY;
```

```
ENTITY classification_association_relationship;
    related           : classification_association;
    relating          : classification_association;
    relationship_type : STRING;
END_ENTITY;

ENTITY person;
    person_name                : STRING;
    preferred_business_address : OPTIONAL STRING;
  INVERSE
    associated_organization    : SET [1:?] OF person_in_organization FOR
associated_person;
END_ENTITY;

ENTITY person_in_organization;
    associated_organization : organization;
    associated_person       : person;
    id                      : OPTIONAL STRING;
    location                : OPTIONAL STRING;
    role                    : STRING;
END_ENTITY;

ENTITY document_assignment;
    assigned_document : assigned_document_select;
    is_assigned_to    : documented_element_select;
    role              : STRING;
END_ENTITY;

ENTITY effectivity;
    concerned_organization : SET OF organization;
    description            : OPTIONAL string_select;
    effectivity_context    : OPTIONAL STRING;
    end_definition         : OPTIONAL date_time;
    id                     : OPTIONAL STRING;
    period                 : OPTIONAL duration;
    start_definition       : OPTIONAL date_time;
    version_id             : OPTIONAL STRING;
END_ENTITY;

ENTITY date_time;
    date : STRING;
    time : OPTIONAL STRING;
END_ENTITY;

ENTITY duration;
    time      : STRING;
    time_unit : STRING;
END_ENTITY;

ENTITY effectivity_assignment;
    assigned_effectivity   : effectivity;
    effective_element      : effective_element_select;
    effectivity_indication : BOOLEAN;
    role                   : STRING;
END_ENTITY;

ENTITY realized_item_association;
    physical_item : physical_item;
```

```
      realized_item : item;
  END_ENTITY;

  ENTITY physical_item_state_association;
      associated_physical_item : physical_item_definition;
      associated_state         : state;
      role                     : STRING;
  END_ENTITY;

  ENTITY state;
      name : STRING;
  END_ENTITY;

  ENTITY physical_item_location_association;
      located_item : physical_item_definition;
      location     : location;
  END_ENTITY;

  ENTITY location;
      location_id   : STRING;
      location_name : OPTIONAL STRING;
      location_type : STRING;
  END_ENTITY;

  ENTITY effectivity_relationship;
      description   : OPTIONAL string_select;
      related       : effectivity;
      relating      : effectivity;
      relation_type : STRING;
  END_ENTITY;

  ENTITY person_organization_assignment;
      assigned_person_organization : person_organization_select;
      description                  : OPTIONAL string_select;
      is_applied_to                : SET [1:?] OF
  general_organizational_data_select;
      role                         : STRING;
  END_ENTITY;

  ENTITY document_version_relationship;
      description   : OPTIONAL string_select;
      related       : document_version;
      relating      : document_version;
      relation_type : STRING;
  END_ENTITY;

  ENTITY grade_relationship;
      related       : grade;
      relating      : grade;
      relation_type : STRING;
  END_ENTITY;

