

BS EN 61360-2:2013



BSI Standards Publication

Standard data element types with associated classification scheme for electric components

Part 2: EXPRESS dictionary schema

bsi.

...making excellence a habit.
AVA Documents™
Document Sharing Hub

National foreword

This British Standard is the UK implementation of EN 61360-2:2013. It is identical to IEC 61360-2:2012. It supersedes BS EN 61360-2:2002 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee GEL/3, Documentation and graphical symbols.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2013.

Published by BSI Standards Limited 2013.

ISBN 978 0 580 61285 5

ICS 31.020

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 28 February 2013.

Amendments issued since publication

Date	Text affected
------	---------------

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 61360-2

January 2013

ICS 31.020

Supersedes EN 61360-2:2002 + A1:2004

English version

**Standard data element types with associated classification scheme for
electric components -
Part 2: EXPRESS dictionary schema
(IEC 61360-2:2012)**

Types normalisés d'éléments de données
avec plan de classification pour
composants électriques -
Partie 2: Schéma d'un dictionnaire
EXPRESS
(CEI 61360-2:2012)

Genormte Datenelementtypen mit
Klassifikationsschema für elektrische
Bauteile -
Teil 2: EXPRESS-Datenmodell
(IEC 61360-2:2012)

This European Standard was approved by CENELEC on 2012-11-06. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

© 2013 CENELEC - All rights of exploitation in any form and by any means reserved worldwide for CENELEC members.

Ref. No. EN 61360-2:2013 E

AVA Documents
Document Sharing Hub

Foreword

The text of document 3D/196/FDIS, future edition 3 of IEC 61360-2, prepared by IEC/SC 3D "Product properties and classes and their identification", of IEC/TC 3 "Information structures, documentation and graphical symbols" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61360-2:2013.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2013-08-06
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2015-11-06

This document supersedes EN 61360-2:2002 + A1:2004.

EN 61360-2:2012 includes the following significant technical changes with respect to EN 61360-2:2002:

- separation of concepts between characterization class and categorization class;
- introduction of value constraints on classes and properties;
- addition of various new subtypes for data types, including `rational_type`;
- improvement on the representation of unit of measurement.

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

Endorsement notice

The text of the International Standard IEC 61360-2:2012 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC 60027 series	NOTE	Harmonized in EN 60027 series.
IEC 61360-5	NOTE	Harmonized as EN 61360-5.
ISO 80000 series	NOTE	Harmonized in EN ISO 80000 series.

Annex ZA

(normative)

Normative references to international publications with their corresponding European publications

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

NOTE When an international publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 61360-1	2009	Standard data elements types with associated classification scheme for electric items - Part 1: Definitions - Principles and methods	EN 61360-1	2010
IEC 61360-4	Data-base	Standard data element types with associated classification scheme for electric components - Part 4: IEC reference collection of standard data element types and component classes	-	-
ISO/IEC 8859-1	1998	Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No.1	-	-
ISO/IEC 10646-1	-	Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane	-	-
ISO/IEC 14977	-	Information technology - Syntactic metalanguage - Extended BNF	-	-
ISO 639	Series	Codes for the representation of names of languages	-	-
ISO 843	1997	Information and documentation - Conversion of Greek characters into Latin characters	-	-
ISO 3166-1	-	Codes for the representation of names of countries and their subdivisions - Part 1: Country codes	EN ISO 3166-1	-
ISO 4217	2008	Codes for the representation of currencies and funds	-	-
ISO 8601	2004	Data elements and interchange formats - Information interchange - Representation of dates and times	-	-
ISO 10303-11	2004	Industrial automation systems and integration - - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual	-	-
ISO 10303-21	2002	Industrial automation systems and integration - - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure	-	-

