INTERNATIONAL STANDARD

ISO 13584-42

First edition
1998-07-01

Industrial automation systems and integration — Parts library — Part 42: Description methodology: Methodology for structuring part families

Systèmes d'automatisation industrielle et intégration — Bibliothèque de composants —

Partie 42: Méthodologie descriptive: Méthodologie appliquée à la structuration des familles de pièces

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Reference number
ISO 13584-42:1998(E)

Contents

Page

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

© ISO 1998

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet iso@iso.ch

Printed in Switzerland

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Figures

Tables

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

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 organisations, 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 standardisation.

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

International Standard ISO 13584-42 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration,* Subcommittee SC 4, *Industrial data.*

ISO 13584 consists of the following parts, under the general title *Industrial automation systems and integration — Parts library*:

— *Part 1: Overview and fundamental principles*

— *Part 10: Conceptual description: Conceptual model of parts library*

— *Part 20: Logical resource: Logical model of expressions*

— *Part 24, Logical resource: Logical model of supplier library*

— *Part 26: Logical resource: Supplier identification*

— *Part 31: Implementation resource: Geometric programming interface*

— *Part 42: Description methodology: Methodology for structuring part families*

— *Part 101: View exchange protocol: Geometric view exchange protocol by parametric program*

— *Part 102: View exchange protocol: View exchange protocol by ISO 10303 conforming specification*

The structure of ISO 13584 is described in ISO 13584-1. The numbering of the parts of ISO 13584 reflects its structure:

— Parts 10 to 19 specify the conceptual descriptions,

— Parts 20 to 29 specify the logical resources,

— Parts 30 to 39 specify the implementation resources,

— Parts 40 to 49 specify the description methodology,

— Parts 50 to 59 specify the conformance testing,

— Parts 100 to 199 specify the view exchange protocol,

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

— Parts 500 to 599 specify the standardised content.

Should further parts of ISO 13584 be published, they will follow the same numbering pattern.

Annexes A, B and C form an integral part of this part of ISO 13584. Annexes D, E, G and H are for information only.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Introduction

ISO 13584 is an International Standard for the computer-interpretable representation and exchange of part library data. The objective is to provide a neutral mechanism capable of transferring parts library data, independent of any application that is using a parts library data system. The nature of this description makes it suitable not only for the exchange of files containing parts, but also as a basis for implementing and sharing databases of parts library data.

ISO 13854 is organised as a series of parts, each published separately. The parts of ISO 13854 fall into one of the following series: conceptual descriptions, logical resources, implementation resources, description methodology, conformance testing, view exchange protocol, and standardised content. The series are described in ISO 13584-1. This part of ISO 13584 is a member of the description methodology series.

This part of ISO 13584 provides rules and guidelines for library data suppliers to create hierarchies of families of parts according to a common methodology intended to enable multi-supplier consistency. These rules pertain to the following: the method for grouping parts into families of parts to form a hierarchy; the dictionary elements that describe the families and properties of parts.

This part of ISO 13584 refers as a normative reference to the data model that specifies the exchange of dictionary data. This EXPRESS specification was developed as a common model for ISO 13584 and IEC 61360. It is intended to be published as IEC 61360-2. For convenience this common model is provided in this part of ISO 13584 as an informative annex that duplicates the normative content of IEC 61360-2. This part of ISO 13584 also provides the mapping of the concepts described here onto the common model.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Industrial automation systems and integration — Parts library — Part 42: Description methodology: Methodology for structuring part families

1 Scope

This part of ISO 13584 specifies the principles that shall be used for defining families of parts and properties of parts to fully characterize parts and associated properties.

The rules and guidelines provided in this part of ISO 13584 are mandatory for the standardisation committees responsible for creating standardised identification hierarchies.

The use of these rules by suppliers and users is recommended as a methodology for building their own hierarchies.

The following are within the scope of this part of ISO 13584:

— the rule to group parts into generic families of parts and simple families of parts;

— the rules for the choice of the appropriate properties that shall be associated with the families of parts;

— the attributes that shall be provided by library data suppliers to describe the families and properties of parts.

— the specifications of those attributes in the EXPRESS information model that provide for the exchange of such dictionary data.

NOTE 1 The EXPRESS information model for the exchange of dictionary data is defined in IEC 61360-2.

NOTE 2 The content of this EXPRESS information model is in the informative Annex D of this part of ISO 13584 that duplicates the normative content of IEC 61360-3.

The following are outside the scope of this part of ISO 13584:

— libraries of assembled parts with aggregate structure (Level 3 libraries);

— the description of the parts themselves;

— the descriptions of the functional models that may refer to some family of parts;

— the description of tables, program libraries and documents that may refer to some family of parts;

— the description of the systems intended to manage parts libraries; and

— the EXPRESS resource constructs for references between libraries.

NOTE The EXPRESS resource constructs for tables, programs libraries, documents and for references between libraries are defined in ISO 13584-24.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

The structure of the information and the methodology defined in this part of ISO 13584 enable the following:

— integration in the same data repository of different parts libraries originating from different library data suppliers with an uniform access mechanism provided by a dictionary;

— integration in the same data repository of different definitions of materials originating from different suppliers with an uniform access mechanism provided by a dictionary;

— referencing another supplier library assumed to be available on the receiving system;

— referencing a standardised identification hierarchy when such a hierarchy exists;

— definition by an end-user of a local classification or search hierarchy, and the mapping of these hierarchies onto the supplier libraries available on its system.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 13584. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 13584 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 31-0:1992, *Quantities and units — Part 0: General principles.*

ISO 4217, *Codes for the representation of names of currencies and funds.*

ISO 6093:1985, *Information processing — Representation of numerical values in character strings for information interchange.*

ISO 8601:1988, *Data elements and interchange formats — Information interchange — Representations of dates and times.*

ISO/IEC 8824-1:1995, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

ISO 8879-1:1987, *Information processing — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1.*

ISO 8879:1986, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML).*

ISO 9735:1988, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules.*

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

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

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

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

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

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

ISO 12083:1992, *Information and documentation — Electronic manuscript preparation and markup.*

ISO/IEC 11179-3:1994, *Information technology — Specification and standardization of data elements — Basic attributes of data elements.*

ISO 13584-10:—[1], *Industrial automation systems and integration — Parts library — Part 10: Conceptual model.*

International Classification of Standards (ICS), 1993.

IEC 61360-1:1995, *Standard data element types with associated classification scheme for electric components — Part 1: Definitions — Principles and methods.*

IEC 61360-2:—[1], *Standard data element types with associated classification scheme for electric components — Part 2: EXPRESS Dictionary Schema.*

IEC 61360-3:1995, *Standard data element types with associated classification scheme for electric components — Part 3: Maintenance and validation procedures.*

IEC 61360-4:1997, *Standard data element types with associated classification scheme for electric components — Part 4: IEC reference collection of standard data element types, component classes and terms.*

3 Terms and definitions

For the purposes of this part of ISO 13584, the following terms and definitions apply.

3.1 Terms and definitions from ISO 10303-1

For the purposes of this part of ISO 13584, the following terms and definitions given in ISO 10303-1:1994 (which are repeated below for convenience) apply.

3.1.1

information
facts, concepts or instructions.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

1 To be published

3.1.2

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

3.2 Terms and definitions from ISO 13584-1

For the purposes of this part of ISO 13584, the definitions given in ISO 13584-1 for the following terms apply.

— Library data supplier;

— Part;

— Parts library;

— Resource construct;

— Supplier library.

NOTE These definitions are duplicated in the informative Annex H of this part of ISO 13584.

3.3 Terms and definitions from ISO 13584-10

For the purposes of this part of ISO 13584, the definitions given in ISO 13584-10 for the following terms apply.

— Applicable property;

— Characteristic of a part (part characteristic);

— Context dependent characteristic;

— Context parameter;

— Family of parts;

— Generic family of parts;

— Property;

— Simple family of parts;

— Visible property.

NOTE These definitions are duplicated in the informative Annex I of this part of ISO 13584.

3.4 Other terms and definitions

For the purposes of this part of ISO 13584, the following apply.

3.4.1

basic semantic unit
the entity that provides an absolute and universally unique identification of certain objects of the application domain (e.g. classes, data element types).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

3.4.2

class valued property
a property that has one single value for a whole family of parts. Its value is not defined individually for every single part of that family, but for the family itself.

NOTE Class valued properties may be used to capture some commonality between different families when such a commonality is not captured by the hierarchy structure (see example of RULE 5b).

3.4.3

common dictionary schema
the information model for a dictionary, using the information modelling language EXPRESS, resulting from a joint effort between ISO TC 184/SC4/WG2 and IEC SC3D;

NOTE The common dictionary schema is specified in IEC 61360-2, and its content is provided in the informative annex D of this part of ISO 13584.

3.4.4

data element type
unit of data for which the identification, the description and value representation have been specified.

3.4.5

data type
the specification of the type of a property that specifies the allowed set of values of that property.

3.4.6

dictionary data
the set of data that describes hierarchies of families of parts and properties of these parts.

NOTE The dictionary data should be exchanged using the common dictionary schema.

3.4.7

dictionary element
the set of attributes that constitutes the dictionary description of certain objects of the application domain (e.g. classes, data element types).

3.4.8

quantitative data element type
data element type with a numerical value representing a physical quantity, a quantity of information or a count of objects.

3.4.9

non-quantitative data element type
data element type that identifies or describes an object by means of codes, abbreviations, names, references or descriptions.

3.4.10

property
an information that may be represented by a data element type.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

3.4.11

standardized identification hierarchy
dictionary data that is defined by a standardisation organisation.

4 Abbreviated terms

For the purpose of this part of ISO 13584 the following abbreviated terms apply.

— BSU : Basic Semantic Unit;

— DET: Data Element Type;

— ICS: International Classification of Standards;

— SI: Système International d'Unités (International System of Units).

5 Description of a hierarchy of families of parts

5.1 Simultaneous description of families of parts and parts properties

The hierarchy shall be organised as a tree-structure with single inheritance. At each level of the hierarchy, the sub-families of a generic family of parts shall be mutually exclusive. The hierarchy shall include two sections:

— the upper section classifies technical fields and so defines the first levels of generic families of parts. These generic families of parts are used for classification purposes. They are not intended to be instanciated.

— the lower section splits these families into generic sub-families to obtain simple families of parts. In choosing these families the instanciation rule (RULE 3) is important. These families are intended to be instanciated.

Both the standardised identification hierarchy and the supplier hierarchies shall simultaneously define the following:

— (hierarchical) families of parts;

— (non hierarchical) properties.

The meaning of each family of parts and of each property shall be such that a human observer shall be able to determine, for a given part:

— those families of parts to which it belongs and those to which it does not belong;

— that characteristic of the part to which every applicable property corresponds.

The simultaneous definition of families of parts and of properties related to every family of parts improves their definition. The composition of a family of parts becomes clearer through the properties that are related; the meaning of a property is explained by the family of parts defining its field of application.

By defining families of parts through a hierarchy with inheritance, the definition of each property can be factored to the highest family level for which this property is meaningful and may be applicable in some sub-families.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

5.2 Applicable and visible properties

The properties defined at each (generic) family level are divided into two categories:

— the applicable properties corresponding to a characteristic property that every part (belonging to the generic or simple family of parts) shall possess;

— the visible properties corresponding to a characteristic property that the parts (belonging to the generic or simple family of parts) may or may not possess according to the sub-families they belong to.

Only those properties that are specified as visible for some family (possibly through inheritance) may be also specified as applicable.

5.3 Purpose of the standardised identification hierarchy

Due to the diversity of the real world, the families of parts and the properties forming the standardised identification hierarchy are not intended to exhaustively characterise and describe every part. It is only aimed at identifying and progressively characterising the common properties to provide:

— multi-supplier search;

— partial multi-supplier characterisation;

— some interchangeability between different parts.

5.4 Use of the standardised identification hierarchy

When describing his own parts a supplier may choose:

— to make reference to standardised families of parts and properties, when they are defined by the standardised identification hierarchy;

— to add his own families of parts and properties following guidance of the rules stated in this standard.

Thus every supplier can connect his families of parts to the level(s) of the standardised identification hierarchy that seem to be the most suitable and may complete the standardised identification hierarchy with his own hierarchy. The higher the level of the connection, the less the suppliers will be able to take advantage of the standardised identification hierarchy and the corresponding properties.

5.5 Class valued property

Due to the diversity of the perspectives that may be adopted for a set of objects of the real world, different structures for a generic family of parts may emerge (see example in 6.1.5).

In this case, the structure that permits the maximum applicability of the factored properties shall be selected for the standardised identification hierarchy. Each other perspective considered relevant is represented as a property that is assigned one single value to each simple family of the subtree and is valid for the whole family. The assignment of the property to a family may be defined at any level of the subtree. The value assigned this way is inherited by all the subfamilies.

The data type of such a property is represented as a set of unique codes. Each code is associated with different human readable and translatable representations.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

5.6 Extensions of the common dictionary schema addressing IEC 61360 needs

The common dictionary schema combines the requirements of IEC 61360 and ISO 13584. Therefore it contains resources to accommodate the specific requirements of both International Standards. These resources are provided either as an optional capability or as subtypes of the types defined to fulfil the common requirements. The IEC 61360 specific extensions shall not be used for the definition of the standardised identification hierarchy conforming to this part of ISO 13584. These extensions are as follows:

a) In the **item_class** entity the optional **value_code** attribute;

b) In the **value_domain** entity the optional **terms** attribute.

5.7 Extension of the common dictionary schema addressing ISO 13584 needs

This part of ISO 13584 addresses two requirements:

— the definition of a methodology for defining standardised identification hierarchies or supplier hierarchies of parts and the properties associated with these parts.

— the specification of an information model that provides for the exchange of dictionary data that describes hierarchies of families of parts and properties of these parts.

Some resources in the common dictionary schema are intended to be used when modelling and exchanging other types of data (e.g., library content) as specified by other parts of ISO 13584. These resources are provided either as optional capabilities or as subtypes of the types defined to fulfil the requirements of this part of ISO 13584.

These extensions shall not be used when defining the standardised identification hierarchy or supplier hierarchies. These resources are as follows (the specification of these resources is defined in the informative Annex E) :

a) content_item;

b) supplier_related_BSU;

c) class_related_BSU;

d) supplier_BSU_relationship;

e) class_BSU_relationship.

5.8 Compatibility between IEC 61360 and ISO 13584

The scope of IEC 61360 is to specify standard data element types with associated classification schema for electric components. This is done through the specification of the principles in IEC 61360-1, the common dictionary schema in IEC 61360-2, the set of data elements and an initial hierarchy for electronic components in IEC 61360-4, and through the specification of a Maintenance and Validation Agency intended to up-date this set of data elements and its hierarchy, to be published as IEC 61360-4.

The scope of ISO 13584 is twofold:

— to specify and to exchange identification hierarchies of parts of any kind;

— to specify and to exchange libraries of parts that may contain both dictionary data and class extensions (i. e., the set of allowed parts within a family of parts).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

The exchange context of identification hierarchies is specified, for both Standards, by the common dictionary schema. Therefore exchange contexts of identification hierarchies are completely compatible. The only restriction is that ISO 13584 conforming hierarchies shall not use the two optional resources specified in 5.6 and that the date of original definition, date of current version and date of current revision that are defined as OPTIONAL in the common dictionary schema are mandatory for ISO 13584 conforming hierarchies. Moreover, the use of the ISO/IEC ICS hierarchy as a root hierarchy in the user system (see 5.1.2) enables the integration of the IEC 61360-4 hierarchy[2] and of the various standardised identification hierarchies intended to be developed by the ISO Standardisation Committees.

For the exchange of electric component libraries referring to the IEC 61360-4 dictionary data, the electric component suppliers can use the exchange format specified in ISO 13584-24 (in preparation). This exchange format enables, through the **component_case_of** entity, to refer to any standardised identification hierarchy conforming to the common dictionary schema.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

6 Rules for creating hierarchies of families of parts

To simplify the analysis when creating a standardised identification hierarchy and supplier hierarchies, the following rules, that allow the splitting of the analysis into two phases, shall be applied. The first set of rules refers to the choice of the family hierarchy. The second set of rules refers to the association of properties with families of parts.

6.1 Choice of family hierarchy

6.1.1 Field of application

RULE 1 - Field covered by the hierarchy.

The standardised identification hierarchy shall be able to cover all the parts dealt with in ISO or IEC standards.

A part supplier hierarchy shall be able to cover all the parts supplied by this supplier.

6.1.2 Upper section of the hierarchy

RULE 2 - Classification.

The upper section of the standardised identification hierarchy shall be based upon the International Classification of Standards (ICS).

Properties can be related to the upper section according to the factoring rule (RULE 8).

NOTE 1 The ICS provides a common high level classification that shall be used or referred to by the standardisation committees through at least the root of the classification that corresponds to their scope. For those standardisation committees whose scope cannot be mapped onto one class of the ICS classification, new classes shall be created as subclasses of an existing class that encompasses their scope.

NOTE 2 When the scope of a standardisation committee encompasses a whole subtree of the ICS classification, this committee may redefine the classification subtree for its own purpose provided that it refers to the root of that subtree.

NOTE 3 The role of the standardised identification hierarchy is to improve the definition of properties. It is not intended to specify a selection mechanism.

2 The IEC 1360-4 hierarchy covering several subclasses of the ISO/IEC ICS hierarchy, the former shall be connected directly to the root of the latter

6.1.3 Lower section of the hierarchy

RULE 3 - Instanciation rule.

Below the upper section of the hierarchy, a generic family of parts shall be created only when it is possible and advisable to associate a functional model (e.g. geometrical model, schematic model) to such a family of parts, i.e. when an user could reasonably choose a generic part of such a family to represent a significant state (phase) of his design process.

EXAMPLE To group screws for metal, wood and sheet metal within one family "screws", is contradictory to the rule of instanciation. During a design process a designer would never select a screw without knowing if this screw shall be a machine screw or a wood screw. Therefore, it is necessary to define below the upper section "bolts/screws/studs" different generic families of parts for the different types of screws.

6.1.4 Simple family of parts

In paper documents, standards or catalogues, sets of parts are sometimes grouped even if the sets of properties applicable to each part of the group are not exactly the same. This shall be prohibited for a simple family of parts.

RULE 4 - Homogeneity of simple family of parts.

Every property applicable to a simple family of parts shall be applicable for every part within this family.

6.1.5 Multiple perspectives on the hierarchy

It may appear that at some point in the hierarchy several properties of a family of parts can be used to define a substructure for that hierarchy.

RULE 5a - Maximum applicability

When the application of the instanciation rule (RULE 3) leads to different possible structures of some generic families depending upon a point of view, the structure that enables the maximum applicability of the factored properties shall be selected.

RULE 5b - Class valued properties

The other points of views considered essential for selection purposes – not selected however for the structure according to the maximum applicability rule (RULE 5a) – shall be represented by means of class valued properties (see 5.5). Those are intended to be queried by the user to select a set of families that correspond to the other points of view.

RULE 5c - Class valued assignment level

A class valued property shall be assigned at the level of a family where that property applies for *every* simple family of the corresponding subtree and does not apply for the immediate parent family.

EXAMPLE The family of circular bearings may split up into ball, needle and conical bearings or into sealed bearings and non-sealed bearings. The first perspective shall be selected for structuring purposes (RULE 5a).A class valued property, named "is_sealed", that is intended to take a constant BOOLEAN value in all the lower simple families shall be visible at the circular bearing family level (RULE 5b).That property shall be assigned at the circular bearing family level if the "is_sealed" property does not apply to bearings that are non-circular bearings. That property shall be assigned at least at the bearings level if the "is_sealed" property applies to any kind of bearings (RULE 5c).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

6.2 Association of properties

6.2.1 Properties to be considered

RULE 6 - Choice of properties.

> As a minimum, those properties that characterise a generic family of parts and can be used when searching parts for every sub-family of this generic family of parts, shall be associated with the standardised identification hierarchy.
>
> Those properties that are not used (or rarely used) for searching purposes can be added as appropriate.

6.2.2 Semantic identification of properties

The following rule gives two criteria for deciding when two properties have the same semantics.

RULE 7 - Semantic identification

> Two characteristic properties of two different parts shall be factored to a higher level of the hierarchy, if and only if, they are considered to have the same semantic meaning. Such a decision shall be justified by satisfying one of the two following criteria:

— interchangeability criterion: the two parts can, in some circumstances, be interchangeable and when interchanged the two characteristic properties shall have identical values;

— criterion of homogeneity in processing: the two properties play an identical role with respect to some process, automatic or non automatic, that might be performed over a set of parts.

Otherwise, two different properties shall be defined.

EXAMPLE 1 The properties thread diameter of hexagon head screws and cylindrical head screws fit the interchangeability criterion.

EXAMPLE 2 The properties mass (calculation of the mass of a part or its inertia), designation (for part lists), diameter of conical attachment of tools for automatic tool exchange or the outside diameter of cylindrical electronic components for automatic placement fit the criterion of homogeneity in automatic processing.

EXAMPLE 3 A thread diameter of a screw for wood or of a screw for metal are characteristic properties which shall not be the same property.

6.2.3 Factoring rule

It is not possible in every case to define a hierarchy in which:

— every property having the same semantics in several sub-families is factored (i.e. defined only once and inherited) as an unique property at the level of the ascendant family;

— any property defined at the level of a generic family of parts applies actually (i.e. has a value) to every part in every sub-family.

To create an intermediate family in order to factorise a property may be prevented by the instanciation rule (RULE 3) that is used for defining the family hierarchy.

The following rule provides guidelines for systematically factoring of properties having the same semantics while maintaining the instanciation rule.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

RULE 8 - Applicability of inherited properties.

Two properties having the same semantics (RULE 7) in two families shall be factored as an unique property defined at the level of the common ascendant family. If this property does not apply to some sub-families, it shall be defined as a visible property. This property may be specified as applicable in the various sub-families where it is visible. The definition of this property shall be such that for any sub-family there is no doubt about its applicability, and when it applies, there is no doubt about the characteristic property to which it corresponds.

EXAMPLE 1 The property "material", the definition of which would be "material from which the part is made" cannot be factored; we do not know for example to which property it corresponds for a turning tool with carbide brazed tip.

EXAMPLE 2 The property "single material", whose definition would be "property only applicable to parts made from one single material; the value of this property is the code of the single material", corresponds with the semantic identification rule (RULE 7).

7 Dictionary elements that describe properties of parts

Following the conceptual model defined in ISO 13584-10, the properties of a part are classified into:

— part characteristics

— context dependent characteristics

— context parameters

7.1 Mapping of properties onto the common dictionary schema

A property is mapped in the common dictionary schema onto a **property_BSU** that carries its identification and a **property_DET** that provides its description. A part characteristic is mapped onto the **non_dependent_P_DET** subtype (of **property_DET**), a context dependent characteristic is mapped onto the **dependent_P_DET** subtype (of **property_DET**) and a context parameter is mapped onto the **condition_DET** subtype (of **property_DET**).NOTE Context parameters and context dependent characteristic shall be represented at least when they prove useful for selection purposes.

The family that defines the visibility of a property is mapped onto the **property_BSU.name_scope** attribute. The applicability of a property is defined by the class(es) to which it is applicable through the **class.described_by** attribute.

In this section, only the attributes linked directly to a property are described.

7.2 Attributes

Each part property is described by a dictionary element that contains the following attributes:

Code

Definition Class

Data Type

Preferred Name

Short Name

Preferred Letter Symbol

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Synonymous Letter Symbols

Synonymous Name

Property Type Classification

Definition

Source Document of Definition

Note

Remark

Unit

Condition

Formula

Value Format

Date of Original Definition

Date of current version

Date of current revision

Version Number

Revision Number

The entry for each attribute is structured in the following way (entries may be omitted if not applicable):

—	Name of the attribute	
—	Obj:	Objective.
—	Descr:	Description.
—	Oblig:	Obligation.
—	Trans:	Need for translation.
—	For:	Representation format or maximal number and type of characters if it is a string.
—	Mapp:	Mapping onto the attributes used in the common dictionary schema.
—	Expl:	Example.

NOTE For internal purposes within a company, other attributes and/or other codes for some attributes can be defined but they cannot be used for exchange purposes.