END_SCHEMA;
```

Annex D
(informative)

ARM EXPRESS-G diagrams

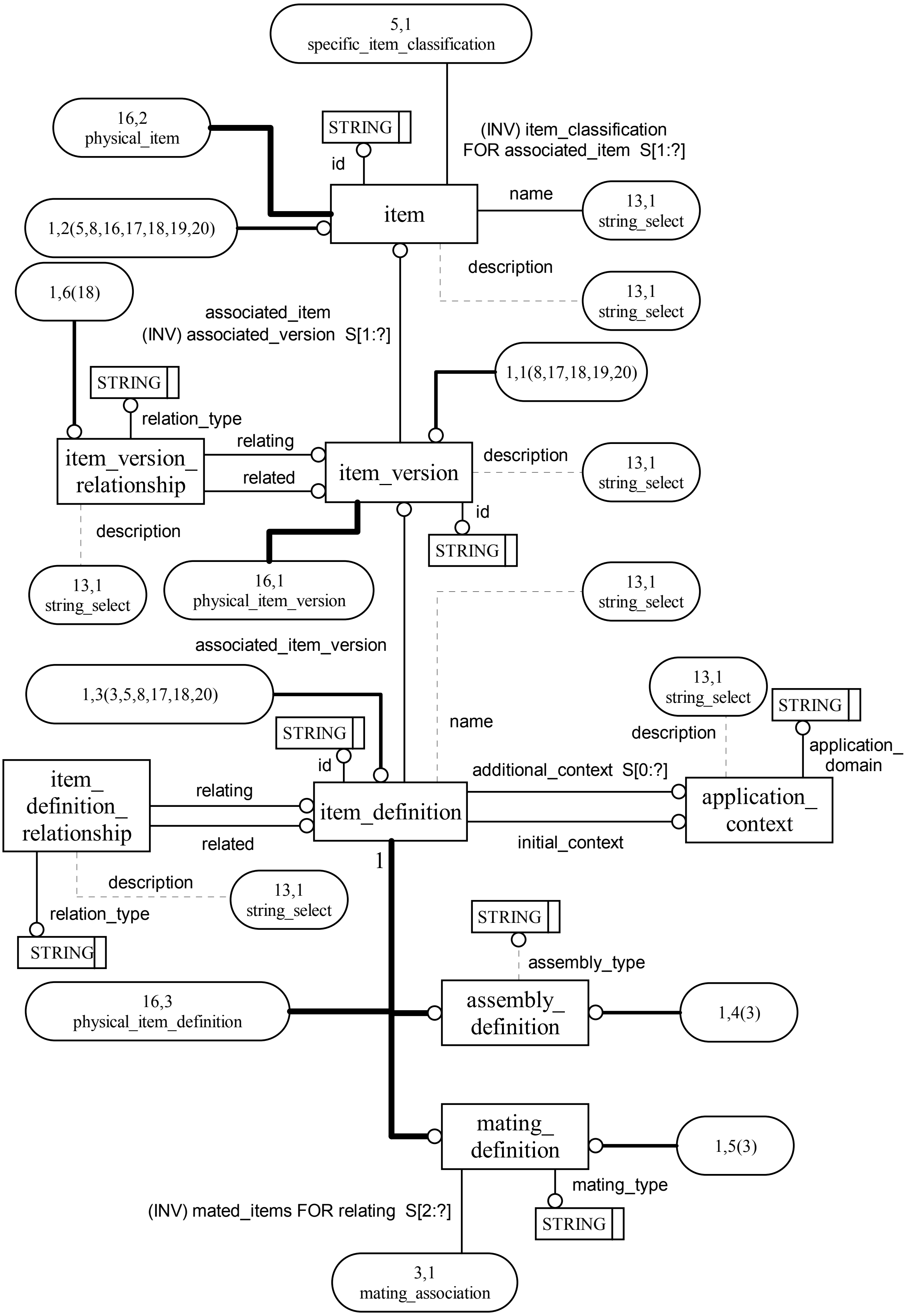

Figure D.1 — ISO 13999-1 EXPRESS-G diagram 1 of 20

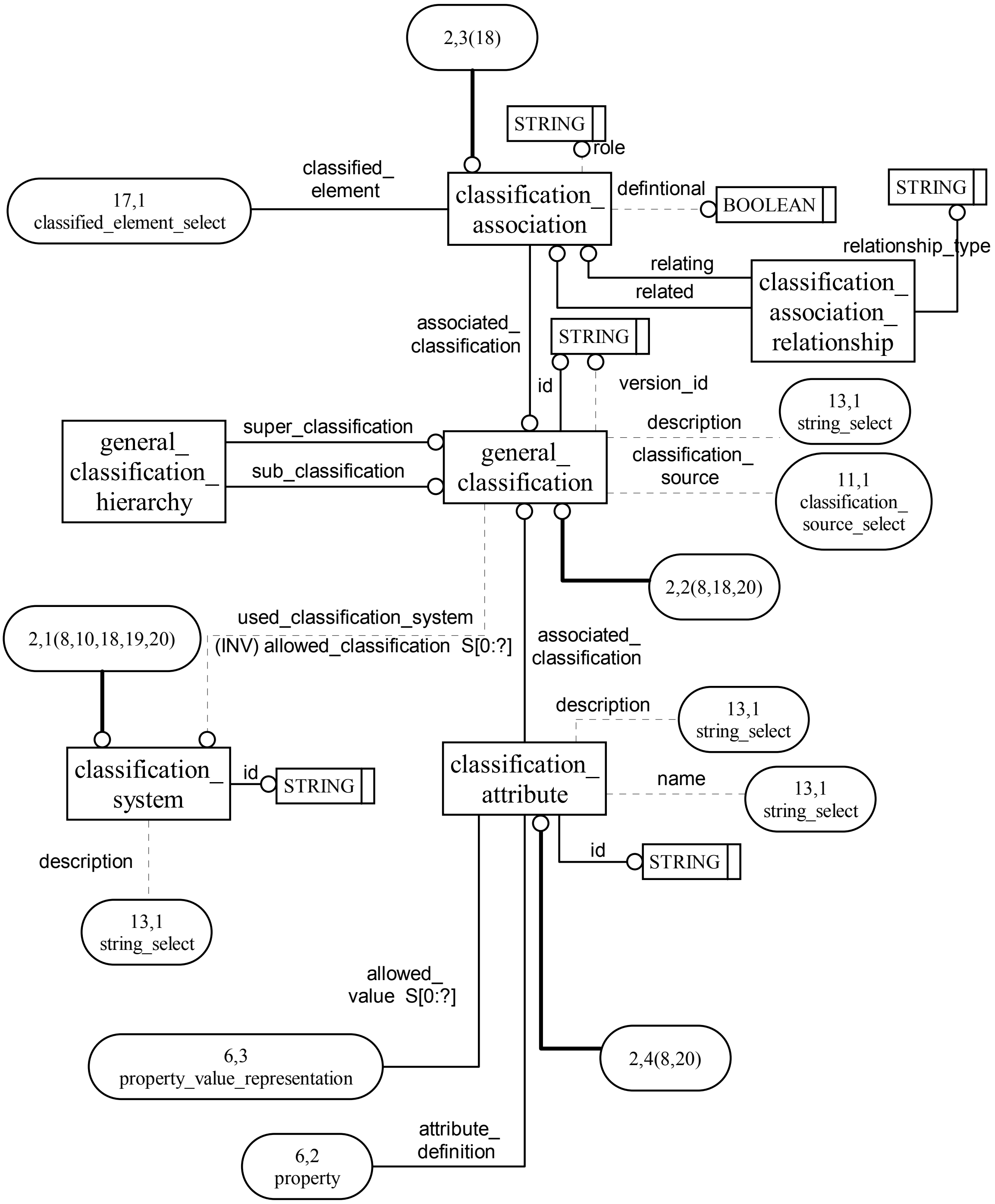

Figure D.2 — ISO13399-1 EXPRESS-G diagram 2 of 20

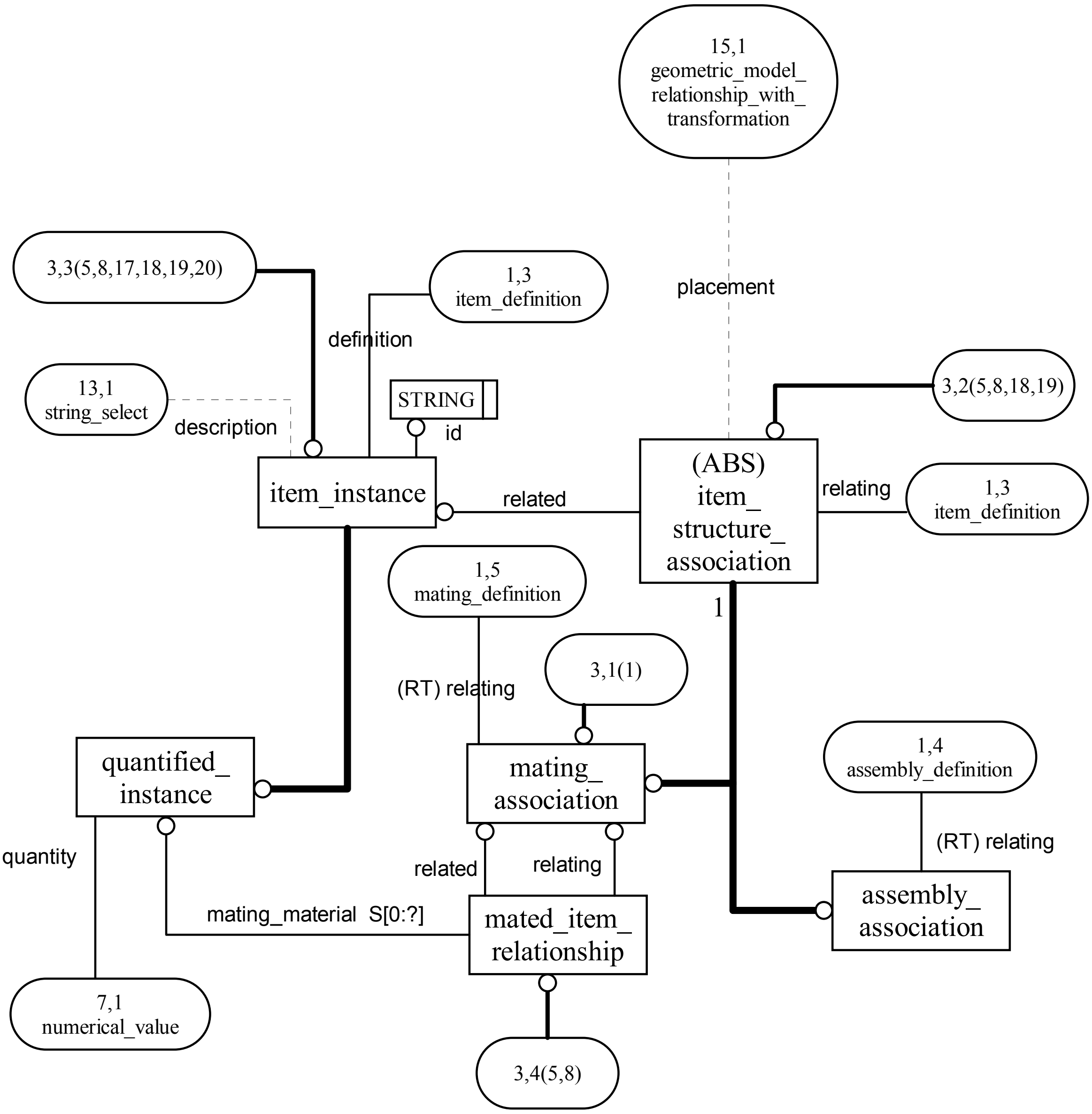

Figure D.3 — ISO13399-1 EXPRESS-G diagram 3 of 20

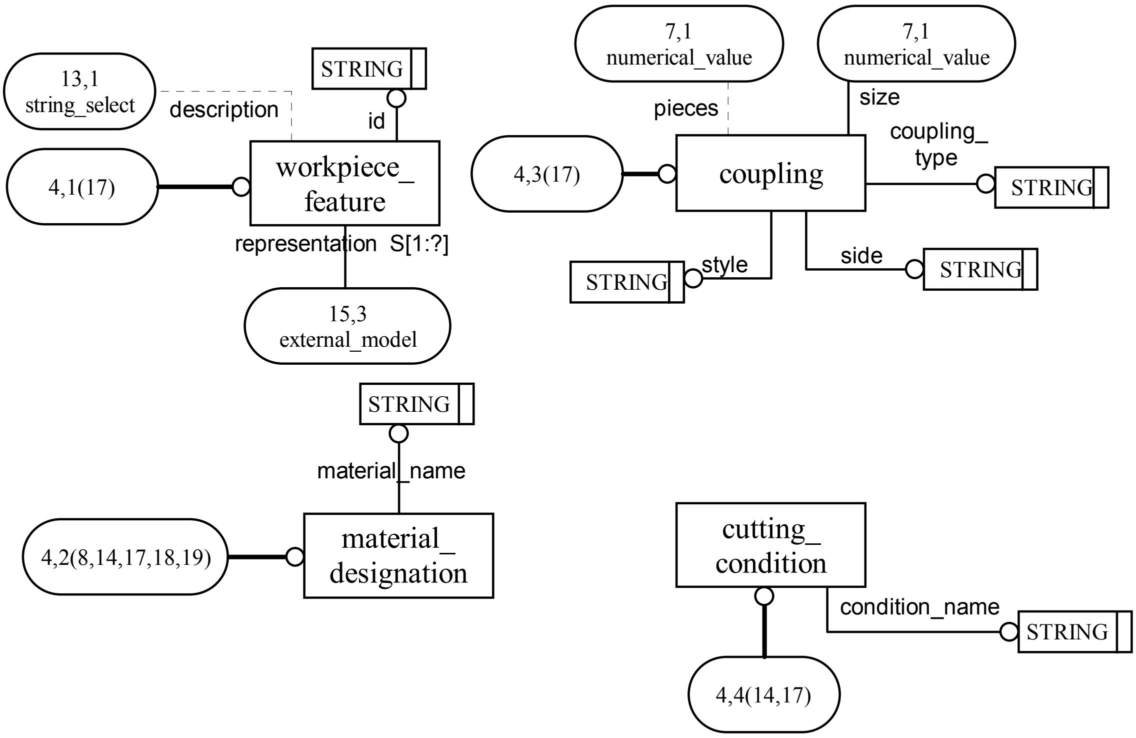

Figure D.4 — ISO13399-1 EXPRESS-G diagram 4 of 20

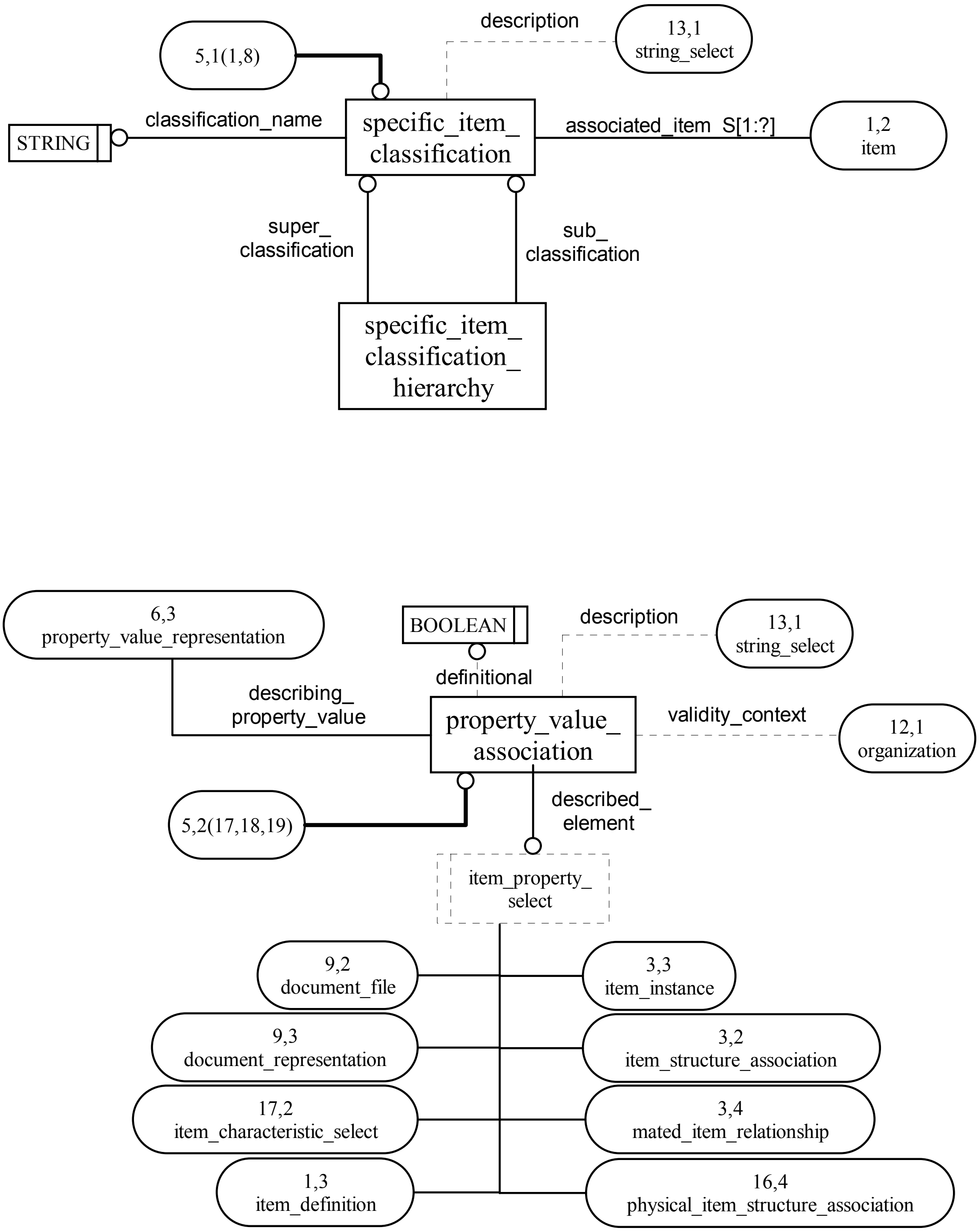

Figure D.5 — ISO13399-1 EXPRESS-G diagram 5 of 20

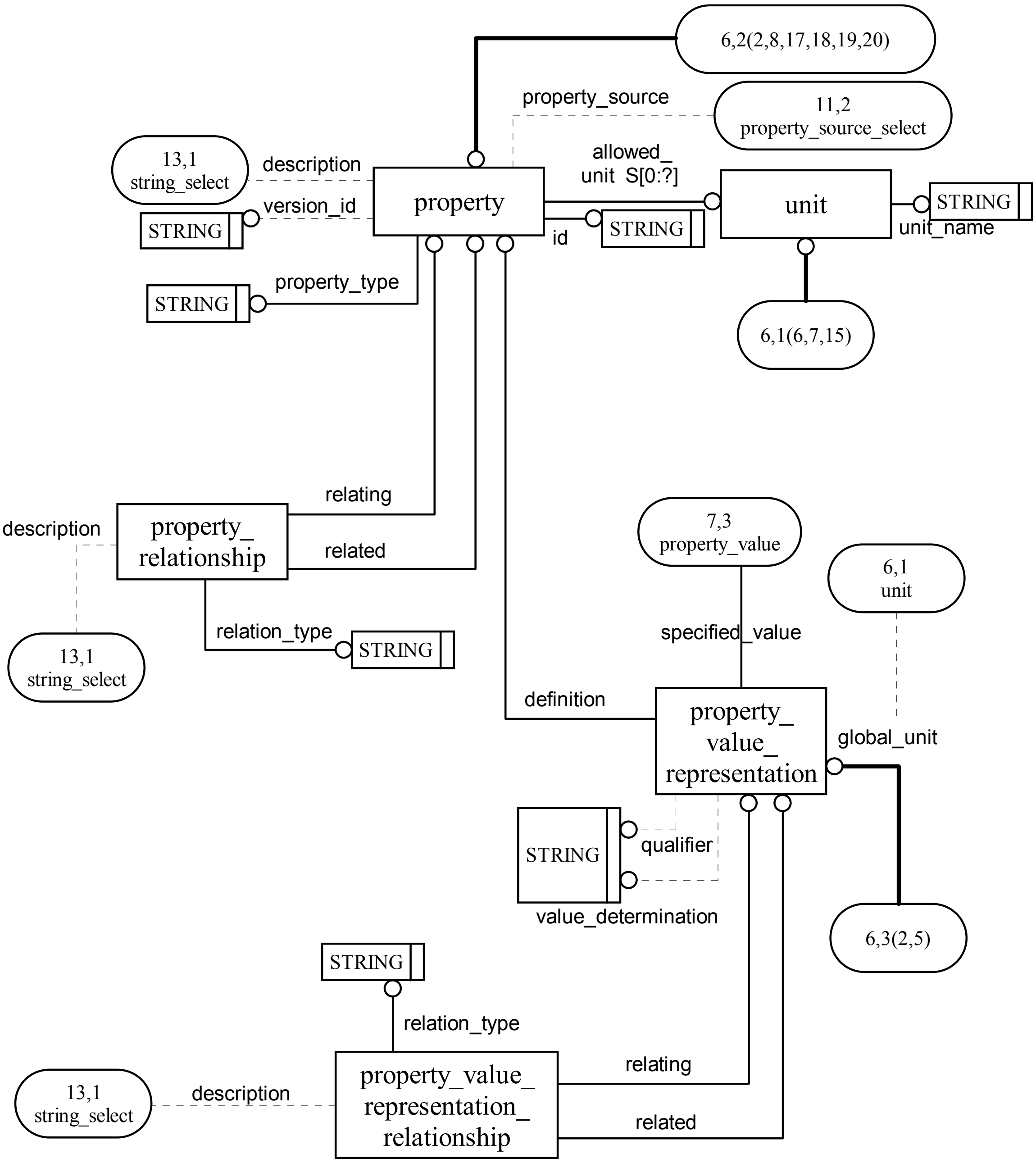

Figure D.6 — ISO13399-1 EXPRESS-G diagram 6 of 20

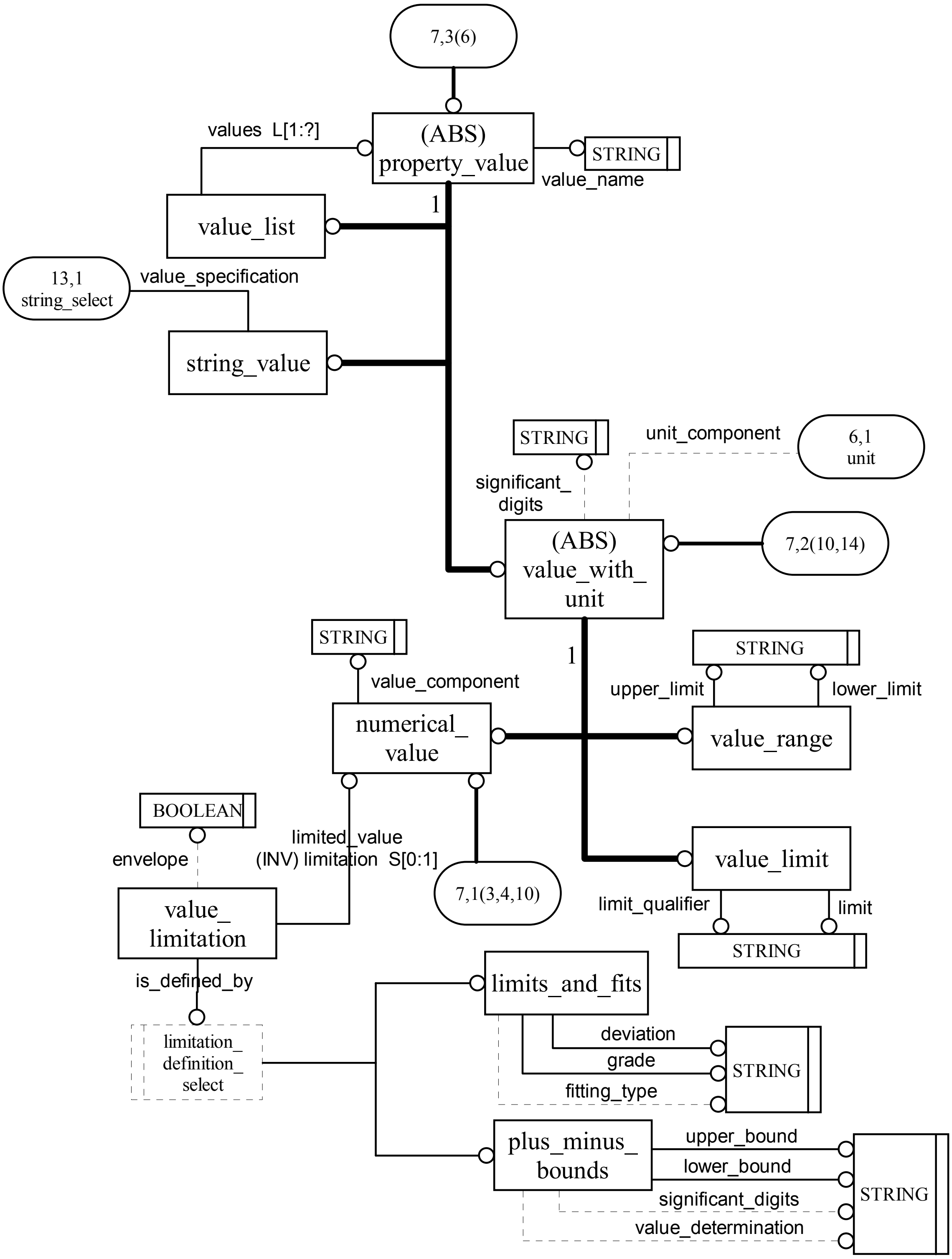

Figure D.7 — ISO13399-1 EXPRESS-G diagram 7 of 20

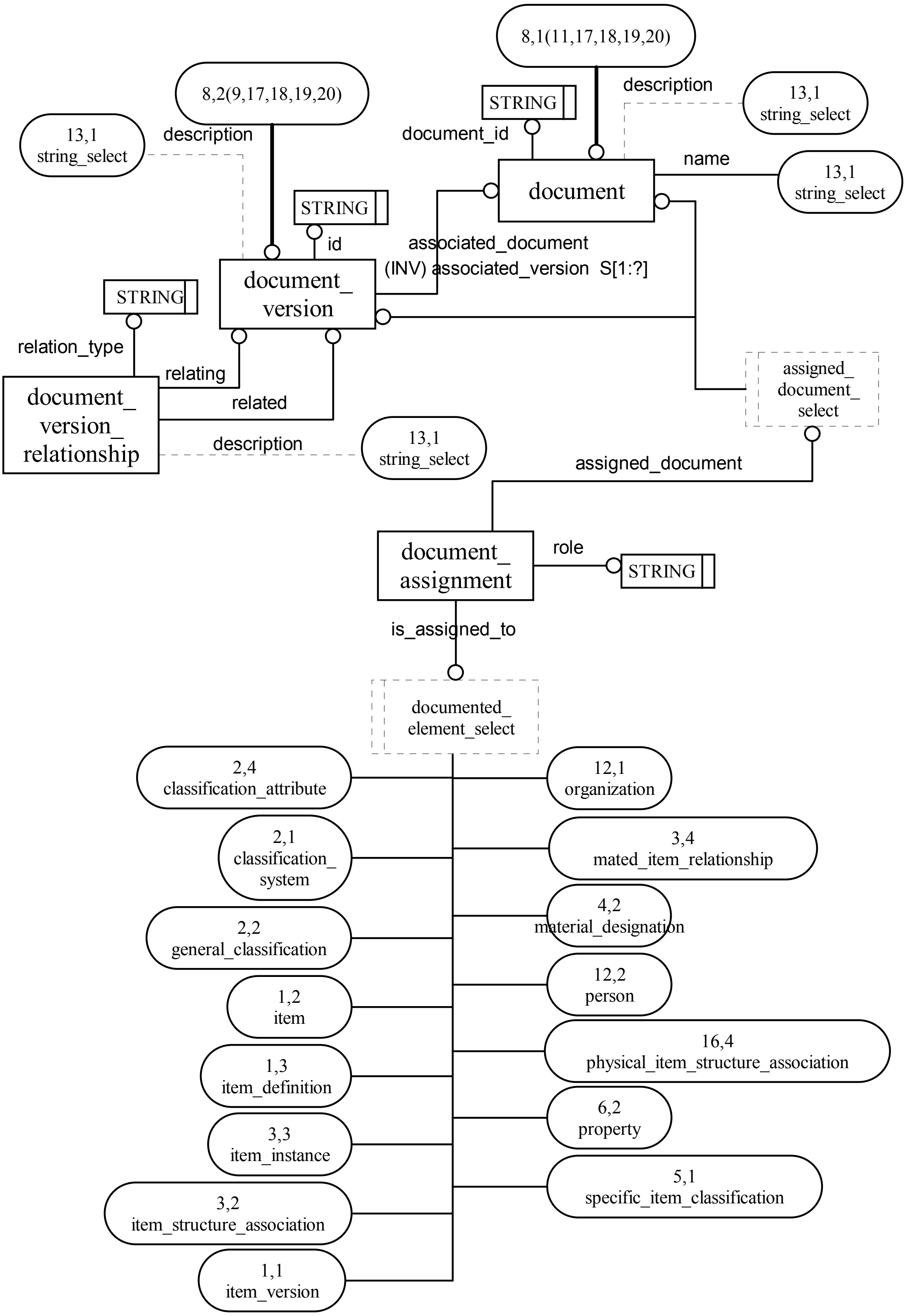

Figure D.8 — ISO13399-1 EXPRESS-G diagram 8 of 20

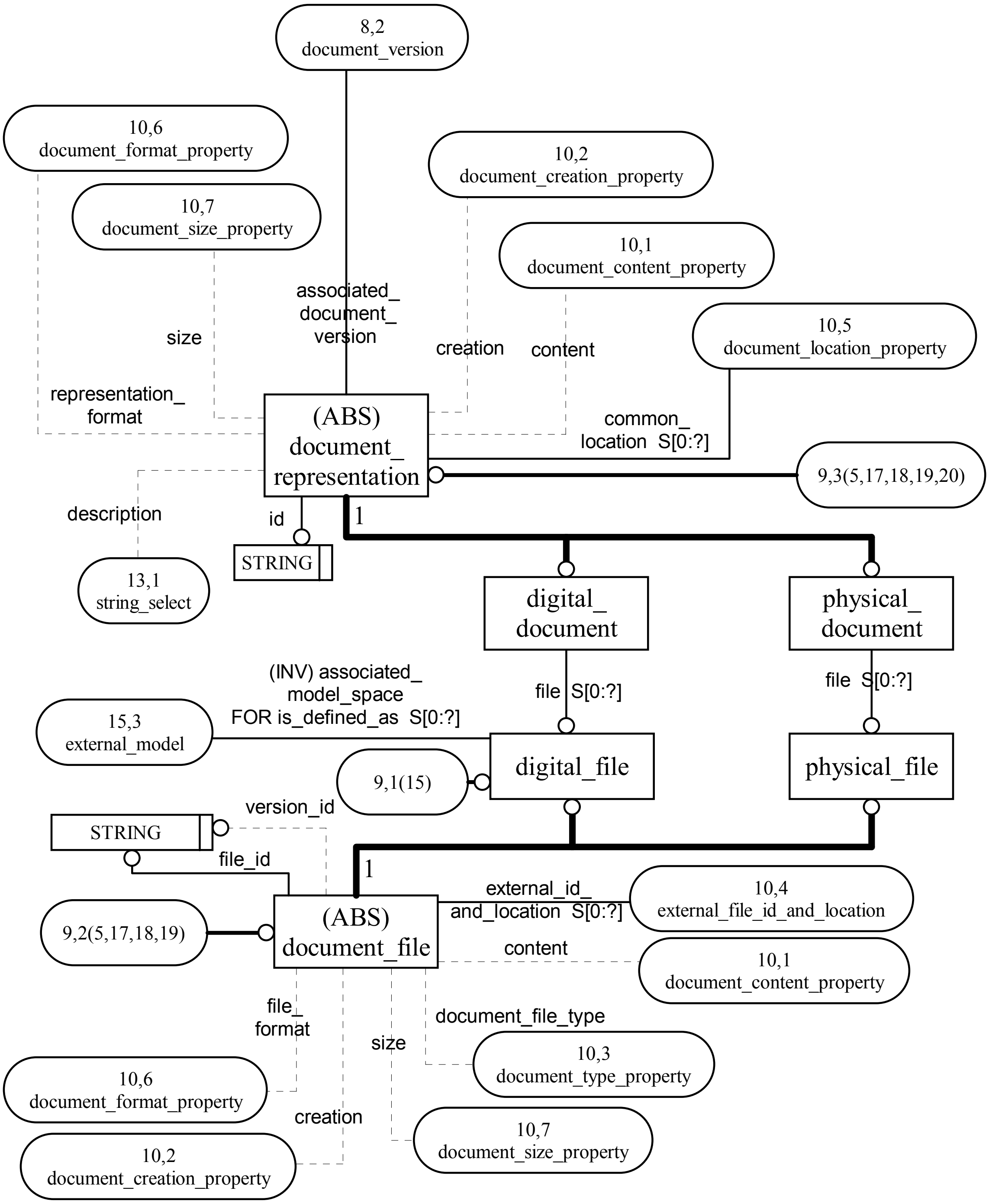

Figure D.9 — ISO13399-1 EXPRESS-G diagram 9 of 20

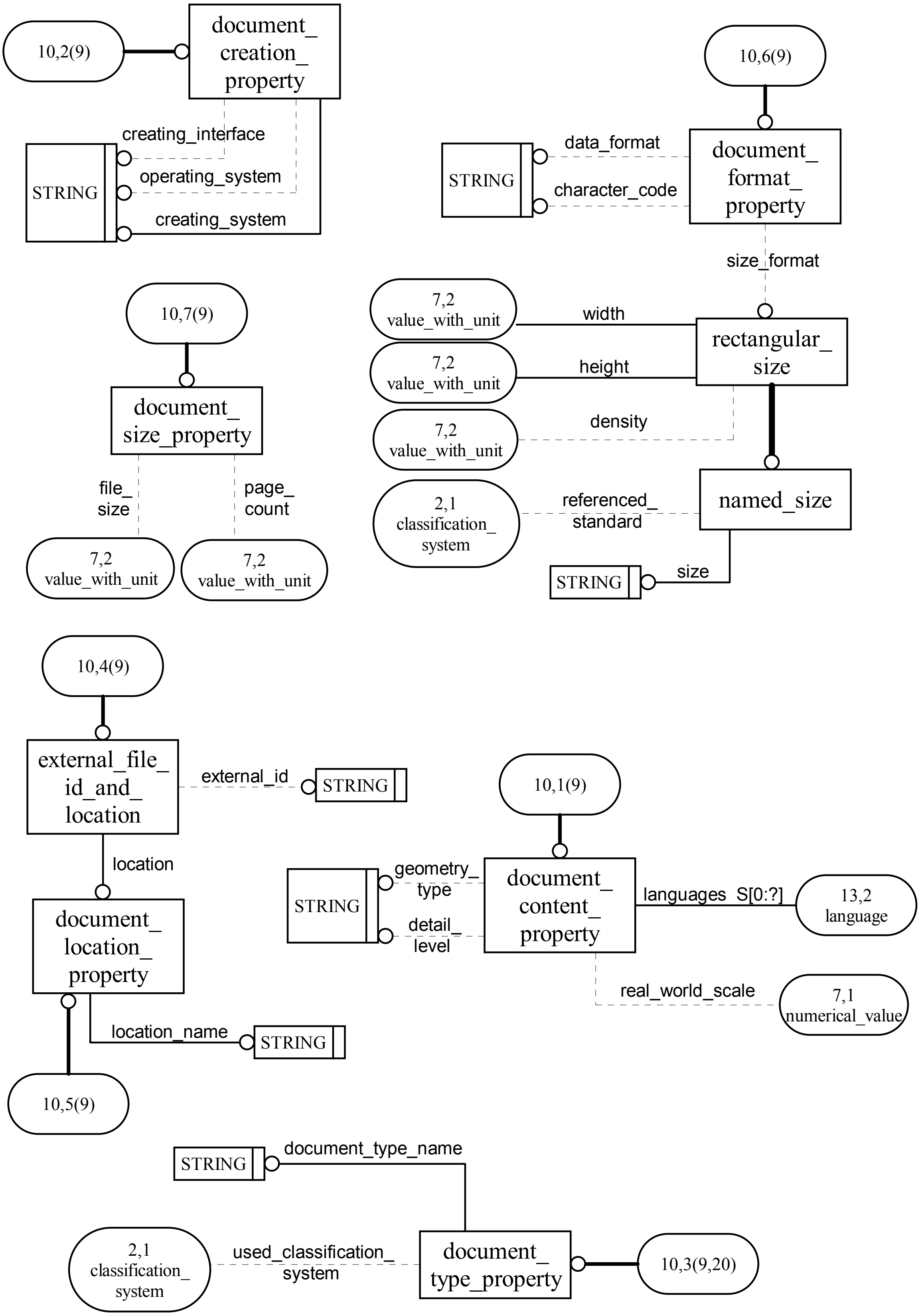

Figure D.10 — ISO13399-1 EXPRESS-G diagram 10 of 20

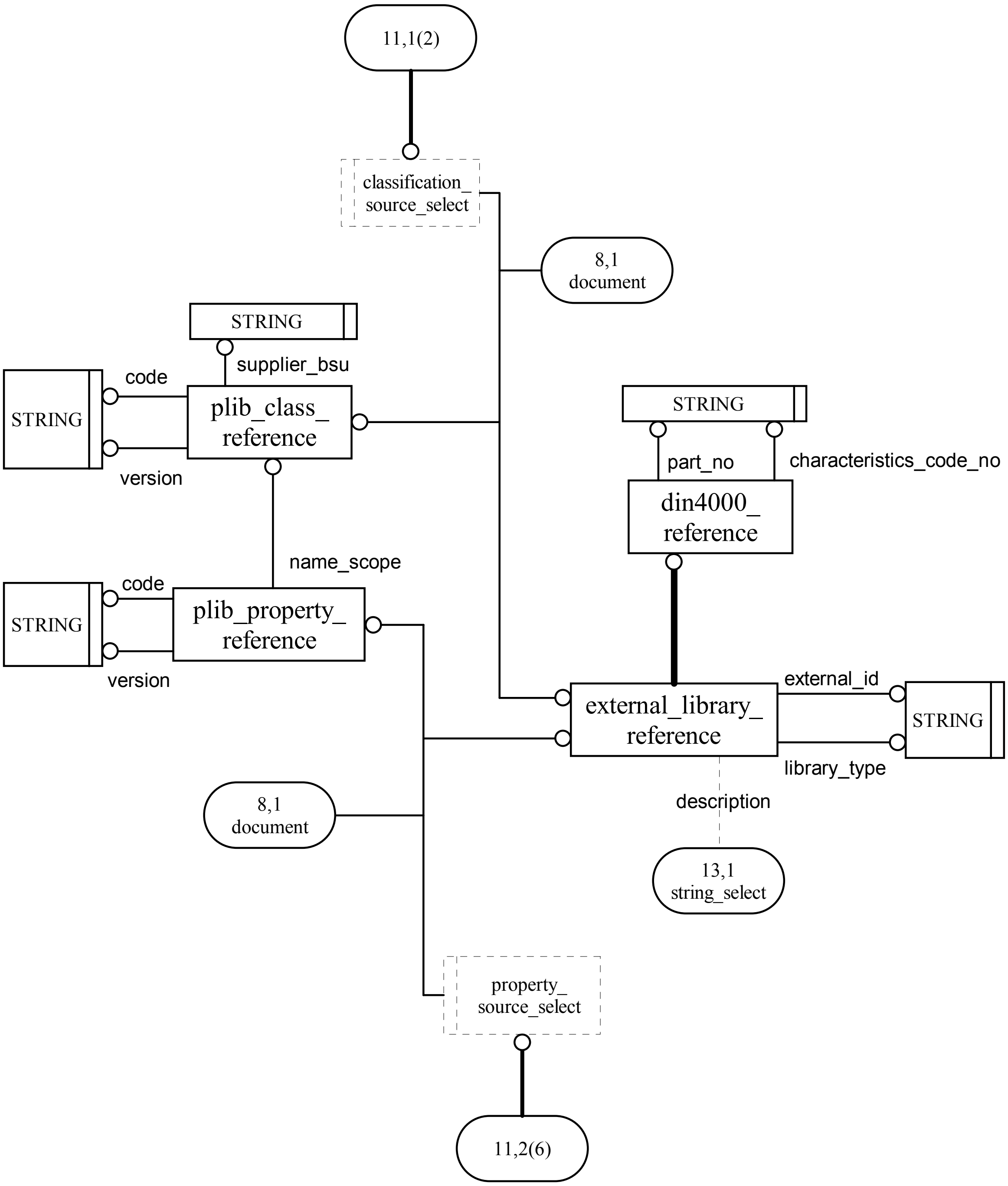

Figure D.11 — ISO13399-1 EXPRESS-G diagram 11 of 20

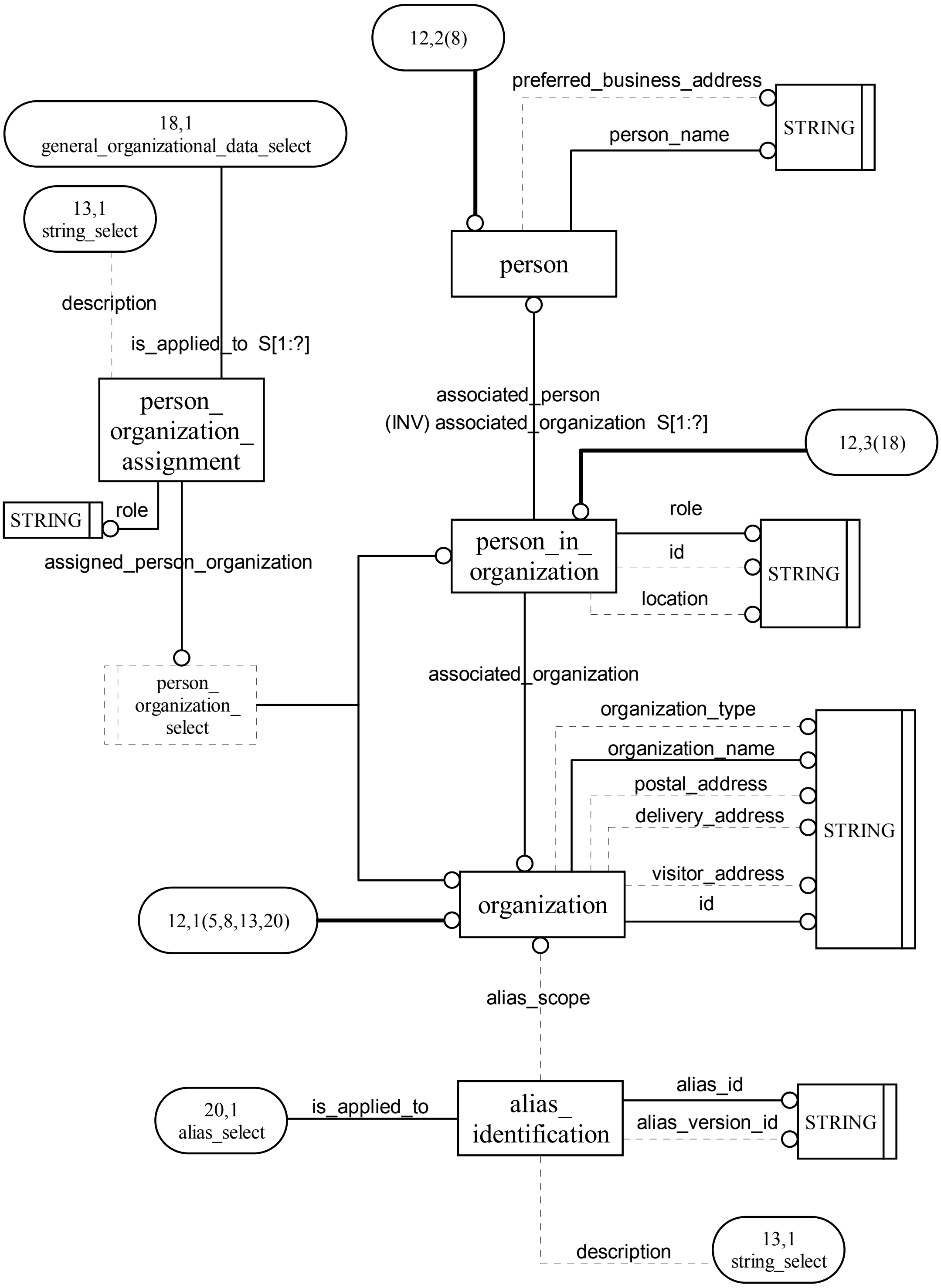

Figure D.12 — ISO13399-1 EXPRESS-G diagram 12 of 20

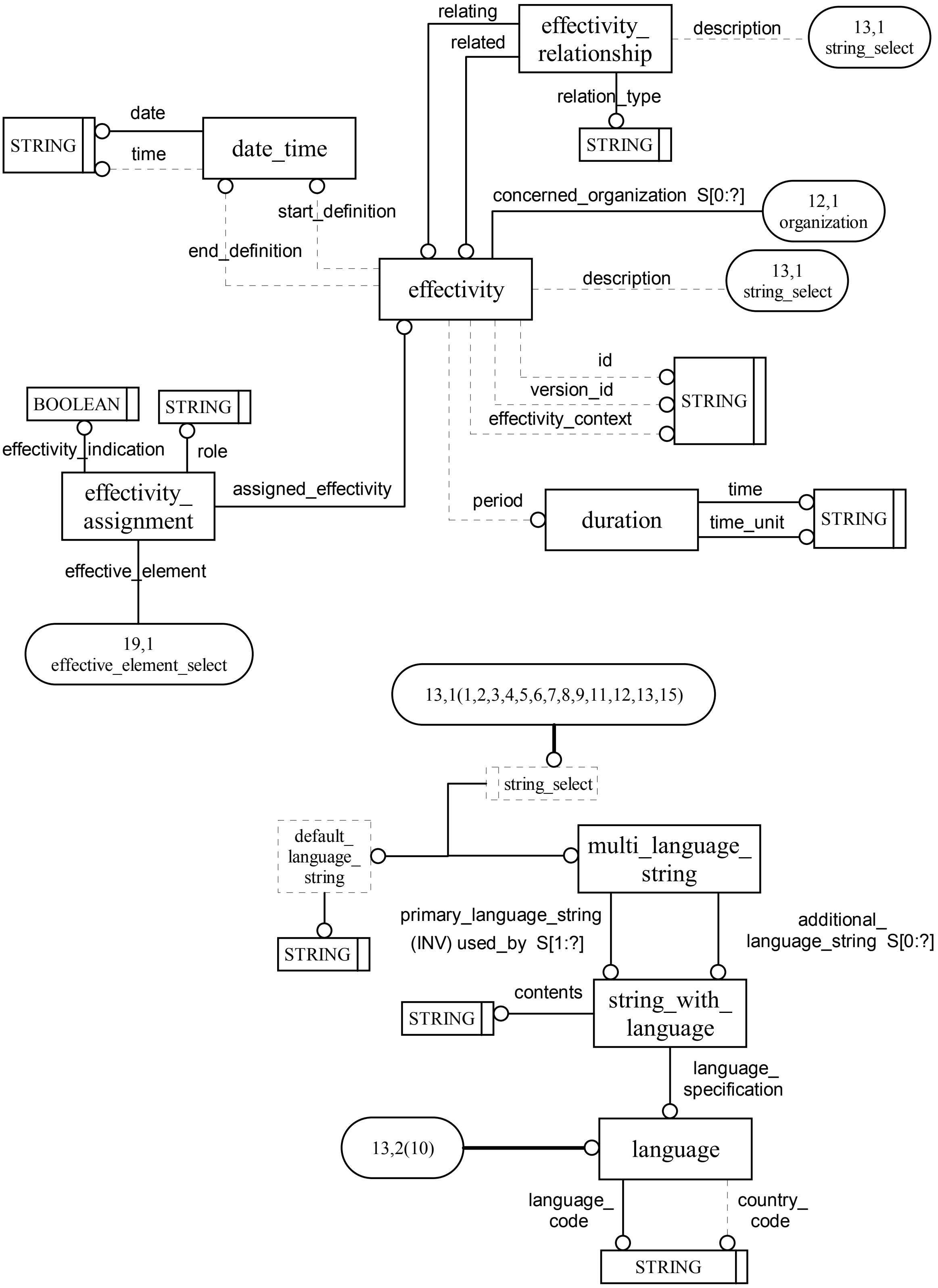

Figure D.13 — ISO13399-1 EXPRESS-G diagram 13 of 20