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
ISO 10303-41	2000	Industrial automation systems and integration - - Product data representation and exchange - Part 41: Integrated generic resource: Fundamentals of product description and support	-	-
ISO 13584-26	2000	Industrial automation systems and integration - - Parts library - Part 26: Logical resource: Information supplier identification	-	-
ISO 13584-42	2010	Industrial automation systems and integration - - Parts library - Part 42: Description methodology: Methodology for structuring parts families	-	-

CONTENTS

INTRODUCTION	8
1 Scope	9
2 Normative references	9
3 Terms and definitions	10
4 Overview of the common dictionary schema and compatibility with ISO13584_IEC61360_dictionary_schema	19
4.1 General	19
4.2 Use of the common dictionary schema to exchange IEC 61360-1 compliant data	19
4.3 Compatibility with ISO 13584-42	20
4.4 Naming correspondence between IEC 61360-1 and IEC 61360-2	20
4.5 Main structure of the common dictionary schema	21
5 ISO13584_IEC61360_dictionary_schema	22
5.1 General	22
5.2 Dictionary schema	22
5.3 References to other schemata	22
5.4 Constant definitions	23
5.5 Identification of a dictionary	23
5.6 Basic Semantic Units: defining and using the dictionary	24
5.6.1 Requirements for exchange	24
5.6.2 Three levels architecture of the dictionary data	25
5.6.3 Overview of basic semantic units and dictionary elements	29
5.6.4 Identification of dictionary elements: three levels structure	30
5.6.5 Extension possibilities for other types of data	30
5.7 Supplier data	32
5.7.1 General	32
5.7.2 Supplier_BSU	32
5.7.3 Supplier_element	33
5.8 Class data	33
5.8.1 General	33
5.8.2 Structural detail	35
5.8.3 Item_class	41
5.8.4 Categorization_class	42
5.9 Data element type / properties data	44
5.9.1 General	44
5.9.2 Property_BSU	44
5.9.3 Property_DET	45
5.9.4 Condition, dependent and non-dependent Data Element Types	47
5.9.5 Structural detail	48
5.9.6 Class_value_assignment	49
5.10 Domain data: the type system	50
5.10.1 General	50
5.10.2 Structural detail	50
5.10.3 The type system	52
5.10.4 Values	69

5.10.5	Structural detail	69
5.10.6	Extension to ISO 10303-41 unit definitions	74
5.11	Basic type and entity definitions	75
5.11.1	Basic type definitions.....	75
5.11.2	Structural detail	75
5.11.3	Basic entity definitions.....	85
5.12	Function definitions	89
5.12.1	General	89
5.12.2	Acyclic_superclass_relationship function	89
5.12.3	Check_syn_length function.....	90
5.12.4	Codes_are_unique function	90
5.12.5	Definition_available_implies function	91
5.12.6	Is_subclass function	91
5.12.7	String_for_derived_unit function	92
5.12.8	String_for_named_unit function	94
5.12.9	String_for_SI_unit function	94
5.12.10	String_for_unit function	96
5.12.11	All_class_descriptions_reachable function.....	96
5.12.12	Compute_known_visible_properties function	97
5.12.13	Compute_known_visible_data_types function	97
5.12.14	Compute_known_applicable_properties function	98
5.12.15	Compute_known_applicable_data_types function	99
5.12.16	List_to_set function	100
5.12.17	Check_properties_applicability function	100
5.12.18	Check_datatypes_applicability function	101
5.12.19	One_language_per_translation function.....	102
5.12.20	Allowed_values_integer_types function	102
5.12.21	Is_class_valued_property function.....	103
5.12.22	Class_value_assigned function.....	103
6	ISO13584_IEC61360_language_resource_schema	104
6.1	Overview	104
6.2	ISO13584_IEC61360_language_resource_schema type and entity definitions.....	105
6.2.1	general.....	105
6.2.2	Language_code	105
6.2.3	Global_language_assignment.....	106
6.2.4	Present_translations	106
6.2.5	Translatable_label	107
6.2.6	Translated_label.....	107
6.2.7	Translatable_text.....	107
6.2.8	Translated_text.....	108
6.3	ISO13584_IEC61360_language_resource_schema function definitions	108
6.3.1	General	108
6.3.2	Check_label_length function.....	108
6.4	ISO13584_IEC61360_language_resource_schema rule definition	109
7	ISO13584_IEC61360_class_constraint_schema	109
7.1	General.....	109
7.2	Introduction to the ISO13584_IEC61360_class_constraint_schema.....	110
7.3	ISO13584_IEC61360_class_constraint_schema entity definitions.....	111
7.3.1	General	111