7.2.1 Code

Obj: To identify a property within a family of parts and to allow an absolute identification within the data dictionary when it is associated with the code of the family of parts where the property is defined, and with the library data supplier code.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Descr: A Basic Semantic Unit associated with the property.

Oblig: Mandatory.

Trans: No translation.

Mapp: **property_DET\dictionary_element.identified_by\basic_semantic_unit.code**

7.2.2 Definition Class

Obj: To specify for which family the property is defined.

Descr: The code of the family that is the root of the tree where this property is visible.

Oblig: Mandatory.

Trans: No translation.

Mapp: **property_DET\dictionary_element.identified_by\property_BSU.name_scope**

7.2.3 Data Type

Obj: To specify the data type of the property. The data type describes the set of values that may be assigned to a property.

Descr: A type conforming to the type system specified in the common dictionary schema that specifies the data type of the property.

Oblig: Mandatory.

Trans: No translation.

Mapp: **property_DET.domain**

NOTE **property_DET.domain** refers to **data_type** which again will be subtyped for the various possible data types. Thus, the data type is directly attached to the property. However, an identification mechanism using a **named_type** and a **data_type_BSU** is also available for the cases where the same data type is intended to be used for different properties. When a **data_type** is intended to be used by different properties, the definition of this **data_type** as a **named_type** provides for separate up-dating of the data-type and of the properties that refer to it.

7.2.4 Preferred Name

Obj: To give a name of the property (in full length where possible). It is used for communication and understanding.

Descr: The preferred name is identical to the name that is used in International Standards if available. If the preferred name in the International Standard is longer than the maximum length allowed for this attribute it shall be meaningfully abbreviated.

Oblig: Mandatory.

Trans: To be translated.

For: 30, alphanumeric.

Mapp: **property_DET\class_and_property_elements.names\item_names.preferred_name**

Expl: Threaded diameter

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

7.2.5 Short Name

Obj: To give a name of the property for representation in limited space.

Descr: It is a meaningful abbreviation of the preferred name. If standardised abbreviations exist they should be used. It can be identical to the preferred name or letter symbol.

Oblig: Mandatory.

Trans: To be translated, in the case that the short name is identical to the letter symbol, it shall be the same in all languages.

For: 15, alphanumeric.

Mapp: **property_DET\class_and_property_elements.names\item_names.short_name**

Expl: thread_diam.

7.2.6 Preferred Letter Symbol

Obj: To allow a shorter name of the property. When it exists, it is used in place of the short name for representation in tables, formula, drawings etc...

Descr: The letter symbol is unique within a simple family of parts. It shall be derived from International Standards (e. g. ISO 31, IEC 27, IEC 148 and product standards). The preferred letter symbol is always provided in a text representation. It may also be provided in an SGML ISO 8879 representation.

Oblig: Optional.

Trans: No translation.

For: Alphanumeric and SGML format (if provided).

Mapp: **property_DET.preferred_symbol**

7.2.7 Synonymous Letter Symbol

Obj: To allow further shorter name of the property. It is used for representation in tables, formula, drawings etc…

Descr: The letter symbol is unique within a simple family of parts. It shall be derived from International Standards (e.g. ISO 31, IEC 27, IEC 148 and product standards). None, one or more synonym letter symbols are allowed. The synonym letter symbol is always provided in a text representation. It may also be provided in an SGML representation.

Oblig: Optional.

Trans: No translation.

For: Alphanumeric and SGML format (if provided).

Mapp: **property_DET.synonymous_symbols**

7.2.8 Synonymous Name

Obj: Synonyms to the preferred name may be provided to facilitate transition from names used for local or historical reasons.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Descr:	Alternative designation that differs from the given preferred name but represents the same concept. None, one or more synonymous names are allowed.
Oblig:	Optional.
Trans:	To be translated.
For:	30, alphanumeric.
Mapp:	**property_DET\class_and_property_elements.names\item_names .synonymous_names**

7.2.9 Property Type Classification

Obj:	To classify the different properties defined in order to make large collections of property definitions more manageable.
Descr:	The whole set of properties are divided into subsets according to the categories defined in ISO 31. The property type classification attribute is the reference to the ISO 31 category that is relevant for the property.
Oblig:	Mandatory.
Trans:	No translation.
For:	One capital letter and two digits.
Mapp:	**property_DET.DET_Classification**

NOTE 1 A survey of the main classes and categories of properties in ISO 31 is given in annex E. The ISO 31 classification of quantitative measure is given in Annex F. The ISO 31 classification of non quantitative properties, also called identifications and indicators, is given in Annex A.

NOTE 2 In IEC 61360, the term unit_of_measure is used to denote this attribute.

7.2.10 Definition

Obj:	To describe the meaning of the property.
Descr:	Statement that describes the meaning of the property and permits its differentiation from all other properties. It shall be a definition in the sense that it shall be complete and unambiguous. All significant words are free from homonym and synonymy.
Oblig:	Mandatory.
Trans:	To be translated.
For:	Unlimited alphanumeric string.
Mapp:	**property_DET\class_and_property_elements.definition**

7.2.11 Source Document of Definition

Obj:	A reference to the source document from which the property definition was derived.
Descr:	As a minimum the reference shall be given with the document number and date of issue of the document.
Oblig:	Optional.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Trans:	No translation.
For:	Alphanumeric. An identifier of the document.
Mapp:	**property_DET\class_and_property_elements.source_doc_of_definition\ identified_document.document_identifier**

NOTE In an exchange conforming to ISO CD 13584 part 24, the document itself may be exchanged.

7.2.12 Note

Obj:	To provide further information on any part of the terminological record, that is essential to the understanding of that record.
Descr:	It shall be copied from the definition in the source document into the definition of the property.
Oblig:	Optional.
Trans:	To be translated.
For:	Unlimited alphanumeric string.
Mapp:	**property_DET\class_and_property_elements.note**

7.2.13 Remark

Obj:	Explanatory text to further clarify the meaning of the usage of the property.
Descr:	Free text remarks. It shall not influence the meaning.
Oblig:	Optional.
Trans:	To be translated.
For:	Unlimited alphanumeric string.
Mapp:	**property_DET\class_and_property_elements.remark**

7.2.14 Unit

Obj:	Prescription of the unit in which the value of a quantitative property is expressed.
Descr:	Only the symbols of basic SI units shall be used.
Oblig:	Mandatory (for quantitative data).
Trans:	No translation.
For:	The unit is represented as specified in ISO 10303-41 using, if required, the extensions specified in the common dictionary schema. A mathematical string may be provided. A mathematical string is always provided in a text representation. It may also be provided in an SGML representation.
Mapp:	**property_DET.domain\int_measure_type.unit.structured_representation** or **property_DET.domain\real_measure_type.unit.structured_representation**

NOTE In IEC 61360, the term **unit_of_measure** is used to denote this attribute.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

7.2.15 Condition

Obj: To formally identify the context parameter on which a context dependent characteristic depends.

Descr: To give the code(s) of the adequate context parameter(s).

Oblig: Mandatory for context dependent characteristics.

Trans: No translation.

For: A reference to a set of Basic Semantic Units.

Mapp: **property_DET\dependent_P_DET.depends_on**

7.2.16 Formula

Obj: Rule or statement in mathematical form expressing semantics of a quantitative property.
A formula shall not change any essential information of the meaning of that definition.

Descr: It is a mathematical expression of the property definition.

Oblig: Optional.

Trans: No translation.

For: A mathematical string (a mathematical string is always provided in a text representation; it may also be provided in an SGML representation).

Mapp: **property_DET.formula**

7.2.17 Value Format

Obj: Specification of the type and length of the representation of the value of a property. It is intended as a maximum value format for communication and database storage.

Descr: The value format shall be defined according to the definition below:

a. Non-quantitative data value format types [ISO 9735: 1988]:

A = alphabetic characters, letters only

M = mixed, all characters allowed

N = numeric characters, digits only

X = alphanumeric characters, letters or digits only.

b. Quantitative data value format types [ISO 6093: 1985]:

NR1 = integers,

NR2 = rational numbers with decimal-mark (real),

NR3 = rational numbers with decimal-mark and exponent-mark (floating point),

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

S[3] = signed (positive or negative),

. = decimal-mark,

E = exponent-mark, base 10: (A)E(B) represents the value $Ax10^B$

c. Field length

The field length of a non-quantitative data value shall be indicated by a number (e.g. 17).

The following preferred standard formats -derived from ISO 9735 and ISO 6093 have been defined:

A..3	N..3	X..3	M..3
A..8	N..8	X..8	M..8
A..17	N..17	X..17	M..17
A..35	N..35	X..35	M..35
A..(nx35)	N..(nx35)	X..(nx35)	M..(nx35)

In these formats no special characters shall be allowed. A variable field length shall start with two dots. A fixed field length shall start with one space (examples: A 3 - N 8 - X 17 - M 35 etc.).

The field length of a quantitative data value shall be indicated by a combination of digits and characters (e.g. 3.3ES2). The following preferred standard formats -derived from ISO 9735 and ISO 6093- have been defined:

NR1..4	positive integers
NR1 S..4	positive or negative integers
NR2..3.3	positive reals
NR2 S..3.3	positive or negative reals
NR3..3.3ES2	floating point, positive
NR3 S..3.3ES2	floating point, positive or negative

In these formats no special characters shall be allowed. A variable field length shall start with two dots. A fixed field length shall start with one space (examples: NR1 4 - NR1 S 4 etc.).

Oblig: Mandatory.

Trans: No translation.

For: 15, alphanumeric.

Mapp: **property_DET.domain\simple_type.value_format**

NOTE In case of named types the mapping may be done through a chain of **named_types.**

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

3 In addition to ISO 6093 the sign ' S ' is used as a marker for ' signed ' (should read ' Sign ').

7.2.18 Date of Original Definition

Obj:	To show when the property was defined by the library data supplier and thus when it was declared as valid by this supplier. This date will never be changed and can be used for verification purpose.
Descr:	The entry shall be in accordance with ISO 8601.
Oblig:	Mandatory.
Trans:	No translation.
For:	10, alphanumeric.
Mapp:	**property_DET\dictionary_element.time_stamps\dates.date_of_original_definition**
Expl:	1967-08-20.

7.2.19 Date of Current Version

Obj:	To show the date when the current version was defined.
Descr:	The entry shall be in accordance with ISO 8601.
Oblig:	Mandatory.
Trans:	No translation.
For:	10, alphanumeric.
Mapp:	**property_DET\dictionary_element.time_stamps\dates.date_of_current_version**

7.2.20 Date of Current Revision

Obj:	To show the date of the last revision number change.
Descr:	The entry shall be in accordance with ISO 8601.
Oblig:	Mandatory.
Trans:	No translation.
For:	10, alphanumeric.
Mapp:	**property_DET\dictionary_element.time_stamps\dates.date_of_current_revision**

7.2.21 Version Number

Obj:	To characterise each version of a property. A new version number of a property shall be created whenever a change in some attribute that describes this property influences its use.
NOTE 1	No change is allowed that affects the meaning of a property.
NOTE 2	The changes in a property that affect its version number are defined in 7.3.1, Table 1.
Descr:	A string that contains a natural number to indicate the different versions of a property during the life cycle. Consecutive version numbers shall be issued in ascending order. A new version of the property shall be generated according to the definitions given in table 1 (see 7.3).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Oblig:	Mandatory.
For:	3, numerical.
Trans:	No translation.
Mapp:	**property_DET\dictionary_element.identified_by\basic_semantic_unit.version**

7.2.22 Revision Number

Obj: To characterise each revision of the same version of a property. A new revision number of a property shall be created when a change in some attribute that describes this property influence neither its meaning nor its use.

NOTE 1 No change is allowed that affects the meaning of a property.

NOTE 2 The changes in a property that affect its revision number are defined in 7.3.1, table 1.

Descr: A string that contains a natural number used for administrative control of a property. Consecutive revision numbers shall be issued in ascending order for each value of the version of a property. Per property, unique by its identifier, only one revision number is current at any time. A new revision number of the property shall be generated according to the definitions given in table 1 (see 7.3).

Oblig:	Mandatory.
For:	3, numerical.
Trans:	No translation.
Mapp:	**property_DET\dictionary_element.revision**

7.3 Rules for defining new versions and/or revision of properties

7.3.1 Changes in the attribute of properties

Two concepts are provided to mark updating operations for properties. The concept of version number is a tag for identifying data (it is related to a BSU) whereas the concept of revision number is one for description data of properties (it is related to a Dictionary Element).

The following table (see table 1) gives an overview of how updating operations (add, modify, delete of each attribute of a property) have effects on the version number (**V**) or the revision number (**R**) concept for the given property. There are cases where the denoted operation is meaningless on the conceptual level, and syntactically constrained in the common dictionary schema (**–**), or forbidden at all (**X**). The table does not reflect the creation of a property. To modify means to modify the value of the attribute.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Table 1 — Overview of versioning in property updating operations

Attribute	Add	Modify	Delete
Code	–	X	X
Definition Class	X	X	X
Data type	–	V[4]	X
Preferred Name	–	R	X
Short Name	–	R	X
Preferred Letter Symbol	R	R	X
Synonymous Letter Symbol	R	R	R
Synonymous Name	R	R	R
Property Type Classification	–	R[5]	X
Definition	–	R/V[6]	X
Source Document of Definition	R[7]	R[8]	R
Note	R[8]	R[8]	R
Remark	R[8]	R[5]	R
Unit	–	X	X
Condition	X	X/R[8]	X
Formula	R[8]	R[8]	R[8]
Format	–	V	X
Date of Original Definition	–	X	X
Date of current version	–	V	X
Date of Current Revision	–	R / V	X
Version Number	–	V	X
Revision Number	–	R	X

7.3.2 Propagated changes in property version numbers

The following table (see table 2) specifies which changes in a property attribute shall lead to new version of this property.

4 The SUBTYPE of the **data_type** entity that defines the domain can not be changed (X). The only allowed changes consist of

1 - Adding new allowed values in an enumerated type, modelled either as a **non_quantitative_code_type** entity or a **non_quantitative_int_type** entity; or

2 - Changing the version of the class that constitutes the domain of the property (is_part_of relationship for assemby modelling).

3 - Changing the version of the **data_type_BSU** that identifies the domain of the property

These three changes modify the version (V) of the property.

5 Only allowed in case of revision of ISO 31 or of this part of the ISO 13584 series.

6 The changes in definition shall not affect the abstract property referred to by the definition. Nevertheless it may change the specification of the measure process to be used to evaluate the value of this property. If the changes in the definition affect this measure process it leads to a new version, else it leads to a new revision.

7 This change shall not affect the abstract property referred to by the property attributes.

8 The context parameters (modelled as **condition_DET**) on which a context dependant characteristic modelled as **dependent_P_DET** depend shall not be changed. Nevertheless, change in the version of some of these **condition_DET**s leads to change of version of the **dependent_P_DET**.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Table 2 — Propagated changes in version numbers of properties

Attribute	Changed item
Data type	**property_DET.domain\non_quantitative_code_type.domain.its_values**[9] **property_DET.domain\non_quantitative_int_type.domain.its_values**[6] **property_DET.domain\Class_Instance_Type.domain.version** **property_DET.domain\Named_Type.referred_type.version**
Condition	**property_DET\dependent_P_DET.depends_on[.].version**[10]

8 Dictionary elements that describe families of parts

8.1 Mapping of families onto the common dictionary schema

A family is mapped in the common dictionary schema onto a **class_BSU** that carries its identification and a **component_class** that provides its description.

In this section, only the attributes linked directly to a class are described.

8.2 Attributes

Each family of parts is described by a dictionary element that contains the following attributes:

Code

Superclass

Preferred Name

Short Name

Synonymous Name

Visible Types

Applicable Types

Sub-class Selection Properties

Visible Properties

Applicable Properties

Class Value Assignment

Definition,

Source Document of Definition

9 Values shall only be added
10 Version change of any property in the set

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Note

Remark

Simplified Drawing

Date of Original Definition

Date of Current Version

Date of Current Revision

Version Number

Revision Number

The entry for each attribute is structured in the following way (entries may be omitted if not applicable):

—	Name of the attribute;	
—	Obj:	Objective;
—	Descr:	Description;
—	Oblig:	Obligation;
—	Trans:	Need for translation;
—	For:	Representation format or maximal number and type of characters if it is a string;
—	Mapp:	Mapping onto the resource constructs used in the common dictionary schema;
—	Expl:	Example.

8.2.1 Code

Obj:	To identify a family of parts and to allow an absolute identification within the data dictionary when associated with the library data supplier code.
Descr:	A Basic Semantic Unit associated with the family of parts.
Oblig:	Mandatory.
Trans:	No translation.
Mapp:	**component_class\dictionary_element.identified_by\basic_semantic_unit.code**

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

8.2.2 Superclass

Obj:	To reference the immediate (unique) parent family of a family.
Descr:	A Basic Semantic Unit of the immediate parent family of the current family.
Oblig:	Optional (if it does not exist, the class has no superclass).
Trans:	No translation.
For:	A Class Basic Semantic Unit.

Mapp: **component_class\class.its_superclass**

NOTE 1 When defining a standardised identification hierarchy, all the defined classes shall have a superclass. The root of such a tree shall be either one class of the ISO/IEC International Classification for Standards (ICS) predefined tree, or a class already defined by another standardised identification hierarchy.

NOTE 2 The code of the class that correspond to each node of the tree defined in the ISO/IEC ICS shall be equal to the code defined in ICS: 1993, where dots are replaced by underscores. The code of the ICS: 1993 root shall be 'OO'.

NOTE 3 The version of the class that corresponds to each node of the tree defined in the ISO/IEC ICS shall be equal to '001', unless other versions be specified by an ISO/IEC standard that integrates, in its normative references, both ISO 13584-42 and IEC 61360-2.

NOTE 4 The **Class_BSU**s that carry the identity of the classes that correspond to the node of the tree defined in the ISO/IEC ICS shall refer in their **defined_by** attribute, to the special code defined in ISO 13584-26 for ISO/IEC ICS..

8.2.3 Preferred Name

Obj:	To give a meaningful description of the family of parts (in full length where possible). It is used for communication and understanding.
Descr:	The preferred name is identical to the name that is used in International Standards if available. If the preferred name in the International Standards is longer than the max. length allowed for this attribute it shall be meaningfully abbreviated.
Oblig:	Mandatory.
Trans:	To be translated.
For:	30, alphanumeric.
Mapp:	**component_class\class_and_property_elements.names\item_names .preferred_name**
Expl:	Screw threads.

8.2.4 Short Name

Obj:	To give a name to the family of parts for representation in limited space.
Descr:	It is a meaningful abbreviation of the preferred name. If standardised abbreviations exist they should be used. It can be identical to the preferred name.
Oblig:	Mandatory.
Trans:	To be translated, in the case that the short name is identical to the letter symbol, it will be the same in all languages.
For:	15, alphanumeric.
Mapp:	**component_class\class_and_property_elements.names\item_names.short_name**

8.2.5 Synonymous name

Obj:	Synonyms to the preferred name may be provided to facilitate transition from names used for local or historical reasons.
Descr:	Alternative designation that defers given preferred name but represents the same concept. None, one or more synonymous names are allowed.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Oblig:	Optional.
Trans:	To be translated.
For:	30, alphanumeric.
Mapp:	**component_class\class_and_property_elements.names \item_names.synonymous_names**

8.2.6 Visible Types

Obj: To define the new named types that may be referred to, as their data type, by the visible properties of this family, or of any of its subfamilies (visible type).

Descr: Different properties may have the same data type of values (e.g., a set of codes that identify materials). Such data types may be defined as named types, independently of any property. They may be, later on, referred to as their data type by different properties. The definition of named type shall contain the code of the named type, the version number of the named type, and the data type of the named type specified by using the resource constructs of the common dictionary schema. For the **non_quantitative_code_type** and **non_quantitative_int_type** the data type shall be specified as a set of **dic_value**s each one consisting of an unique code and a set of names (possibly translated)**.**

Oblig: Optional.

Trans: In case of **non_quantitative_code_type** and **non_quantitative_int_type** value names are to be translated

For: **data_type_BSU**s and **data_type_element**s

Mapp: **component_class\dictionary_element.identified_by \class_BSU.added_visible_data_types**

NOTE 1 Visible types are inherited.

NOTE 2 Visible types that are not also applicable types may only be referenced to define the data type of the visible properties of the family (or of any of its subfamilies).

EXAMPLE A **named_type** "material" may be defined at the level of some family as a **non_quantitative_code_type**. The set of values of this named type consists of a set of codes associated with different (translated) names. This named type may be referenced in some (sub-) family to define the data type of the property that corresponds to the material the parts of the family consist of. It may be referenced in some other (sub-) family to define the data type of the property that corresponds to the coating of the parts of the family.

8.2.7 Applicable Types

Obj: To define which visible types are allowed as data type for the properties applicable to the family (or any of its sub-families).

Descr: The list of the code of the defined or inherited visible types of the family that become applicable types for the family (and any of its sub-families).

Oblig: Optional.

Trans: No translation.

For: A list of **data_type_BSU**s.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Mapp: **component_class\class.defined_types**.

8.2.8 Sub-class Selection Properties

Obj: To define which (new) class valued properties shall be assigned a value in any simple part family that is a sub-family of the current family.

Descr: A class valued property enables representation of different perspectives on the set of parts that belongs to the current family. When a class valued property is defined at the level of some family, the dictionary user may query all the simple sub-families of this family for which this class valued property is assigned some value.

Oblig: Optional.

Trans: No translation.

For: A list of **property_BSU**s.

Mapp: **component_class\item_class.sub_class_properties**

NOTE 1 Sub-class selection properties are inherited. They shall not appear in the sub-class selection properties list of any sub-family.

NOTE 2 In each simple family of parts, each class valued property defined as sub-class selection properties in any upper family shall be assigned a value.

8.2.9 Visible Properties

Obj: To specify the new properties that are defined at the level of this family and may be thus specified as applicable to the family or any of its sub-families (visible properties).

Descr: The new properties (with respect to inheritance) that the parts belonging to the generic or simple family of parts may or may not possess according to the subfamilies they belong to.

For: A set of **property_BSU**s.

Oblig: Mandatory (possibly empty).

Mapp: **component_class\dictionary_element.identified_by \class_BSU.added_visible_properties**

8.2.10 Applicable Properties

Obj: To specify the new properties that are specified as applicable for that family and for any of its sub-families.

Descr: The new properties (with respect to inheritance) that the parts belonging to the generic or simple family of parts shall possess. The properties of this list shall be visible for the family, i.e. they shall be defined as visible either by this family or by a family higher in the hierarchy. The LIST order shall correspond to the default presentation order of the properties in any case where such an order shall be defined (e.g., to display the properties of some classes on a screen).

For: List of **property_BSU**s.

Oblig: Mandatory (possibly empty).

Mapp: **component_class\class.described_by**

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

8.2.11 Class Value Assignment

Obj:	To define the values assigned to some class valued properties specified as Sub-Class Selection Properties in the famlily or any family higher in the hierarchy.
Descr:	Class valued properties are intended to be queried by the user to select a set of families for which such a property has a given value. Once a value is assigned to a class valued property, this value is inherited and shall not be redefined.
For:	A set of **Class_Value_Assignment**s.
Oblig:	Optional.
Mapp:	**component_class\item_class.class_constant_values**

8.2.12 Definition

Obj:	To clarify the meaning of the family by giving its intention textually.
Descr:	Statement that describes the meaning of the family and permits its differentiation from all other families. It shall be a definition in the sense that it shall be complete and unambiguous. All significant words are free from homonymy (homonym) and synonymy.
Oblig:	Mandatory.
Trans:	To be translated.
For:	Unlimited alphanumeric string.
Mapp:	**component_class\class_and_property_elements.definition**

8.2.13 Source Document of Definition

Obj:	A reference to the source document from which the family definition was derived.
Descr:	As a minimum the reference shall be given with the document number and date of issue of the document.
Oblig:	Optional.
Trans:	No translation.
For:	Alphanumeric. An identifier of the document.
Mapp:	**component_class\class_and_property_elements .source_doc_of_definition\identified_document.document_identifier**

NOTE In an exchange conforming to ISO CD 13584 part 24, the document itself may be exchanged.