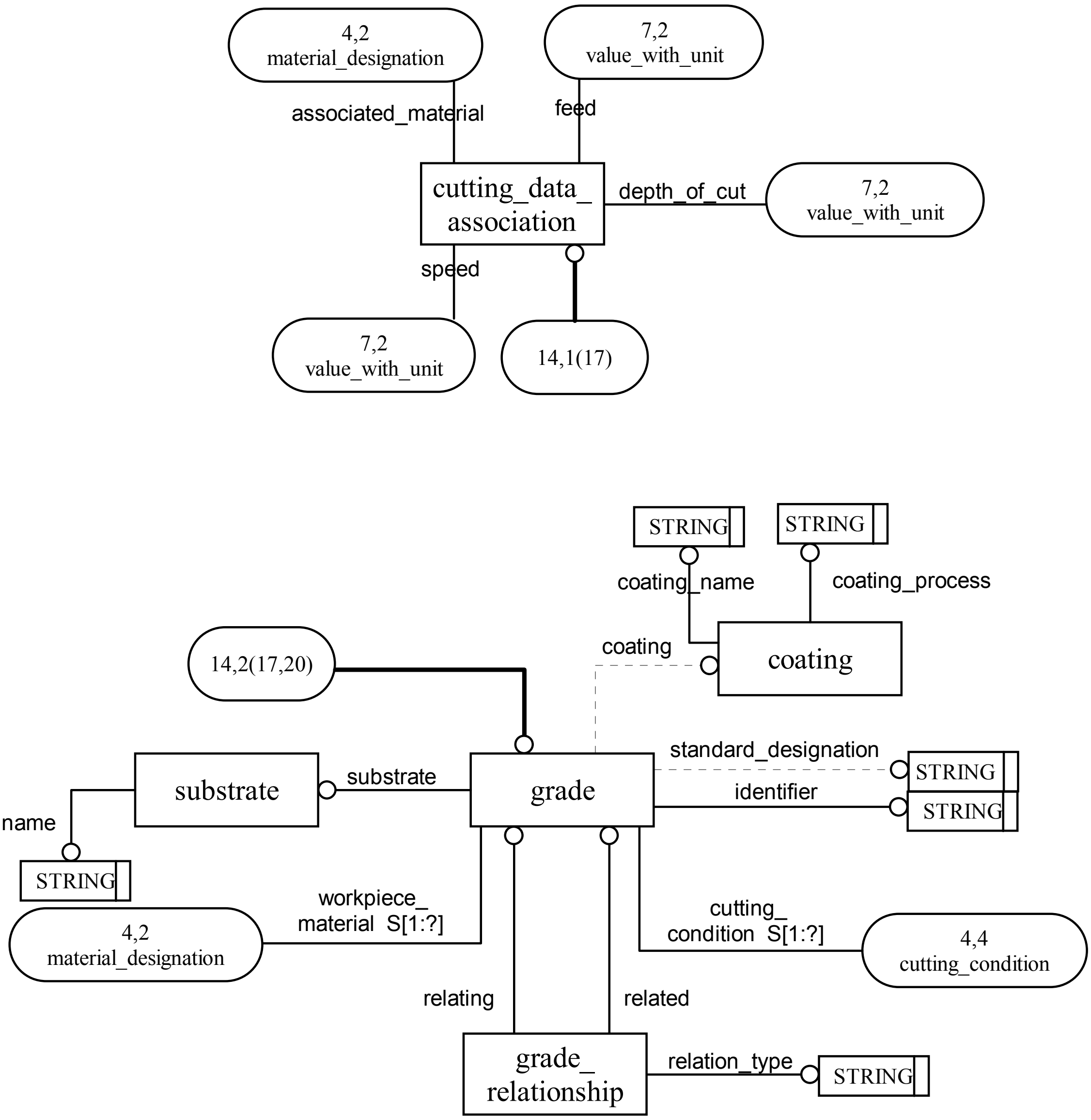

Figure D.14 — ISO13399-1 EXPRESS-G diagram 14 of 20

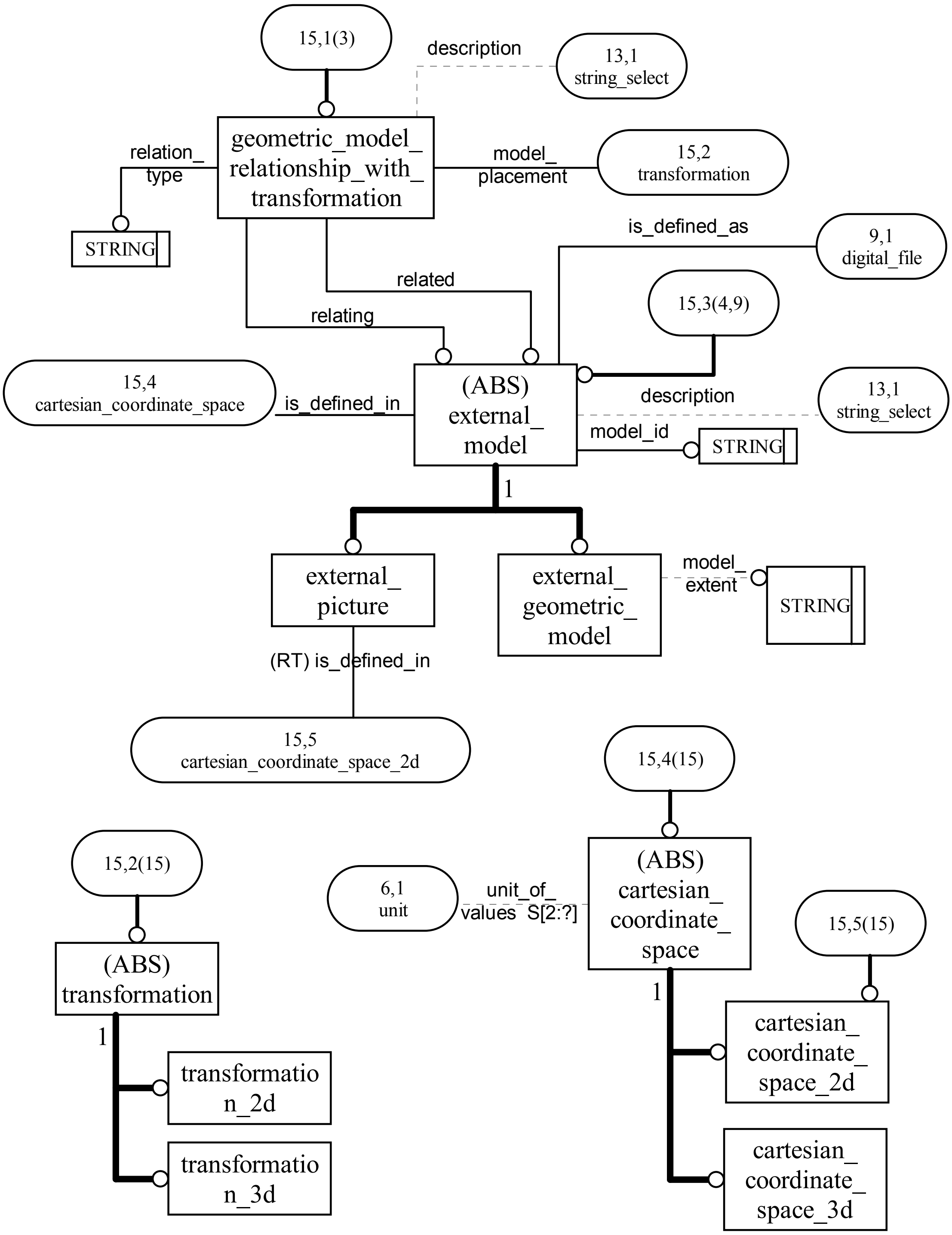

Figure D.15 — ISO13399-1 EXPRESS-G diagram 15 of 20

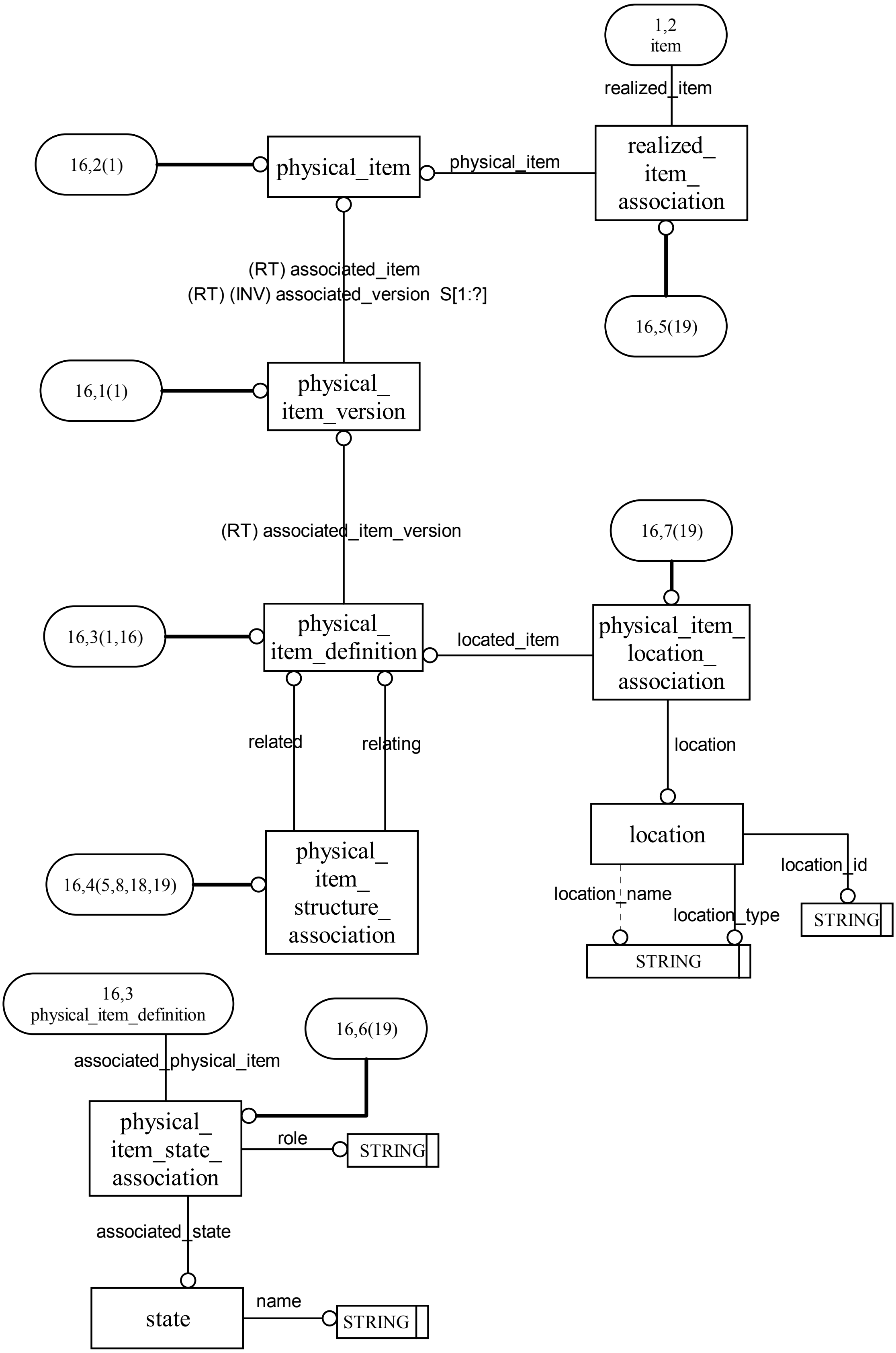

Figure D.16 — ISO13399-1 EXPRESS-G diagram 16 of 20

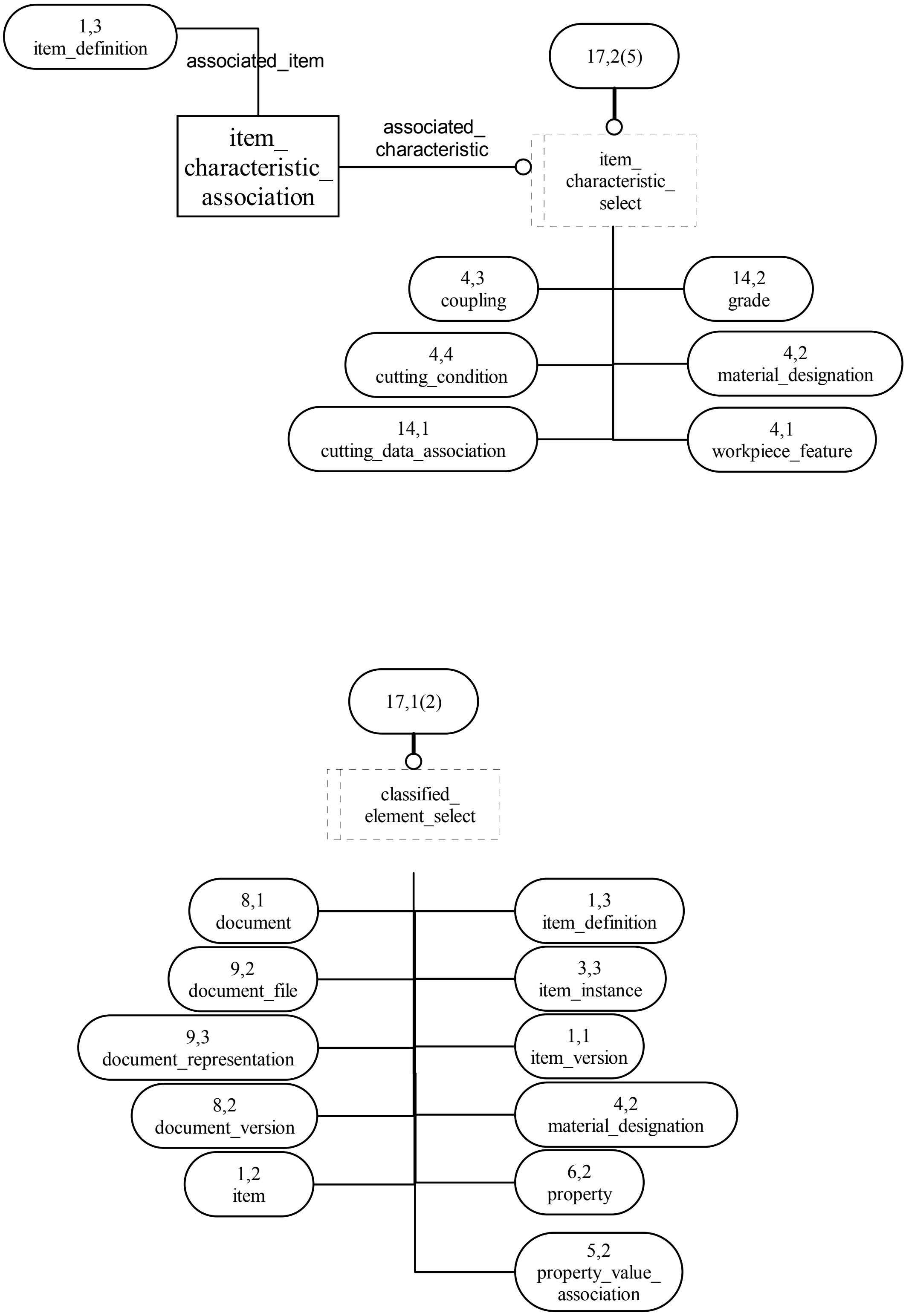

Figure D.17 — ISO13399-1 EXPRESS-G diagram 17 of 20

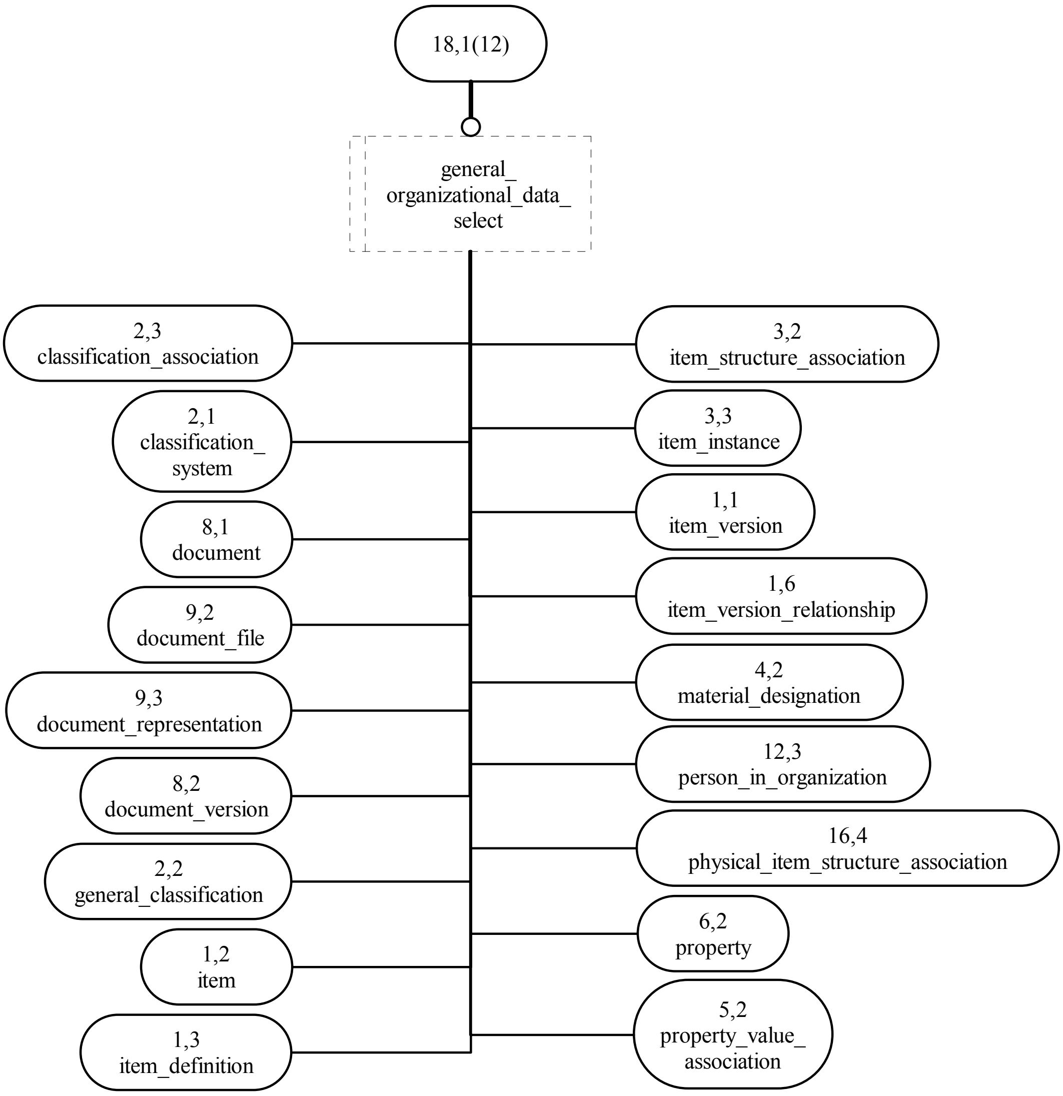
18,1(12)
general_ organizational_data_ select
2,3 classification_association
2,1 classification_ system
8,1 document
9,2 document_file
9,3 document_representation
8,2 document_version
2,2 general_classification
1,2 item
1,3 item_definition
3,2 item_structure_association
3,3 item_instance
1,1 item_version
1,6 item_version_relationship
4,2 material_designation
12,3 person_in_organization
16,4 physical_item_structure_association
6,2 property
5,2 property_value_ association

Figure D.18 — ISO13399-1 EXPRESS-G diagram 18 of 20

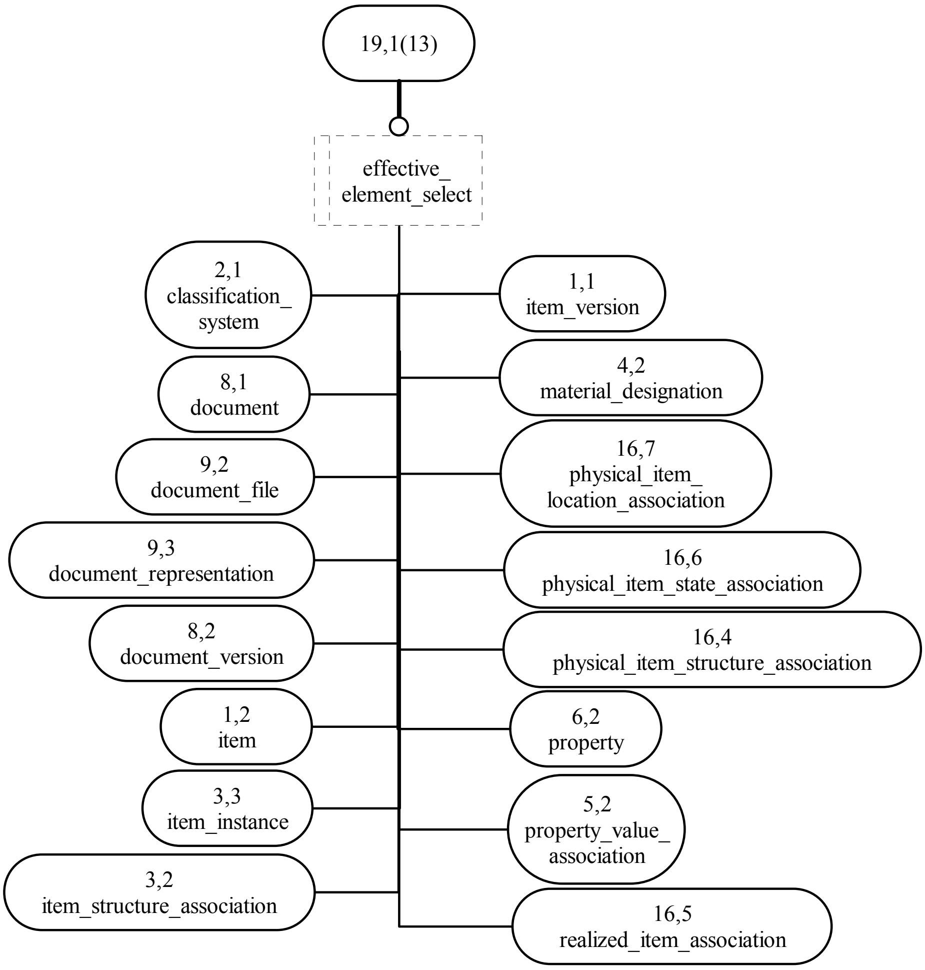
19,1(13)
effective_ element_select
2,1 classification_ system
8,1 document
9,2 document_file
9,3 document_representation
8,2 document_version
1,2 item
3,3 item_instance
3,2 item_structure_association
1,1 item_version
4,2 material_designation
16,7 physical_item_ location_association