7.3.2	Constraint.....	111
7.3.3	Property_constraint	112
7.3.4	Class_constraint.....	112
7.3.5	Configuration_control_constraint	112
7.3.6	Filter.....	113
7.3.7	Integrity_constraint.....	114
7.3.8	Context_restriction_constraint	115
7.3.9	Domain_constraint.....	115
7.3.10	Subclass_constraint	116
7.3.11	Entity_subtype_constraint.....	116
7.3.12	Enumeration_constraint.....	116
7.3.13	Range_constraint	118
7.3.14	String_size_constraint	119
7.3.15	String_pattern_constraint	119
7.3.16	Cardinality_constraint.....	120
7.4	ISO13584_IEC61360_class_constraint_schema type definitions	121
7.4.1	General	121
7.4.2	Constraint_or_constraint_id.....	121
7.5	ISO13584_IEC61360_class_constraint_schema function definition	121
7.5.1	General	121
7.5.2	Integer_values_in_range function	121
7.5.3	Correct_precondition function	122
7.5.4	Correct_constraint_type function	122
7.5.5	Compatible_data_type_and_value function.....	125
7.6	ISO13584_IEC61360_class_constraint_schema rule definition.....	129
7.6.1	General	129
7.6.2	Unique_constraint_id.....	129
8	ISO13584_IEC61360_item_class_case_of_schema	129
8.1	Overview	129
8.2	Introduction to the ISO13584_IEC61360_item_class_case_of_schema	130
8.3	ISO13584_IEC61360_item_class_case_of_schema entity definitions	130
8.3.1	A priori semantic relationship.....	130
8.3.2	Item_class_case_of.....	133
8.4	ISO13584_IEC61360_item_class_case_of_schema function definitions	135
8.4.1	General	135
8.4.2	Compute_known_property_constraints function	135
8.4.3	Compute_known_referenced_property_constraints function	136
8.4.4	Superclass_of_item_is_item function.....	137
8.4.5	Check_is_case_of_referenced_classes_definition function	138
8.5	ISO13584_IEC61360_item_class_case_of_schema rule definitions.....	138
8.5.1	General	138
8.5.2	Imported_properties_are_visible_or_applicable_rule rule	138
8.5.3	Imported_data_types_are_visible_or_applicable_rule rule	139
8.5.4	Allowed_named_type_usage_rule rule.....	139
Annex A (informative)	Example physical file.....	141
Annex B (informative)	EXPRESS-G Diagram	146
Annex C (informative)	Partial dictionaries	157
Annex D (normative)	Value format specification	158