8.2.14 Note

Obj:	To provide further information on any part of the terminological record, that is essential to the understanding of that record.
Descr:	It shall be copied from the definition in the source document into the definition of the family of parts.
Oblig:	Optional.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Trans: To be translated.

For: Unlimited alphanumeric string.

Mapp: **component_class\class_and_property_elements.note**

8.2.15 Remark

Obj: Explanatory text to further clarify the meaning of the usage of the family of parts.

Descr: Free text remarks. It shall not influence the meaning.

Oblig: Optional.

Trans: To be translated.

For: Unlimited alphanumeric string.

Mapp: **component_class\class_and_property_elements.remark**

8.2.16 Simplified Drawing

Obj: To provide a visualisation, on request of the user, that shows an image of the class of parts.

Descr: A drawing including at least the reference coordinate system of the part (that is to be used for all the representations of this part), and the letter symbols of the main applicable properties.

Oblig: Mandatory for all the families that may be instanciated.

Trans: No translation.

For: An ABSTRACT SUPERTYPE.

Mapp: **component_class\item_class.simplified_drawing**

NOTE ISO CD 13584 part 24 provides an exchange format for such drawings.

8.2.17 Date of Original Definition

Obj: To show when the family of parts was defined by the library data supplier and thus when it was declared as valid by this supplier. This date will never be changed and can be used for verification purpose.

Descr: The entry shall be in accordance with ISO 8601.

Oblig: Mandatory.

Trans: No translation.

For: 10, alphanumeric.

Mapp: **component_class\dictionary_element.time_stamps.date_of_original_definition**

Expl: 1967-08-20

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

8.2.18 Date of Current Version

Obj:	To show the date when the current version was defined.
Descr:	The entry shall be in accordance with ISO 8601.
Oblig:	Mandatory.
Trans:	No translation.
For:	10, alphanumeric.
Mapp:	**component_class\dictionary_element.time_stamps.date_of_current_version**

8.2.19 Date of Current Revision

Obj:	To show the date of the last revision number change.
Descr:	The entry shall be in accordance with ISO 8601.
Oblig:	Mandatory.
Trans:	No translation.
For:	10, alphanumeric.
Mapp:	**component_class\dictionary_element.time_stamps. date_of_current_revision**

8.2.20 Version Number

Obj: To characterise each version of a class. A new version number of a class shall be defined when ever a change in the attributes that describes this class influences its use.

NOTE 1 No change is allowed that affects the meaning of a class.

NOTE 2 The changes in a class that affect its version number are defined in 7.3.1, table 3.

Descr: A string that contains a natural number to indicate the different versions of a family during the life cycle. The version number string of a property shall consist of three numeric characters. Consecutive version numbers shall be issued in ascending order. A new version number of the property shall be generated according to the definitions given in table 3 (see 7.3).

Oblig:	Mandatory.
For:	3, numeric.
Trans:	No translation.
Mapp:	**component_class\dictionary_element.identified_by\basic_semantic_unit.version**

8.2.21 Revision Number

Obj: To characterise each revision of a version of a class. A new revision number of a class shall be defined when ever a change in the attributes that describes this class influences neither its meaning nor its use.

NOTE 1 No change is allowed that affects the meaning of a class.

NOTE 2 The changes in a class that affect its revision number are defined in 7.3.1, table 3.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Descr:	A string that contains a natural number used for administrative control of a class. Consecutive revision numbers shall be issued in ascending order for each value of the version number of a class. Per class, unique by its identifier, only one revision number is current at any time. A new revision number of the class shall be generated according to the definitions given in table 3 (see 7.3).
For:	3, numeric.
Trans:	No translation.
Mapp:	**component_class\dictionary_element.revision**

8.3 Rules for defining new version or revision of classes

8.3.1 Change in the attribute of class

Two concepts are provided to mark updating operations for families of parts. The concept of version number is a tag for identifying data (it is related to a BSU) whereas the concept of revision number is one for description data of families of parts (it is related to a Dictionary element).

The following table gives an overview of how updating operations (add, modify, delete of some attributes of a class) have effects on the version number (**V**) or the revision number (**R**) concept for the given class. There are cases, where the denoted operation is meaningless , and syntactically constrained in the common dictionary schema (**–**), or forbidden at all (**X**). The tables do not reflect the creation of a family. To modify means to modify the value of the attribute.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Table 3 — Overview of versioning in class updating operations

Attribute	Add	Modify	Delete
Code	–	X	X
Superclass	V[11]	V[8]	V[8]
Preferred Name	–	R	X
Short Name	–	R	X
Visible Types	V	V[12]	X
Applicable Types	V	V[9]	X
Sub-Class Selection Properties	V	V[9]	X
Synonymous name	R	R	R
Visible Properties	V	V[9]	X
Applicable Properties	V	V[9]	X
Class Value Assignment	V	V[9]	X
Definition	–	R[13]	X
Source Document of Definition	R	R	R
Note	R	R	R
Remark	R	R	R
Simplified Drawing	–	R	R
Date of Original Definition	–	X	X
Date of current version	–	V	X
Date of Current Revision	–	R / V	X
Version number	–	V	X
Revision number	–	R	X

8.3.2 Propagated changes in family version numbers

Any change of version number of a family leads to a change of version number of its sub-family (see table 3: Modify Superclass), and of the sub-families of this family, and so on. *Therefore, all the changes that change the version number of the inherited properties or types of a family change the version number of this family.*

When a part family, intentionally defined by a set of dictionary data, belongs to a parts library, the class that constitutes the information model of this part family may also reference other basic semantic units (tables and documents), and it may be associated with a **content_item** that defines the extension of this class.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

11 Modification of superclass shall not remove any inherited visible or applicable property or type. It may consist either in changing the version of the referenced superclass, that defines a new version of the sublass, or, e.g., in adding an intermediate class in between one class and its previous superclass.

12 Modification shall consist only in addition of new items or in change in version of referenced items. No other modification shall take place. No item shall be removed.

13 This change shall not affect the abstract population referred to by the class data element type.

NOTE 1 The concept of extension of a class is defined in ISO CD 13584 part 24.

NOTE 2 The information model of a family of parts in a parts library in defined in ISO 13584 -24[14].

Any change in the **content_version** that characterises the extension of a family of parts in a parts library, and any change in the version number of the tables that are referenced from the class that constitutes the information model of the family of parts or in the version number of the tables that reference this class through their **name_scope** attribute shall lead to a new version number of the family.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

14 To be published

9 Bibliography

1) ISO 639: 1988, *Language Codes.*

2) ISO 843: 1990, *Documentation - International system for the transliteration of Greek characters into Latin characters.*

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Annex A

(normative)

Survey of type classification codes of non-quantitative data element types

(main class A)

Type classification code	Description
A11	Geographical unit (greater than a place)
A12	Geographical location (place or smaller)
A13	Geographical route and network
A21	Organization
A22	Functionary
A31	Date and time period
A32	Time of day
A41	Private person
A51	Product
A52	Product class
A53	Product batch and package (type)
A54	Transport mode, means and unit
A55	Manufacturing process and technology
A56	Product function and application
A57	Material
A58	Product geometry, shape, and size
A59	Product quality, performance, and test
A61	document and message
A62	Information element and information group
A63	Data medium and transmission unit
A71	Measuring unit
A79	Type of measurement
A81	Account
A82	Project, project activity
A83	Procedure
A91	Abstract identification such as language, colour, etc.
A93	Clause

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Annex B

(normative)

Short names of entities

Table B.1 provides the short names of entities specified in this part of ISO 13584. Requirements on the use of short names are found in the implementation methods included in ISO 10303

Table B.1 - Short names of entities

Long name	Short name
AXIS1_PLACEMENT_TYPE	AXPLTY
AXIS2_PLACEMENT_2D_TYPE	AP2T
AXIS2_PLACEMENT_3D_TYPE	AP3T
BASIC_SEMANTIC_UNIT	BSSMUN
BOOLEAN_TYPE	BLNTYP
CLASS	CLASS
CLASS_AND_PROPERTY_ELEMENTS	CAPE
CLASS_BSU	CLSBS
CLASS_BSU_RELATIONSHIP	CLBSRL
CLASS_INSTANCE_TYPE	CLINTY
CLASS_RELATED_BSU	CLRLBS
CLASS_VALUE_ASSIGNMENT	CLVLAS
COMPLEX_TYPE	CMPTYP
COMPONENT_CLASS	CMPCLS
CONDITION_DET	CNDDT
CONTENT_ITEM	CNTITM
DATA_TYPE	DTTYP
DATA_TYPE_BSU	DTTYBS
DATA_TYPE_ELEMENT	DTTYEL
DATES	DATES
DEPENDENT_P_DET	DPPDT
DICTIONARY_ELEMENT	DCTELM
DIC_UNIT	DCUNT
DIC_VALUE	DCVL
DOCUMENT	DCMNT
ENTITY_INSTANCE_TYPE	ENINTY
GLOBAL_LANGUAGE_ASSIGNMENT	GLLNAS

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Table B.1 (concluded)

Long name	Short name
GRAPHICS	GRPHCS
IDENTIFIED_DOCUMENT	IDNDCM
INT_CURRENCY_TYPE	INCRTY
INT_MEASURE_TYPE	INMSTY
INT_TYPE	INTTYP
ITEM_CLASS	ITMCLS
ITEM_NAMES	ITMNMS
LABEL_WITH_LANGUAGE	LBWTLN
LEVEL_TYPE	LVLTYP
MATERIAL_CLASS	MTRCLS
MATHEMATICAL_STRING	MTHSTR
NAMED_TYPE	NMDTYP
NON_DEPENDENT_P_DET	NDPD
NON_QUANTITATIVE_CODE_TYPE	NQCT
NON_QUANTITATIVE_INT_TYPE	NQIT
NON_SI_UNIT	NNSUN
NUMBER_TYPE	NMBTYP
PLACEMENT_TYPE	PLCTYP
PRESENT_TRANSLATIONS	PRSTRN
PROPERTY_BSU	PRPBS
PROPERTY_DET	PRPDT
REAL_CURRENCY_TYPE	RLCRTY
REAL_MEASURE_TYPE	RLMSTY
REAL_TYPE	RLTYP
SIMPLE_TYPE	SMPTYP
STRING_TYPE	STRTYP
SUPPLIER_BSU	SPPBS
SUPPLIER_BSU_RELATIONSHIP	SPBSRL
SUPPLIER_ELEMENT	SPPELM
SUPPLIER_RELATED_BSU	SPRLBS
TRANSLATED_LABEL	TRNLBL
TRANSLATED_TEXT	TRNTXT
VALUE_DOMAIN	VLDMN

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Annex C

(normative)

Information object registration

C.1 Document identification

In order to provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 13584 part (42) version(1) }

is assigned to this part of ISO 13584. The meaning of this value is defined in ISO 8824-1.

C.2 Schema identification

C.2.1 ISO13584_IEC61360_dictionary_schema

The **ISO13584_IEC61360_dictionary_schema** is assigned the object identifier

{ iso standard 13584 part (42) version(1) object(1) ISO13584-IEC61360-dictionary-schema(1) }

C.2.2 ISO13584_IEC61360_language_resource_schema

The **ISO13584_IEC61360_language_resource_schema** is assigned the object identifier

{ iso standard 13584 part (42) version(1) object(1) ISO13584-IEC61360-language-resource-schema(2)}

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Annex D

(informative)

Common IEC/ISO dictionary schema

This annex provides the common information model of ISO 13584-42 and IEC 61360. The normative version is published in IEC 61360-2. This informative annex duplicates the normative content of IEC 61360-2 and provides, by means of NOTEs, some additional explanations referring to this part of ISO 13584.

In this annex references to clauses within ISO 13584-42 are shown in the form "6.2.1" while references to paragraphs in Annex D are prefixed with the capital letter D, thus "D.3.3.3".

D.1 General

D.1.1 Scope and object of the common dictionary schema

The scope the common ISO / IEC dictionary schema based is defined by the intersection of the scopes of the two base standards:

— IEC 61360-1, " Standard data element types with associated classification scheme for electric components - Part 1: Definitions - Principles and methods", generated by IEC TC3 SC3D and

— ISO 13584-42, " Methodology for structuring part families" generated by ISO/TC184/SC4/WG2.

The presented EXPRESS model represents a common formal model for the two documents and facilitates a harmonization of both.

Relevant parts of their scope clauses are cited below.

From **IEC 61360-1**: " This part of IEC 61360 specifies the principles that shall be used for defining technical data element types with associated classification schemes needed to describe electric components, including electronic and electromechanical components and materials used in electro-technical equipment and systems."

From **ISO DIS 13584-42**: " This part of ISO 13584 specifies:

— the attributes that shall be provided by library data suppliers to describe the families and properties of parts.

— the specifications of those attributes in the EXPRESS information model that provides for the exchange of such dictionary data".

The object of this EXPRESS schema is to provide a formal model for data according to the scopes as given above, and thus to provide a means for the computer-sensible representation and exchange of such data.

The intention is to provide a common information model for the work of both committees, thus allowing for the implementation of dictionary systems dealing with data delivered according to either of the standards elaborated by both committees.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

D.1.2 Interoperability of ISO 13584 and IEC 61360

In the ISO 13584 Standard series, the common dictionary schema is used in ISO 13584-24[15] to define the requirements that should be fulfilled by an ISO 13584-compliant implementation.

ISO 13584-24 provides for a number of options that may be supported by an implementation. These options have been grouped into conformance classes. Conformance to a particular conformance class requires that all entities, types, and associated constraints defined as part of that class shall be supported. Support for a particular conformance class requires support of all the options specified in this class.

Conformance class 0 of the view exchange protocol 24-1, documented in ISO 13584-24, corresponds to the common ISO and IEC requirements defined in the **ISO13584_IEC61360_dictionary_schema**. An ISO 13584 compliant implementation conform to that conformance class shall therefore support all the entities, types and associated constraints defined in the common dictionary schema. All the other conformance classes of the view exchange protocol 24-1 including the requirements of conformance class 0, the same capability shall be provided by any ISO 13584-compliant implementation supporting the view exchange protocol 24-1.

In IEC 61360-2, two schemata are provided, these two schemata defining two options that may be selected by an IEC 61360-compliant implementation. Each of these options is referred to as a conformance class.

— The **ISO13584_IEC61360_dictionary_schema** provides for modelling and exchanging technical data element types with associated classification scheme but without modelling the definitions of the terms used in the data element type definitions. It constitutes conformance class 1 of IEC 61360-2;

— The **IEC61360_extended_dictionary_schema** provides for modelling and exchanging technical data element types with associated classification scheme together with modelling definitions of and references to the terms used in the data element type definitions. It constitutes conformance class 2 of IEC 61360-2.

When used together with ISO 10303-21, each schema defines one single exchange format.

The exchange format defined by conformance class 1 of IEC 61360-2 is fully compatible with conformance class 0 of the view exchange protocol 24-1 defined in ISO 13584-24.

Conformance class 2 of IEC 61360-2 including the requirement of its conformance class 1, any ISO 13584-compliant or IEC 61360-compliant implementation shall support all the entities, types and associated constraints defined in the common dictionary schema.

Both committees agreed NOT to change and/or modify the presented EXPRESS model independent of each other in order to guarantee the harmonization and the reusability of the work from both committees. Requests for amendments should therefore be sent to both committees. These requests should be adopted by both committees before modifying the EXPRESS information model.

D.2 Overview of the dictionary schema

This section explains the main resource constructs provided by the common dictionary schema.

15 To be published

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

— **dictionary_element** is any element defined in the dictionary;

— **supplier_element** captures the data of suppliers of dictionary elements (classes, properties, datatypes);

— **class** models the dictionary element of classes (families) that are described by properties;

— **property_DET** is the dictionary element of a property;

— **data_type** specifies the type of a property.

These parts of the dictionary schema are presented in more detail in the following clause D.3 "**ISO13584_IEC61360_dictionary_schema**".

In the presentation of the common dictionary schema, some overview diagrams are provided as planning models (figure D.1 to figure D.11). These planning models use the EXPRESS-G graphical notation for the EXPRESS language.

For clarification of the diagrams, some of the relationships that are defined in the EXPRESS model are omitted. The figure D.1 below outlines as a planning model the main structure of the common dictionary schema.

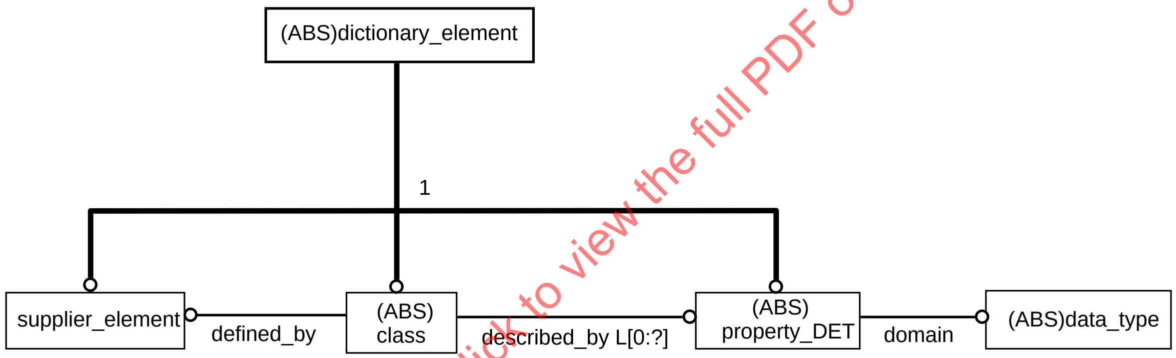


Figure D.1 — Overview of the dictionary schema

Most of these figures contain overview models (or planning models) but show only that level of detail that is appropriate at a certain place.

D.3 ISO13584_IEC61360_dictionary_schema

This clause, that constitutes the main part of the common information model of ISO 13584-42 and IEC 61360, contains the full EXPRESS listing of the dictionary schema, annotated with comments and explanatory text. The order of text in this clause is determined primarily by the order imposed by the EXPRESS language, secondarily by importance.

EXPRESS specification:

```
*)
SCHEMA ISO13584_IEC61360_dictionary_schema;
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

D.3.1 References to other schemata

This subclause contains references to other EXPRESS schemata that are used in the dictionary schema. Their source is indicated in the respective comment.

EXPRESS specification:

```
*)
REFERENCE FROM support_resource_schema (identifier, label, text);
   (* from  ISO  10303-41:  STEP  Part  41:  "Fundamentals  of  Product
       Description and Support" *)
REFERENCE FROM person_organization_schema (organization, address);
   (* from  ISO  10303-41:  STEP  Part  41:  "Fundamentals  of  Product
       Description and Support" *)
REFERENCE FROM measure_schema;
   (* from  ISO  10303-41:  STEP  Part  41:  "Fundamentals  of  Product
       Description and Support" *)
REFERENCE FROM ISO13584_IEC61360_language_resource_schema;
   (* see clause D.4 "ISO13584_IEC61360_language_resource_schema" *)
(*
```

D.3.2 Constant definitions

This subclause contains constant definitions used later in type definitions (see D.3.8 "Basic Type and Entity Definitions").

EXPRESS specification:

```
*)
CONSTANT
property_code_len: INTEGER := 14;
class_code_len: INTEGER := 14;
data_type_code_len:INTEGER := 14;
supplier_code_len: INTEGER := 18;
version_len: INTEGER := 3;
revision_len: INTEGER := 3;
value_code_len: INTEGER :=18;
pref_name_len: INTEGER := 30;
short_name_len: INTEGER := 15;
syn_name_len: INTEGER := pref_name_len;
DET_classification_len: INTEGER := 3;
source_doc_len: INTEGER := 80;
value_format_len: INTEGER := 80;
sep_cv: STRING := '-';
sep_id: STRING := '.';
END_CONSTANT;
(*
```

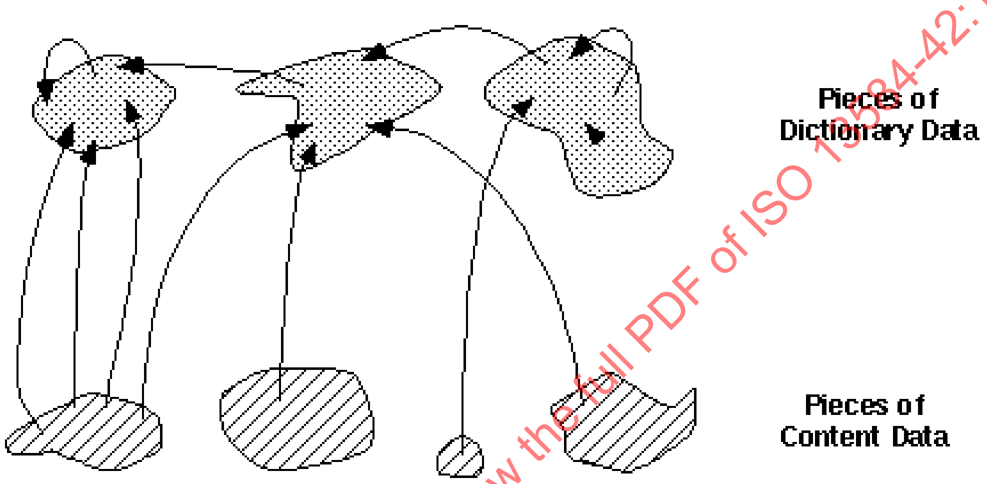
STANDARDSISO.COM: Click to view the full PDF of ISO 13584-42:1998

D.3.3 Basic Semantic Units: defining and using the dictionary

D.3.3.1 Requirements for exchange

In the exchange of dictionary and part library data it is customary to partition the data. For example, a dictionary could be updated with some classes that specify their superclass by a reference to a pre-existing class, or when the content of a library is exchanged, dictionary elements are only referenced and not included every time. It must be possible to refer unambiguously and consistently to the dictionary data.

Thus, it is a clear requirement first, to be able to exchange pieces of data, and second, to have relationships between these pieces. This is depicted in figure D.2.



STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Figure D.2 — Pieces of data with relationships

Every one of these pieces corresponds to a Physical File (according to ISO 10303-21). EXPRESS (ISO 10303-11) attributes can only contain references to data within the same Physical File. Thus it is impossible to use EXPRESS attributes directly to implement inter-piece references.

D.3.3.2 Three levels architecture of the dictionary data

In this clause the concept of **basic_semantic_unit** (BSU) is introduced as a means to implement these inter-piece references. A BSU provides an universally unique identification for dictionary descriptions. This is depicted in figure D.3.

Assume some piece of content (**content_item**) wants to refer a certain dictionary description, e.g. to convey the value of a property of a component. It does this by referring to a Basic Semantic Unit through the attribute **dictionary_definition**.

A dictionary description (**dictionary_element**) refers to a Basic Semantic Unit through the attribute **identified_by**. From the correspondence of the absolute identifiers of the Basic Semantic Units this indirect relation is established.

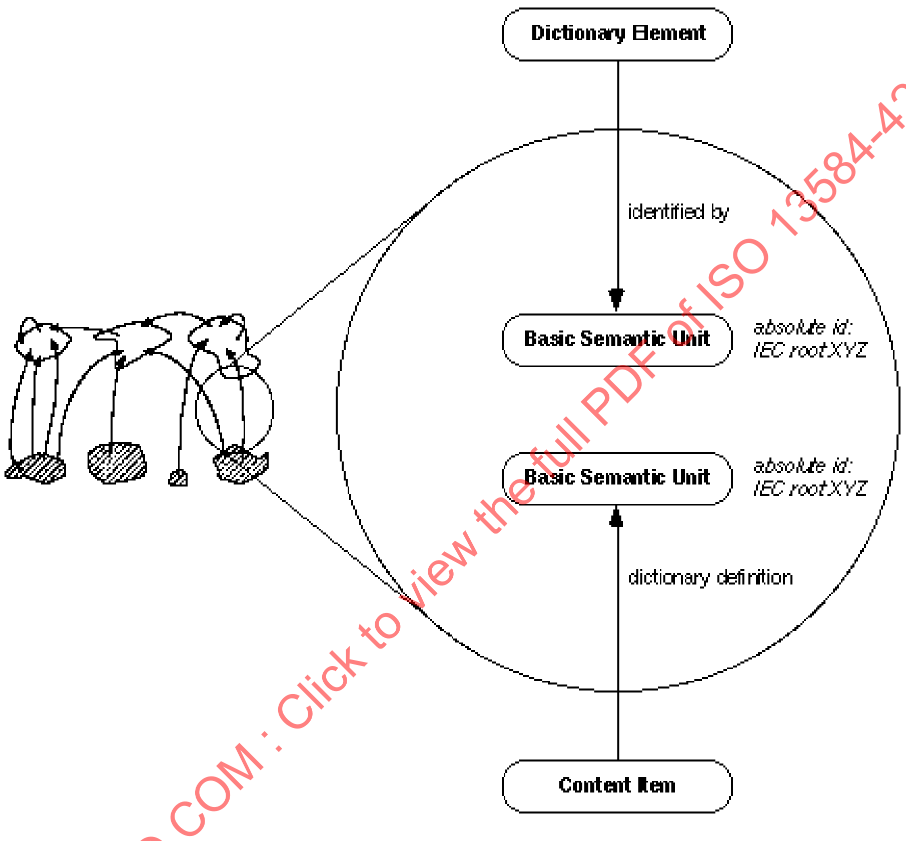
Note that:

— both dictionary element and content item can be present in the same Physical File, but need not be;

— the dictionary element does not need to be present for the exchange of some Content Item referring to it. In this case it is assumed to be present in the dictionary of the target system already. Conversely, dictionary data can be exchanged without any content data;

— the basic semantic unit can be one single instance in the case where both dictionary element and content item instances are in the same Physical file;

— the same mechanism applies also to references between various dictionary elements (e.g. between a component class and the associated **property_DET**s).



STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Figure D.3 — Implementation of "inter-piece" relationships using basic semantic units

A BSU provides a reference to a dictionary description in any place where this is needed, e. g. dictionary delivery, update delivery, library delivery, component data exchange. The data associated with a property e.g. could be exchanged as a couple (**property_BSU**, <value>).

Figure D.3 outlines the implementation of this general mechanism.

D.3.3.2.1 Basic_semantic_unit

A **basic_semantic_unit** is an unique identification of a **dictionary_element**. BSU is the abbreviation of Basic Semantic Unit.

EXPRESS specification:

```
*)
ENTITY basic_semantic_unit
ABSTRACT SUPERTYPE OF ( ONEOF (
           supplier_BSU,
           class_BSU,
           property_BSU,
           data_type_BSU,
           supplier_related_BSU,
           class_related_BSU));
   code: code_type;
   version: version_type;
DERIVE
   dic_identifier: identifier := code + sep_cv + version;
INVERSE
   definition: SET [ 0 : 1 ] OF dictionary_element FOR
   identified_by;
   referenced_by: SET[ 0 : 1 ] OF content_item
                                     FOR dictionary_definition;
END_ENTITY; -- basic_semantic_unit
(*
```

Attribute definitions:

code: the code assigned to identify a certain dictionary element.

version: the version number of a certain dictionary element.

dic_identifier: the full identification, consisting of concatenation of code and version.

definition : a reference to the dictionary element identified by this BSU. If not present in some exchange context, it is assumed to be present in the dictionary of the target system already.

referenced_by: items making use of the dictionary element associated with this BSU.

D.3.3.2.2 Dictionary_element

A **dictionary_element** is a full definition of the data required to be captured in the semantic dictionary for some concept. For every concept a separate subtype is to be used. The **dictionary_element** is associated with a **basic_semantic_unit** (BSU), that serves to uniquely identify this definition in the dictionary.

By including the version attribute in the **basic_semantic_unit** entity, it forms part of the identification of a dictionary element (in contrast to the **revision** and **time_stamps** attributes).

EXPRESS specification:

```
    *)
    ENTITY dictionary_element
    ABSTRACT SUPERTYPE OF ( ONEOF (
                supplier_element,
                class_and_property_elements,
                data_type_element));
        identified_by: basic_semantic_unit;
        time_stamps: OPTIONAL dates;
        revision: revision_type;
    END_ENTITY;
    (*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Attribute definitions:

identified_by: the BSU identifying this dictionary element.

time_stamps: the optional dates of creation and update of this dictionary element.

revision: the revision number of this dictionary element.

NOTE 1 The type of the **identified_by** attribute will be redeclared later to **property_BSU** and **class_BSU** and will then be used to encode together with the code attribute of the BSUs the "Code" attribute for properties (see 6.2) and classes (see 7.2) respectively. It will also be used to encode the "Version Number" attribute for properties and classes respectively.

NOTE 2 The **time_stamps** attribute will be used as a starting point to encode in the **dates** entity the property and class attributes "Date of Original Definition", "Date of Current Version" and "Date of Current Revision" (see 6.2, 7.2 and D.3.8.2).

NOTE 3 The **revision** attribute will be used to encode the property and class attribute "Revision Number" (see 6.2 and 7.2).

Figure D.4 presents a planning model of the relationship between basic semantic unit and the dictionary element.

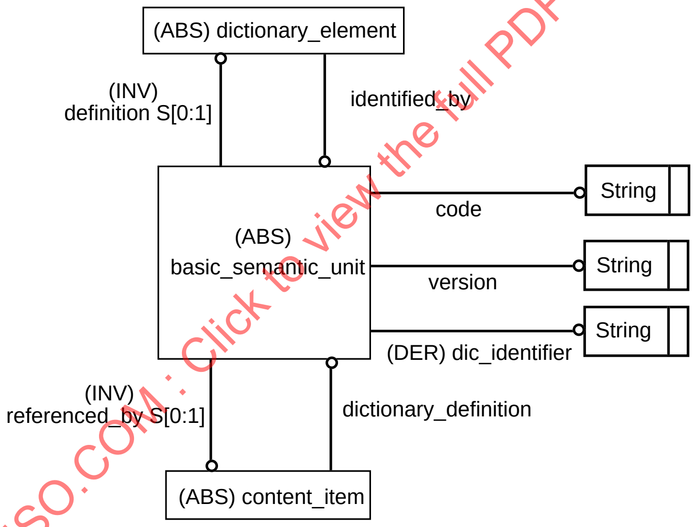


Figure D.4 — Relationship between basic semantic unit and dictionary element

D.3.3.2.3 Content_item

A **content_item** is a piece of data referring to its description in the dictionary. It shall be subtyped.

EXPRESS specification:

```
*)
ENTITY content_item
ABSTRACT SUPERTYPE;
     dictionary_definition: basic_semantic_unit;
END_ENTITY;
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Attribute definitions:

dictionary_definition: the Basic Semantic Unit to be used for referring to the definition in the dictionary.

D.3.3.3 Overview of basic semantic units and dictionary elements

For every kind of dictionary data a pair of **basic_semantic_unit** and **dictionary_element** subtypes must be defined. Figure D.5 outlines, as a planning model, the Basic Semantic Units (BSU) and Dictionary Elements defined later. Note that the relationship between BSU and Dictionary Elements is redefined for each type of data, so that only corresponding pairs can be related. This is not graphically depicted here, however.

Every kind of dictionary data is treated in one of the following subclauses:

— for suppliers see D.3.4 "Supplier data";

— for classes see D.3.5 "Class data";

— for properties / data element types see D.3.6 "Data element type / properties data";

— for data types see D.3.7 "Domain data: the type System".

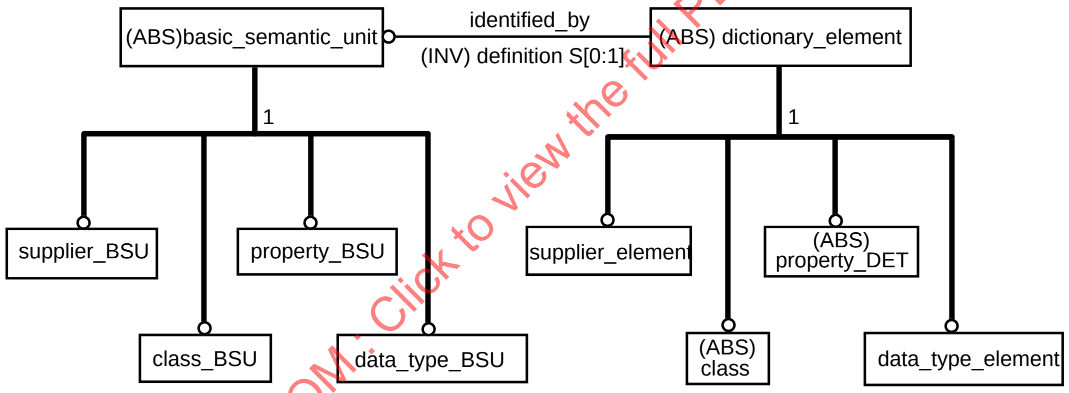


Figure D.5 — Current BSUs and dictionary elements

D.3.3.4 Identification of dictionary elements: three levels structure

The absolute identification of basic semantic units is based on the following three levels structure:

— supplier (of dictionary data);

— class ;

— class-related dictionary elements (any dictionary element defined in the context of a class; in this document class-related dictionary elements are **property_DET** and **data_type_element,** but there are provisions to extend this mechanism to other items).

An absolute identification can be achieved by concatenation of the applicable code for each level.

This identification scheme is appropriate within a multi-supplier context. If in a certain application area, only data of one single (data-) supplier are relevant, the corresponding parts of the identification, that are then constant, can be eliminated. For the purpose of exchange, however, all the levels must be present, to avoid clashes of identifiers.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

This identification scheme is described formally in the ..._BSU entities in the D.3.3 through D.3.6, attribute **absolute_id**.

D.3.3.5 Extension possibilities for other types of data

The BSU - Dictionary Element mechanism is very general and not limited to the four kinds of data used here (see figure D.5). This clause specifies some facilities that allow for extensions for other kinds. Depending on whether the scope of the identifier is given by a class or a supplier, the corresponding **..._related_BSU** entity has to be subtyped. It is necessary to redefine the **identified_by** attribute of the entity **dictionary_element** (as is done in the D.3.4 through D.3.7 for the current kinds of data).

D.3.3.5.1 Supplier_related_BSU

The **supplier_related_BSU** provides for the dictionary elements to be associated with suppliers, e.g. for ISO 13584: program libraries.

EXPRESS specification:

```
*)
ENTITY supplier_related_BSU
ABSTRACT SUPERTYPE
SUBTYPE OF (basic_semantic_unit);
END_ENTITY;
(*
```

D.3.3.5.2 Class_related_BSU

The **class_related_BSU** provides for the dictionary elements to be associated with classes, e.g. for ISO 13584 tables, documents, etc ...

EXPRESS specification:

```
*)
ENTITY class_related_BSU
ABSTRACT SUPERTYPE
SUBTYPE OF (basic_semantic_unit);
END_ENTITY;
(*
```

D.3.3.5.3 Supplier_BSU_relationship

The **supplier_BSU_relationship** is a provision for association of BSUs with suppliers.

EXPRESS specification:

```
*)
ENTITY supplier_BSU_relationship
ABSTRACT SUPERTYPE;
     relating_supplier: supplier_element;
     related_tokens: SET [ 1 : ? ] OF supplier_related_BSU;
END_ENTITY;
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Attribute definitions:

relating_supplier: the **supplier_element** that identifies the data supplier.

related_tokens: the set of dictionary elements associated to the supplier identified by the **relating_supplier** attribute.

D.3.3.5.4 Class_BSU_relationship

The **class_BSU_relationship** entity is a provision for association of BSUs with classes.

EXPRESS specification:

```
*)
ENTITY class_BSU_relationship
ABSTRACT SUPERTYPE;
    relating_class: class;
    related_tokens: SET [ 1 : ? ] OF class_related_BSU;
END_ENTITY;
(*
```

Attribute definitions:

relating_class: the **class** that identifies the dictionary element.

related_tokens: the set of dictionary elements associated to the class identified by the **relating_class** attribute**.**

D.3.4 Supplier Data

This clause contains definitions for the representation of data about a supplier itself. In a multi-supplier environment it is necessary to be able to identify the source of a certain dictionary element. Figure D.6 presents a planning model of the data associated with suppliers, followed by the EXPRESS definition.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

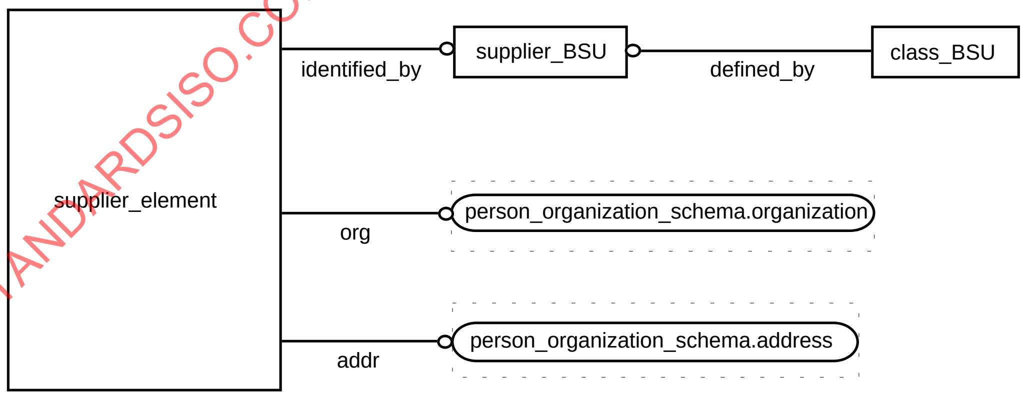

supplier_element
identified_by
supplier_BSU
defined_by
class_BSU
org
person_organization_schema.organization
addr
person_organization_schema.address

Figure D.6 — Overview of supplier data and relationships

D.3.4.1 Supplier_BSU

The **supplier_BSU** entity provides for unique identification of suppliers of dictionary data.

<u>EXPRESS specification:</u>

```
    *)
    ENTITY supplier_BSU
    SUBTYPE OF (basic_semantic_unit);
         SELF\basic_semantic_unit.code: supplier_code_type;
    DERIVE
         SELF\basic_semantic_unit.version: version_type:='001';
         absolute_id: identifier := SELF\basic_semantic_unit.code ;
    UNIQUE
         UR1: absolute_id;
    END_ENTITY;
    (*
```

<u>Attribute definitions:</u>

code: the supplier's code assigned according to ISO 13584-26.

version: the version number of a supplier code shall be equal to 001.

absolute_id: the absolute identification of the supplier.

<u>Formal propositions:</u>

UR1: the supplier identifier defined by the **absolute_id** attribute is unique.

D.3.4.2 Supplier_element

The **supplier_element** entity gives the dictionary description of suppliers.

<u>EXPRESS specification:</u>

```
    *)
    ENTITY supplier_element
    SUBTYPE OF (dictionary_element);
         SELF\dictionary_element.identified_by: supplier_BSU;
         org: organization;
         addr: address;
    INVERSE
         associated_items: SET [ 0 : ? ] OF supplier_BSU_relationship
                FOR relating_supplier;
    END_ENTITY;
    (*
```

<u>Attribute definitions:</u>

identified_by: the **supplier_BSU** used to identify this **supplier_element**.

org: the organizational data of this supplier.

addr: the address of this supplier.

associated_items: allows access to other kinds of data via the BSU mechanism (e.g. program library in ISO 13584).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

D.3.5 Class Data

This clause contains definitions for the representation of dictionary data of Classes.

D.3.5.1 General

Figure D.7 outlines, as a planning model, the data associated with Classes and their relationship to other Dictionary Elements.

As indicated in the figure with the **its_superclass** attribute, classes form an inheritance tree. It is important to note that throughout this document the terms "inheritance" and "to inherit" stand for this relationship between classes (defined in the dictionary), although EXPRESS has an inheritance concept, too. These must be clearly distinguished to avoid misunderstandings.

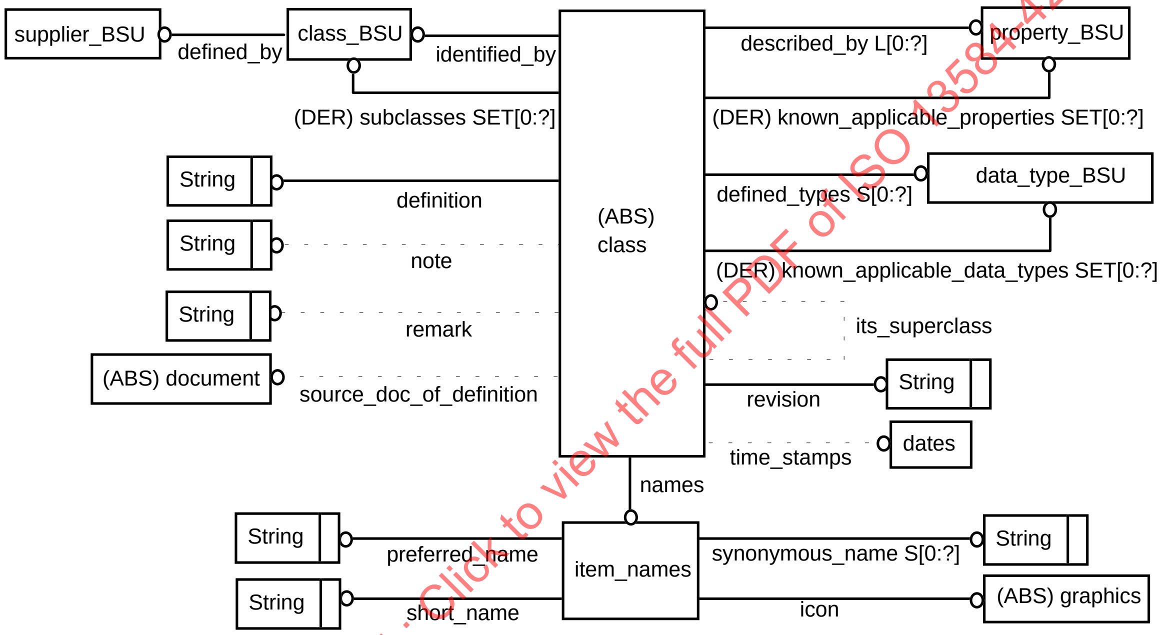


Figure D.7 — Overview of class data and relationships

The dictionary data for Component Classes (as shown in Figure D.7 - Overview of class data and relationships") is spread over four inheritance levels:

— **class_and_property_elements** defines data common to both classes and **property_DETs**;

— **class** allows for other kinds of classes to be specified later (e.g. in ISO 13584-24);

— **item_class** is the entity to hold data of different classes of application domain objects (e. g., components, materials,...);

— **component_class** is the entity that models family of parts and **material_class** is the entity that models class of materials.

D.3.5.1.1 Class_BSU

The **class_BSU** entity provides for the identification of classes.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
ENTITY class_BSU
SUBTYPE OF (basic_semantic_unit);
     SELF\basic_semantic_unit.code: class_code_type;
     defined_by: supplier_BSU;
DERIVE
     absolute_id: identifier
            := defined_by.absolute_id + sep_id + dic_identifier;
     known_visible_properties : SET [0 : ?]OF property_BSU
        :=compute_known_visible_properties(SELF);
     known_visible_data_types: SET [0 : ?]OF data_type_BSU
        :=compute_known_visible_data_types(SELF);
INVERSE
     subclasses: SET [0 : ?] OF class FOR its_superclass;
     added_visible_properties:SET [0 : ?] OF property_BSU
            FOR name_scope;
     added_visible_data_types:SET [0 : ?] OF data_type_BSU
            FOR name_scope;
UNIQUE
     UR1: absolute_id;
END_ENTITY; -- class_BSU
(*
```

Attribute definitions:

code: the code assigned to this class by its supplier.

defined_by: the supplier defining this class and its dictionary element

absolute_id: the unique identification of this class.

known_visible_properties: the set of **property_BSU**s that refer to the class as their **name_scope** attribute or to any known super-class of this class and that are therefore visible for the class (and any of its sub-class).

NOTE When some class **dictionary_definition** is not present in the same exchange context (a PLib exchange context is never assumed to be complete), the super-class of a class may not be known. Therefore the properties defined as visible by this super-class do not belong to the **known_visible_properties** attribute. Only on the receiving system all the **dictionary_definition**s of the BSUs are required to be available. Therefore, on the receiving sytem, the **kown_visible_properties** attribute contains all properties visible for the class.

known_visible_data_types: the set of **data_type_BSU**s that refer to the class as their **name_scope** attribute or to any known super-class of this class and that are therefore visible for the class (and any of its sub-class).

NOTE When some class **dictionary_definition** is not present in the same exchange context (a PLib exchange context is never assumed to be complete), the super-class of a class may not be known. Therefore the **data_type**s defined as visible by this super-class do not belong to the **known_visible_data_types** attribute. Only on the receiving sytem all the **dictionary_definition**s of the BSUs are required to be available. Therefore, on the receiving sytem, the **kown_visible_data_types** attribute contains all **data_type**s visible for the class.

subclasses: the set of classes specifying this class as their superclass.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

added_visible_properties: the set of **property_BSU**s that refer to the class as their **name_scope** and that are therefore visible for the class (and any of its sub-class).

NOTE 1 Only the **property_BSU**s that belongs to the same exchange context are referenced by this inverse attribute. On the receiving system they may already exit other **property_BSU**s that refer to this class (a PLib exchange context is never assumed to be complete).

NOTE 2 The **added_visible_properties** attribute will be used to encode the class attribute "Visible Properties" (see 7.2).

added_visible_data_types: the set of **data_type_BSU**s that refer to the class as their **name_scope** and that are therefore visible for the class (and any of its sub-class).

NOTE 1 Only the **data_type_BSU**s that belongs to the same exchange context are referenced by this inverse attribute. On the receiving system they may already exit other **data_type_BSU**s that refer to this class (a PLib exchange context is never assumed to be complete).

NOTE 2 The **added_visible_data_types** attribute will be used to encode the class attribute "Visible Types" (see 7.2).

<u>Formal propositions:</u>

UR1: the concatenation of supplier code and class code is unique.

D.3.5.1.2 Class_and_property_elements

The **class_and_property_elements** entity captures the attributes that are common to both **class**es and **property_DET**s.

<u>EXPRESS specification:</u>

```
*)
ENTITY class_and_property_elements
ABSTRACT SUPERTYPE OF ( ONEOF (
            property_DET,
            class))
SUBTYPE OF (dictionary_element);
     names: item_names;
     definition: definition_type;
     source_doc_of_definition: OPTIONAL document;
     note: OPTIONAL note_type;
     remark: OPTIONAL remark_type;
END_ENTITY;
(*
```

<u>Attribute definitions:</u>

names: the names describing this dictionary element.

definition: the text describing this dictionary element.

source_doc_of_definition: the source document of this textual description.

note: further information on any part of the dictionary element, which is essential to the understanding.

remark: explanatory text further clarifying the meaning of this dictionary element.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

NOTE 1 The **names** attribute will be used as a starting point to encode in the **item_names** entity the property and class attributes "Preferred Name", "Short Name" and "Synonymous Name" (see 6.2, 7.2 and D.3.8.2).

NOTE 2 The **definition** attribute will be used to encode the property attribute "Definition" (see 6.2) and the class attribute "Definition," (see 7.2).

NOTE 3 The **source_of_doc_definition** attribute will be used to encode the property attribute "Source Document of Definition" (see 6.2) and the class attribute "Source Document of Definition," (see 7.2).

NOTE 4 The **note** attribute will be used to encode the property and class attribute "Note" (see 6.2 and 7.2).

NOTE 5 The **remark** attribute will be used to encode the property and class attribute "Remark" (see 6.2 and 7.2).

D.3.5.1.3 Class

The **class** entity is an abstract resource for all kinds of classes.

EXPRESS specification:

```
*)
ENTITY class
ABSTRACT SUPERTYPE OF (item_class)
SUBTYPE OF (class_and_property_elements);
     SELF\dictionary_element.identified_by: class_BSU;
     its_superclass: OPTIONAL class_BSU;
     described_by: LIST [ 0 : ? ] OF UNIQUE property_BSU;
     defined_types: SET [ 0 : ? ] OF data_type_BSU;
DERIVE
     subclasses: SET [ 0 : ? ] OF class := identified_by.subclasses;
     known_applicable_properties : SET [ 0 : ? ] OF property_BSU
        := compute_known_applicable_properties(
             SELF\dictionary_element.identified_by);
     known_applicable_data_types : SET [ 0 : ? ] OF data_type_BSU
        := compute_known_applicable_data_types(
             SELF\dictionary_element.identified_by);
INVERSE
     associated_items: SET [ 0 : ? ] of class_BSU_relationship
             FOR relating_class;
WHERE
     WR1: acyclic_superclass_relationship ( SELF.identified_by, [ ] );
     WR2: NOT all_class_descriptions_reachable (
                    SELF\dictionary_element.identified_by)
             OR (list_to_set (SELF. described_by) <=
        SELF\dictionary_element.identified_by
                              \class_BSU.known_visible_properties);
     WR3: NOT all_class_descriptions_reachable (
                    SELF\dictionary_element.identified_by)
             OR (SELF. defined_types <=
        SELF\dictionary_element.identified_by
                              \class_BSU.known_visible_data_types);
     WR4: list_to_set (SELF. described_by) *
             NVL (SELF.its_superclass.definition[1]
                   \class.known_applicable_properties, [])= [] ;
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
        WR5: SELF.defined_types *
                NVL (SELF.its_superclass.definition[1]
                    \class.known_applicable_data_types, [])= []  ;
    END_ENTITY;
    (*
```

Attribute definitions:

identified_by: the **class_BSU** identifying this class.

its_superclass: reference to the class the current one is a subclass of.

described_by: the list of references to the additional properties available for use in the description of the parts within the class, and any of its subclasses

NOTE In ISO 13584-24, a property may also be applicable to a class when this property is imported from another class through the is-case-of or is-view-of semantics relationships. Therefore the properties referenced by the **described_by** attribute do not define all the applicable properties for an ISO 13584-defined class

defined_types: the set of references to the types that can be used for various **property_DET**s throughout the inheritance tree descending from this class.

NOTE In ISO 13584-24, a **data_type** may also be applicable to a class when this **data_type** is imported from another class through the is-case-of or is-view-of semantics relationships. Therefore the properties referenced by the **defined_types** attribute do not define all the applicable data types for an ISO 13584-defined class

subclasses: the set of classes specifying this class as their superclass.

known_applicable_properties : the **property_BSU**s that are referenced by the class or any of its known super-class by their **described_by** attribute and that are therefore applicable to this class (and to any of its sub-class).

NOTE When some class **dictionary_definition** is not present in the same exchange context (a PLib exchange context is never assumed to be complete), the super-class of a class may not be known. Therefore the properties defined as applicable by this super-class do not belong to the **known_applicable_properties** attribute. Only on the receiving sytem all the **dictionary_definitions** of the **BSU**s are required to be available. Therefore, on the receiving sytem, the **known_applicable_properties** attribute contains all the properties that are applicable to a class by virtue of being referenced by a **described_by** attribute

known_applicable_data_types: the **data_type_BSU**s that are referenced by the class or any of its known super-class by their **defined_types** attribute and that are therefore applicable to this class (and to any of its sub-class).

NOTE When some **class dictionary_definition** is not present in the same exchange context (a PLib exchange context is never assumed to be complete), the super-class of a class may not be known. Therefore the **data_type**s defined as applicable by this super-class do not belong to the **known_applicable_data_types** attribute. Only on the receiving sytem all the **dictionary_definition**s of the **BSU**s are required to be available. Therefore, on the receiving sytem, the **known_applicable_data_types** attribute contains all the **data_type**s that are applicable to a class by virtue of being referenced by a **defined_types** attribute

associated_items: allows to access other kinds of data using the BSU mechanism.

Formal propositions:

WR1: the inheritance structure defined by the class hierarchy does not contain cycles.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

WR2: only those properties that are visible for a class may become applicable to this class by virtue of being referenced by the **described_by** attribute.

WR3: only those data types that are visible for a class may become applicable to this class by virtue of being referenced by the **defined_types** attribute.

WR4: only those properties that are not applicable for a class by inheritance may become applicable to this class by virtue of being referenced by the **described_by** attribute.

WR5: only those data types that are not applicable for a class by inheritance may become applicable to this class by virtue of being referenced by the **defined_types** attribute.

NOTE 1 The **its_superclass** attribute will be used to encode the class attribute "Superclass" (see 7.2).

NOTE 2 The **described_by** attribute provides the encoding for the "Applicable Properties" of a class (see 7.2).

NOTE 3 The **defined_types** attribute is used to encode the "Applicable Types" attribute of a class (see 7.2).

D.3.5.2 Item_class

The entity **item_class** enables the modelling of any type of entity of the application domain that corresponds to an autonomous and stand-alone abstraction as a class. It is a supertype intended to be sub-typed to define the nature of the objects. Nevertheless, it is not defined as ABSTRACT to enable its instanciation to model the classes that are super-classes of two classes corresponding to two different kinds of objects (e. g., components and materials).

EXAMPLE 1 A material is an autonomous and stand-alone abstraction of an object of the parts library application domain. It is represented as a specific sub-class of **item_class**.

EXAMPLE 2 A feature is an autonomous and stand-alone abstraction of an object of the parts library application domain. It might be represented as a specific sub-class of **item_class**.

EXAMPLE 3 A product representation is not an autonomous and stand-alone abstraction of an object of the parts library application domain: it may only exist with a relation to a product. In ISO 13584-24, part representations are represented as specific sub-class of **class**.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
ENTITY item_class
SUPERTYPE OF (ONEOF(component_class, material_class))
SUBTYPE OF (class);
     simplified_drawing: OPTIONAL graphics;
     sub_class_properties: SET [ 0 : ? ] OF property_BSU;
     class_constant_values: SET [ 0 : ? ]
            OF class_value_assignment;
     coded_name: OPTIONAL value_code_type;
WHERE
     WR1: QUERY (p <* sub_class_properties
            | NOT (p IN SELF.described_by)) = [ ];
     WR2: NOT all_class_descriptions_reachable(SELF.identified_by) OR
            (QUERY (va <* class_constant_values | SIZEOF (QUERY (c <*
            va.super_class_defined_property.describes_classes |
            is_subclass (SELF, c)
            AND (va.super_class_defined_property
            IN  c\item_class.sub_class_properties))) <> 1) = []);
```

```
    END_ENTITY;
    (*
```

Attribute definitions:

simplified_drawing: optional drawings (**graphics**) that can be associated to the described class.

sub_class_properties: declares properties as class-valued, i.e. in subclasses one single value will be assigned per class. See D.3.6.4 "Class-Valued Properties ".

class_constant_values: assignments in the current class for class-valued properties declared in superclasses. See D.3.6.4 "Class-Valued Properties ".

coded_name: to be used in the value domain of the Classifying DET of the superclass.

NOTE This is only used in IEC 61360.

Formal propositions:

WR1: the **class_valued_properties** shall belong to the **described_by** list.

WR2: the properties referenced in **class_constant_values** were declared as class-valued in some superclass of the current class.

NOTE 1 The **simplified_drawing** attribute of the **item_class** entity is used to encode the "Simplified Drawing" attribute for classes (see 7.2).

NOTE 2 The **sub_class_properties** attribute of the **item_class** entity is used to encode the "Sub-class Selection Properties" attribute for classes (see 7.2).

NOTE 3 The **class_constant_values** attribute of the **item_class** entity is used to encode the "Class Value Assignment " for classes (see 7.2).

D.3.5.3 Component_class

The entity **component_class** captures the dictionary description of a class of items that represent, at some level of abstraction, parts or components. A property of which the data type is defined by a **component_class** stands for the aggregation relationship .

EXPRESS specification:

```
    *)
    ENTITY component_class
    SUBTYPE OF (item_class);
    END_ENTITY;
    (*
```

D.3.5.4 Material_class

The entity **material_class** captures the dictionary description of a class of materials. Materials are used to define properties of parts or components. Materials are associated with an idea of amount, they may not be counted. A property of which the data type is defined by a **material_class** captures that some (part of a) product is made of, or contains, some material.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
ENTITY material_class
SUBTYPE OF (item_class);
END_ENTITY;
(*
```

D.3.6 Data Element Type / properties data

This clause contains definitions for the dictionary data for properties.

D.3.6.1 Property_BSU

The entity **property_BSU** provides for identification of a property.

EXPRESS specification:

```
*)
ENTITY property_BSU
SUBTYPE OF (basic_semantic_unit);
     SELF\basic_semantic_unit.code: property_code_type;
     name_scope: class_BSU;
DERIVE
     absolute_id: identifier :=
            name_scope.defined_by.absolute_id        (* Supplier *)
            + sep_id + name_scope.dic_identifier     (* Class *)
            + sep_id + dic_identifier;               (* Property*)
INVERSE
     describes_classes: SET OF class FOR described_by;
UNIQUE
     UR1: absolute_id;
WHERE
     WR1: QUERY ( c <* describes_classes |
        NOT ( is_subclass (c, name_scope.definition[1]\class )))= [ ];
END_ENTITY;
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Attribute definitions:

code: to allow for unique identification within the scope indicated by the **name_scope** attribute.

name_scope: is the reference to the class at which or below which the property element is available for reference by the **described_by** attribute.

absolute_id: the unique identification of this property.

describes_classes: the classes declaring this property as available for use in the description of a part.

Formal propositions:

WR1: the class used in the **name_scope** attribute is a superclass of the one(s) where this **property_DET** is defined.

Formal propositions:

UR1: the property identifier **absolute_id** is unique.

NOTE The **name_scope** attribute of the **property_BSU** entity will be used to encode the "Definition Class" attribute for properties (see 6.2).

D.3.6.2 Property_DET

The **property_DET** entity captures the dictionary description of properties.

EXPRESS specification:

```
*)
ENTITY property_DET
ABSTRACT SUPERTYPE OF ( ONEOF (
           condition_DET,
           dependent_P_DET,
           non_dependent_P_DET))
SUBTYPE OF (class_and_property_elements);
     SELF\dictionary_element.identified_by: property_BSU;
     preferred_symbol: OPTIONAL mathematical_string;
     synonymous_symbols: SET [ 0 : 2 ] OF mathematical_string;
     figure: OPTIONAL graphics;
     det_classification: OPTIONAL DET_classification_type;
     domain: data_type;
     formula: OPTIONAL mathematical_string;
DERIVE
     describes_classes: SET [ 0 : ? ] OF class
           := identified_by.describes_classes;
END_ENTITY;
(*
```

Attribute definitions:

identified_by: the **property_BSU** identifying this property.

preferred_symbol: a shorter description of this property.

synonymous_symbols: synonymous for the shorter description of the property.

figure: an optional **graphics** that describes the property.

det_classification: the ISO 31 class for this property.

domain: the reference to the **data_type** associated to the property.

formula: a mathematical expression for explaining the property.

describes_classes: the classes declaring this property as available for use in the description of a part.

NOTE 1 The **preferred_symbol** attribute is used to encode the "Preferred Letter Symbol" attribute for properties (see 6.2).

NOTE 2 The **synonymous_symbols** attribute is used to encode the "Synonymous Letter Symbol" attribute for properties (see 6.2).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

NOTE 3 The **det_classification** attribute is used to encode the "Property Type Classification" attribute of a property (see 6.2).

NOTE 4 The **domain** attribute is used as a starting point for the encoding of the property attribute "Data Type" (see 6.2). The entity **data_type** will be subtyped for various possible data types.

NOTE 5 The **formula** attribute is used to encode the "Formula" attribute for properties (see 6.2).

Figure D.8 presents a planning model of the data associated with **property_DETs**.

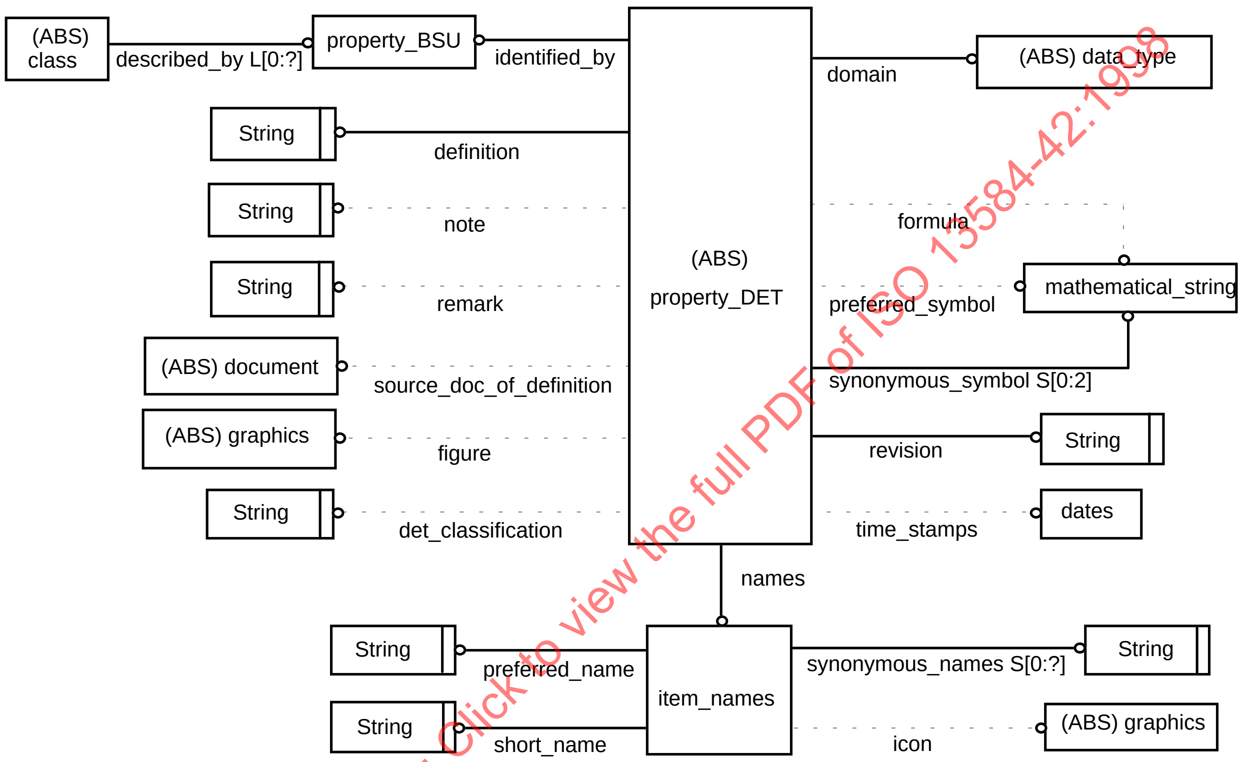


Figure D.8 — Overview of property data element type data and relationships

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

D.3.6.3 Condition, dependent and non-dependent Data Element Types

Figure D.9 depicts the various kinds of Data Element Types in the format of a planning model.

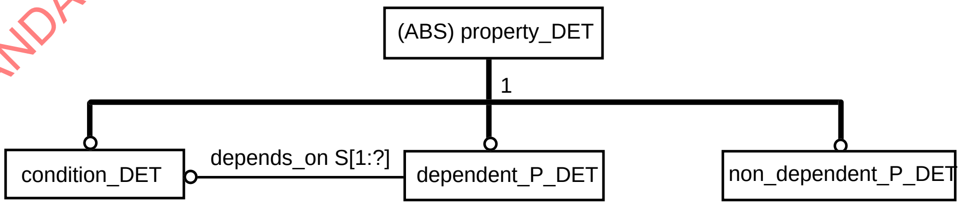


Figure D.9 — Kinds of data element types

Note that this figure (like others) is simplified: the "depends_on" relation essentially is implemented with a BSU reference, but a constraint is specified that the referred-to **property_DET** must be a **condition_DET** (see entity **dependent_P_DET**).

D.3.6.3.1 Condition_DET

A **condition_DET** is a property on which other properties may depend upon.

EXPRESS specification:

```
*)
ENTITY condition_DET
SUBTYPE OF (property_DET);
END_ENTITY;
(*
```

D.3.6.3.2 Dependent_P_DET

A **dependent_P_DET** is a property whose value depends explicitly on the value(s) of some condition(s), like e.g. ambient temperature.

EXPRESS specification:

```
*)
ENTITY dependent_P_DET
SUBTYPE OF (property_DET);
     depends_on: SET [ 1 : ? ] OF property_BSU;
WHERE
     WR1: QUERY ( p <* depends_on | NOT (definition_available_implies
(p,         ('ISO13584_IEC61360_DICTIONARY_SCHEMA.CONDITION_DET'
       IN TYPEOF (p.definition)))) ) = [ ];
END_ENTITY;
(*
```

Attribute definitions:

depends_on: the set of basic semantic units identifying the properties on which this property depends on.

Formal propositions:

WR1: only **condition_DET**s shall be used in the **depends_on** list.

D.3.6.3.3 Non_dependent_P_DET

A **non_dependent_P_DET** is a property that does not depend explicitly on certain conditions.

EXPRESS specification:

```
*)
ENTITY non_dependent_P_DET
SUBTYPE OF (property_DET);
END_ENTITY;
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

NOTE 1 The three subtypes **condition_DET**, **dependent_P_DET** and **non_dependent_P_DET** of the entity **property_DET** are used to encode the "Category" attribute for properties (see 6.2). **condition_DET** is used for context parameters, **dependent_P_DET** is used for context dependent characteristics and the **non_dependent_P_DET** entity is used for part characteristics.

NOTE 2 The **depends_on** attribute of the **dependent_P_DET** entity is used to encode the "Condition" attribute for properties (see 6.2).

D.3.6.4 Class-valued properties

Class-valued properties are those properties to which in each subclass one single value, valid for the whole subclass, is assigned, not for every instance within the class individually. By being included in the list **sub_class_properties** in entity **item_class** a property is distinguished as being of that type. In subclasses that inherit this property, values may be assigned using the attribute **class_constant_values** that contains a set of **class_value_assignment**s.

EXPRESS specification:

```
*)
ENTITY class_value_assignment;
     super_class_defined_property: property_BSU;
     assigned_value: value_code_type;
WHERE
     WR1: definition_available_implies (super_class_defined_property,
             ('ISO13584_IEC61360_DICTIONARY_SCHEMA'
             +'.NON_QUANTITATIVE_CODE_TYPE' IN TYPEOF (
             super_class_defined_property.
             definition[1]\property_DET.domain)));
     WR2: definition_available_implies (super_class_defined_property,
             ( SIZEOF ( QUERY ( v <*
             super_class_defined_property.
             definition[1]\property_DET.domain
             \non_quantitative_code_type.domain.its_values |
             assigned_value = v.value_code)) = 1));
END_ENTITY;
(*
```

Attribute definitions:

super_class_defined_property: the reference to the property (defined in a superclass as being a **class_valued_property**) to which a certain value is assigned.

assigned_value: the value assigned to the property, valid for the whole class referring this **class_value_assignment** instance in the **class_constant_values** list.

Formal propositions:

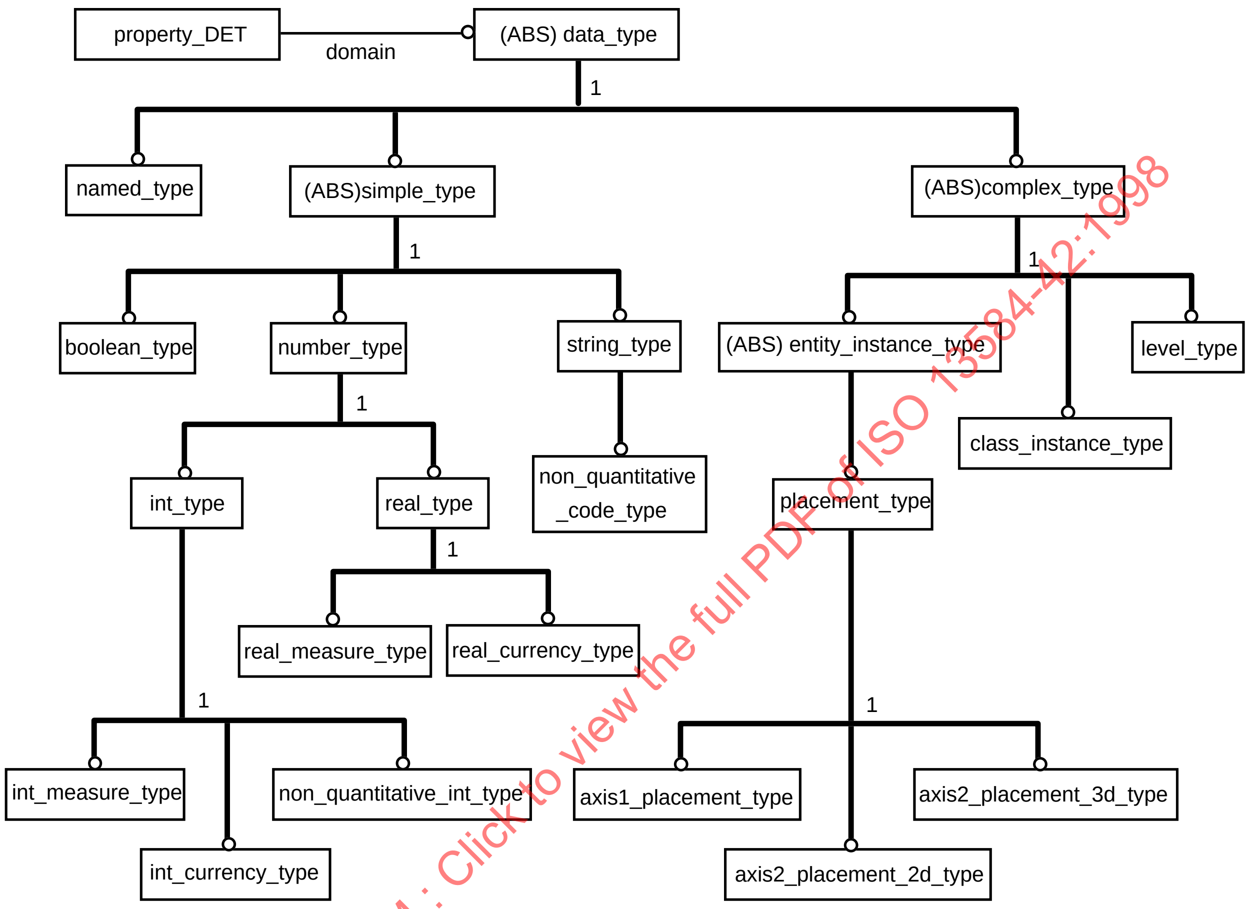
WR1: the **super_class_defined_property** must be of type non-quantitative with codes (strings) as values.

WR2: the value assigned to the **super_class_defined_property** shall be type compatible, i.e. it occurs in the value domain of the **super_class_defined_property**.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

D.3.7 Domain data: the type system

This clause contains definitions for the representation of the data types of a **property_DET**. Figure D.10 outlines, as a planning model, the entity hierarchy for data types.



STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Figure D.10 — Entity hierarchy for the type system

D.3.7.1 General

In contrast to the other dictionary elements (Suppliers, Classes, Properties) an identification with the Basic Semantic Unit concept is not mandatory for **data_type**, since it will be attached directly to the **property_DET** in many cases, and thus doesn't need an identification. However, the entities **data_type_BSU** and **data_type_element** allow for an unique identification where this is suitable. It provides for re-using the same type definition in another **property_DET** definition, even outside the current Physical File.

D.3.7.1.1 Data_type_BSU

The **data_type_BSU** entity provides for identification of **data_type_element**s.

EXPRESS specification:

```
    *)
    ENTITY data_type_BSU
    SUBTYPE OF (basic_semantic_unit);
         SELF\basic_semantic_unit.code: data_type_code_type;
         name_scope: class_BSU;
    DERIVE
         absolute_id: identifier :=
                name_scope.defined_by.absolute_id        (* Supplier*)
                + sep_id + name_scope.dic_identifier     (* Class*)
                + sep_id + dic_identifier;               (* Data_type *)
    INVERSE
         defining_class: SET [ 0 : 1 ] OF class FOR defined_types;
    UNIQUE
         absolute_id;
    WHERE
         WR1: is_subclass ( defining_class[1], name_scope.definition[1] );
    END_ENTITY;
    (*
```

Attribute definitions:

code: to allow for unique identification within the scope indicated by the **name_scope** attribute.

name_scope: the reference to the class at which or below which the data type element is available for reference by the **defined_types** attribute.

absolute_id: the unique identification of this property.

defining_class: the classes declaring this **data_type** as available for use in the description of a part.

Formal propositions:

WR1: the class used in the **name_scope** attribute is a superclass of the one where this **data_type** is defined.

NOTE The **name_scope** attribute is used to encode the reference to a class the related data type belongs to. This itself, beside the **data_type_element** entity (see below) , is part of the encoding of the class attribute "Visible Types" (see 7.2).

D.3.7.1.2 Data_type_element

The **data_type_element** entity describes the dictionary element for types. Note that it is not necessary in every case to have BSU and **dictionary_element** for a certain **data_type**, because a **property_DET** can refer to the **data_type** directly. Usage of the BSU relation is only necessary when a supplier wants to refer to the same type in a different Physical File.

EXPRESS specification:

```
    *)
    ENTITY data_type_element
    SUBTYPE OF (dictionary_element);
         SELF\dictionary_element.identified_by: data_type_BSU;
         names: item_names;
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
         type_definition: data_type;
     END_ENTITY;
     (*
```

Attribute definitions:

identified_by: the BSU that identifies the described **data_type_element**.

names: the names that allow the description of the defined **data_type_element**.

type_definition: the description of the type carried by the **data_type_element**.

NOTE The redeclared attribute **identified_by** is used to encode the reference to the BSU, this **data_type_element** is related to. This itself, beside the **data_type_BSU** entity (see above) is used to encode the class attribute "Visible Types" (see 7.2).

D.3.7.2 The type system

D.3.7.2.1 Data_type

The **data_type** entity serves as a common supertype for the entities used to indicate the type of the associated DET.

EXPRESS specification:

```
     *)
     ENTITY data_type
     ABSTRACT SUPERTYPE OF ( ONEOF (
                 simple_type,
                 complex_type,
                 named_type));
     END_ENTITY;
     (*
```

D.3.7.2.2 Simple_type

The **simple_type** entity serves as a common supertype for the entities used to indicate a simple type of the associated DET.

EXPRESS specification:

```
     *)
     ENTITY simple_type
     ABSTRACT SUPERTYPE OF ( ONEOF (
                 number_type,
                 boolean_type,
                 string_type))
     SUBTYPE OF(data_type);
         value_format: value_format_type;
     END_ENTITY;
     (*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Attribute definitions:

value_format: the encoding of the format of values for properties.