16,6 physical_item_state_association
16,4 physical_item_structure_association
6,2 property
5,2 property_value_ association
16,5 realized_item_association

Figure D.19 — ISO13399-1 EXPRESS-G diagram 19 of 20

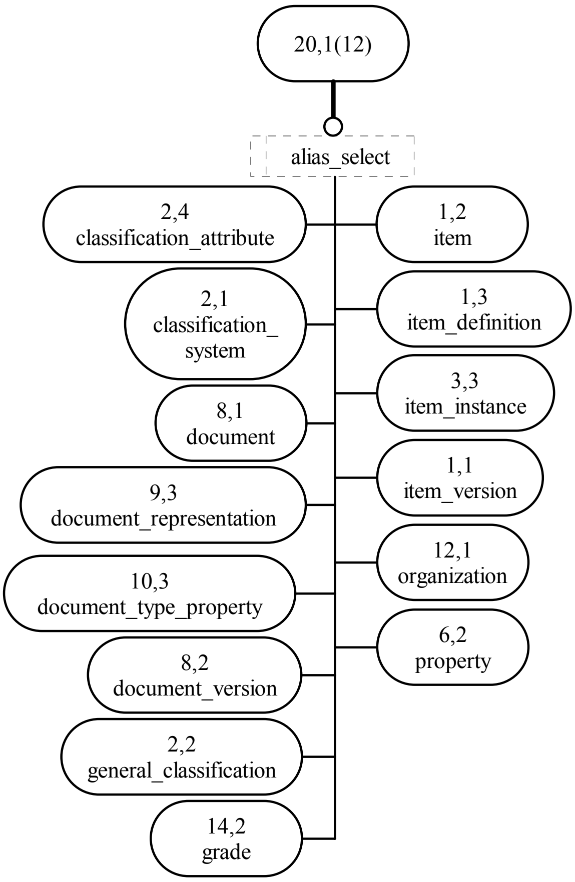

Figure D.20 — ISO13399-1 EXPRESS-G diagram 20 of 20

Annex E
(informative)

AIM EXPRESS-G diagrams

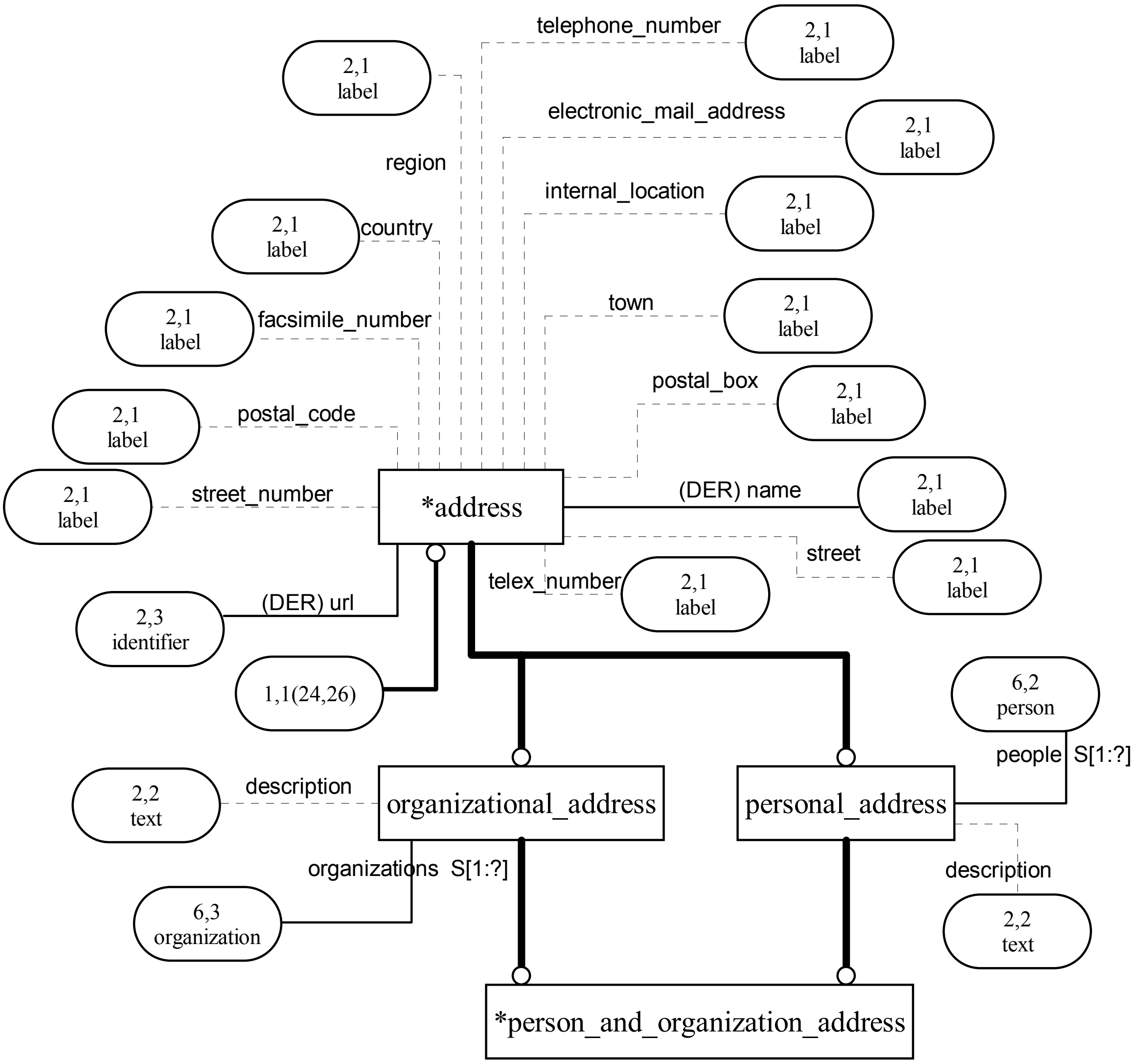

Figure E.1 — ISO13399-1 AIM EXPRESS-G diagram 1 of 31

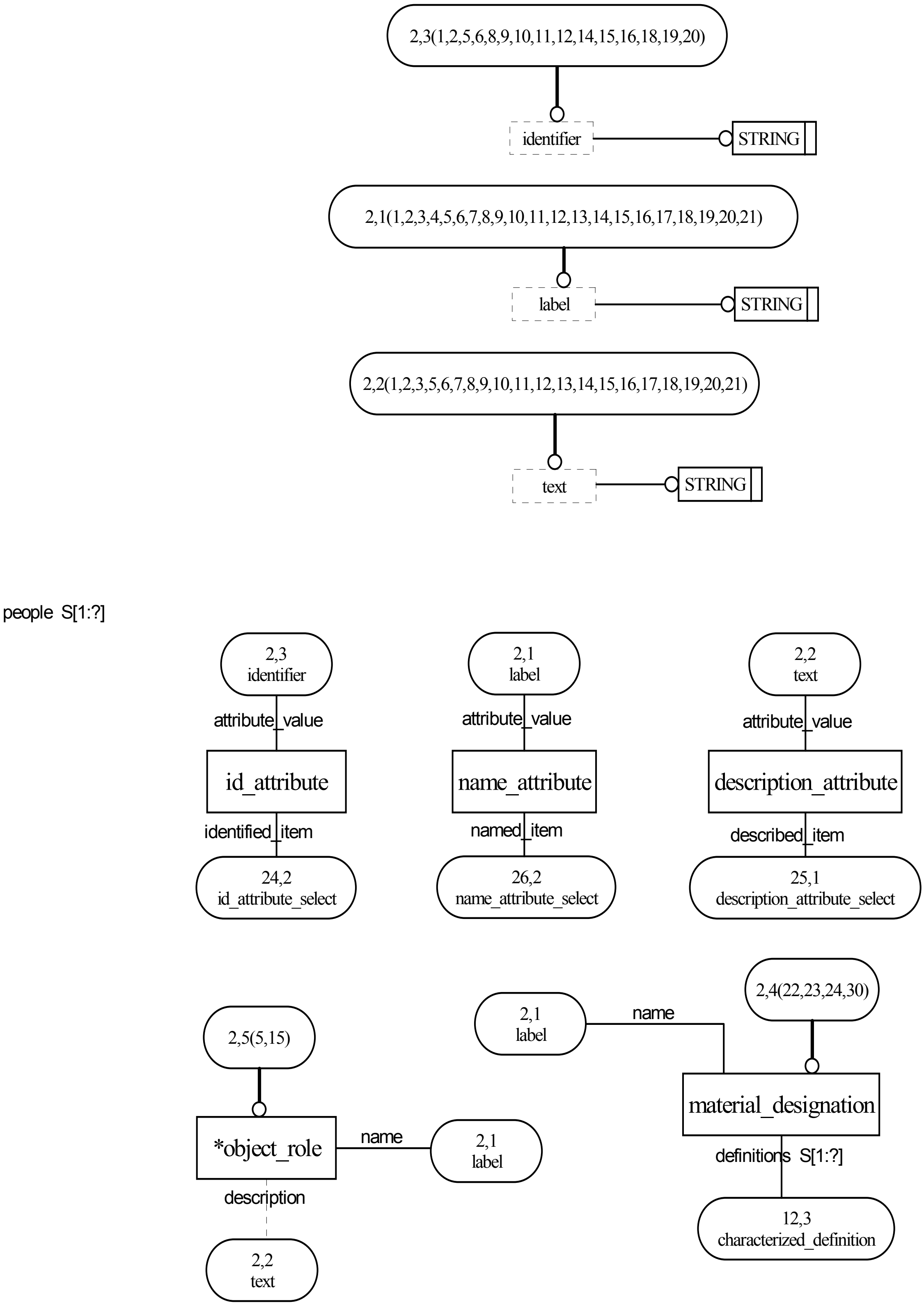

Figure E.2 — ISO13399-1 AIM EXPRESS-G diagram 2 of 31

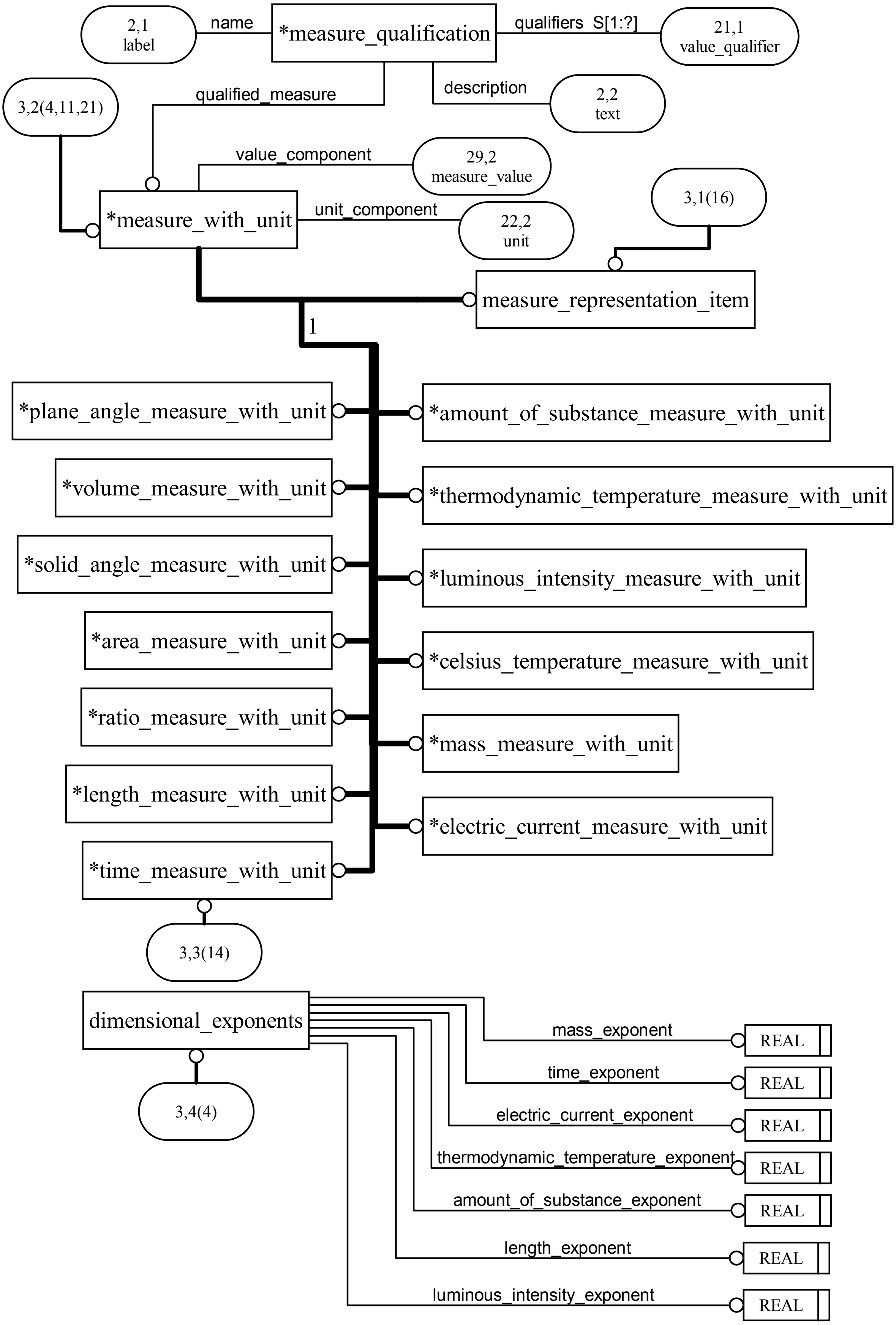
2,1
label
name
*measure_qualification
qualifiers S[1:?]
21,1
value_qualifier
3,2(4,11,21)
qualified_measure
description
2,2
text
value_component
29,2
measure_value
3,1(16)
*measure_with_unit
unit_component
22,2
unit
measure_representation_item
1
*plane_angle_measure_with_unit
*amount_of_substance_measure_with_unit
*volume_measure_with_unit
*thermodynamic_temperature_measure_with_unit
*solid_angle_measure_with_unit
*luminous_intensity_measure_with_unit
*area_measure_with_unit
*celsius_temperature_measure_with_unit
*ratio_measure_with_unit
*mass_measure_with_unit
*length_measure_with_unit
*electric_current_measure_with_unit
*time_measure_with_unit
3,3(14)
dimensional_exponents
mass_exponent