Bibliography.....	173
Figure 1 – Overview of the dictionary schema.....	21
Figure 2 – Pieces of data with relationships	25
Figure 3 – Implementation of "inter-piece" relationships using basic semantic units	26
Figure 4 – Relationship between basic semantic unit and dictionary element	29
Figure 5 – Current BSUs and dictionary elements	30
Figure 6 – Overview of supplier data and relationships	32
Figure 7 – Overview of class data and relationships.....	34
Figure 8 – Example of a supplier ontology	43
Figure 9 – Overview of property data element type data and relationships	47
Figure 10 – Kinds of data element types	47
Figure 11 – Entity hierarchy for the type system	50
Figure 12 – Overview of non-quantitative data element types.....	69
Figure 13 – ISO13584_IEC61360_language_resource_schema and support_resource_schema	105
Figure B.1 – ISO13584_IEC61360_dictionary_schema – EXPRESS-G diagram 1 of 7	147
Figure B.2 – ISO13584_IEC61360_dictionary_schema – EXPRESS-G diagram 2 of 7	148
Figure B.3 – ISO13584_IEC61360_dictionary_schema – EXPRESS-G diagram 3 of 7	149
Figure B.4 – ISO13584_IEC61360_dictionary_schema EXPRESS-G diagram 4 of 7	150
Figure B.5 – ISO13584_IEC61360_dictionary_schema – EXPRESS-G diagram 5 of 7	151
Figure B.6 – ISO13584_IEC61360_dictionary_schema – EXPRESS-G diagram 6 of 7	152
Figure B.7 – ISO13584_IEC61360_dictionary_schema – EXPRESS-G diagram 7 of 7	153
Figure B.8 – ISO13584_IEC61360_language_resource_schema – EXPRESS-G diagram 1 of 1.....	154
Figure B.9 – ISO13584_IEC61360_constraint_schema – EXPRESS-G diagram 1 of 1	155
Figure B.10 – ISO13584_IEC61360_item_class_case_of_schema – EXPRESS-G diagram 1 of 1.....	156
Table 1 – Cross reference table.....	20
Table D.1 – ISO/IEC 14977 EBNF syntactic metalanguage	159
Table D.2 – Transposing European style digits into Arabic digits	166
Table D.3 – Number value examples.....	167
Table D.4 – Characters from other rows of the Basic Multilingual Plane of ISO/IEC 10646-1	168

INTRODUCTION

The common ISO/IEC dictionary schema presented here is based on the intersection of the scopes of the following standards:

- IEC 61360-1;
- ISO 13584-42.

Relevant parts of the scope clauses of these standards include the following:

IEC 61360-1:2009

“This part of IEC 61360 provides a firm basis for the clear and unambiguous definition of characteristic properties (data element types) of all elements of electrotechnical systems from basic components to subassemblies and full systems. Although originally conceived in the context of providing a basis for the exchange of information on electric/electronic components, the principles and methods of this standard may be used in areas outside the original conception such as assemblies of components and electrotechnical systems and subsystems.”

ISO 13584-42:2010

“This part of ISO 13584 specifies the principles to be used for defining characterization classes of parts and properties of parts which provide for characterizing a part independently of any particular supplier-defined identification.

The rules and guidelines provided in this part of ISO 13584 are mandatory for the standardization committees responsible for creating standardized characterization hierarchies.

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

IEC SC3D and ISO TC184/SC4 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

STANDARD DATA ELEMENT TYPES WITH ASSOCIATED CLASSIFICATION SCHEME FOR ELECTRIC COMPONENTS –

Part 2: EXPRESS dictionary schema

1 Scope

This part of IEC 61360 series provides a formal model for data according to the scope as given in IEC 61360-1 and ISO 13584-42, and thus provides 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 IEC SC3D and ISO TC184/SC4, thus allowing for the implementation of dictionary systems dealing with data delivered according to either of the standards elaborated by both committees.

The scope of this part of IEC 61360 is the common ISO/IEC dictionary schema based on the intersection of the scopes of the two base standards IEC 61360-1 and ISO 13584-42.

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

The IEC 61360-2 forms the master document. ISO 13584-42 contains a copy of the IEC 61360-2 EXPRESS model in an informative annex

In a number of clauses, where the common EXPRESS model allows more freedom, IEC has defined more restrictions which are found in the methodology part of IEC 61360-1.

Two schemas are provided in this part of IEC 61360 defining the two options that may be selected for an implementation. Each of these options is referred to as a conformance class.

- The ISO13584_IEC61360_dictionary_schema2 provides for modelling and exchanging technical data element types with associated classification scheme used in the data element type definitions. It constitutes conformance class 1 of this part of IEC 61360.
- The ISO13584_IEC61360_language_resource_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 10303-41:2000, 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.