NOTE - The **value_format** attribute of the **simple_type** entity is used to encode the "Value Format" attribute for properties (see 6.2).

D.3.7.2.3 Number_type

The **number_type** entity provides for values of DETs that are of type NUMBER.

EXPRESS specification:

```
*)
ENTITY number_type
SUPERTYPE OF ( ONEOF (
           int_type,
           real_type))
SUBTYPE OF (simple_type);
END_ENTITY;
(*
```

D.3.7.2.4 Int_type

The **int_type** entity provides for values of DETs that are of type INTEGER.

EXPRESS specification:

```
*)
ENTITY int_type
SUPERTYPE OF ( ONEOF (
           int_measure_type,
           int_currency_type,
           non_quantitative_int_type))
SUBTYPE OF (number_type);
END_ENTITY;
(*
```

D.3.7.2.5 Int_measure_type

The **int_measure_type** entity provides for values of DETs that are measures of type INTEGER.

EXPRESS specification:

```
*)
ENTITY int_measure_type
SUBTYPE OF (int_type);
    unit: dic_unit;
END_ENTITY;
(*
```

Attribute definitions:

unit: the unit associated to the described measure.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

NOTE The attribute **unit** is used to encode the "Unit" attribute for properties (see 6.2).

D.3.7.2.6 Int_currency_type

The **int_currency_type** entity provides for values of DETs that are integer currencies.

EXPRESS specification:

```
*)
ENTITY int_currency_type
SUBTYPE OF (int_type);
    currency: OPTIONAL currency_code;
END_ENTITY;
(*
```

Attribute definitions:

currency: the associated code of the described currency according to ISO 4217. If not present, the currency code has to be exchanged together with the data (values).

D.3.7.2.7 Non_quantitative_int_type

The **non_quantitative_int_type** entity is an enumeration type where elements of the enumeration are represented with an INTEGER value (see also ENTITY **non_quantitative_code_type** and figure D.11.).

EXPRESS specification:

```
*)
ENTITY non_quantitative_int_type
SUBTYPE OF (int_type);
    domain: value_domain;
WHERE
    WR1: QUERY (v <* domain.its_values |
       'ISO13584_IEC61360_DICTIONARY_SCHEMA.VALUE_CODE_TYPE' IN
        TYPEOF ( v.value_code )) = [ ];
END_ENTITY;
(*
```

Attribute definitions:

domain: the set of enumerated values described in the **value_domain** entity.

Formal propositions:

WR1: the values associated with the **domain.its_values** list shall not contain a **value_code_type**.

D.3.7.2.8 Real_type

The **real_type** entity provides for values of DETs that are of type REAL.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
ENTITY real_type
SUPERTYPE OF ( ONEOF (
            real_measure_type,
            real_currency_type))
SUBTYPE OF (number_type);
END_ENTITY;
(*
```

D.3.7.2.9 Real_measure_type

The **real_measure_type** entity provides for values of DETs that are measures of type REAL.

EXPRESS specification:

```
*)
ENTITY real_measure_type
SUBTYPE OF (real_type);
     unit: dic_unit;
END_ENTITY;
(*
```

Attribute definitions:

unit: the unit associated to the described measure.

NOTE The attribute **unit** is used to encode the "Unit" attribute for properties (see 6.2).

D.3.7.2.10 Real_currency_type

The **real_currency_type** entity defines real currencies.

EXPRESS specification:

```
*)
ENTITY real_currency_type
SUBTYPE OF (real_type);
     currency: OPTIONAL currency_code;
END_ENTITY;
(*
```

Attribute definitions:

currency: the associated code of the described currency according to ISO 4217. If not present, the currency code has to be exchanged together with the data (values).

D.3.7.2.11 Boolean_type

The **boolean_type** entity provides for values of DETs that are of type BOOLEAN.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
    *)
    ENTITY boolean_type
    SUBTYPE OF (simple_type);
    END_ENTITY;
    (*
```

D.3.7.2.12 String_type

The **string_type** provides for values of DETs that are of type STRING.

EXPRESS specification:

```
    *)
    ENTITY string_type
    SUBTYPE OF (simple_type);
    END_ENTITY;
    (*
```

D.3.7.2.13 Non_quantitative_code_type

The **non_quantitative_code_type** entity is an enumeration type where elements of the enumeration are represented with a STRING value (see also ENTITY **non_quantitative_int_type** and figure D.11).

EXPRESS specification:

```
    *)
    ENTITY non_quantitative_code_type
    SUBTYPE OF (string_type);
         domain: value_domain;
    WHERE
         WR1: QUERY (v <* domain.its_values |
            NOT ('ISO13584_IEC61360_DICTIONARY_SCHEMA.VALUE_CODE_TYPE' IN
            TYPEOF ( v.value_code ))) = [ ];
    END_ENTITY;
    (*
```

Attribute definitions:

domain: the set of enumerated values described in the **value_domain** entity.

Formal propositions:

WR1: the values associated with the **domain.its_values** list shall only contain elements of type **value_code_type**.

D.3.7.2.14 Complex_type

The **complex_type** entity provides for the definition of types of which the values are represented as EXPRESS instances.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
    *)
    ENTITY complex_type
    ABSTRACT SUPERTYPE OF ( ONEOF (
                level_type,
                class_instance_type,
                entity_instance_type))
    SUBTYPE OF(data_type );
    END_ENTITY;
    (*
```

D.3.7.2.15 Level_type

The **level_type** entity provides an indicator to qualify the values of a quantitative data element type.

EXPRESS specification:

```
    *)
    ENTITY level_type
    SUBTYPE OF (complex_type);
                levels: LIST [ 1 : 4 ] OF UNIQUE level;
                value_type: simple_type;
    WHERE
         WR1: 'ISO13584_IEC61360_DICTIONARY_SCHEMA.NUMBER_TYPE'
             IN TYPEOF ( value_type );
    END_ENTITY;
    (*
```

Attribute definitions:

levels: the list of unique elements that specifies which qualified values shall be associated with the property.

value_type: the type of value of the different qualified values.

Formal propositions:

WR1: the **SELF.value_type** shall be of type **number_type**

D.3.7.2.16 Level

The **level** type provides an abbreviated name of the different qualified values that may be associated with a physical quantity, distinguishing it from other possible or allowed values of the same quantity.

EXPRESS specification:

```
    *)
    TYPE level = ENUMERATION OF (
         min,   (*corresponds  to  the  minimal  value  of  the  physical
    quantity*)
         nom,   (*corresponds  to  the  nominal  value  of  the  physical
    quantity*)
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
        typ,   (*corresponds  to  the  typical  value  of  the  physical
    quantity*)
        max);  (*corresponds  to  the  maximal  value  of  the  physical
    quantity*)
    END_TYPE;
    (*
```

D.3.7.2.17 Class_instance_type

The **class_instance_type** entity provides for values of DETs that are represented as instances of a **class**. It is used, in particular, for the description of assemblies or to describe the material a (part of a) component consists of.

EXPRESS specification:

```
    *)
    ENTITY class_instance_type
    SUBTYPE OF (complex_type);
        domain: class_BSU;
    END_ENTITY;
    (*
```

Attribute definitions:

domain: the **class_BSU** referring to the **class** representing the described type.

D.3.7.2.18 Entity_instance_type

The **entity_instance_type** entity provides for values of DETs that are represented as instances of some EXPRESS entity data types. A **type_name** attribute enables the specification what are the allowed data types. This attribute, together with the EXPRESS TYPEOF function applied to the value, permits strong type checking and polymorphism. This entity will be subtyped below for some data types that are allowed for use in the dictionary schema.

EXPRESS specification:

```
    *)
    ENTITY entity_instance_type
    ABSTRACT SUPERTYPE
    SUBTYPE OF (complex_type);
        type_name: SET OF STRING;
    END_ENTITY;
    (*
```

Attribute definitions:

type_name: the set of strings that describe, in the format of the EXPRESS TYPEOF function, the EXPRESS entity data type names that shall belong to the result of the EXPRESS TYPEOF function when it is applied to a value that references the present entity as its data type.

D.3.7.2.19 Placement_type

The **placement_type** entity provides for values of DETs that are instances of **placement** entity data type. (See ISO 10303-42 for details).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
ENTITY placement_type
SUPERTYPE OF ( ONEOF (
           axis1_placement_type,
           axis2_placement_2d_type,
           axis2_placement_3d_type))
SUBTYPE OF (entity_instance_type);
WHERE
    WR1: 'GEOMETRY_SCHEMA.PLACEMENT'
           IN SELF\entity_instance_type.type_name;
END_ENTITY;
(*
```

Formal propositions:

WR1: the string 'GEOMETRY_SCHEMA.PLACEMENT' shall be contained in the set defined by the **SELF\entity_instance_type.type_name** attribute.

D.3.7.2.20 Axis1_placement_type

The **axis1_placement_type** entity provides for values of DETs that are instances of **axis1_placement**. entity data type. (See ISO 10303-42 for details).

EXPRESS specification:

```
*)
ENTITY axis1_placement_type
SUBTYPE OF (placement_type);
WHERE
    WR1: 'GEOMETRY_SCHEMA.AXIS1_PLACEMENT' IN
            SELF\entity_instance_type.type_name;
END_ENTITY;
(*
```

Formal propositions:

WR1: the string 'GEOMETRY_SCHEMA.AXIS1_PLACEMENT' shall be contained in the set defined for the **SELF\entity_instance_type.type_name** attribute.

D.3.7.2.21 Axis2_placement_2d_type

The **axis2_placement_2d_type** entity provides for values of DETs that are instances of **axis2_placement_2d** entity data type (See ISO 10303-42 for details).

EXPRESS specification:

```
*)
ENTITY axis2_placement_2d_type
SUBTYPE OF (placement_type);
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
    WHERE
         WR1: 'GEOMETRY_SCHEMA.AXIS2_PLACEMENT_2D'
                IN SELF\entity_instance_type.type_name;
    END_ENTITY;
    (*
```

Formal propositions:

WR1: the string 'GEOMETRY_SCHEMA.AXIS2_PLACEMENT_2D' shall be contained in the set defined for the **SELF\entity_instance_type.type_name** attribute.

D.3.7.2.22 Axis2_placement_3d_type

The **axis2_placement_3d_type** entity provides for values of DETs that are instances of **axis2_placement_3d** entity data type. (See ISO 10303-42 for details).

EXPRESS specification:

```
    *)
    ENTITY axis2_placement_3d_type
    SUBTYPE OF (placement_type);
    WHERE
         WR1: 'GEOMETRY_SCHEMA.AXIS2_PLACEMENT_3D'
            IN SELF\entity_instance_type.type_name;
    END_ENTITY;
    (*
```

Formal propositions:

WR1: the string 'GEOMETRY_SCHEMA.AXIS2_PLACEMENT_3D' shall be contained in the set defined for the **SELF\entity_instance_type.type_name** attribute.

D.3.7.2.23 Named_type

The **named_type** entity provides for referring to other types via the BSU mechanism.

EXPRESS specification:

```
    *)
    ENTITY named_type
    SUBTYPE OF (data_type );
         referred_type: data_type_BSU;
    END_ENTITY;
    (*
```

Attribute definitions:

referred_type: the BSU identifying the **data_type** the present entity refers to.

D.3.7.3 Values

This clause contains definitions for non-quantitative data element types (see entity **non_quantitative_int_type** and entity **non_quantitative_code_type)**.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Figure D.11 outlines, as a planning model, the data associated with non-quantitative data element types.

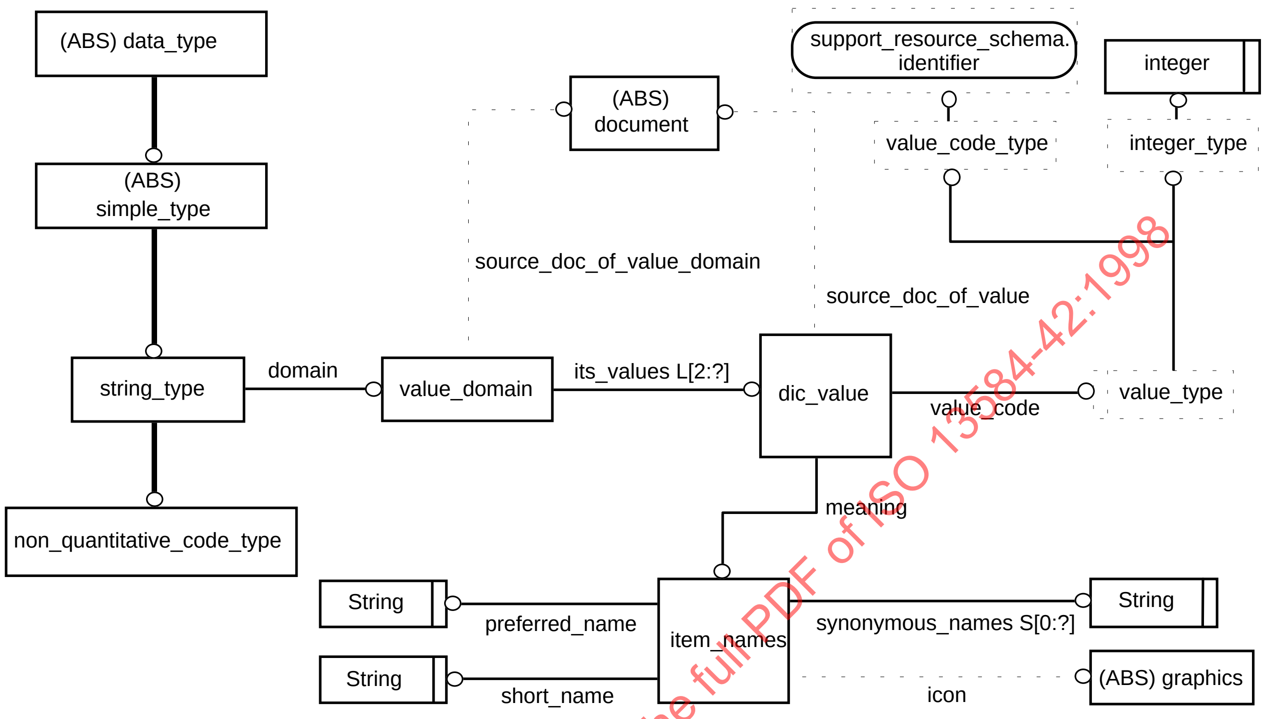


Figure D.11 — Overview of non-quantitative data element Types

D.3.7.3.1 Value_domain

The **value_domain** entity describes the set of allowed values for a non-quantitative data element type.

EXPRESS specification:

```
*)
ENTITY value_domain;
     its_values: LIST [ 2 : ? ] OF dic_value;
     source_doc_of_value_domain: OPTIONAL document;
     languages: OPTIONAL present_translations;
     terms: LIST [ 0 : ? ] OF item_names;
WHERE
     WR1: NOT EXISTS ( languages ) XOR ( QUERY ( v <* its_values |
            languages :<>: v.meaning.languages ) = [ ]);
     WR2: codes_are_unique (its_values);
END_ENTITY;
(*
```

Attribute definitions:

its_values: the enumeration list of values contained in the described domain.

source_doc_of_value_domain: the document describing the domain associated to the described **value_domain** entity.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

languages: the optional languages in which the translations are provided.

terms: the list of **item_names** to allow for IEC 61360 the link to the terms dictionary.

Formal propositions:

WR1: if the value meanings are provided in more than one language, then the set of languages used must be the same for the whole set of values.

WR2: value codes must be unique within this data type.

D.3.7.3.2 Value_type

Each value of a non-quantitative data element is associated with a code, that characterizes the value. A **value_type** may be either an INTEGER or a **value_code_type**.

EXPRESS specification:

```
    *)
    TYPE integer_type = INTEGER; END_TYPE;
    TYPE value_type = SELECT (value_code_type, integer_type); END_TYPE;
    (*
```

D.3.7.3.3 Dic_value

The **dic_value** entity is one of the values of a **value_domain** entity.

EXPRESS specification:

```
    *)
    ENTITY dic_value;
        value_code: value_type;
        meaning: item_names;
        source_doc_of_value: OPTIONAL document;
    END_ENTITY;
    (*
```

Attribute definitions:

value_code: the code associated to the described value. It can be either a **value_code_type** or an INTEGER.

meaning: the meaning associated to this value. It is provided by names.

source_doc_of_value: the optional source document in which the value is defined.

D.3.7.4 Extension to ISO 10303-41 unit definitions

This clause defines the resources for description of units in a dictionary. It extends the resources defined in ISO 10303-41.

D.3.7.4.1 Non_si_unit

The **non_si_unit** entity extends the unit model of ISO 10303-41 to allow for the representation of non-SI-units that are neither context dependent, nor conversion-based (See ISO 10303-41 for details).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
ENTITY non_si_unit
SUBTYPE OF (named_unit);
     name: label;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** used to name the described unit.

D.3.7.4.2 Assert_ONEOF rule

The **assert_ONEOF** rule asserts that ONEOF holds between the following subtypes of **named_unit**: **si_unit**, **context_dependent_unit**, **conversion_based_unit**, **non_si_unit**.

EXPRESS specification:

```
*)
RULE assert_ONEOF FOR (named_unit);
WHERE
     QUERY (u <* named_unit |
        ('ISO13584_IEC61360_DICTIONARY_SCHEMA.NON_SI_UNIT'
        IN TYPEOF(u)) AND
        ('MEASURE_SCHEMA.SI_UNIT' IN TYPEOF(u))
     OR ('ISO13584_IEC61360_DICTIONARY_SCHEMA.NON_SI_UNIT'
        IN TYPEOF(u)) AND
        ('MEASURE_SCHEMA.CONTEXT_DEPENDENT_UNIT' IN TYPEOF(u))
     OR ('ISO13584_IEC61360_DICTIONARY_SCHEMA.NON_SI_UNIT'
        IN TYPEOF(u)) AND
        ('MEASURE_SCHEMA.CONVERSION_BASED_UNIT' IN TYPEOF(u))
     ) = [];
END_RULE;
(*
```

D.3.7.4.3 Dic_unit

The basic representation of units is in structured form according to ISO 10303-41. But since one of the purposes of storing units in the dictionary is for the presentation to the user, a structured representation alone is not sufficient, it must be supplemented by a string representation.

The present definitions allow various possibilities:

— the function **string_for_unit** (see D.3.9 "Function Definitions") can be used. For a given structured representation of an unit it returns a string representation corresponding to the one used in Annex B of IEC 61360-1;

— a string representation can be supplied in plain text form (entity **mathematical_string**, attribute **text_representation**);

— an SGML representation can be supplied to allow for an enhanced presentation of the unit including sub- and superscripts etc... (entity **mathematical_string**, attribute **SGML_representation**).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

The **dic_unit** entity describes an unit to be stored in a dictionary.

EXPRESS specification:

```
*)
ENTITY dic_unit;
     structured_representation: unit;
     string_representation: OPTIONAL mathematical_string;
END_ENTITY;
(*
```

Attribute definitions:

structured_representation: structured representation, from ISO 10303-41, including extension defined in D.3.7.4 "Extension to ISO 10303-41 definitions".

string_representation: the function **string_for_unit** can be used to compute a string representation from the **structured_representation**, for the case where no **string_representation** is present.

NOTE The **structured_representation** attribute of the entity **dic_unit** is used to encode the property attribute "Unit" (see 6.2).

D.3.8 Basic type and entity definitions

This subclause contains the basic type and entity definitions that were used in the main part of the model.

D.3.8.1 Basic type definitions

This subclause contains the basic type and entity definitions, sorted alphabetically.

D.3.8.1.1 Class_code_type

The **class_code_type** identifies the allowed values for a class code.

EXPRESS specification:

```
*)
TYPE class_code_type = code_type;
WHERE
     WR1: LENGTH(SELF) <= class_code_len;
END_TYPE;
(*
```

Formal propositions:

WR1: the length of values corresponding to **class_code_type** shall be less or equal to the length of **class_code_len** (i.e. 14).

D.3.8.1.2 Code_type

The **code_type** identifies the allowed values for a code type.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
TYPE code_type = identifier;
WHERE
     WR1: NOT (SELF LIKE '*.*');
     WR2: NOT (SELF LIKE '*-*');
     WR3: NOT( SELF LIKE '* *');
END_TYPE;
(*
```

Formal propositions:

WR1: the '.' shall not be contained in a **code_type** value. '.' is used to concatenate identifiers, (see: CONSTANT **sep_id**).

WR2: the '-' shall not be contained in a **code_type** value.'-' is used to compute the identifier as the concatenation of code and version, (see: CONSTANT **sep_cv**).

WR3: spaces are not allowed, to avoid problems with leading and trailing blanks when concatenating codes.

D.3.8.1.3 Currency_code

The **currency_code** identifies the values allowed for a currency code.

These values are defined according to ISO 4217. Values are e.g., "CHF" for Swiss Francs, "CNY" for Yuan Renminbi (Chinese), "DEM" for German Mark, "FRF" for French Francs, "JPY" for Yen (Japanese), "SUR" for SU Rouble, "USD" for US Dollars, "XEU" for ECU (European Community Unit) etc ...

EXPRESS specification:

```
*)
TYPE currency_code = identifier;
WHERE
     WR1: LENGTH(SELF) = 3;
END_TYPE;
(*
```

Formal propositions:

WR1: the length of a **currency_code** value shall be equal to 3.

D.3.8.1.4 Date_type

The **date_type** identifies the values allowed for a date.

These values are defined according to ISO 8601 (e.g. "1994-03-21").

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
TYPE date_type = STRING(10) FIXED;
END_TYPE;
(*
```

D.3.8.1.5 Definition_type

The **definition_type** identifies the values allowed for a definition.