REAL
time_exponent
REAL
electric_current_exponent
REAL
thermodynamic_temperature_exponent
REAL
amount_of_substance_exponent
REAL
length_exponent
REAL
luminous_intensity_exponent
REAL
3,4(4)

Figure E.3 — ISO13399-1 AIM EXPRESS-G diagram 3 of 31

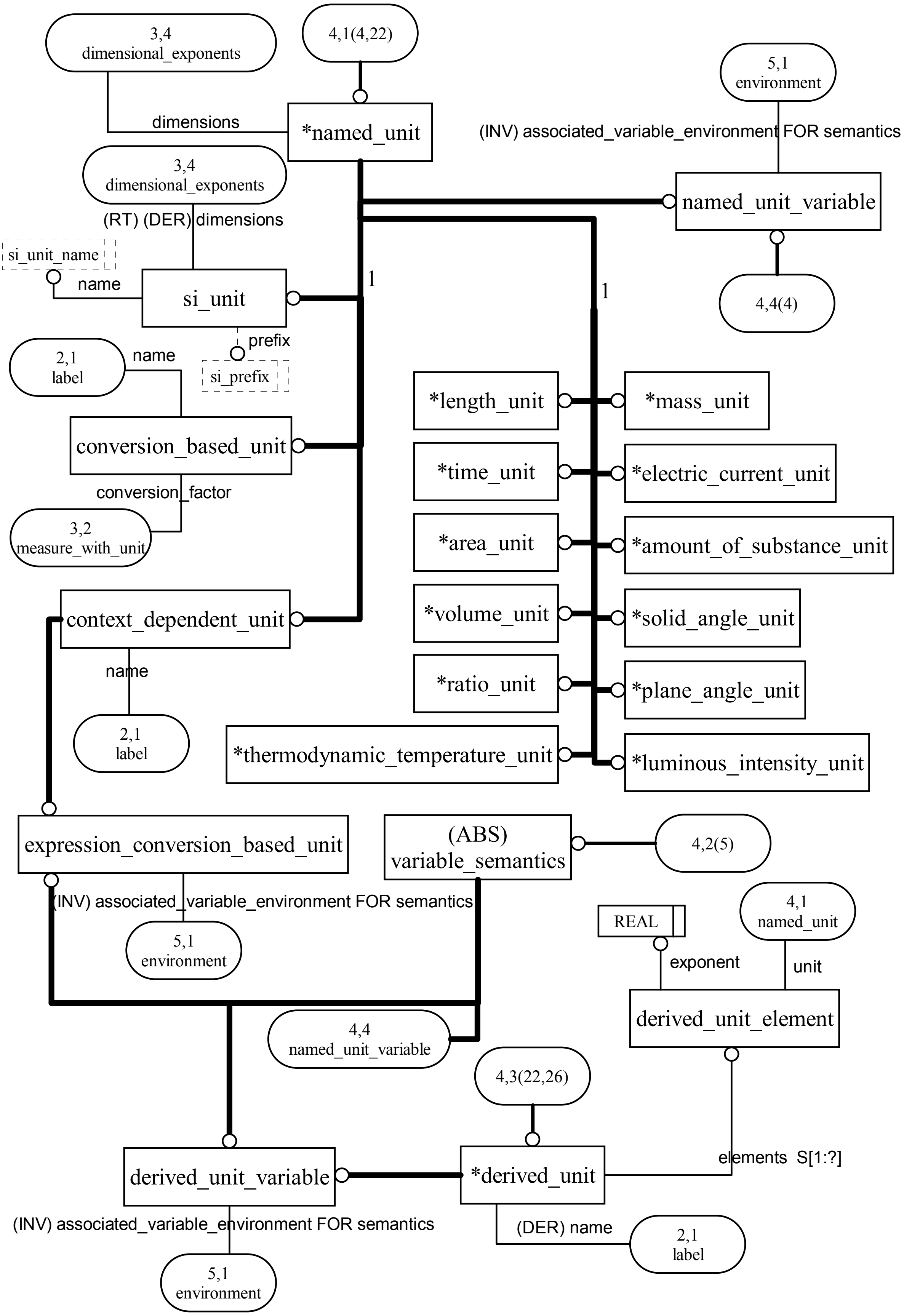

Figure E.4 — ISO13399-1 AIM EXPRESS-G diagram 4 of 31

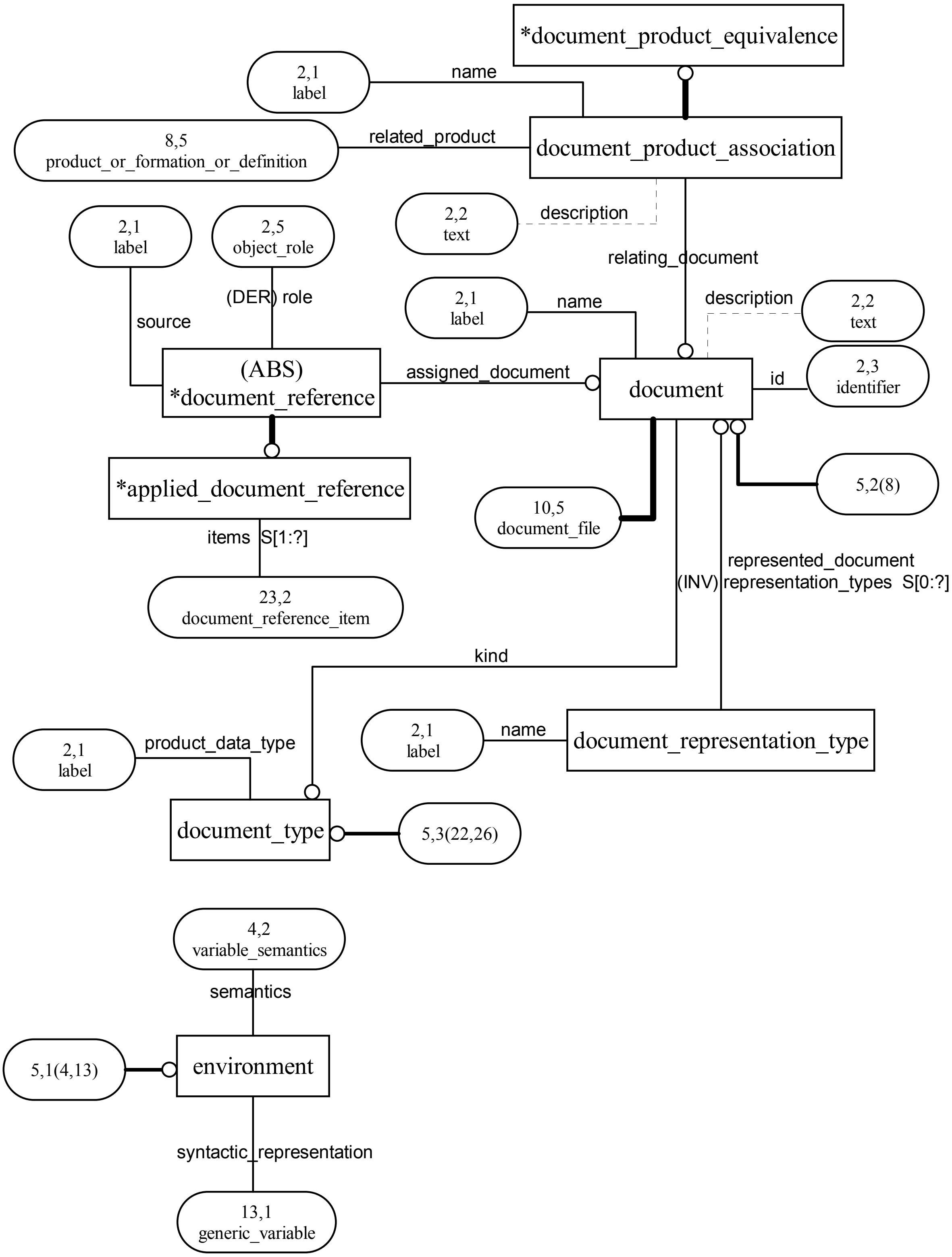

Figure E.5 — ISO13399-1 AIM EXPRESS-G diagram 5 of 31

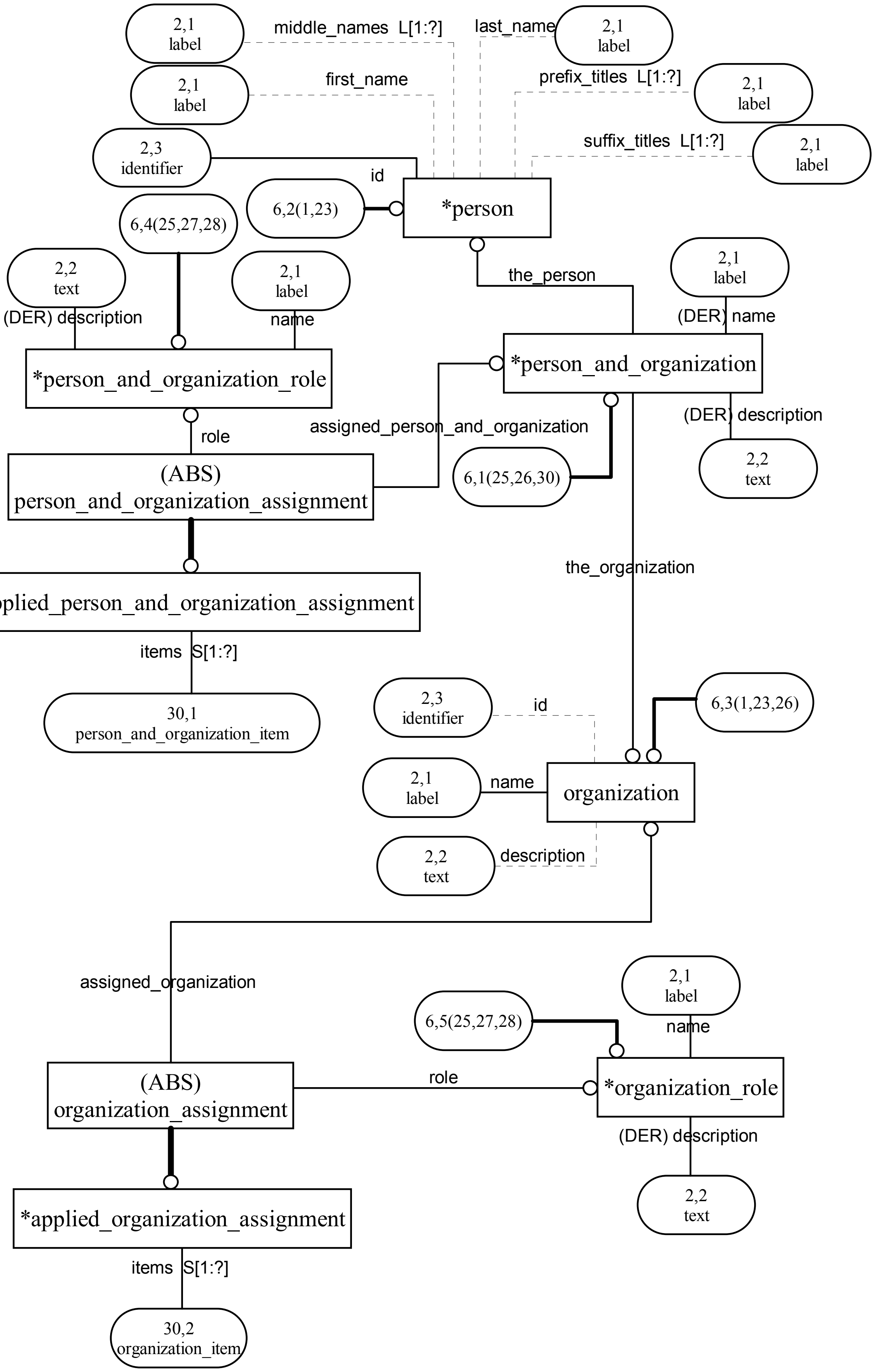

Figure E.6 — ISO13399-1 AIM EXPRESS-G diagram 6 of 31

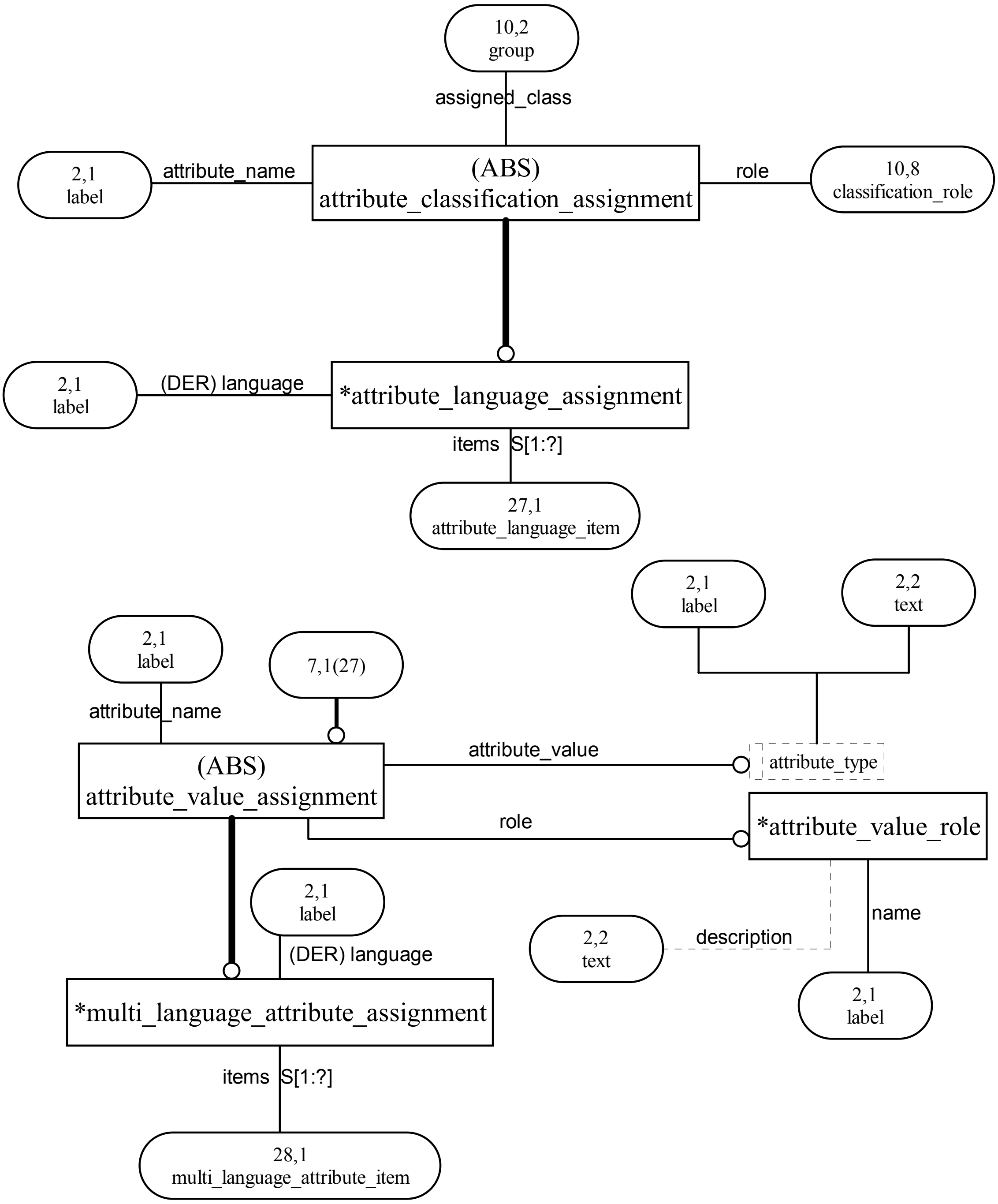

Figure E.7 — ISO13399-1 AIM EXPRESS-G diagram 7 of 31

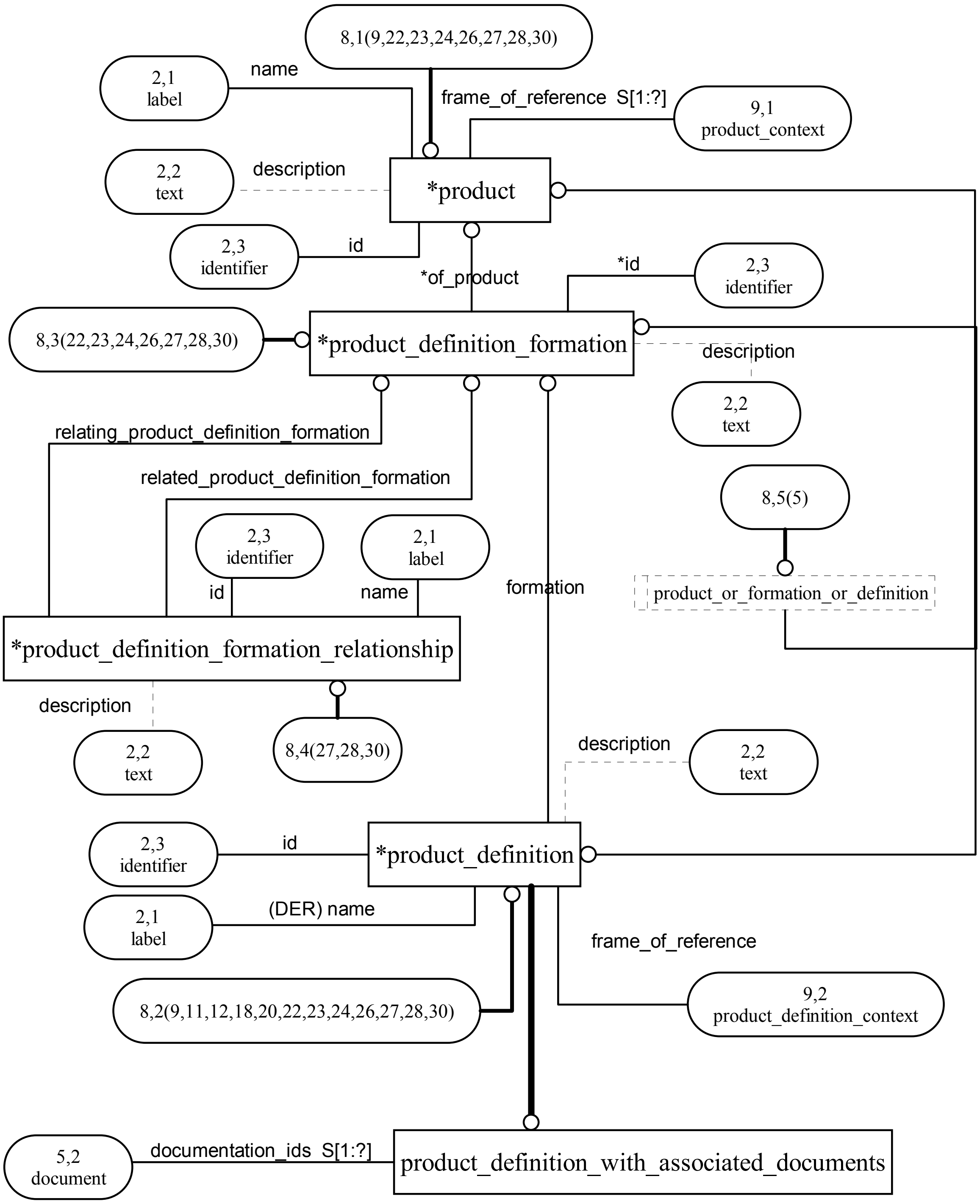

Figure E.8 — ISO13399-1 AIM EXPRESS-G diagram 8 of 31

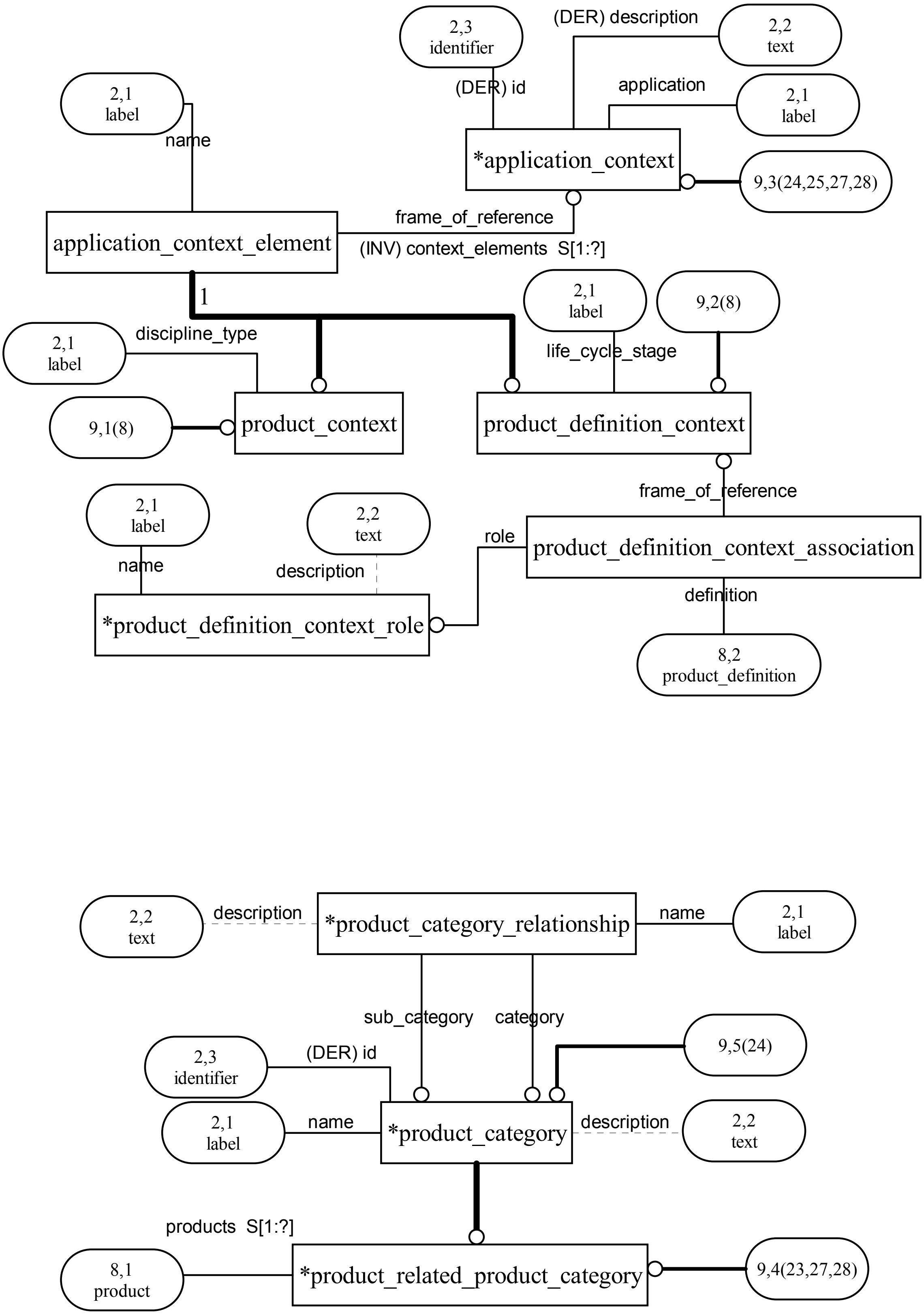

Figure E.9 — ISO13399-1 AIM EXPRESS-G diagram 9 of 31

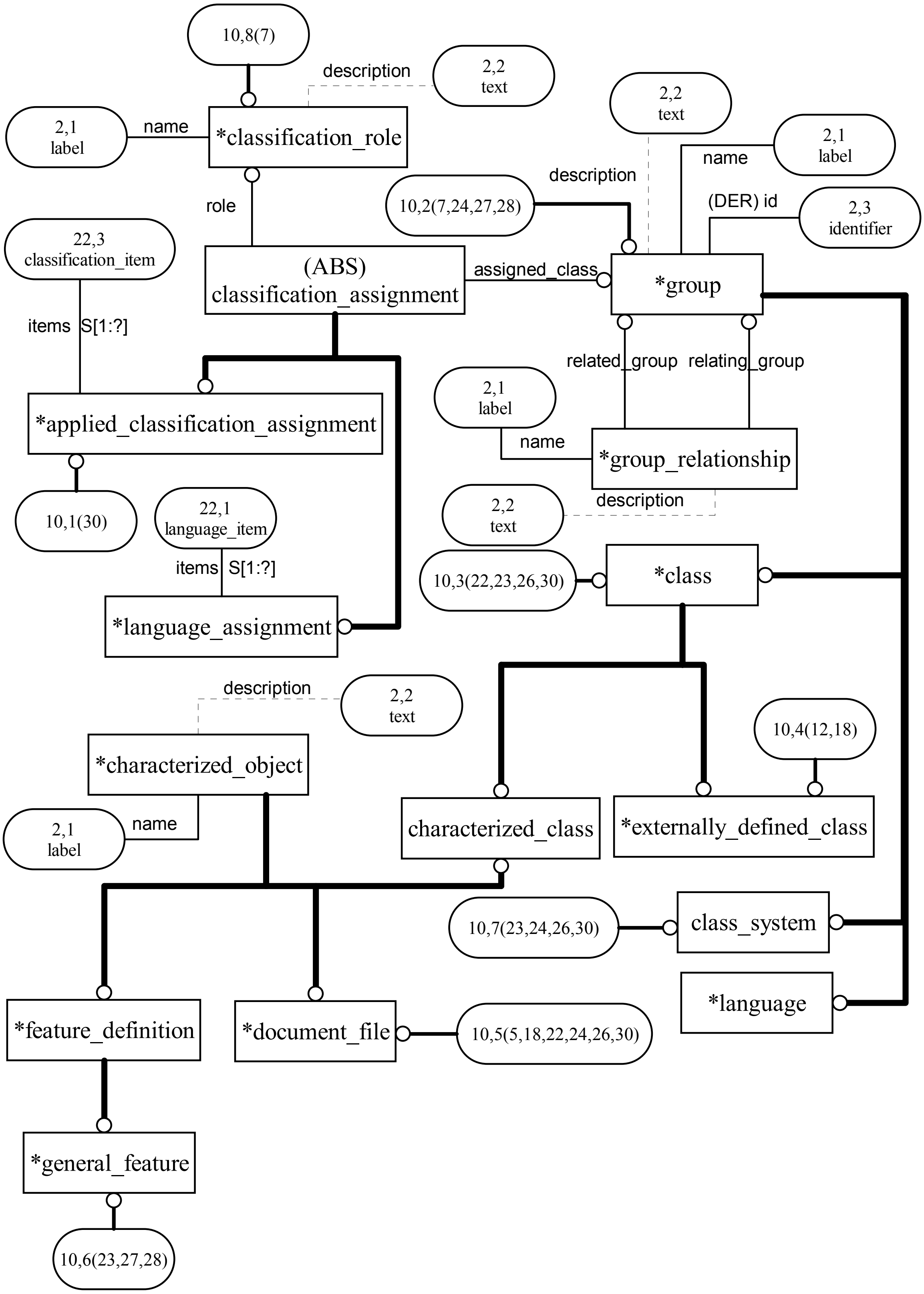

Figure E.10 — ISO13399-1 AIM EXPRESS-G diagram 10 of 31

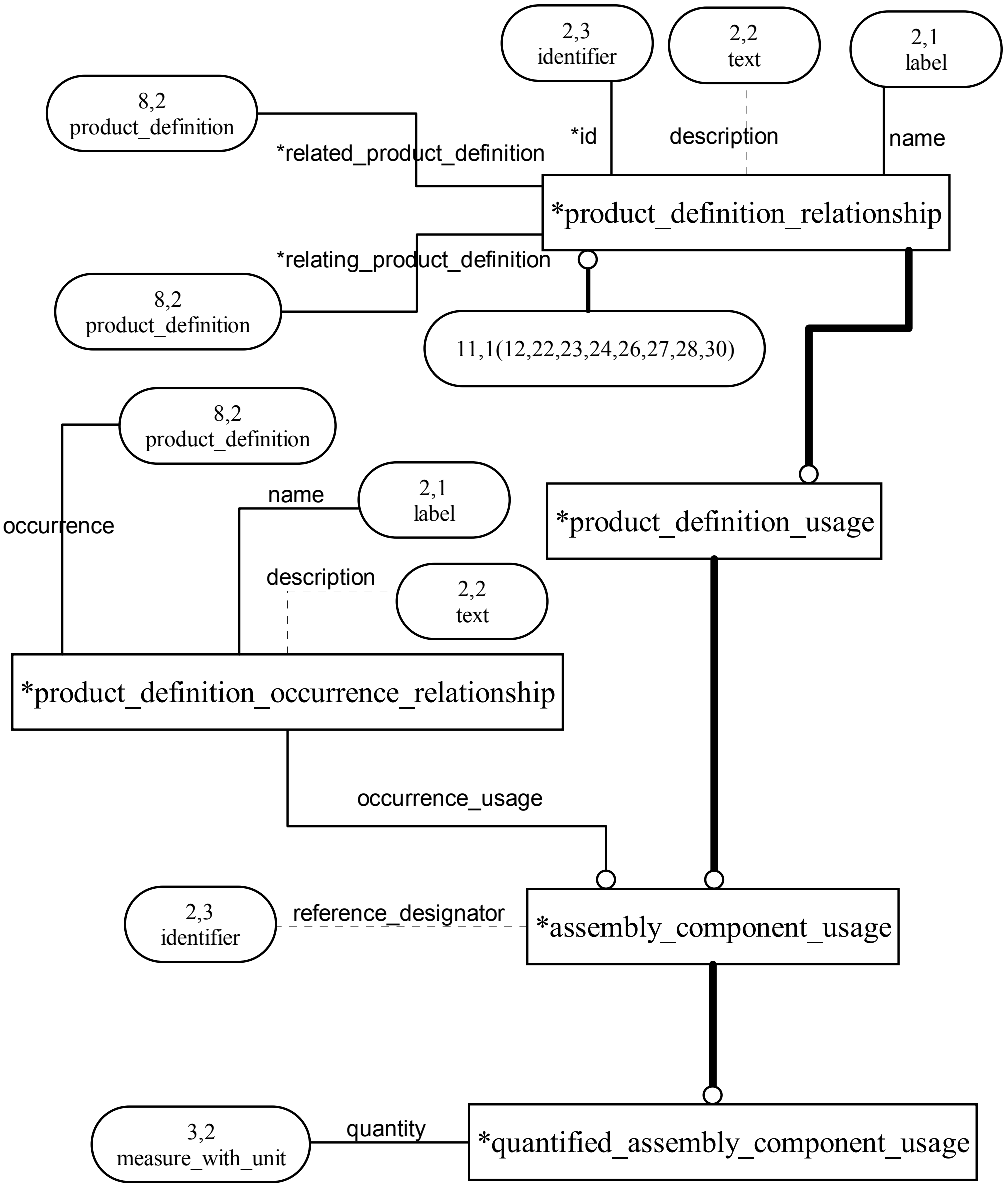

Figure E.11 — ISO13399-1 AIM EXPRESS-G diagram 11 of 31

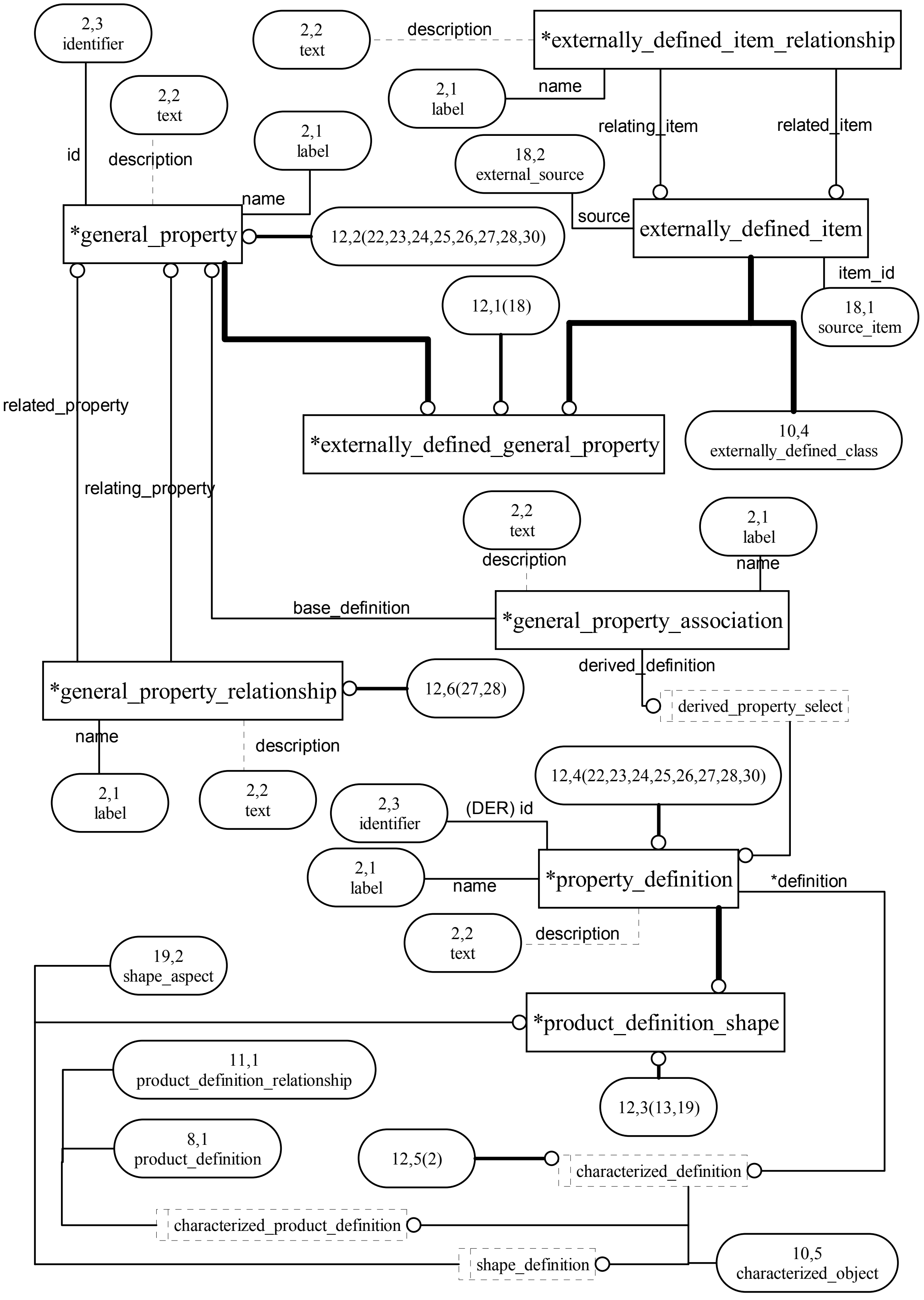

Figure E.12 — ISO13399-1 AIM EXPRESS-G diagram 12 of 31

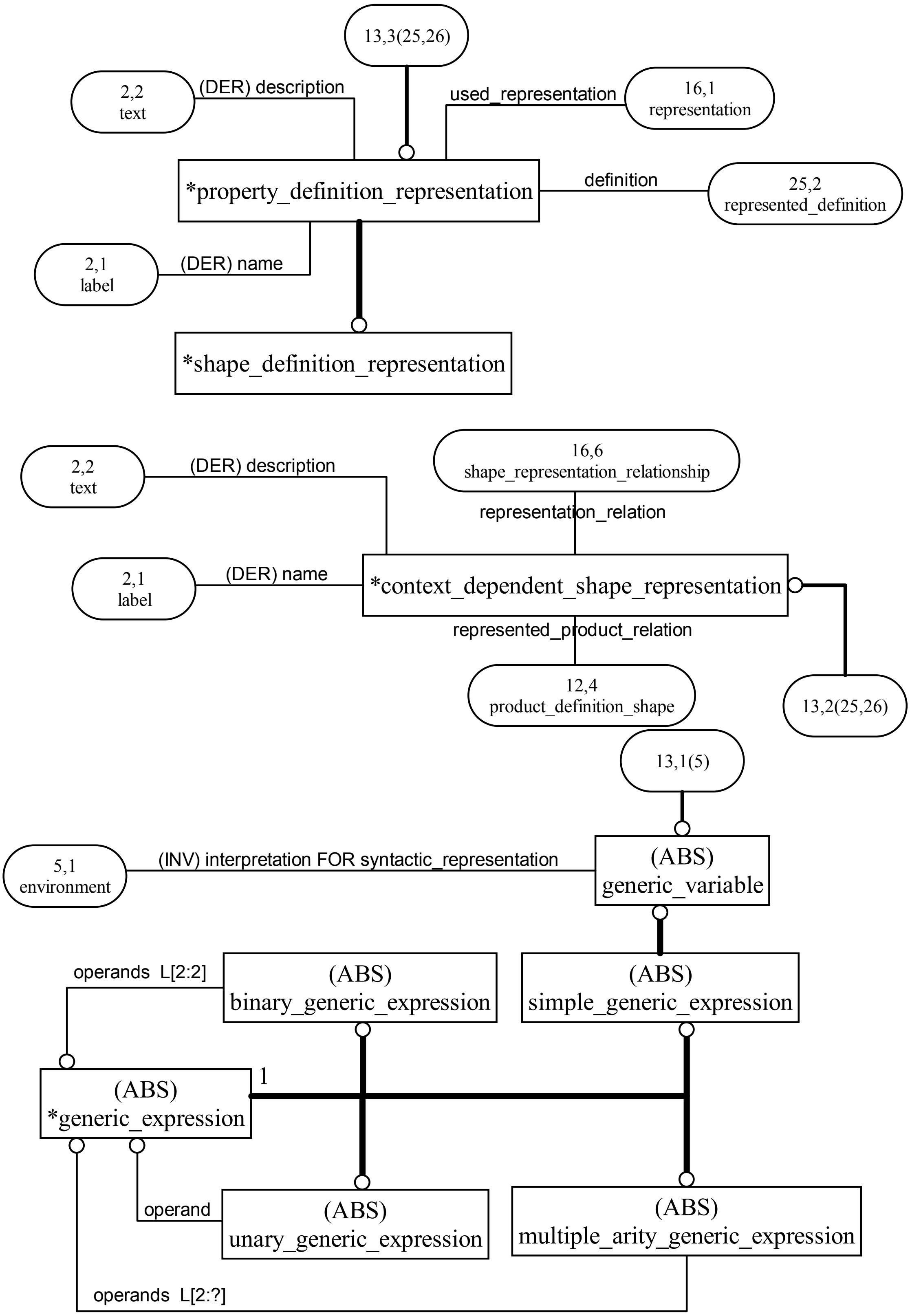

Figure E.13 — ISO13399-1 AIM EXPRESS-G diagram 13 of 31

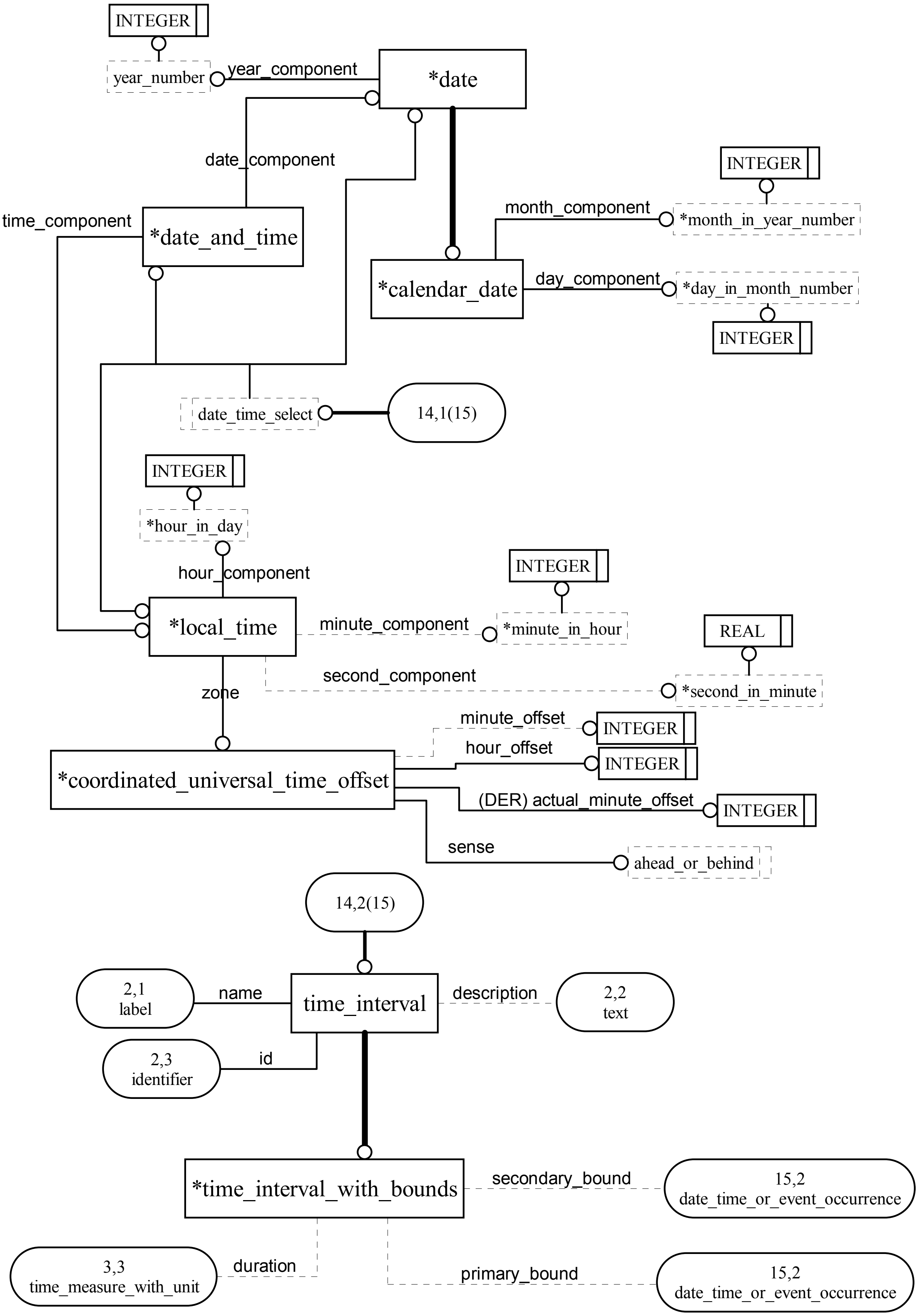