When used together with ISO 10303-21, each schema defines one single exchange format. The exchange format defined by conformance class 1 is fully compatible with the ISO 13584 series.

2 Normative references

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

IEC 61360-1:2009, *Standard data elements types with associated classification scheme for electric items – Part 1: Definitions – Principles and methods*

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

ISO/IEC 8859-1:1998, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

ISO/IEC 10646-1, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

ISO/IEC 14977, *Information technology – Syntactic metalanguage – Extended BNF*

ISO 639 (all parts), *Codes for the representation of names of languages*

ISO 843:1997, *Information and documentation – Conversion of Greek characters into Latin characters*

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

ISO 4217:2008, *Codes for the representation of currencies and funds*

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

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

ISO 10303-21:2002, *Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure*

ISO 10303-41:2000, *Industrial automation systems and integration – Product data representation and exchange – Part 41: Integrated generic resources: Fundamentals of product description and support*¹

ISO 13584-26:2000, *Industrial automation systems and integration – Parts library – Part 26: Logical resource: Information supplier identification*

ISO 13584-42:2010, *Industrial automation systems and integration – Parts library – Part 42: Description methodology: Methodology for structuring parts families*

3 Terms and definitions

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

3.1

abstract class

class of which all members are also members of one of its subclasses

¹ A new edition of ISO 10303-41 was published in 2005.

Note 1 to entry: Abstract classes are used when it is needed to group different kinds of objects in a class of a class inclusion hierarchy.

Note 2 to entry: In the common ISO13584/IEC61360 dictionary model, both abstract categorization classes and abstract characterization classes can be defined. The fact of being abstract is only a conceptual characteristic of a class. This characteristic is not explicitly represented in the model.

Note 3 to entry: Through inheritance, abstract characterization class allows to share, for example, some visible properties between different subclasses that correspond to different kinds of items.

3.2

applicable property of a class

applicable property necessarily possessed by each part that is member of a characterization class

Note 1 to entry: Each part that is member of a characterization class possesses an aspect corresponding to each applicable property of this characterization class.

Note 2 to entry: The above definition is conceptual, there is no requirement that all the applicable properties of a class should be used for describing each part of this class at the data model level.

Note 3 to entry: All the applicable properties of a superclass are also applicable properties for the subclasses of this superclass.

Note 4 to entry: Only properties defined or inherited as visible and imported properties of a class may be applicable properties.

Note 5 to entry: To facilitate integration of component libraries and electronic catalogues based on ISO 13584-24:2003 and ISO 13584-25, these parts of ISO 13584 request that only properties that are applicable to a class be used to characterize their instances in component libraries and electronic catalogues.

3.3

attribute

data element for the computer-sensible description of a property, a relation or a class

Note 1 to entry: An attribute describes only one single detail of a property, of a class or of a relation.

EXAMPLE The name of a property, the code of a class, the measure unit in which values of a property are provided.

3.4

basic semantic unit

entity that provides an absolute and universally unique identification of a certain object of the application domain that is represented as a dictionary element

EXAMPLE 1 A dictionary compliant with this part of IEC 61360 provides for the identification of classes, properties, information sources and datatypes.

EXAMPLE 2 A dictionary compliant with ISO 13584-24:2003 provides for the identification of classes, properties, information sources, datatypes, tables, documents and program libraries.

EXAMPLE 3 In ISO 13584-511, the class of the hexagon head bolts is identified by a BSU, the property thread tolerance grade is also identified by a BSU.

Note 1 to entry: The content of a basic semantic unit may also be represented as an IRDI.

3.5

characteristic of a product

product characteristic

invariable property, characteristic of a product, whose value is fixed once the product is defined

Note 1 to entry: Changing the value of a characteristic of a product would mean changing the product.

EXAMPLE For a ball bearing, the inner diameter and the outer diameter are product characteristics.