EXPRESS specification:

```
*)
TYPE definition_type = translatable_text;
END_TYPE;
(*
```

D.3.8.1.6 DET_classification_type

The **DET_classification_type** identifies the values allowed for a the DET classification. These values are used for DET classification according to ISO 31.

EXPRESS specification:

```
*)
TYPE DET_classification_type = identifier;
WHERE
     WR1: LENGTH(SELF) = DET_classification_len;
END_TYPE;
(*
```

Formal propositions:

WR1: the length of a **DET_classification_type** value shall be equal to the value of a **DET_classification_len** (i.e. 3).

D.3.8.1.7 Data_type_code_type

The **data_type_code_type** identifies the values allowed for a data type code.

EXPRESS specification:

```
*)
TYPE data_type_code_type = code_type;
WHERE
     WR1: LENGTH(SELF) = data_type_code_len;
END_TYPE;
(*
```

Formal propositions:

WR1: the length of a **data_type_code_type** value shall be equal to the value of a **data_type_code_len** (i.e. 14).

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

D.3.8.1.8 Note_type

The **note_type** identifies the values allowed for a note.

EXPRESS specification:

```
    *)
    TYPE note_type = translatable_text;
    END_TYPE;
    (*
```

D.3.8.1.9 Pref_name_type

The **pref_name_type** identifies the values allowed for a preferred name.

EXPRESS specification:

```
    *)
    TYPE pref_name_type = translatable_label;
    WHERE
         WR1: check_label_length (SELF, pref_name_len);
    END_TYPE;
    (*
```

Formal propositions:

WR1: the length of a **pref_name_type** value shall not exceed the length of **pref_name_len** (i.e. 30).

D.3.8.1.10 Property_code_type

The **property_code_type** identifies the values allowed for a property code.

EXPRESS specification:

```
    *)
    TYPE property_code_type = code_type;
    WHERE
         WR1: LENGTH(SELF) <= property_code_len;
    END_TYPE;
    (*
```

Formal propositions:

WR1: the length of a **property_code_type** value shall be less or equal to the value of a **property_code_len** (i.e. 14).

D.3.8.1.11 Remark_type

The **remark_type** identifies the values allowed for a remark.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
    *)
    TYPE remark_type = translatable_text;
    END_TYPE;
    (*
```

D.3.8.1.12 Revision_type

The **revision_type** identifies the values allowed for a revision.

EXPRESS specification:

```
    *)
    TYPE revision_type = code_type;
    WHERE
         WR1: LENGTH(SELF) <= revision_len;
    END_TYPE;
    (*
```

Formal propositions:

WR1: the length of a **revision_type** value shall not exceed the length of **revision_len** (i.e. 3).

D.3.8.1.13 Short_name_type

The **short_name_type** identifies the values allowed for a short name.

EXPRESS specification:

```
    *)
    TYPE short_name_type = translatable_label;
    WHERE
         WR1: check_label_length (SELF, short_name_len);
    END_TYPE;
    (*
```

Formal propositions:

WR1: the length of a **short_name_type** value shall not exceed the length of **short_name_len** (i.e. 15).

D.3.8.1.14 Supplier_code_type

The **supplier_code_type** identifies the values allowed for a supplier code.

EXPRESS specification:

```
    *)
    TYPE supplier_code_type = code_type;
    WHERE
         WR1: LENGTH(SELF) <= supplier_code_len;
    END_TYPE;
    (*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Formal propositions:

WR1: the length of a **supplier_code_type** value shall be less or equal to the value of **supplier_code_len** (i.e. 18).

D.3.8.1.15 Syn_name_type

The **syn_name_type** identifies the values allowed for a synonymous name.

EXPRESS specification:

```
*)
TYPE syn_name_type = SELECT (label_with_language, label);
WHERE
     WR1: check_syn_length (SELF, syn_name_len);
END_TYPE;
(*
```

Formal propositions:

WR1: the length of a **syn_name_type** value shall be equal to the value of **syn_name_len** (i.e. 30).

D.3.8.1.16 Value_code_type

The **value_code_type** identifies the values allowed for a value code.

EXPRESS specification:

```
*)
TYPE value_code_type = identifier;
WHERE
     WR1: LENGTH(SELF) <= value_code_len;
END_TYPE;
(*
```

Formal propositions:

WR1: the length of a **value_code_type** value shall not exceed the length of **value_code_len** (i.e. 18).

D.3.8.1.17 Value_format_type

The **value_format_type** identifies the values allowed for a value format. These values are defined according to ISO 6093 and ISO 9735.

EXPRESS specification:

```
*)
TYPE value_format_type = identifier;
WHERE
     WR1: LENGTH(SELF) <= value_format_len;
END_TYPE;
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Formal propositions:

WR1: the length of a **value_format_type** value shall not exceed the length of **value_format_len** (i.e. 80).

D.3.8.1.18 Version_type

The **version_type** identifies the values allowed for a version.

EXPRESS specification:

```
    *)
    TYPE version_type = code_type;
    WHERE
         WR1: LENGTH(SELF) = version_len;
         WR2:SELF LIKE '###';
    END_TYPE;
    (*
```

Formal propositions:

WR1: the length of a **version_type** value shall be equal to **version_len** (i.e. 3).

WR2: the **version_type** shall contain digits only.

D.3.8.1.19 Source_doc_type

The **source_doc_type** identifies the values allowed for a source document.

EXPRESS specification:

```
    *)
    TYPE source_doc_type = identifier;
    WHERE
         WR1: LENGTH(SELF) <= source_doc_len;
    END_TYPE;
    (*
```

Formal propositions:

WR1: the length of a **source_doc_type** value shall not exceed the length of **source_doc_len** (i.e. 80).

D.3.8.2 Basic entity definitions

This subclause contains the basic entity definitions, sorted alphabetically.

D.3.8.2.1 Dates

The **dates** entity describes the three dates associated respectively to the first version, the current version and the current revision for a given description.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
ENTITY dates;
    date_of_original_definition: date_type;
    date_of_current_version: date_type;
    date_of_current_revision: OPTIONAL date_type;
END_ENTITY;
(*
```

Attribute definitions:

date_of_original_definition: the date associated to version 001.

date_of_current_version: the date associated to the present version.

date_of_current_revision: the date associated to the last revision.

D.3.8.2.2 Document

The **document** entity is an abstract resource that stands for a document. The dictionary schema only provides for exchanging the identification of documents (see below). The **document** entity may also be subtyped with entities implementing a means for exchanging document data, e.g. by reference to an external file and exact specification of the format of the file.

EXPRESS specification:

```
*)
ENTITY document
ABSTRACT SUPERTYPE;
END_ENTITY;
(*
```

D.3.8.2.3 Graphics

The **graphics** entity is to be subtyped with entities implementing a means for exchanging graphical data, e.g. by reference to an external file and exact specification of the format of the file.

EXPRESS specification:

```
*)
ENTITY graphics
ABSTRACT SUPERTYPE;
END_ENTITY;
(*
```

D.3.8.2.4 Identified_document

The **identified_document** entity describes a document identified by its code.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
    *)
    ENTITY identified_document
    SUBTYPE OF (document);
         document_identifier: source_doc_type;
    END_ENTITY;
    (*
```

Attribute definitions:

document_identifier: the code of the described document.

D.3.8.2.5 Item_names

The **item_names** entity identifies the names that can be associated to a given description. It states the preferred name, the set of synonymous names, the short name and the **languages** in which the different names are provided. It may be associated with an icon.

EXPRESS specification:

```
    *)
    ENTITY item_names;
         preferred_name: pref_name_type;
         synonymous_names: SET OF syn_name_type;
         short_name: short_name_type;
         languages: OPTIONAL present_translations;
         icon : OPTIONAL graphics;
    WHERE
         WR1: NOT ( EXISTS ( languages ) XOR (
            ('ISO13584_IEC61360_LANGUAGE_RESOURCE_SCHEMA'
                +'.TRANSLATED_LABEL' IN TYPEOF (preferred_name))
            AND (languages :=: preferred_name\translated_label.languages)
            AND ('ISO13584_IEC61360_LANGUAGE_RESOURCE_SCHEMA'
                +'.TRANSLATED_LABEL' IN TYPEOF (short_name))
            AND (languages :=: short_name\translated_label.languages)
            AND (QUERY ( s <* synonymous_names |
            NOT ('ISO13584_IEC61360_DICTIONARY_SCHEMA.LABEL_WITH_LANGUAGE’
                IN TYPEOF (s))) = [ ])));
         WR2: NOT EXISTS(languages) XOR (QUERY ( s <* synonymous_names |
            EXISTS(s.language) AND NOT (s.language IN
            QUERY ( l <* languages.language_codes | TRUE
            (* i.e. take all *) ))) = [ ]);
         WR3: at_most_two_synonyms_per_language
            (languages, synonymous_names);
    END_ENTITY;
    (*
```

Attribute definitions:

preferred_name: the name that is preferred for use.

synonymous_names: the set of synonymous names.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

short_name: the abbreviation of the preferred name.

languages: the optional list of languages in which the different names are provided.

icon : an optional **icon** which graphically represents the description associated with the **item_names**.

Formal propositions:

WR1: if preferred and short names are provided in more than one language, then all the "languages" attributes of the **translated_label**s must contain the **present_translations** instance as in the languages attribute of this **item_names** instance.

WR2: if synonymous names are provided in more than one language, then only languages indicated in the **present_translations** instance in the "languages" attribute of this **item_names** instance can be used.

WR3: at most two synonymous shall be defined in the **languages** attribute, if it exists.

NOTE 1 The attributes **preferred_name**, **synonymous_names** and **short_name** are used to encode the "Preferred Name", "Synonymous Name" and "Short Name" attributes respectively for properties and classes (see 6.2 and 7.2).

NOTE 2 The attribute **languages** is used to define the sequence of translations (if requested for attributes).

D.3.8.2.6 Label_with_language

The **label_with_language** entity provides resources for associating a label to a language.

EXPRESS specification:

```
*)
ENTITY label_with_language;
     l: label;
     language: language_code;
END_ENTITY;
(*
```

Attribute definitions:

l: the label associated to a language.

language: the code of the labelled language.

D.3.8.2.7 Mathematical_string

The **mathematical_string** entity provides resources defining a representation for mathematical strings. It also allows a representation in the SGML format.

EXPRESS specification:

```
*)
ENTITY mathematical_string;
     text_representation: text;
     SGML_representation: OPTIONAL text;
END_ENTITY;
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Attribute definitions:

text_representation: "linear" form of a mathematical string, using ISO 843-2: "Transliteration of Greek characters into Latin characters", if necessary.

SGML_representation: SGML-Text (ISO 8879), marked up according to the Math DTD (document Type Definition) in ISO 12083: "Electronic Manuscript Preparation and Markup". The SGML text must be processed so that it will be treated as one single string during the exchange (see ISO 10303-21).

D.3.9 Function definitions

This subclause contains functions that are referenced in WHERE clauses to assert data consistency, or that provide resources for application development.

D.3.9.1 Acyclic_superclass_relationship function

The **acyclic_superclass_relationship** function checks that there is no cycle in the superclass relationship. By the cardinality of the **its_superclass** attribute in ENTITY class, it is ensured that there is an inheritance tree, no acyclic graph. Thus, this function merely has to check that no class instance refers in the **its_superclass** attribute to another one that is essentially a subclass.

EXPRESS specification:

```
*)
FUNCTION acyclic_superclass_relationship (
            current: class_BSU;
            visited: SET OF class) : LOGICAL;

IF SIZEOF (current.definition) = 1 THEN
     IF current.definition[1]\class IN visited THEN
        RETURN (FALSE);
     (* wrong: current declares a subclass as its superclass *)
     ELSE
        IF EXISTS
        (current.definition[1]\class.its_superclass)
        THEN
             RETURN (acyclic_superclass_relationship (
                  current.definition[1]\class.its_superclass,
                  visited + current.definition[1]\class));
        ELSE
             RETURN (TRUE);
        END_IF;
     END_IF;
ELSE
     RETURN (UNKNOWN);
END_IF;
END_FUNCTION; -- acyclic_superclass_relationship
(*
```

D.3.9.2 At_most_two_synonyms_per_language function

The **at_most_two_synonyms_per_language** function checks that there are at most two synonymous names in the **synonymous_names** parameter that correspond to each language of the **languages** parameter.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
FUNCTION at_most_two_synonyms_per_language (
       languages: present_translations;
       synonymous_names: SET OF syn_name_type): BOOLEAN;

IF EXISTS (languages) THEN
     REPEAT i := 1 TO SIZEOF (languages.language_codes);
       IF SIZEOF( QUERY ( s <* synonymous_names |
            s.language = languages.language_codes[i] )) > 2
       THEN
            RETURN (FALSE);
       END_IF;
     END_REPEAT;
     RETURN (TRUE);
ELSE
     RETURN (SIZEOF (synonymous_names) <= 2);
END_IF;
END_FUNCTION; -- at_most_two_synonyms_per_language
(*
```

D.3.9.3 Check_syn_length function

The **check_syn_length** function checks that the length of **s** doesn't exceed the length indicated by **s_length**.

EXPRESS specification:

```
*)
FUNCTION check_syn_length (
            s: syn_name_type;
            s_length: INTEGER) : BOOLEAN;

IF 'ISO13584_IEC61360_DICTIONARY_SCHEMA.LABEL_WITH_LANGUAGE'
       IN TYPEOF(s) THEN
     RETURN (LENGTH(s.l) <= s_length);
ELSE
     RETURN (LENGTH(s) <= s_length);
END_IF;
END_FUNCTION; -- check_syn_length
(*
```

D.3.9.4 Codes_are_unique function

The **codes_are_unique** function returns TRUE if the **value_code**s are unique within this list of **values**.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
FUNCTION codes_are_unique (values: LIST OF dic_value): BOOLEAN;

LOCAL
     l: SET OF STRING := [ ];
END_LOCAL;

REPEAT i := 1 TO SIZEOF (values);
     l := l + values[i].value_code;
     (* duplicates are eliminated. *)
END_REPEAT;

RETURN (SIZEOF(values) = SIZEOF(l));
END_FUNCTION; -- codes_are_unique
(*
```

D.3.9.5 Definition_available_implies function

The **definition_available_implies** function checks whether the definition corresponding to the **BSU** parameter exists or not. Then, if this definition exists, the **expression** parameter is returned.

EXPRESS specification:

```
*)
FUNCTION definition_available_implies (
           BSU: basic_semantic_unit;
           expression: LOGICAL): LOGICAL;

RETURN (NOT (SIZEOF(BSU.definition) = 1) OR expression);

END_FUNCTION; -- definition_available_implies
(*
```

D.3.9.6 Is_subclass function

The function **is_subclass** returns TRUE if **sub** is defined as a subclass of **super**.

EXPRESS specification:

```
*)
FUNCTION is_subclass ( sub, super: class): LOGICAL ;

IF (NOT EXISTS (sub)) OR (NOT EXISTS (super)) THEN
     RETURN (UNKNOWN);
END_IF;
IF NOT EXISTS (sub.its_superclass) THEN
     RETURN (FALSE);
     (* end of chain reached, didn't meet super so far *)
END_IF;
IF SIZEOF(sub.its_superclass.definition) = 1 THEN
     (* definition available *)
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
          IF (sub.its_superclass.definition[1]\class = super) THEN
             RETURN ( TRUE );
          ELSE
             RETURN   (is_subclass   (sub.its_superclass.definition[1]\class,
     super));
          END_IF;
     ELSE
          RETURN (UNKNOWN);
     END_IF;
     END_FUNCTION; -- is_subclass
     (*
```

D.3.9.7 String_for_derived_unit function

The function **string_for_derived_unit** returns a STRING representation of the **derived_unit** (according to ISO 10303-41) passed as parameter. First, the elements of the derived unit are separated according to the sign of the exponent. If there are elements of both kinds, the '/' notation is used to separate those with positive from those with negative sign. If there are only negative exponents, the u-e notation is used. A dot '.' (decimal code 46 according to ISO 8859-1, see ISO 10303-21) is used to separate individual elements.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
     *)
     FUNCTION string_for_derived_unit (u: derived_unit): STRING;

          FUNCTION string_for_derived_unit_element
             (u: derived_unit_element; neg_exp: BOOLEAN
             (* print negative exponents with power -1 *)): STRING;
             (* returns a STRING representation of the derived_unit_element
             (according to ISO 10303-41) passed as parameter *)

          LOCAL
             result: STRING;
          END_LOCAL;

          result := string_for_named_unit(u.unit);
          IF (u.exponent <> 0) THEN
             IF (u.exponent > 0) OR NOT neg_exp THEN
                  result := result + '**' + FORMAT (ABS(u.exponent), '1I');
             ELSE
                  result := result + '**' + FORMAT (u.exponent, '1I');
             END_IF;
          END_IF;
          RETURN (result);
     END_FUNCTION; -- string_for_derived_unit_element

     LOCAL
          pos, neg: SET OF derived_unit_element;
          us: STRING;
     END_LOCAL;

     (* separate unit elements according to the sign of the exponents: *)
```

```
pos := QUERY ( ue <* u.elements | ue.exponent > 0 );
neg := QUERY ( ue <* u.elements | ue.exponent < 0 );
us := '';
IF SIZEOF (pos) > 0 THEN
     (* there are unit elements with positive sign *)
     REPEAT i := LOINDEX (pos) TO HIINDEX (pos);
        us := us + string_for_derived_unit_element(pos[i], FALSE);
        IF i <> HIINDEX (pos)
        THEN
             us := us + '.';
        END_IF;
     END_REPEAT;
     IF SIZEOF (neg) > 0
     THEN
        (* there are unit elements with negative sign, use '/' notation:
*)
        us := us + '/';
        IF SIZEOF (neg) > 1 THEN us := us + '('; END_IF;
        REPEAT i := LOINDEX (neg) TO HIINDEX (neg);
             us := us + string_for_derived_unit_element(neg[i], FALSE);
             IF i <> HIINDEX (neg) THEN us := us + '.'; END_IF;
        END_REPEAT;
        IF SIZEOF (neg) > 1
        THEN
             us := us + ')';
        END_IF;
     END_IF;
ELSE
     (* only negative signs, use u-e notation *)
     IF SIZEOF(neg) > 0 THEN
        REPEAT i := LOINDEX (neg) TO HIINDEX (neg);
             us := us + string_for_derived_unit_element(neg[i],
             TRUE);
             IF i <> HIINDEX (neg)
             THEN
                  us := us + '.';
             END_IF;
        END_REPEAT;
     END_IF;
END_IF;
RETURN (us);
END_FUNCTION; -- string_for_derived_unit
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

D.3.9.8 String_for_named_unit function

The **string_for_named_unit** function returns a STRING representation of the **named_unit** (according to ISO 10303-41 and the extension in D.3.7.4.1) passed as parameter.

EXPRESS specification:

```
*)
FUNCTION string_for_named_unit (u: named_unit): STRING;

IF 'MEASURE_SCHEMA.SI_UNIT' IN TYPEOF(u) THEN
     RETURN (string_for_SI_unit (u\si_unit));
ELSE
     IF 'MEASURE_SCHEMA.CONTEXT_DEPENDENT_UNIT' IN TYPEOF(u)
     THEN
        RETURN (u\context_dependent_unit.name);
     ELSE
        IF 'MEASURE_SCHEMA.CONVERSION_BASED_UNIT' IN TYPEOF(u)
        THEN
             RETURN (u\conversion_based_unit.name);
        ELSE
             IF 'ISO13584_IEC61360_DICTIONARY_SCHEMA'
                                      +'.NON_SI_UNIT' IN TYPEOF(u)
             THEN
                  RETURN (u\non_si_unit.name);
             ELSE
                  (* pure named_unit instance, not subtyped further. *)
                  RETURN ('name_unknown');
             END_IF;
        END_IF;
     END_IF;
END_IF;
END_FUNCTION; -- string_for_named_unit
(*
```

D.3.9.9 String_for_SI_unit function

The **string_for_SI_unit** function returns a STRING representation of the **si_unit** (according to ISO 10303-41) passed as parameter.

EXPRESS specification:

```
*)
FUNCTION string_for_SI_unit (unit: si_unit): STRING;

LOCAL
     prefix_string, unit_string: STRING;
END_LOCAL;

IF EXISTS (unit.prefix) THEN
     CASE unit.prefix OF
          exa      : prefix_string := 'E';
          peta     : prefix_string := 'P';
          tera     : prefix_string := 'T';
          giga     : prefix_string := 'G';
          mega     : prefix_string := 'M';
          kilo     : prefix_string := 'k';
          hecto    : prefix_string := 'h';
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
            deca    : prefix_string := 'da';
            deci    : prefix_string := 'd';
            centi   : prefix_string := 'c';
            milli   : prefix_string := 'm';
            micro   : prefix_string := 'u';
            nano    : prefix_string := 'n';
            pico    : prefix_string := 'p';
            femto   : prefix_string := 'f';
            atto    : prefix_string := 'a';
        END_CASE;
    ELSE
        prefix_string := '';
    END_IF;
    CASE unit.name OF
            metre          : unit_string:= 'm';
            gram           : unit_string := 'g';
            second         : unit_string := 's';
            ampere         : unit_string := 'A';
            kelvin         : unit_string := 'K';
            mole           : unit_string := 'mol';
            candela        : unit_string := 'cd';
            radian         : unit_string := 'rad';
            steradian      : unit_string := 'sr';
            hertz          : unit_string := 'Hz';
            newton         : unit_string := 'N';
            pascal         : unit_string := 'Pa';
            joule          : unit_string := 'J';
            watt           : unit_string := 'W';
            coulomb        : unit_string := 'C';
            volt           : unit_string := 'V';
            farad          : unit_string := 'F';
            ohm            : unit_string := 'Ohm';
            siemens        : unit_string := 'S';
            weber          : unit_string := 'Wb';
            tesla          : unit_string := 'T';
            henry          : unit_string := 'H';
            degree_Celsius : unit_string := 'Cel';
            lumen          : unit_string := 'lm';
            lux            : unit_string := 'lx';
            becquerel      : unit_string := 'Bq';
            gray           : unit_string := 'Gy';
            sievert        : unit_string := 'Sv';
    END_CASE;
    RETURN (prefix_string + unit_string);
    END_FUNCTION; -- string_for_SI_unit
    (*
```

D.3.9.10 String_for_unit function

The **string_for_unit** function returns a STRING representation of the **unit** (according to ISO 10303-41) passed as parameter.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
FUNCTION string_for_unit (u: unit): STRING;
     IF 'MEASURE_SCHEMA.DERIVED_UNIT' IN TYPEOF(u)
     THEN
        RETURN (string_for_derived_unit(u));
     ELSE (* 'MEASURE_SCHEMA.NAMED_UNIT' IN TYPEOF(u) holds true *)
        RETURN (string_for_named_unit(u));
     END_IF;
END_FUNCTION; -- string_for_unit
(*
```

D.3.9.11 All_class_descriptions_reachable function

The **all_class_descriptions_reachable** function checks if the **dictionary_elements** that describe a class, referred by a **class_BSU**, and all its super-classes, can be computed in the inheritance tree defined by the class hierarchy.

EXPRESS specification:

```
*)
FUNCTION all_class_descriptions_reachable (cl:class_BSU): BOOLEAN;

IF SIZEOF(cl.definition) = 0
THEN
     RETURN(FALSE);
END_IF;

IF NOT (EXISTS(cl.definition[1]\class.its_superclass))
THEN
     RETURN (TRUE);
ELSE
     RETURN(all_class_descriptions_reachable(

     cl.definition[1]\class.its_superclass));
END_IF;

END_FUNCTION; -- all_class_descriptions_reachable
(*
```

D.3.9.12 Compute_known_visible_properties function

The **compute_known_visible_properties** function computes the set of properties that are visible for a given class. When a definition is not available, it returns only the visible properties that may be computed.

NOTE When some **class dictionary_definition** is not present in the same exchange context (a PLib exchange context is never assumed to be complete), the super-class of a class may not be known. Therefore the properties defined as visible by this super-class cannot be computed by the **compute_known_visible_properties** function. Only on the receiving sytem all the **dictionary_definitions** of the **BSUs** are required to be available. Therefore, on the receiving sytem, the **compute_known_visible_properties** function computes all the properties that are visible to a class by virtue of referencing it (or any of its superclass) by their **name_scope** attribute.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
FUNCTION compute_known_visible_properties (cl: class_BSU):
                                        SET OF property_BSU;

LOCAL
     s: SET OF property_BSU :=[ ];
END_LOCAL;

s:= USEDIN(cl,
        'ISO13584_IEC61360_DICTIONARY_SCHEMA.PROPERTY_BSU.NAME_SCOPE');
IF SIZEOF(cl.definition)=0
THEN
     RETURN(s);
ELSE
     IF EXISTS (cl.definition[1]\class.its_superclass) THEN
            s := s + compute_known_visible_properties(
                    cl.definition[1]\class.its_superclass);
     END_IF;

     RETURN(s);
END_IF;
END_FUNCTION;
(*
```

D.3.9.13 Compute_known_visible_data_types function

The **compute_known_visible_data_types** function computes the set of data_types that are visible for a given class. When a definition is not available, it returns only the visible **data_type**s that may be computed.