Figure E.14 — ISO13399-1 AIM EXPRESS-G diagram 14 of 31

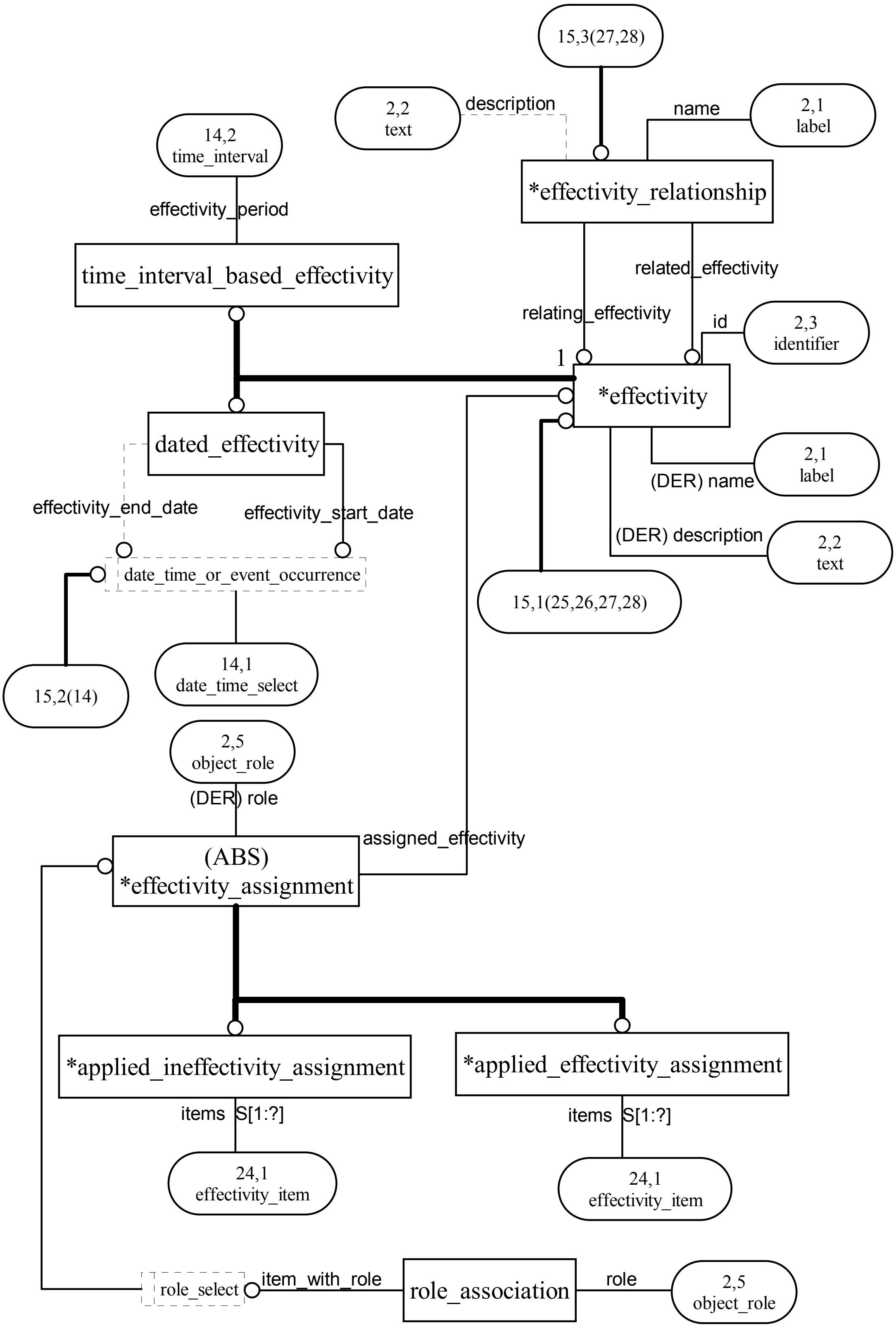

Figure E.15 — ISO13399-1 AIM EXPRESS-G diagram 15 of 31

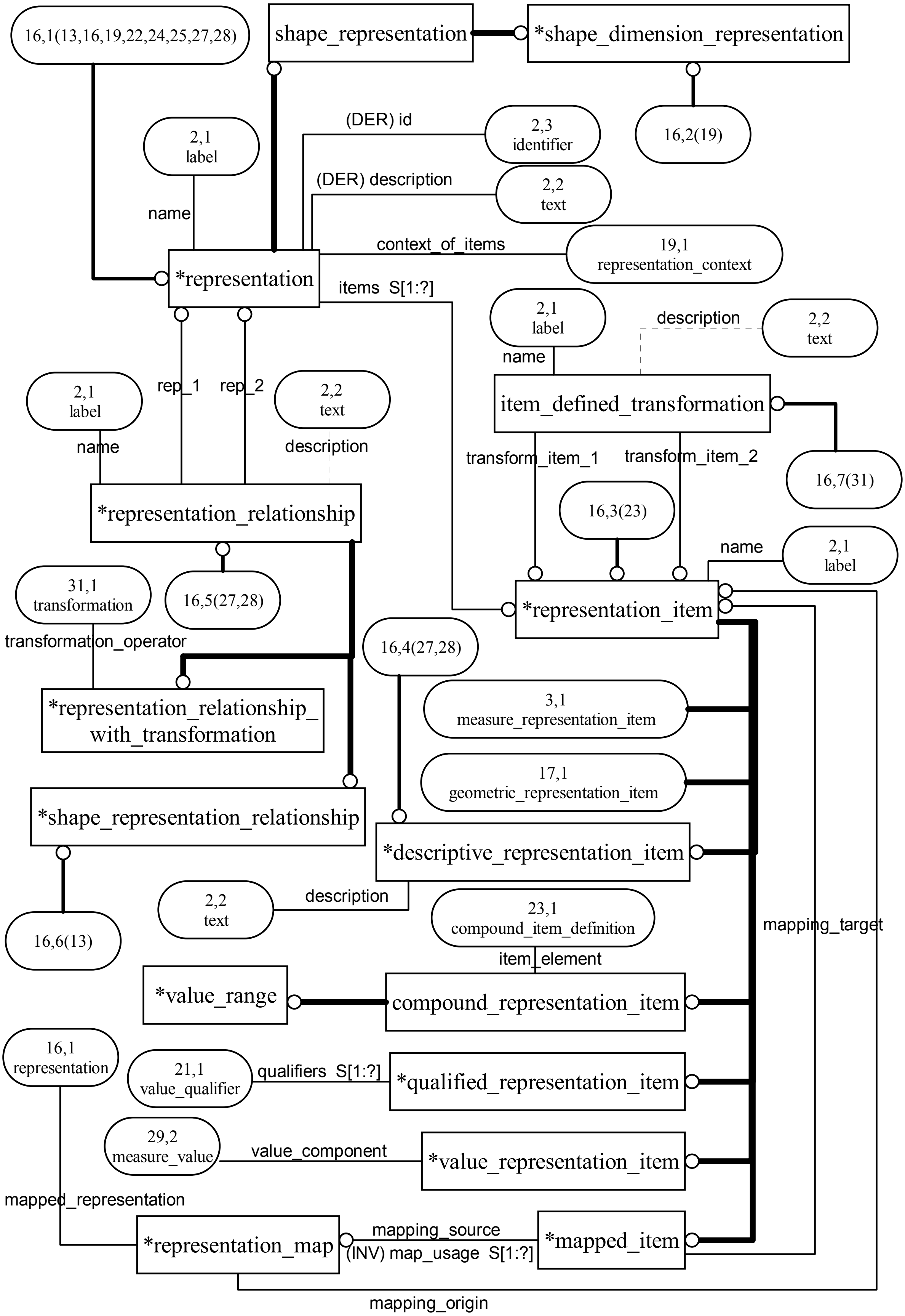

Figure E.16 — ISO13399-1 AIM EXPRESS-G diagram 16 of 31

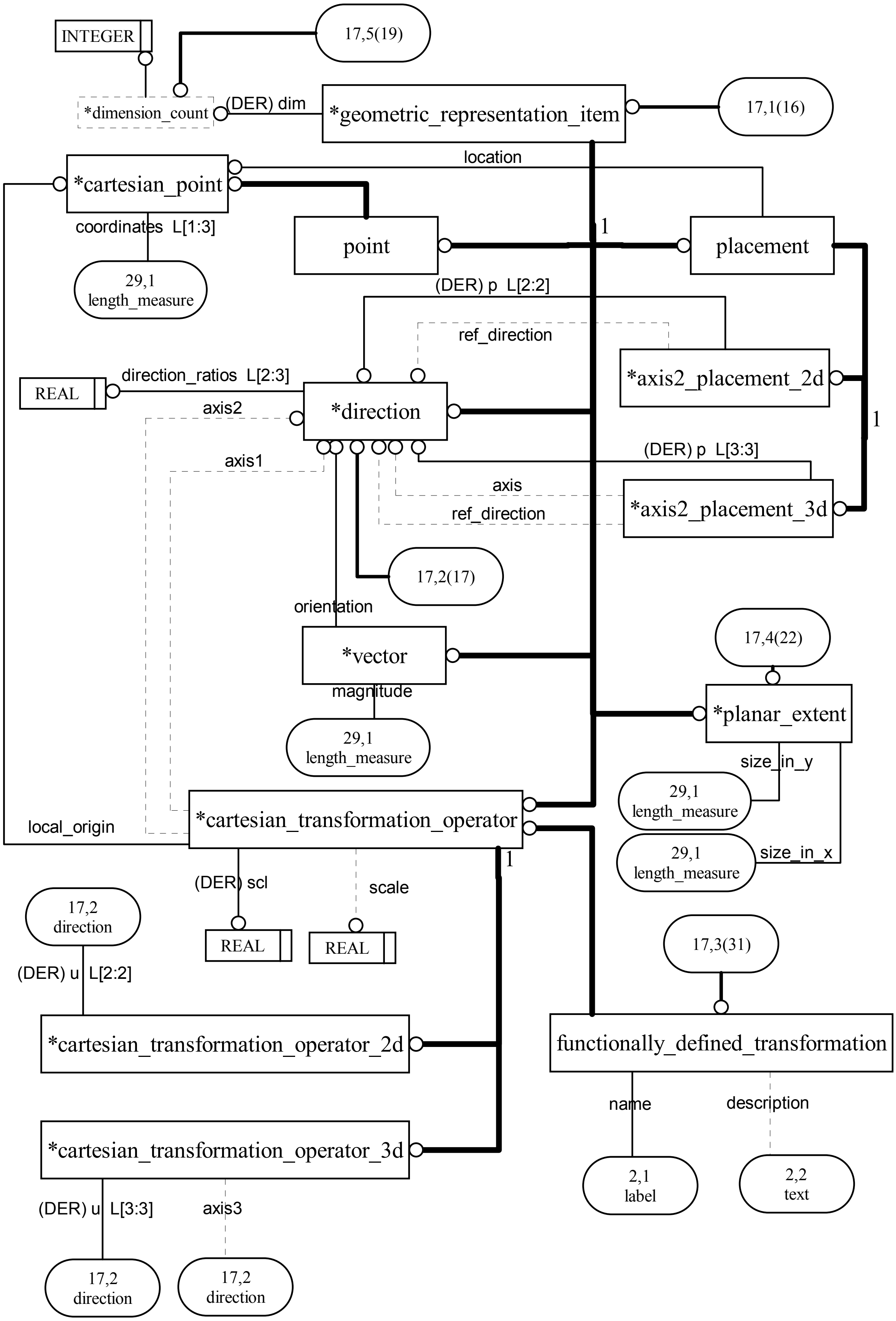

Figure E.17 — ISO13399-1 AIM EXPRESS-G diagram 17 of 31

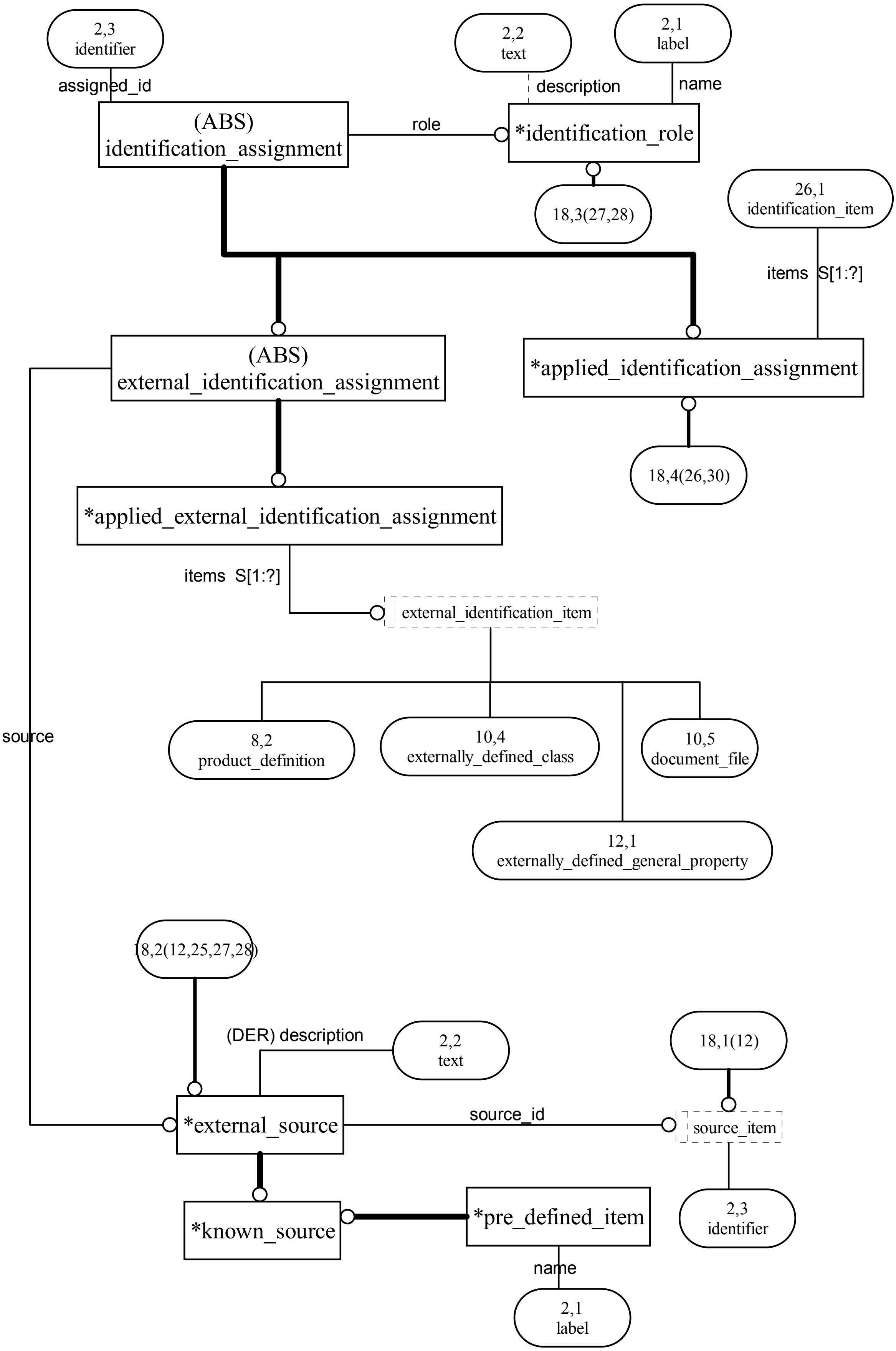

Figure E.18 — ISO13399-1 AIM EXPRESS-G diagram 18 of 31

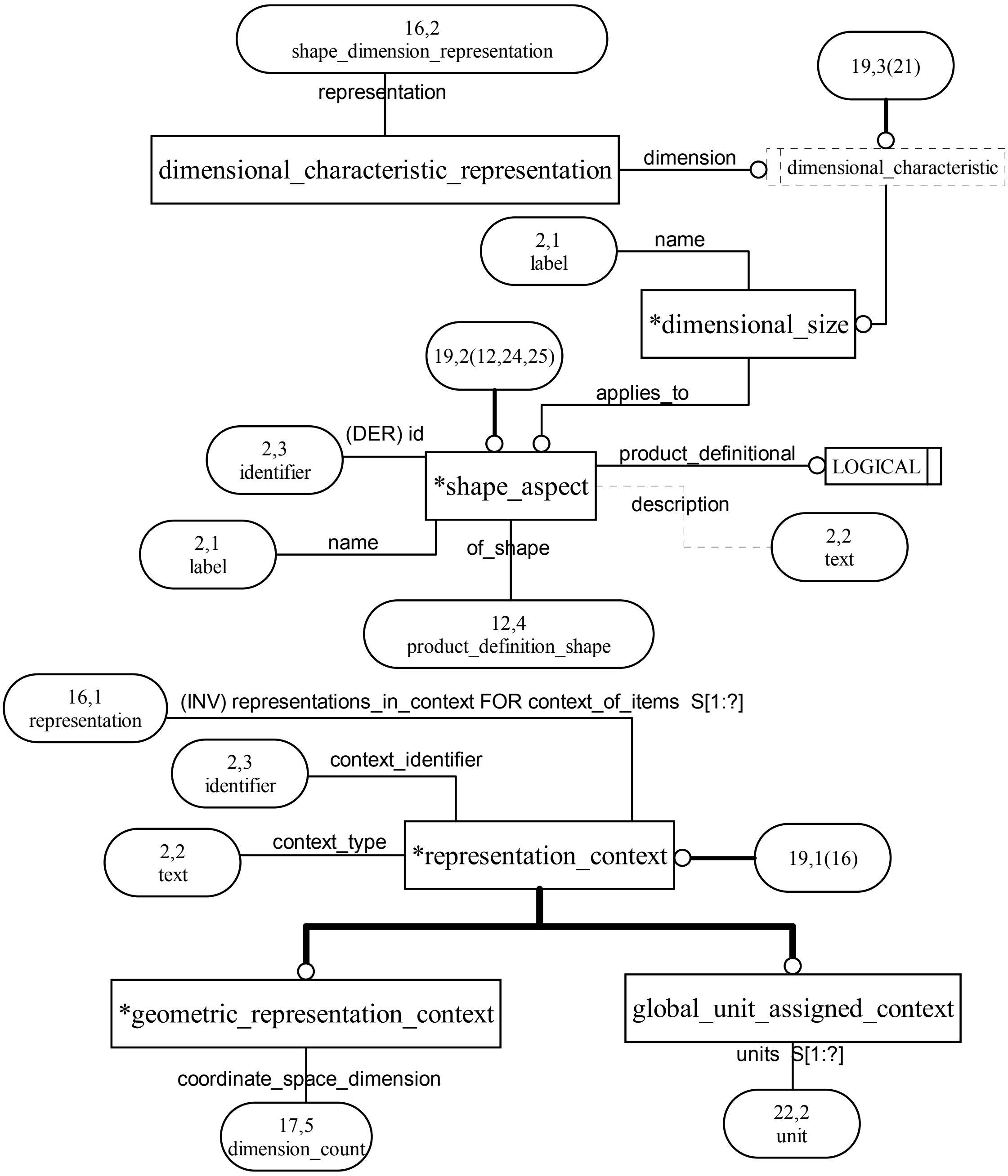

Figure E.19 — ISO13399-1 AIM EXPRESS-G diagram 19 of 31

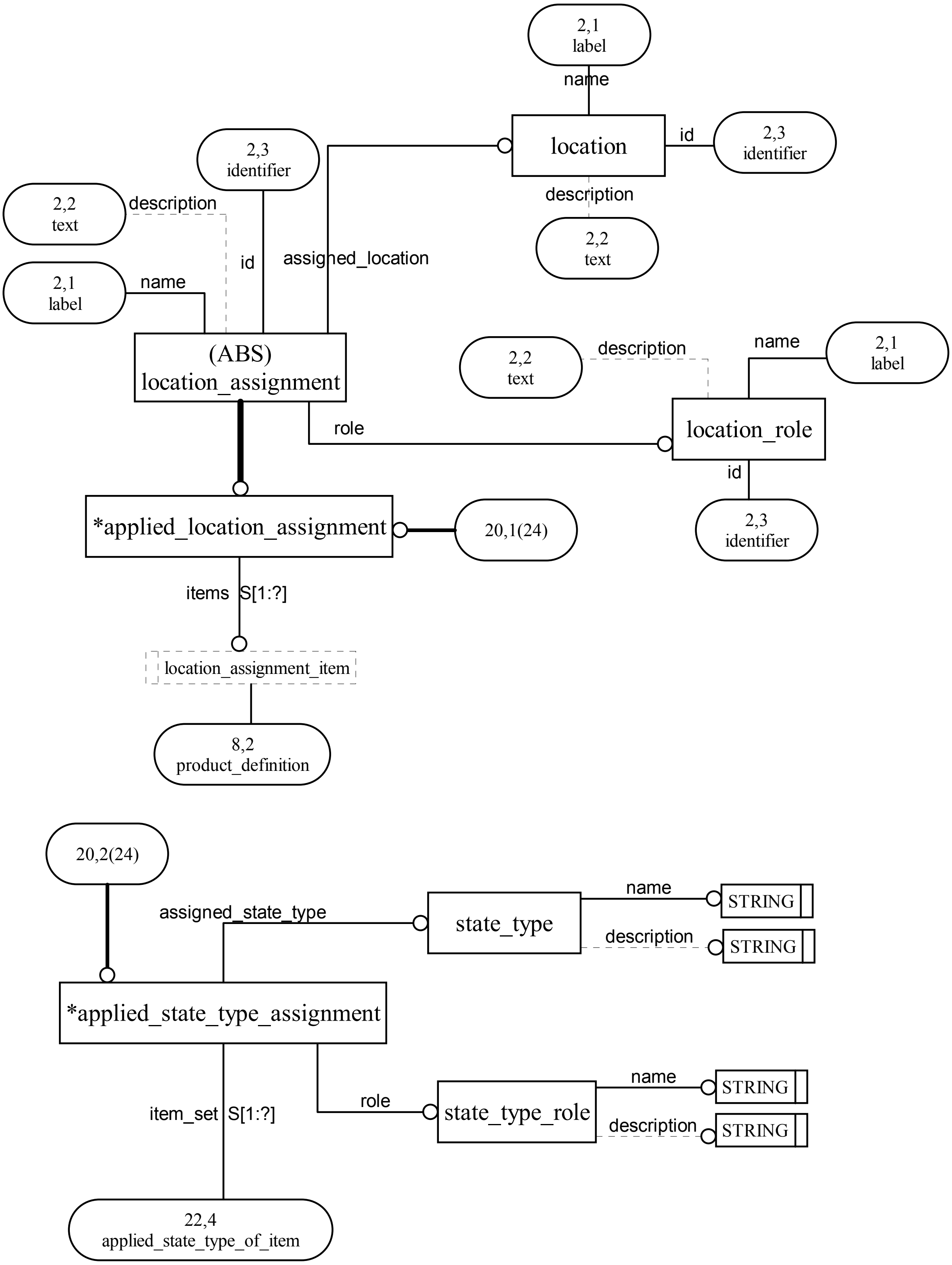

Figure E.20 — ISO13399-1 AIM EXPRESS-G diagram 20 of 31

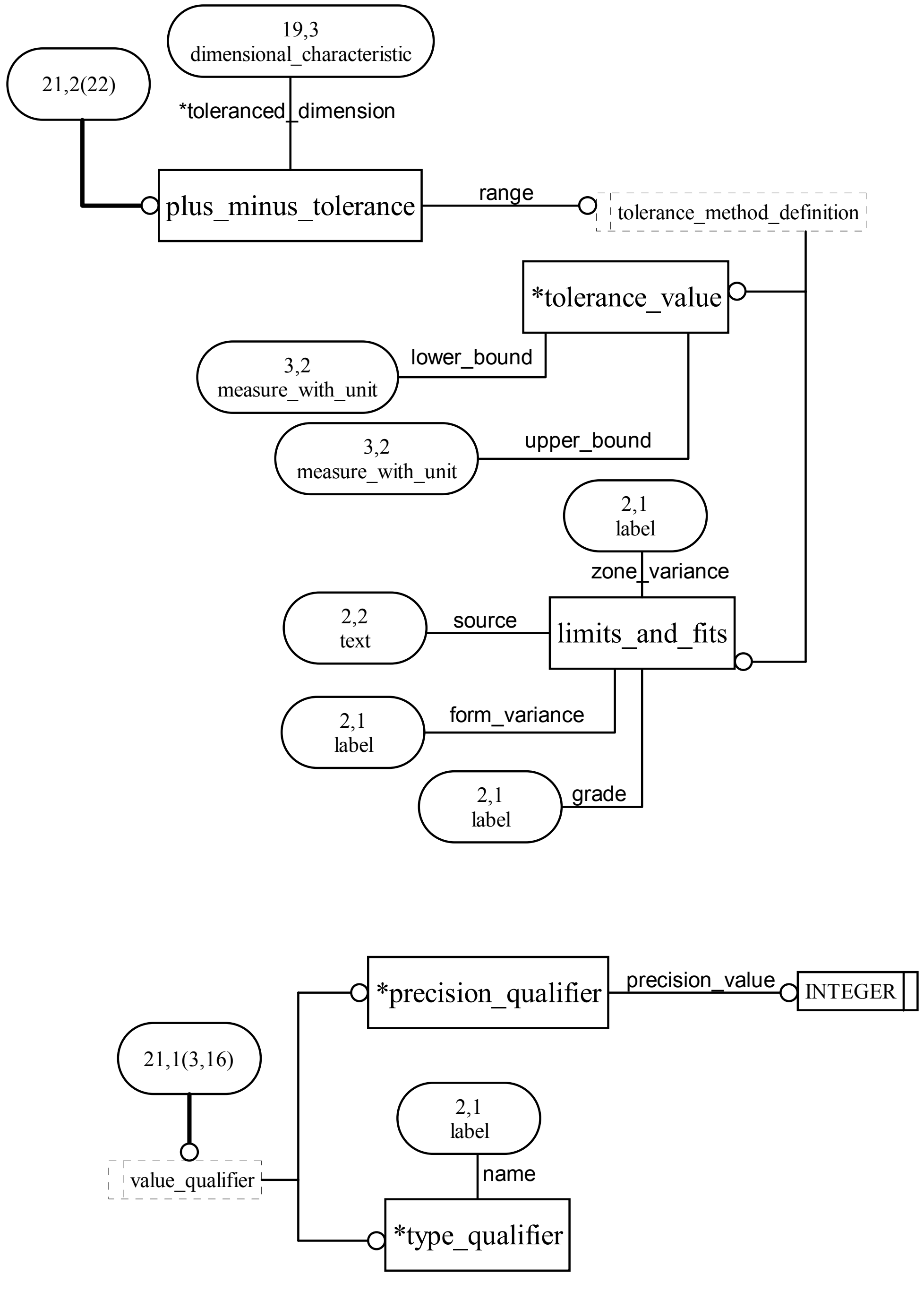

Figure E.21 — ISO13399-1 AIM EXPRESS-G diagram 21 of 31

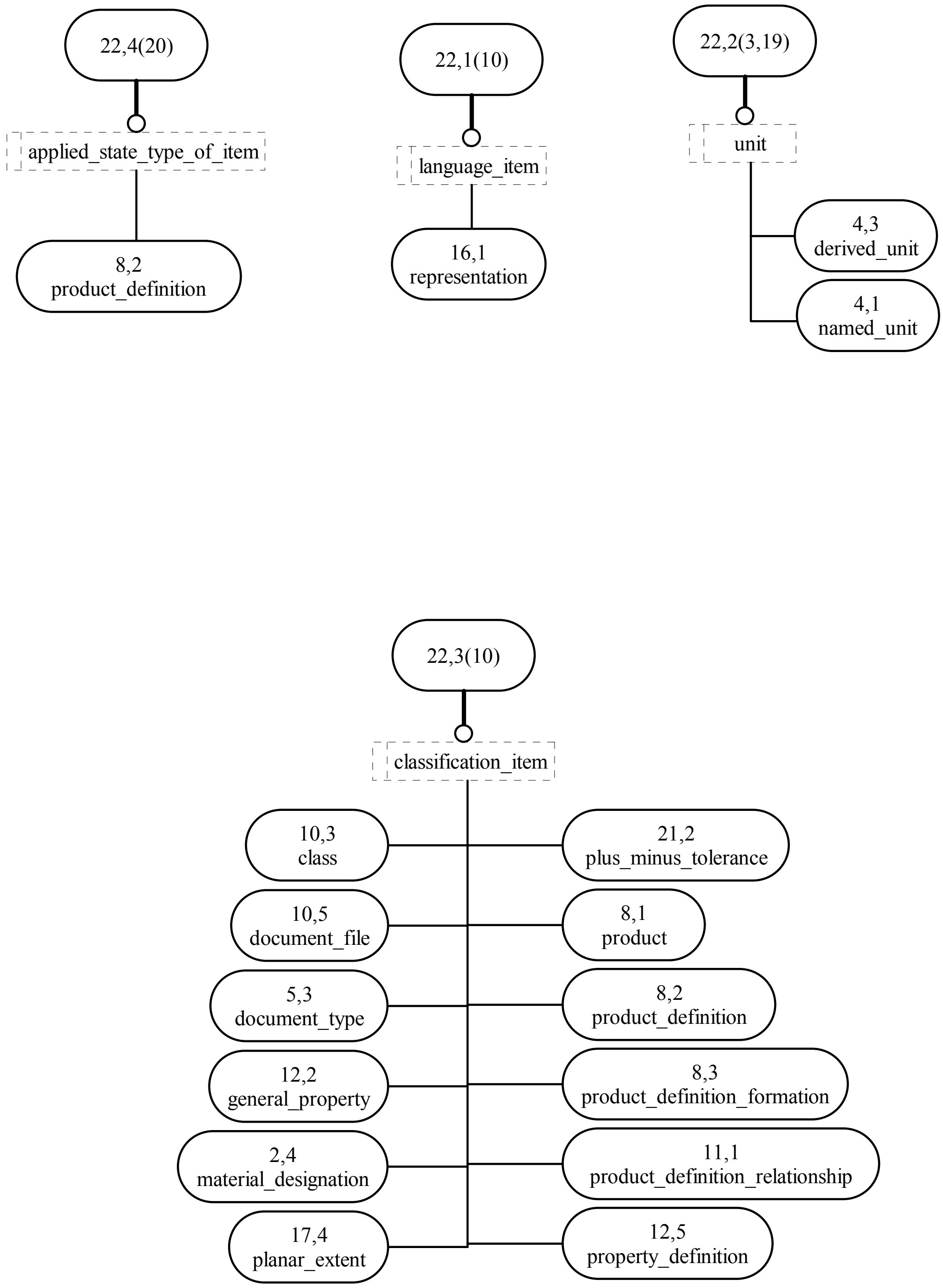

Figure E.22 — ISO13399-1 AIM EXPRESS-G diagram 22 of 31

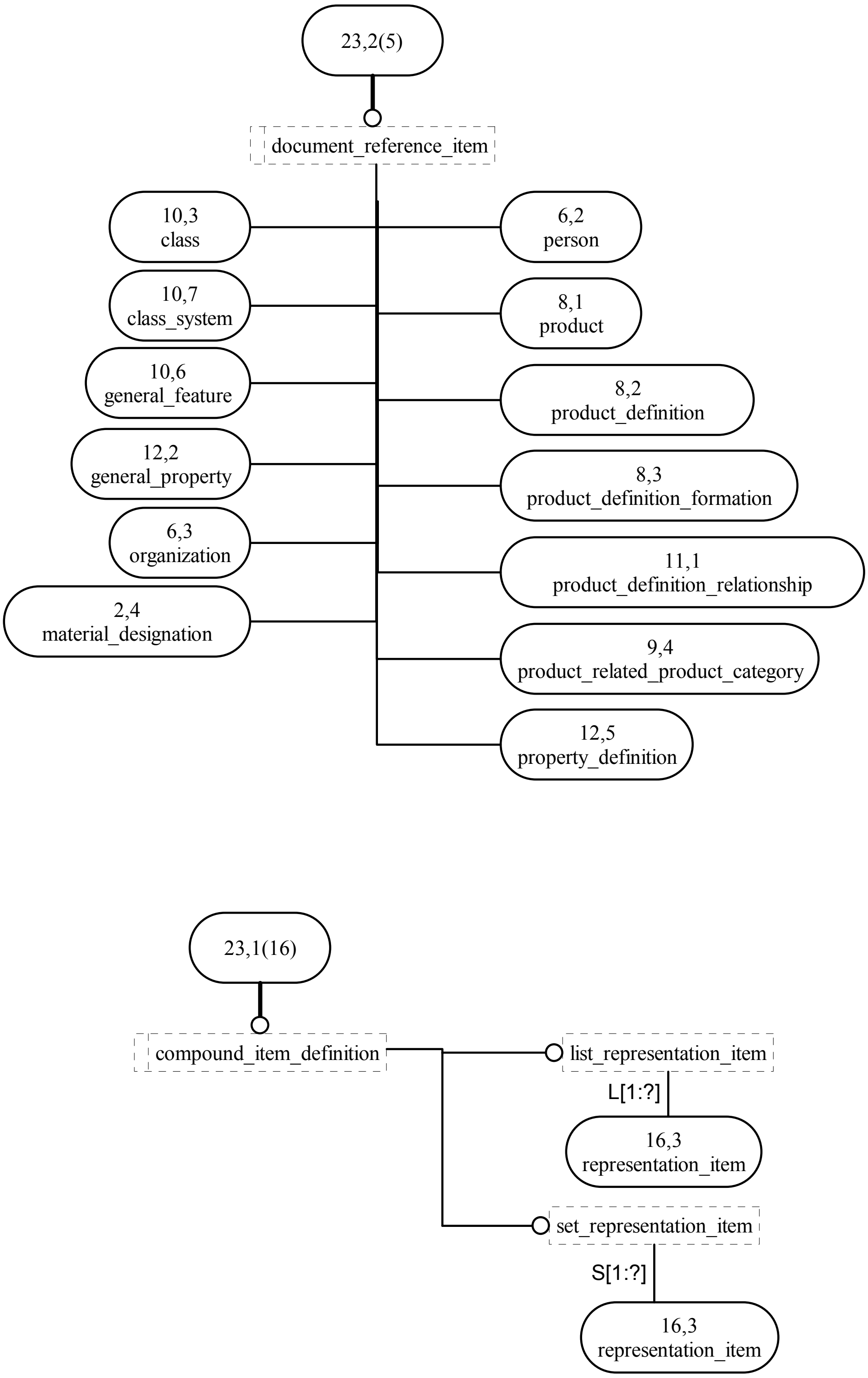

Figure E.23 — ISO13399-1 AIM EXPRESS-G diagram 23 of 31

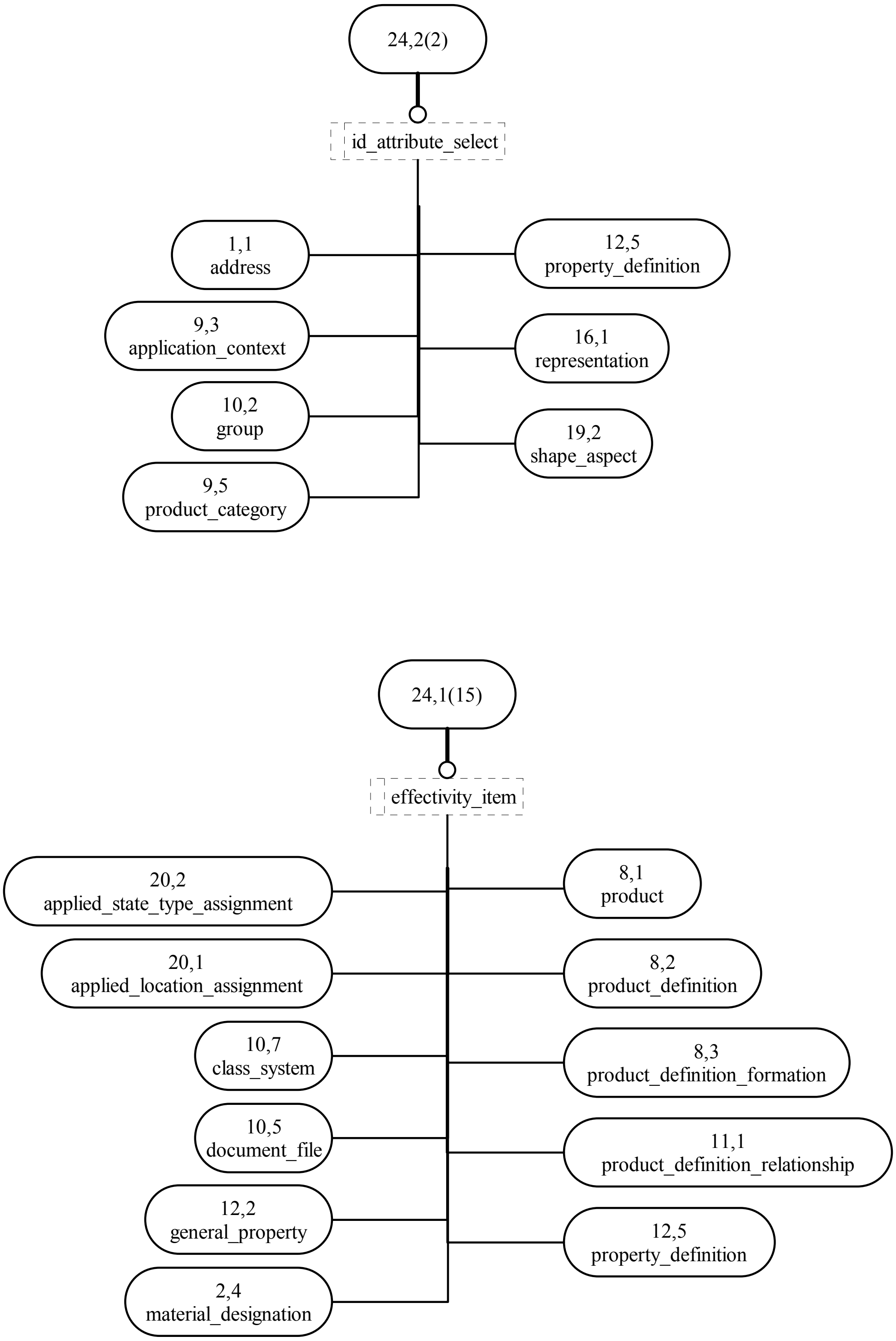

Figure E.24 — ISO13399-1 AIM EXPRESS-G diagram 24 of 31

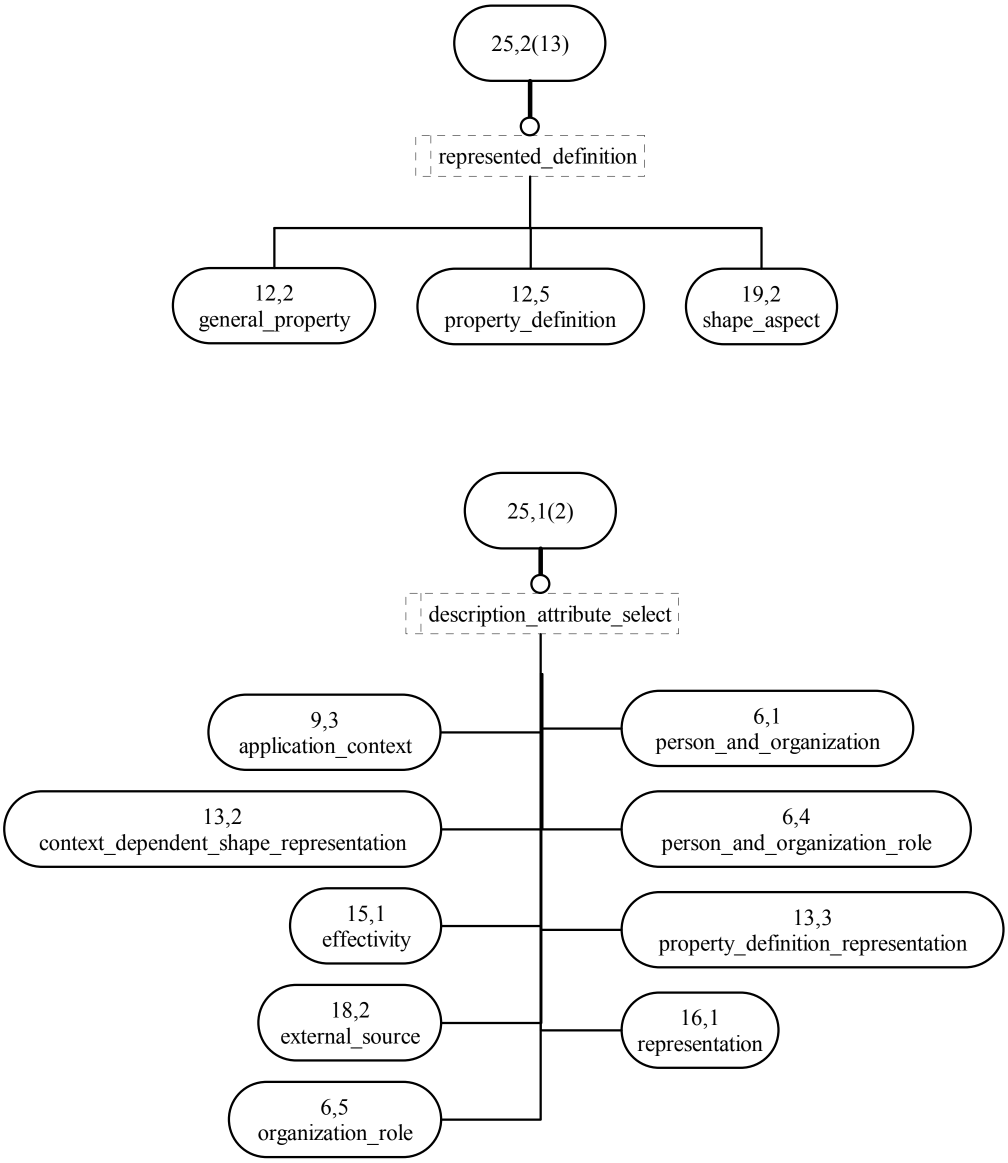

Figure E.25 — ISO13399-1 AIM EXPRESS-G diagram 25 of 31

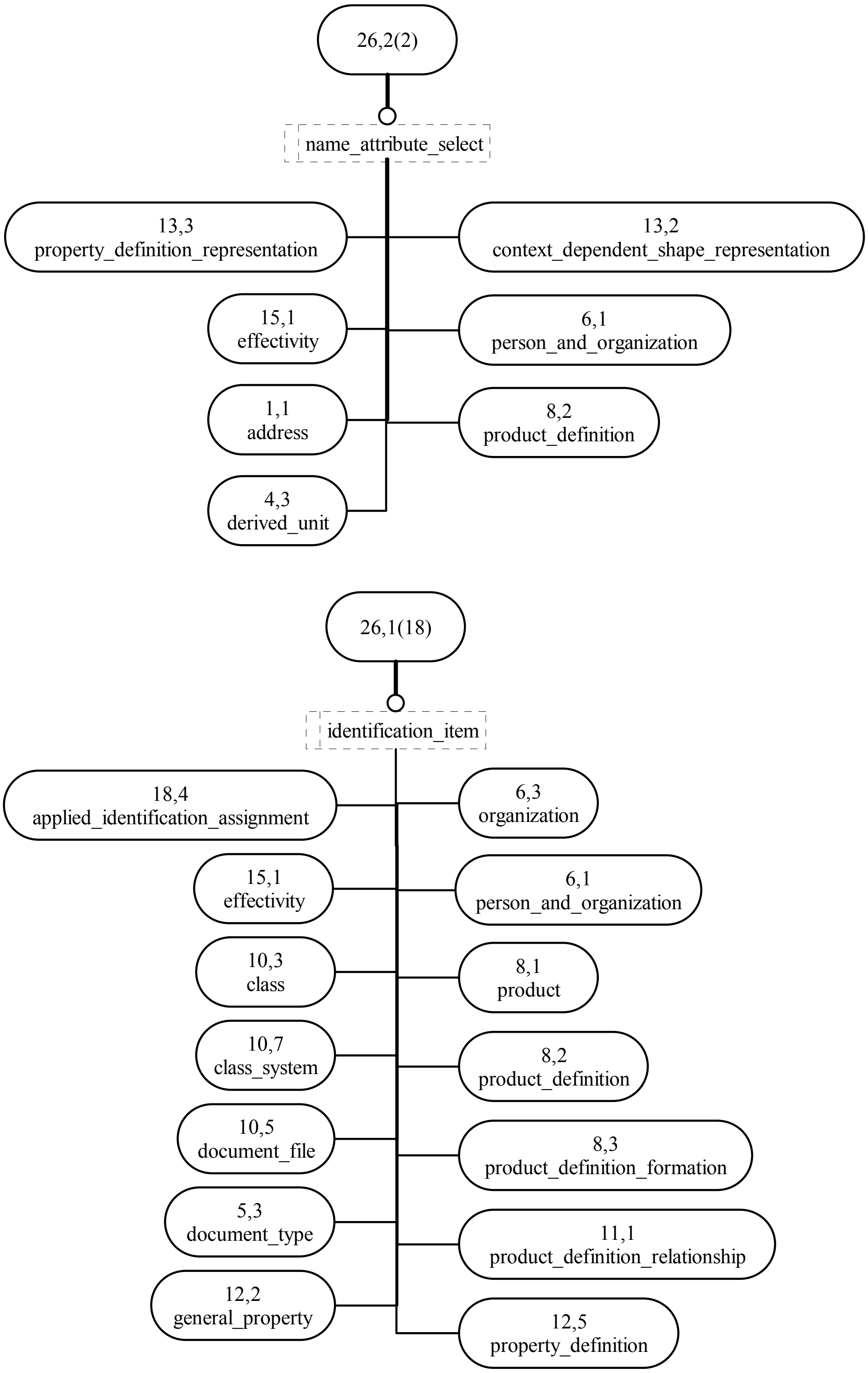

Figure E.26 — ISO13399-1 AIM EXPRESS-G diagram 26 of 31

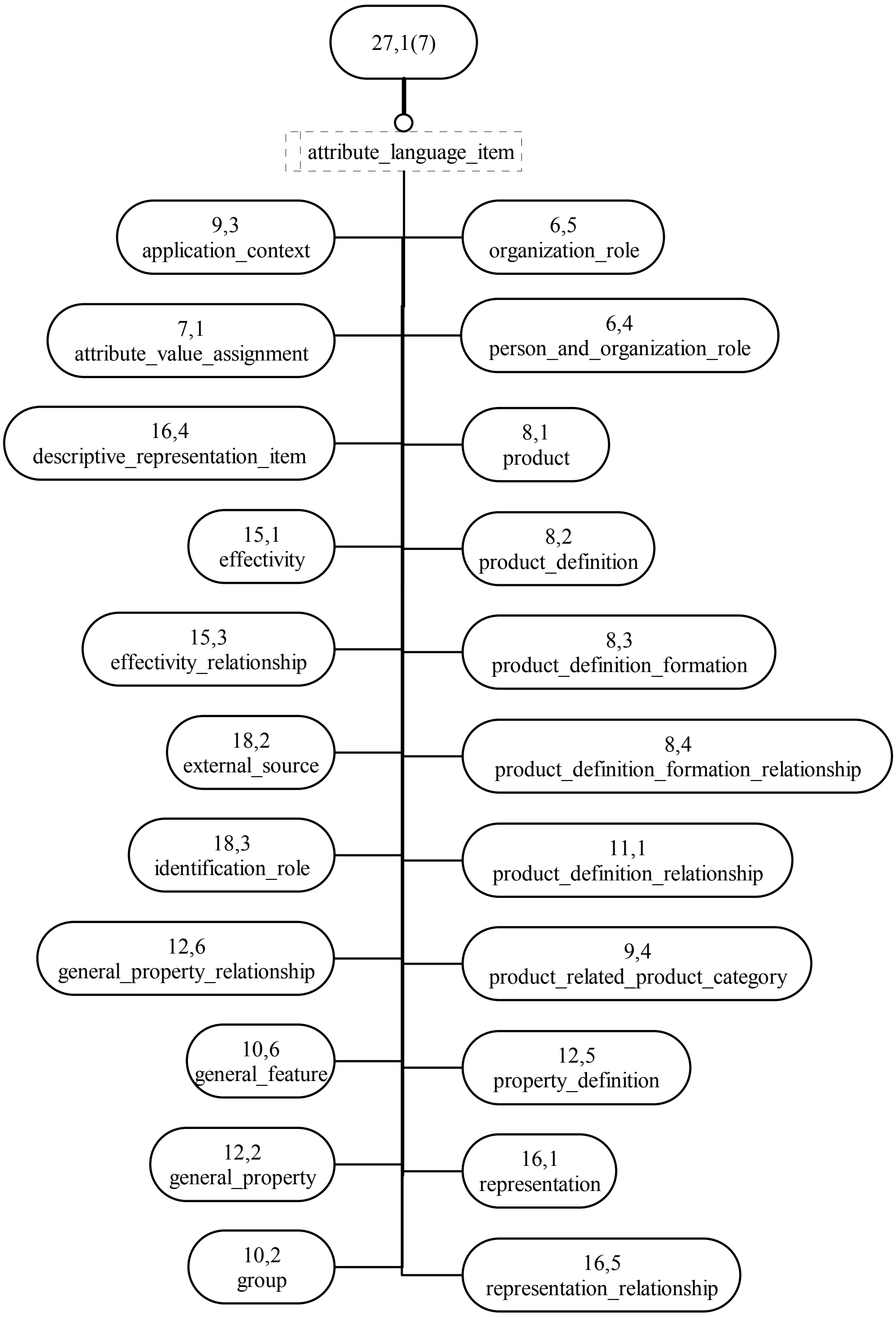