Note 2 to entry: Adapted from ISO 13584-24:2003, definition 3.12.

3.6**class**

abstraction of a set of similar products

Note 1 to entry: A product that complies with the abstraction defined by a class is called a class member.

Note 2 to entry: A class is an intentional concept that can take different extensional meanings in different contexts.

EXAMPLE The set of products used by a particular enterprise and the set of all ISO-standardized products are two examples of contexts. In these two contexts (the particular enterprise and ISO), the set of products that are considered as members of the *single ball bearing* class can be different, in particular because employees of each enterprise ignore a number of existing single ball bearing products.

Note 3 to entry: Classes are structured by class inclusion relationships.

Note 4 to entry: A class of products is a general concept as defined in ISO 1087-1. Thus, it is advisable that the rules defined in ISO 704 be used for defining the designation and definition attributes of classes of products.

Note 5 to entry: In the context of the ISO 13584 series, a class is either a characterization class, associated with properties and usable for characterizing products, or a categorization class, not associated with properties and not usable for characterizing products.

3.7**class inclusion relationship**

relationship between classes that means inclusion of class members: if A is a superclass of A1 this means that, in any context, any member of A1 is also member of A

EXAMPLE 1 The set of products used by a particular enterprise and the set of all ISO-standardized products are two examples of contexts.

EXAMPLE 2 In any context, the class *capacitor* includes the class *electrolytic capacitor*.

Note 1 to entry: Class inclusion defines a hierarchical structure between classes.

Note 2 to entry: Class inclusion is a conceptual relationship that does not prescribe anything at the data representation level. Consequently, it does not prescribe any particular database schema or data model.

Note 3 to entry: In the model defined in this part of IEC 61360, the “is-a” relationship ensures class inclusion. This part of IEC 61360 recommends that the “case-of” relationship also ensure class inclusion.

Note 4 to entry: The class inclusion relationship is also called subsumption.

3.8**class member**

product that complies with the abstraction defined by a class

3.9**class valued property**

property that has one single value for a whole characterization class of products

Note 1 to entry: The value of a class valued property is not defined individually for every single product of a characterization class, but globally for the class itself.

Note 2 to entry: When all products from a characterization class of products have the same value for a particular property, defining this property as a class valued property permits to avoid duplication of the value for each instance.

Note 3 to entry: Class valued properties can also be used to capture some commonality between different characterization classes when such a commonality is not captured by the hierarchy structure.

3.10**common ISO13584/IEC61360 dictionary model**

data model for product ontology, using the information modeling language EXPRESS, resulting from a joint effort between ISO/TC 184/SC 4/WG 2 and IEC SC3D

Note 1 to entry: Several levels of allowed implementations, known as conformance classes, are defined for the common ISO13584/IEC61360 dictionary model. Conformance class 1 consists of the various schemes defined in IEC 61360-2.

in this part of IEC 61360 (that duplicate information contained in this standard), more the ISO13584_IEC61360_dictionary_aggregate_extension_schema documented in ISO 13584-25 (duplicated in IEC 61360-5). Other conformance classes are documented in ISO 13584-25 (conformance classes 2, 3 and 4).

Note 2 to entry: In the ISO 13584 standard series, each particular product ontology addressing a particular product domain and based on the common ISO13584/IEC61360 dictionary model is called a reference dictionary for that domain.

3.11

context dependent characteristic of product

property of a *product* whose value depends on some *context parameters*

Note 1 to entry: For a given product, a context dependent characteristic is mathematically defined as a function whose domain is defined by some context parameters that define the product environment.

EXAMPLE For a *ball bearing*, the *life-time* is a context dependent characteristic that depends on the *radial load*, the *axial load* and the *rotational speed*.

Note 2 to entry: Adapted from ISO 13584-24:2003, definition 3.22.

3.12

context parameter

variable whose value characterizes the context in which a *product* is inserted

EXAMPLE 1 The *dynamic-load* applied to a *bearing* is a context parameter for this *bearing*.