NOTE When some **class dictionary_definition** is not present in the same exchange context (a PLib exchange context is never assumed to be complete), the super-class of a class may not be known. Therefore the **data_type**s defined as visible by this super-class cannot be computed by the **compute_known_visible_data_types** function. Only on the receiving sytem all the **dictionary_definition**s of the **BSU**s are required to be available. Therefore, on the receiving sytem, the **compute_known_visible_data_types** function computes all the data_types that are visible to a class by virtue of referencing it (or any of its superclass) by their **name_scope** attribute.

EXPRESS specification:

```
*)
FUNCTION compute_known_visible_data_types (cl: class_BSU):
                                        SET OF data_type_BSU;

LOCAL
     s: SET OF data_type_BSU :=[ ];
END_LOCAL;

s:= USEDIN(cl,
        'ISO13584_IEC61360_DICTIONARY_SCHEMA.DATA_TYPE_BSU.NAME_SCOPE');
IF SIZEOF(cl.definition)=0
THEN
     RETURN(s);
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
ELSE
     IF EXISTS (cl.definition[1]\class.its_superclass) THEN
             s := s + compute_known_visible_data_types(
                     cl.definition[1]\class.its_superclass);
     END_IF;
     RETURN(s);
END_IF;
END_FUNCTION;
(*
```

D.3.9.14 Compute_known_applicable_properties function

The **compute_known_applicable_properties** function computes the set of properties that are applicable for a given class. When a definition is not available, it returns only the applicable properties that may be computed.

NOTE When some class **dictionary_definition** is not present in the same exchange context (a PLib exchange context is never assumed to be complete), the super-class of a class may not be known. Therefore the properties defined as applicable by this super-class cannot be computed by the **compute_known_applicable_properties** function. Only on the receiving sytem all the **dictionary_definition**s of the **BSU**s are required to be availabe. Therefore, on the receiving sytem, the **compute_known_applicable_properties** function computes all the properties that are applicable to a class by virtue of being referenced by a **described_by** attribute.

EXPRESS specification:

```
*)
FUNCTION compute_known_applicable_properties (cl: class_BSU):
                                                SET OF property_BSU;

LOCAL
     s: SET OF property_BSU :=[ ];
END_LOCAL;

IF SIZEOF(cl.definition)=0
THEN
     RETURN(s);
ELSE
     REPEAT i:=1 TO SIZEOF(cl.definition[1]\class.described_by);
        s := s + cl.definition[1]\class.described_by[i];
     END_REPEAT;
     IF EXISTS (cl.definition[1]\class.its_superclass) THEN
        s := s + compute_known_applicable_properties(
                    cl.definition[1]\class.its_superclass);
     END_IF;

     RETURN(s);
END_IF;
END_FUNCTION;
(*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

D.3.9.15 Compute_known_applicable_data_types function

The **compute_known_applicable_data_types** function computes the set of data_types that are applicable for a given class. When a definition is not available, it returns only the applicable **data_type**s that may be computed.

NOTE When some class **dictionary_definition** is not present in the same exchange context (a PLib exchange context is never assumed to be complete), the super-class of a class may not be known. Therefore the **data_type**s defined as applicable by this super-class cannot be computed by the **compute_known_applicable_data_types** function. Only on the receiving sytem all the **dictionary_definition**s of the **BSU**s are required to be available. Therefore, on the receiving sytem, the **compute_known_applicable_data_types** function computes all the **data_type**s that are applicable to a class by virtue of being referenced by a **defined_types** attribute.

<u>EXPRESS specification:</u>

```
*)
FUNCTION compute_known_applicable_data_types (cl: class_BSU):
                                                SET OF data_type_BSU;

LOCAL
     s: SET OF data_type_BSU :=[ ];
END_LOCAL;

IF SIZEOF(cl.definition)=0
THEN
     RETURN(s);
ELSE
     REPEAT i:=1 TO SIZEOF(cl.definition[1]\class.described_by);
        s:=s+cl.definition[1]\class.defined_types[i];
     END_REPEAT;
     IF EXISTS (cl.definition[1]\class.its_superclass) THEN
        s:=s+compute_known_applicable_data_types(
                        cl.definition[1]\class.its_superclass);
     END_IF;

     RETURN(s);
END_IF;
END_FUNCTION;
(*
```

D.3.9.16 List_to_set

The **list_to_set** function creates a SET from a LIST named **l**, the type of element for the SET will be the same as that in the original LIST.

<u>EXPRESS specification:</u>

```
*)
FUNCTION list_to_set(l: LIST [0:?] OF GENERIC:type_elem)
                                       : SET OF GENERIC: type_elem;

LOCAL
     s: SET OF GENERIC: type_elem := [];
END_LOCAL;
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
    REPEAT i := 1 TO SIZEOF(l);
        s := s + l[i];
    END_REPEAT;

    RETURN(s);
    END_FUNCTION; -- list_to_set

    END_SCHEMA; -- ISO13584_IEC61360_dictionary_schema
    (*
```

D.4 ISO13584_IEC61360_language_resource_schema

The following schema provides resources for permitting strings in various languages. It has been extracted from the Dictionary schema, since it could be used in other schemata. It is largely based on the **support_resource_schema** from ISO 10303-41: STEP part 41: "Fundamentals of Product Description and Support", and can be seen as an extension to that. It allows for the usage of one specific language throughout an exchange context (Physical File) without the overhead introduced when multiple languages are used. See figure D.12 - EXPRESS-G diagram of **ISO13584_IEC61360_language_resource_schema** and **support_resource_schema**, for a graphical depiction.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

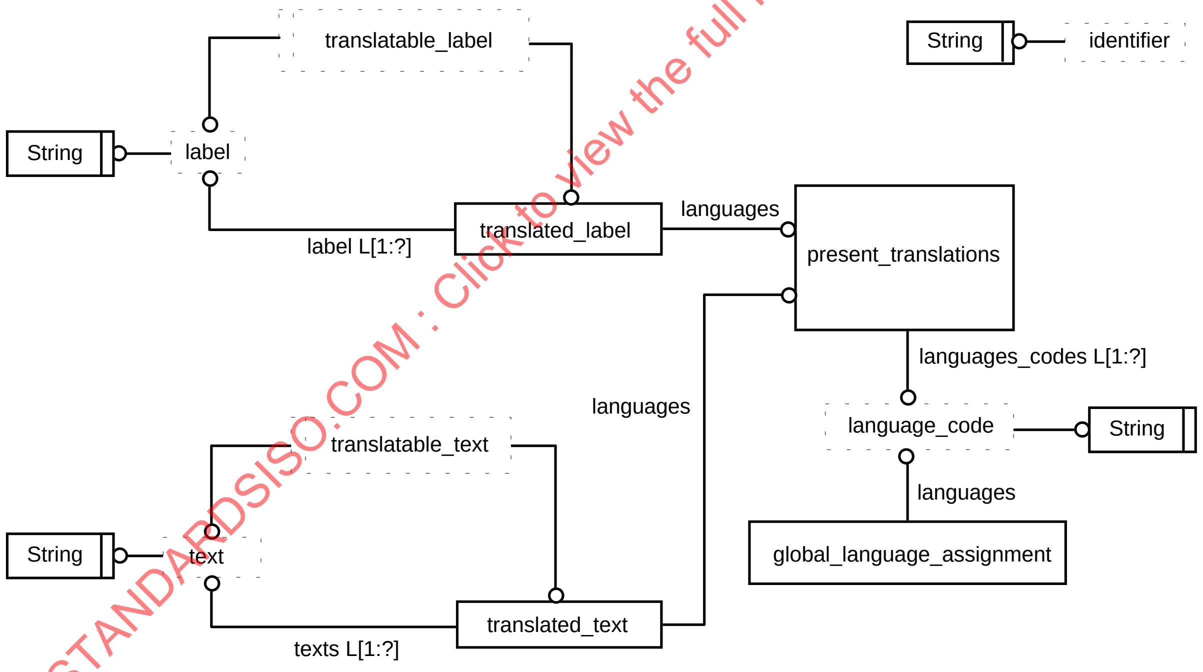


Figure D.12 — ISO13584_IEC61360_language_resource_schema and support_resource_schema

EXPRESS specification:

```
    *)
    SCHEMA ISO13584_IEC61360_language_resource_schema;

    REFERENCE FROM support_resource_schema (identifier, label, text);
```

```
         (* from ISO 10303-41: STEP part 41: "Fundamentals of Product
    Description and Support" *)
    (*
```

D.4.1 ISO13584_IEC61360_language_resource_schema type and entity definitions

This subclause contains the EXPRESS type and entity definitions in the **ISO13584_IEC61360_language_resource_schema.**

D.4.1.1 Language_code

The **language_code** type enables to identify a langage according to ISO 639. Values are e.g. "EN" for English in general, "FR" for French, "RU" for Russian, "DE" for German, "en GB" for UK English, "en US" for US English etc...

EXPRESS specification:

```
    *)
    TYPE language_code = identifier;
    END_TYPE;
    (*
```

D.4.1.2 Global_language_assignment

The **global_language_assignment** entity specifies the language for **translatable_label** and **translatable_text**, if **label** and **text** are selected, respectively (i.e. without explicit language indication as is done in **translated_label** and **translated_text**).

EXPRESS specification:

```
    *)
    ENTITY global_language_assignment;
         language: language_code;
    END_ENTITY;
    (*
```

Attribute definitions:

language: the code of the assigned language.

D.4.1.3 Present_translations

The **present_translations** entity serves to indicate the languages used for **translated_label** and **translated_text**.

EXPRESS specification:

```
    *)
    ENTITY present_translations;
         language_codes: LIST [ 1 : ? ] OF UNIQUE language_code;
    UNIQUE
         UR1: language_codes;
    END_ENTITY;
    (*
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

Attribute definitions:

language_codes: the list of unique language codes corresponding to the language in which a translation is made.

Formal proposition:

UR1: for each list of **language_code** there shall be a unique instance of **present_translations**

D.4.1.4 Translatable_label

A **translatable_label** defines a type of values that can be **label**s or **translated_label**s.

EXPRESS specification:

```
*)
TYPE translatable_label = SELECT (label, translated_label);
END_TYPE;
(*
```

D.4.1.5 Translated_label

The **translated_label** entity defines the labels that are translated and the corresponding languages of translation.

EXPRESS specification:

```
*)
ENTITY translated_label;
    labels: LIST [ 1 : ? ] OF label;
    languages: present_translations;
WHERE
    WR1: SIZEOF (labels ) = SIZEOF (languages.language_codes);
END_ENTITY;
(*
```

Attribute definitions:

labels: the list of **label**s that are translated.

languages: the list of **languages** in which each label is translated.

Formal propositions:

WR1: the number of **label**s contained in the **labels** list shall be equal to the number of languages provided in the **languages.language_codes** attribute.

Informal propositions:

The content of **labels [i]** is in the language identified by **languages.language_codes [i]**.

D.4.1.6 Translatable_text

A **translatable_text** defines a type of values that can be **text**s or **translated_ text**s.

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

EXPRESS specification:

```
*)
TYPE translatable_text = SELECT ( text, translated_text);
END_TYPE;
(*
```

D.4.1.7 Translated_text

The **translated_text** entity defines the **text**s that are translated and the corresponding **language**s of translation.

EXPRESS specification:

```
*)
ENTITY translated_text;
     texts: LIST [ 1 : ? ] OF text;
     languages: present_translations;
WHERE
     WR1: SIZEOF (texts ) = SIZEOF (languages.language_codes);
END_ENTITY;
(*
```

Attribute definitions:

texts: the list of **text**s that are translated.

languages: the list of languages in which each text is translated.

Formal propositions:

WR1: the number of **text**s contained in the **texts** list shall be equal to the number of languages provided in the **languages.language_codes** attribute.

Informal propositions:

IP1: the content of **texts** [i] is in the language identified by **languages.language_codes** [i].

D.4.2 ISO13584_IEC61360_language_resource_schema function definitions

This subclause contains functions that are referenced in WHERE clauses to assert data consistency.

D.4.2.1 Check_label_length function

The **check_label_length** function checks that no label in **l** exceeds the length indicated by **l_length**.

EXPRESS specification:

```
*)
FUNCTION check_label_length (
            l: translatable_label;
            l_length: INTEGER) : BOOLEAN;

IF 'ISO13584_IEC61360_LANGUAGE_RESOURCE_SCHEMA.TRANSLATED_LABEL'
     IN TYPEOF(l)
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
    THEN
         REPEAT i :=1 TO SIZEOF (l.labels);
            IF LENGTH(l.labels[i]) > l_length
            THEN
                 RETURN (FALSE);
            END_IF;
         END_REPEAT;
         RETURN (TRUE);
    ELSE (* the argument l is a single string *)
         RETURN (LENGTH(l) <= l_length);
    END_IF;
    END_FUNCTION; -- check_label_length
    (*
```

D.4.2.2 Check_text_length function

The **check_text_length** function checks that no text in **t** exceeds the length indicated by **t_length**.

EXPRESS specification:

```
    *)
    FUNCTION check_text_length (
                 t: translatable_text;
                 t_length: INTEGER) : BOOLEAN;

    IF 'ISO13584_IEC61360_LANGUAGE_RESOURCE_SCHEMA.TRANSLATED_TEXT'
         IN TYPEOF(t)
    THEN
         REPEAT i :=1 TO SIZEOF (t.texts);
            IF LENGTH(t.texts[i]) > t_length
            THEN
                 RETURN (FALSE);
            END_IF;
         END_REPEAT;
         RETURN (TRUE);
    ELSE (* the argument t is a single string *)
         RETURN (LENGTH(t) <= t_length);
    END_IF;
    END_FUNCTION; -- check_text_length
    (*
```

D.4.3 ISO13584_IEC61360_language_resource_schema rule definition

The rule **single_language_assignment** asserts that only one language may be assigned to be used in **translatable_label** and **translatable_text**, resp.

EXPRESS specification:

```
    *)
    RULE single_language_assignment FOR (global_language_assignment);
    WHERE
         SIZEOF ( global_language_assignment ) <= 1;
    END_RULE;
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
    END_SCHEMA; -- ISO13584_IEC61360_language_resource_schema
    (*
```

D.5 Example Physical File

D.5.1 Some example data

This chapter gives some fragments of a Physical File for exchanging the data of the IEC 61360-4 standard: "International Reference List of Standard Data Element Types for Electric Components". It is intended to show the use of the EXPRESS model in clause D.3 "**ISO13584_IEC61360_dictionary_schema**" together with ISO 10303-21 to exchange corresponding data.

```
ISO-10303-21;
HEADER;
...
ENDSEC;
DATA;
```

D.5.1.1 Supplier data

```
#1=SUPPLIER_BSU('01122//61360-4', *, * );  /* according to ISO 13584-26 */
#2=SUPPLIER_ELEMENT(#1, #3, '01', *, #4, #5);
#3=DATES('1994-09-16', '1994-09-16', $);
#4=ORGANIZATION('IEC', 'IEC Maintenance Agency', 'The IEC Maintenance
Agency as described in IEC 61360-3: "Maintenance and Validation
Procedures"');
#5=ADDRESS('to be determined', $, $, $, $, $, $, $, $, $, $, $);
#10=SUPPLIER_BSU('01123//-00', *, * );     /* ISO/IEC ICS */
```

D.5.1.2 Root class data

The AAA000 IEC root class provides a name scope corresponding to the whole future IEC 61360-4 standard. It covers two trees, one for materials, one for components, therefore the class is defined as an **item_class**. It is a subtype of ICS root.

```
#90=CLASS_BSU('OO', '001', *, #10);   /* ICS root */
#100=CLASS_BSU('AAA000', '001', *, #1);
#101=ITEM_CLASS(#100, #3, '01', #102, 'IEC root class that provides a name
scope corresponding to the whole IEC 61360-4 standard. It covers two trees,
one for materials, one for components', $, $, $, *, #90, (#110), (), $,
(#110), (), $);
#102=ITEM_NAMES('IEC root class', $, 'IEC root', $, $);
#110=PROPERTY_BSU('AAE000', '001', *, #100);
#111=NON_DEPENDENT_P_DET(#110, #3, '01', #112, 'the type of tree: material
or component', $, $, $, *, $, (), $, $, #113, $);
#112=ITEM_NAMES('type of tree', $, 'tree type', $, $);
#113=NON_QUANTITATIVE_CODE_TYPE('A..8', #114);
#114=VALUE_DOMAIN((#120, #122), $, $, ());
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
#120=DIC_VALUE('MATERIAL', #121, $);
#121=ITEM_NAMES('material tree', $, 'mat tree', $, $);
#122=DIC_VALUE('COMPONS', #123, $);
#123=ITEM_NAMES('component tree', $, 'comp tree', $, $);
```

D.5.1.3 Material data

```
#200=CLASS_BSU('AAA218', '001', *, #1);
#201=MATERIAL_CLASS(#200, #3, '01', #202, 'root class of the materials
tree', $, $, $, *, #100, (#210, #230), (), $, (#210), (#205), 'MATERIAL');
#202=ITEM_NAMES('materials root class', $, 'materials root', $, $);
#205=CLASS_VALUE_ASSIGNMENT(#110, 'MATERIAL');
#210=PROPERTY_BSU('AAF311', '005', *, #100);
#211=NON_DEPENDENT_P_DET(#210, #3, '01', #212, 'code of the type of
material', $, $, $, *, $, $, $, 'A57', #213, $);
#212=ITEM_NAMES('material type', $, 'material type', $, $);
#213=NON_QUANTITATIVE_CODE_TYPE('M..3', #214);
#214=VALUE_DOMAIN((#220, #222, #224, #226), $, $, ());
#220=DIC_VALUE('ACO', #221, $);
#221=ITEM_NAMES('acoustical', $, 'acoustical', $, $);
#222=DIC_VALUE('MG', #223, $);
#223=ITEM_NAMES('magnetic', $, 'magnetical', $, $);
#224=DIC_VALUE('OP', #225, $);
#225=ITEM_NAMES('optical', $, 'optical', $, $);
#226=DIC_VALUE('TH', #227, $);
#227=ITEM_NAMES('thermal-electric', $, 'th-electric', $, $);
#230=PROPERTY_BSU('AAF286', '005', *, #100);
#231=NON_DEPENDENT_P_DET(#230, #3, '01', #232, 'The nominal density (in
kg/m**3) of a material.', $, $, $, *, #233, (), $, 'K02', #234, $);
#232=ITEM_NAMES('density', $, 'density', $, $);
#233=MATHEMATICAL_STRING('$r_d', '&rho;<sub>d</sub>');
#234=REAL_MEASURE_TYPE('NR3..3.3ES2', #235);
#235=DIC_UNIT(#236, $);
#236=DERIVED_UNIT((#237, #239));
#237=DERIVED_UNIT_ELEMENT(#238, $);
#238=SI_UNIT( *, .KILO., .GRAM.);
#239=DERIVED_UNIT_ELEMENT(#240, -3.0);
#240=SI_UNIT( *, $, .METRE.);
```

D.5.1.4 Component data

```
#300=CLASS_BSU('EEE000', '001', *, #1);
#301=COMPONENT_CLASS(#300, #3, '01', #302, 'root class of the components
tree', $, $, $, *, #100, (#310, #330, #350), (), $, (#310), (#305),
'COMPONS');
#302=ITEM_NAMES('components root class', $, 'components root', $, $);
#305=CLASS_VALUE_ASSIGNMENT(#110, 'COMPONS');
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998

```
#310=PROPERTY_BSU(’AAE001’, ’005’, *, #100);
#311=NON_DEPENDENT_P_DET(#310, #3, ’01’, #312, 'Code of the main functional
class to which a component belongs', $, $, $, *, $, (), $, 'A52', #313, $);
#312=ITEM_NAMES('main class of component', $, 'main class', $, $);
#313=NON_QUANTITATIVE_CODE_TYPE(’M..3’, #314);
#314=VALUE_DOMAIN((#320, #322, #324, #326), $, $, ());
#320=DIC_VALUE(’EE’, #321, $);
#321=ITEM_NAMES(’EE (electric / electronic)’, $, ’EE’, $, $);
#322=DIC_VALUE(’EM’, #323, $);
#323=ITEM_NAMES(’electromechanical’, $, ’electromech’, $, $);
#324=DIC_VALUE(’ME’, #325, $);
#325=ITEM_NAMES(’mechanical’, $, ’mechanical’, $, $);
#326=DIC_VALUE(’MP’, #327, $);
#327=ITEM_NAMES(’magnetic part’, $, ’magnetic’, $, $);
#330=PROPERTY_BSU(’AAF267’, ’005’, *, #100);
#331=NON_DEPENDENT_P_DET(#330, #3, ’01’, #332, 'The nominal distance (in m)
between the inside of the two tapes used for taped products with axial
leads', $, $, $, *, #333, (), $, 'T03', #334, $);
#332=ITEM_NAMES('inner tape spacing', $, 'inner tape spac', $, $);
#333=MATHEMATICAL_STRING(’b_tape’, ’b<sub>tape</sub>’);
#334=LEVEL_TYPE((.NOM.), #335);
#335=REAL_MEASURE_TYPE('NR3..3.3ES2', #336);
#336=DIC_UNIT(#337, $);
#337=SI_UNIT( *, $, .METRE.);
#350=PROPERTY_BSU(’AAE022’, ’005’, *, #100);
#351=NON_DEPENDENT_P_DET(#350, #3, ’01’, #352, 'The value as specified by
level (miNoMax) of the outside diameter (in m) of a component with a body
of circular cross-section', $, $, $, *, #353, $, $, 'T03', #354, $);
#352=ITEM_NAMES('outside diameter', $, 'outside diam', $, $);
#353=MATHEMATICAL_STRING(’d_out’, ’d<sub>out</sub>’);
#354=LEVEL_TYPE((.MIN.,.NOM.,.MAX.), #355);
#355=REAL_MEASURE_TYPE('NR3..3.3ES2', #356);
#356=DIC_UNIT(#357, ’m’);
#357=SI_UNIT( *, $, .METRE.);
```

D.5.1.5 Electric / Electronic component data

```
#400=CLASS_BSU(’EEE001’, ’001’, *, #1);
#401=COMPONENT_CLASS(#400, #3, ’01’, #402, ’electric / electronic
components’, $, $, $, *, #300, (#410, #470), (), $, (#410), (#405), ’EE’);
#402=ITEM_NAMES(’EE components’, $, ’EE components’, $, $);
#405=CLASS_VALUE_ASSIGNMENT(#310, ’EE’);
#410=PROPERTY_BSU(’AAE002’, ’005’, *, #100);
#411=NON_DEPENDENT_P_DET(#410, #3, ’01’, #412, 'Code of the category to
which an electric/electronic component belongs.', $, $, $, *, $, (), $,
'A52', #413, $);
#412=ITEM_NAMES('category EE component', $, 'categ EE comp', $, $);
```

STANDARDSISO.COM : Click to view the full PDF of ISO 13584-42:1998