27,1(7)
attribute_language_item
9,3 application_context
6,5 organization_role
7,1 attribute_value_assignment
6,4 person_and_organization_role
16,4 descriptive_representation_item
8,1 product
15,1 effectivity
8,2 product_definition
15,3 effectivity_relationship
8,3 product_definition_formation
18,2 external_source
8,4 product_definition_formation_relationship
18,3 identification_role
11,1 product_definition_relationship
12,6 general_property_relationship
9,4 product_related_product_category
10,6 general_feature
12,5 property_definition
12,2 general_property
16,1 representation
10,2 group
16,5 representation_relationship

Figure E.27 — ISO13399-1 AIM EXPRESS-G diagram 27 of 31

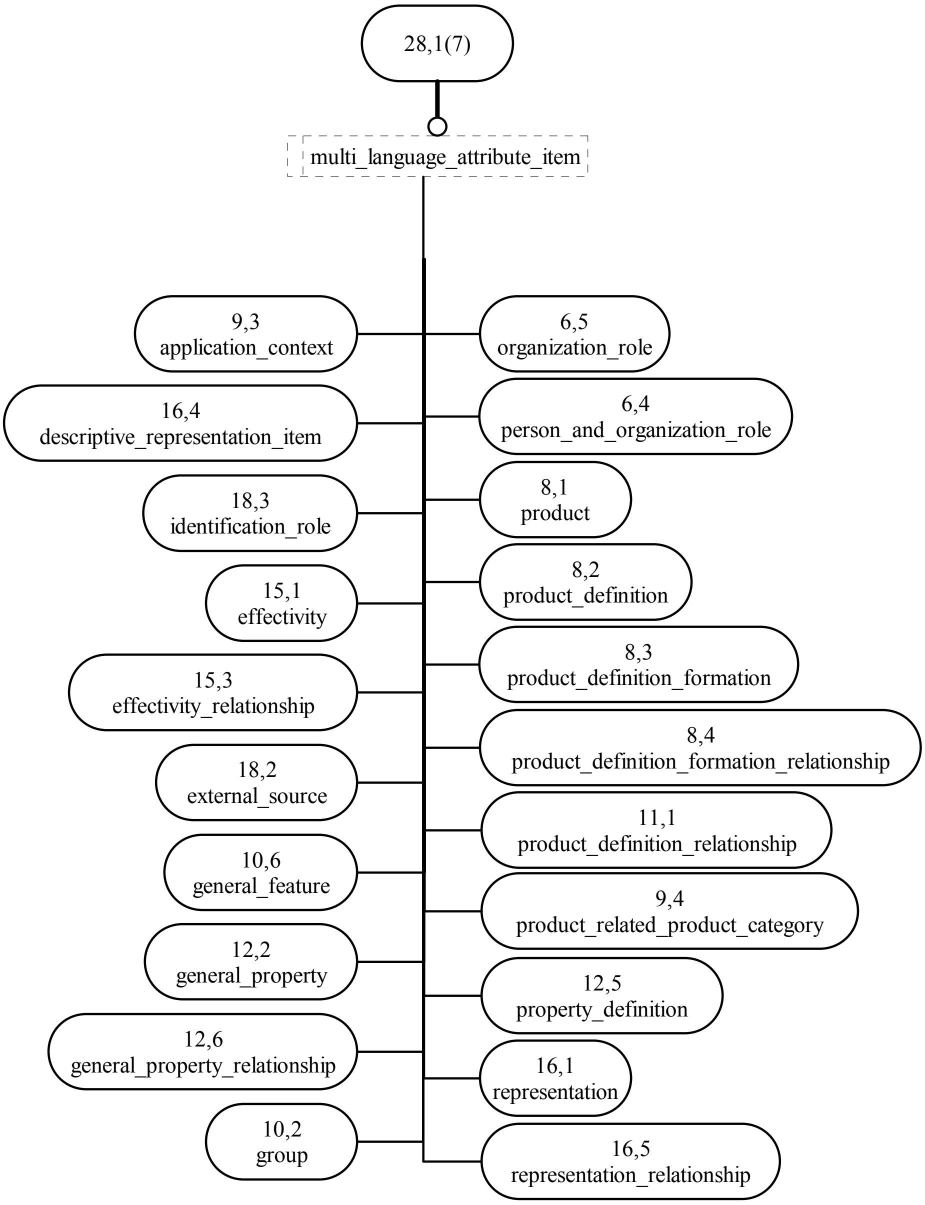

Figure E.28 — ISO13399-1 AIM EXPRESS-G diagram 28 of 31

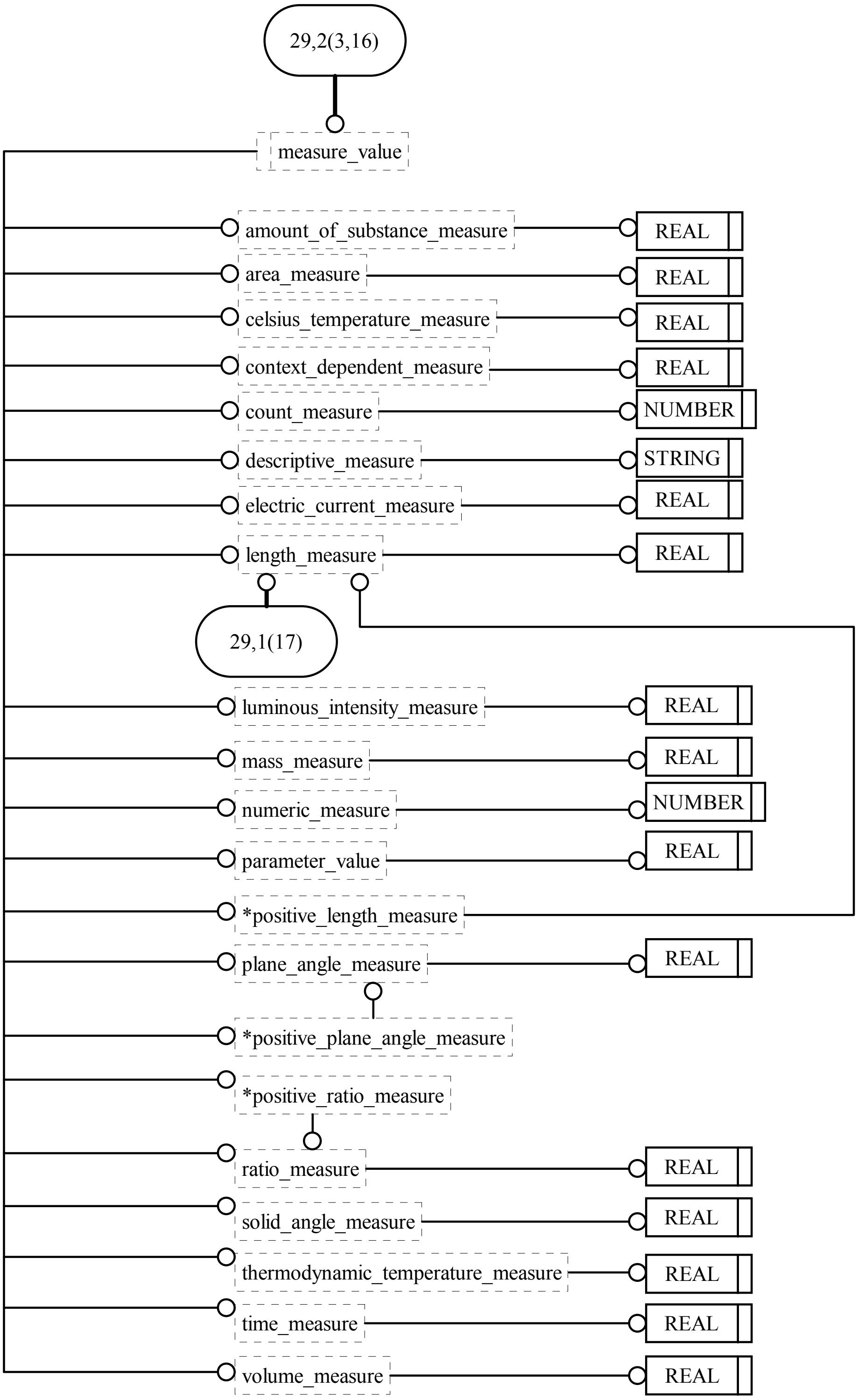

Figure E.29 — ISO13399-1 AIM EXPRESS-G diagram 29 of 31

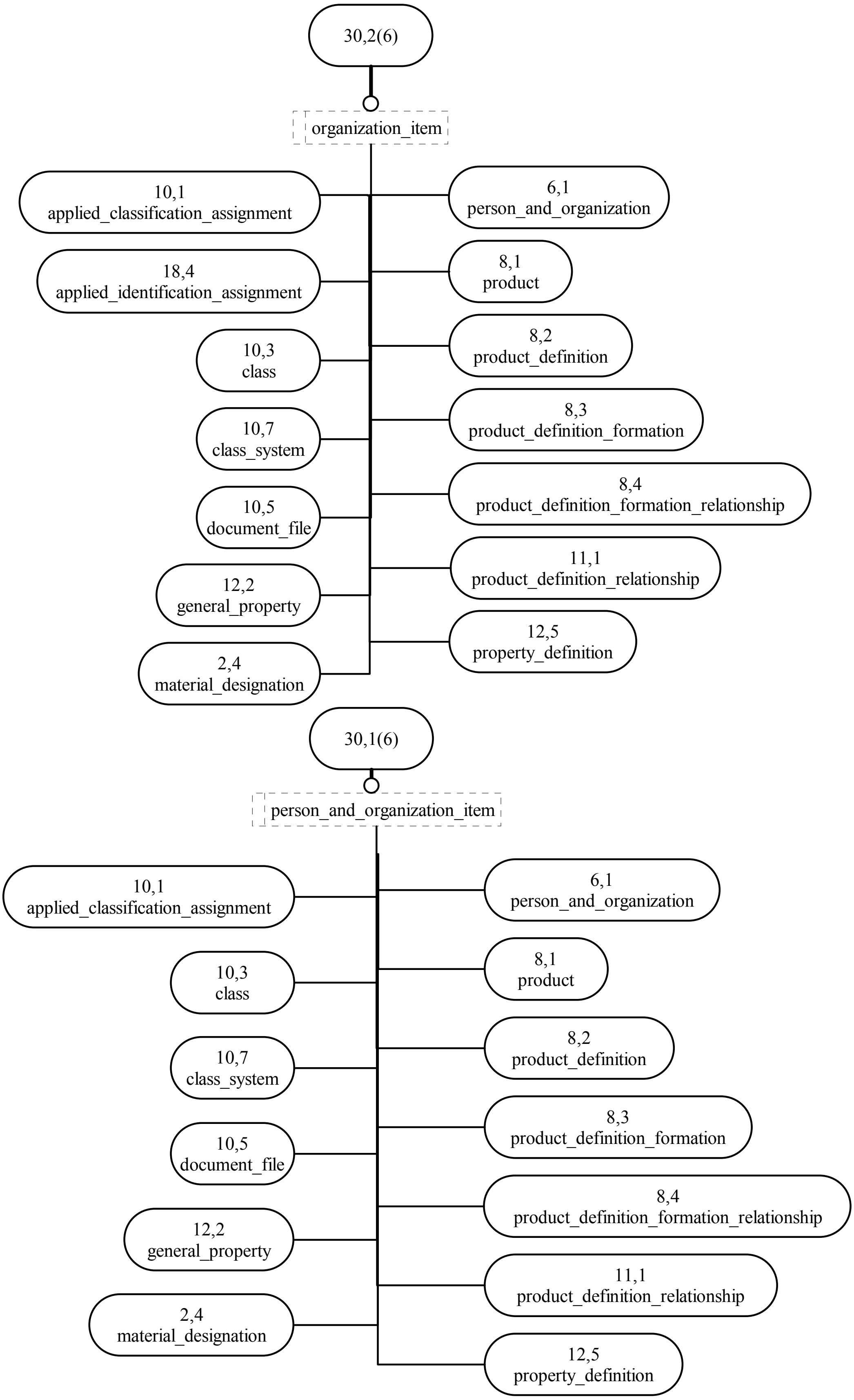

Figure E.30 — ISO13399-1 AIM EXPRESS-G diagram 30 of 31

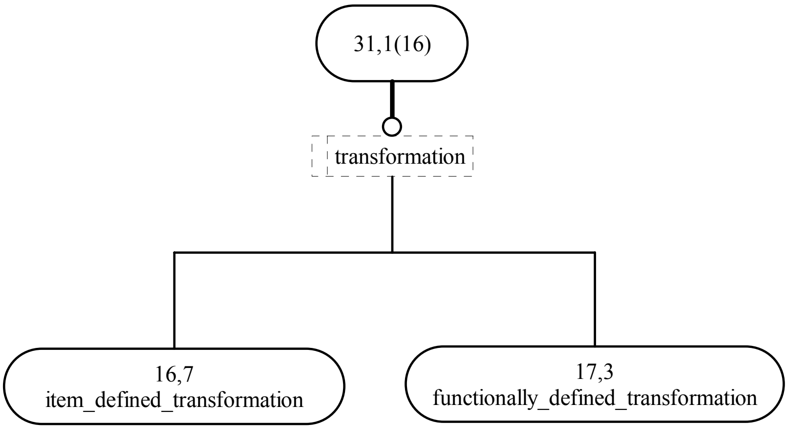
31,1(16)
transformation
16,7
item_defined_transformation
17,3
functionally_defined_transformation

Figure E.31 — ISO13399-1 AIM EXPRESS-G diagram 31 of 31

Bibliography

[1] ISO 3002-2, *Basic quantities in cutting and grinding — Part 2: Geometry of the active part of cutting tools — General conversion formulae to relate tool and working angles*

[2] ISO 8824-1, *Information technology — Open systems interconnection — Abstract Syntax Notation one (ASN.1) — Part 1: Specification of basic notation*

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

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

[5] ISO 10303-21, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Clear text encoding of the exchange structure*

[6] ISO 10303-28, *Industrial automation systems and integration — Product data representation and exchange — Part 28: Implementation methods: XML representations of EXPRESS schemas and data*

[7] ISO 13584-1, *Industrial automation systems and integration — Parts library — Part 1: Overview and fundamental principles*

[8] DIN 4000 (all parts), *Tabular layouts of article characteristics*

[9] SCHENCK, D., WILSON, P., 1994: *Information Modeling the EXPRESS way, Oxford University Press*

[10] NYQVIST, O., NIELSEN J., 2001: *Features as a Part of the Functional Interface Between Product and Resource: An Approach Based on Standards*, International CIRP Design Seminar, Design in the New Economy, 2001, KTH, Stockholm, Sweden, P. 181-186 in Proceedings

ICS 25.100.01; 35.240.50
Price based on 284 pages