EXAMPLE 2 The *ambient temperature* in which the *resistance* of a *resistor* is measured is a context parameter for this *resistor*.

Note 1 to entry: This definition supersedes the definition given in ISO 13584-24:2003, that was the following: “a variable of which the value characterizes the context in which it is intended to insert a *product*”.

Note 2 to entry: In the ISO 13584 standard series, a property value is represented as a data element type.

3.13

data element type

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

Note 1 to entry: In the ISO 13584 standard series, a property value is represented as a data element type.

3.14

dictionary data

set of data that represents product ontologies possibly associated with product categorizations

Note 1 to entry: It is advisable that dictionary data be exchanged using some conformance class of the common ISO/IEC dictionary model.

Note 2 to entry: This definition of dictionary data supersedes the previous definition from the first edition of IEC 61360-2 that was the following: “the set of data that describes hierarchies of characterization classes of products and properties of these products”.

3.15

dictionary element

set of attributes that constitutes the dictionary description of certain objects of the application domain

EXAMPLE 1 A dictionary compliant with this part of IEC 61360 provides for the description of classes, properties, information sources and datatypes.

EXAMPLE 2 A dictionary compliant with ISO 13584-24:2003 provides for the description of classes, properties, information sources, datatypes, tables, documents and program libraries.

3.16

family of products

set of products represented by the same characterization class

Note 1 to entry: This definition supersedes the definition given in ISO 13584-24:2003, that was the following: “a simple or generic family of parts”.

3.17

feature

aspect of a product that can be described by a characterization class and a set of property-value pairs

Note 1 to entry: In the real world, a feature instance only exists embedded within the product of which it is an aspect.

EXAMPLE 1 The head of a screw is a feature described by a head class and a number of head properties, which depends upon the head class. A screw head only exists when it belongs to a screw.

Note 2 to entry: Features are represented by means of **item_class** whose the **instance_sharable** attribute equals *false*.

Note 3 to entry: The **instance_sharable** attribute allows to specify the conceptual status of an item: either a stand-alone item (**instance_sharable** = *true*), or a feature (**instance_sharable** = *false*). It does not imply any constraint at the data representation level. In the common ISO13584/IEC61360 dictionary model, representing several real world instances that share the same EXPRESS representation by a single EXPRESS entity, or by several EXPRESS entities is considered as implementation dependant. There exist no mechanism for specifying whether data value of a feature instance may or may not be shared.

EXAMPLE 2 The same instance of a screw head class can be referenced by several instances of a screw class. It means that there exists several screw heads, but that all these screw heads have the same characterization class and the same set of property values. The **instance_sharable** attribute allows to specify that changing this instance of the screw head class would change several instances of the screw class.

3.18

imported property

property defined in a class that is selected by another class of the same or of a different reference dictionary, by means of the case-of relationship, to become applicable to the latter class

Note 1 to entry: Only properties that are visible and/or applicable in a class can be imported from this class.

Note 2 to entry: Importation between classes of different reference dictionaries allows reusing properties, defined for example in a standard reference dictionary, without redefining them.

Note 3 to entry: Importation between classes of the same reference dictionary acknowledges the fact that some products can perform several functions, requiring the capability to import property from several higher level classes.

Note 4 to entry: When it is imported in a new class, a property keeps its original identifier, thus all the attributes do not need to be duplicated.

Note 5 to entry: An imported property is applicable for the class where it is imported.

3.19

information

facts, concepts or instructions

[SOURCE: ISO 10303-1:1994, definition 3.2.20]

3.20

information model

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

[SOURCE: ISO 10303-1:1994, definition 3.2.21]

3.21

information supplier supplier

organization that delivers an ontology, i.e. a data dictionary, or a supplier library that is responsible for its content

Note 1 to entry: This definition of supplier supersedes the definition of information supplier from ISO 13584-1:2001 that was the following: "organization that delivers a supplier library in the standard format defined in this International Standard and is responsible for its content".

3.22

international registration data identifier

internationally unique identifier for a certain object of the application domain as defined in ISO/IEC 11179-5

Note 1 to entry: Only international registration data identifiers compliant with ISO/TS 29002-5 are used in the context of the ISO 13584 standard series.

Note 2 to entry: An international registration data identifier may be used for representing the content of a basic semantic unit that identifies a dictionary element as a string.

Note 3 to entry: An international registration data identifier may also be used for identifying the content of an attribute of a dictionary element.

EXAMPLE The unit of measure of a property, a value of a property or a constraint over a property may be identified by an IRDI.

3.23

is-a relationship

class inclusion relationship associated with inheritance: if A1 *is-a* A, then each product belonging to A1 belongs to A, and all that is described in the context of A is automatically duplicated in the context of A1

Note 1 to entry: This mechanism is usually called "inheritance".

Note 2 to entry: In the common ISO13584/IEC61360 dictionary model, the is-a relationship can only be defined between characterization classes. It is advisable that it defines a single hierarchy and it ensures that both visible and applicable properties are inherited.

3.24

is-case-of relationship case-of

property importation mechanism: if A1 is *case-of* A, then the definition of A products also covers A1 products, thus A1 can import any property from A

Note 1 to entry: The goal of the *case-of* relationship is to allow connecting together several class inclusion hierarchies while ensuring that referenced hierarchies can be updated independently.

Note 2 to entry: There is no constraint that the case-of relationship is intended to define single hierarchies.

Note 3 to entry: In the common ISO13584/IEC61360 dictionary model, the case-of relationship can in particular be used in four cases: (1) to link a characterization class to a categorization class, (2) to import, in the context of some standardized reference dictionaries, some properties already defined in other standardized reference dictionaries, (3) to connect a user reference dictionary to one or several standardized reference dictionaries, (4) to describe a product using the properties of different classes: when products of class A1 fulfil two different functions, and are thus logically described by properties associated with two different classes, A and B, A1 can be connected by is-a to e. g., A, and by case-of to B.

Note 4 to entry: The EXPRESS resource constructs for modeling the case-of relationships are defined in 4.5 and after.

3.25

item

thing that can be characterized by means of a characterization class to which it belongs and a set of property value pairs

Note 1 to entry: This definition supersedes the definition given in ISO 13584-24:2003, that was the following: "a thing that can be captured by a class structure and a set of properties".

Note 2 to entry: In the ISO 13584 standard series, both products and features of products that correspond to composite properties are items.

3.26

leaf characterization class

characterization class that is not further specialized into more precise characterization classes

EXAMPLE Countersunk flat head screw with cross recess (type Y) and hexagon socket head cap screw with metric fine pitch thread are leaf characterization classes defined in ISO 13584-511.

3.27

non-leaf characterization class

characterization class that is further specialized into more precise characterization classes

EXAMPLE *Externally-threaded component* and *metric threaded bolt/screw* are non-leaf characterization classes defined in ISO 13584-511.

3.28

non-quantitative data element type

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

3.29

part

material or functional element that is intended to constitute a component of different products

[SOURCE: ISO 13584-1:2001, definition 3.1.16]

3.30

parts library

computer-sensible product ontology and computer-sensible description of a set of products by means of references to this ontology

Note 1 to entry: This definition supersedes the definition given in the first edition of this part of IEC 61360, which was the following: "identified set of data and possibly programs which can generate information about a set of parts".

3.31

product

thing or substance produced by a natural or artificial process

Note 1 to entry: In this part of IEC 61360, the term product is taken in its widest sense to include devices, systems and installations as well as materials, processes, software and services.

3.32

product categorization

part categorization

categorization

recursive partition of a set of products into subsets for a specific purpose

Note 1 to entry: Subsets which appear in a product categorization are called product categorization classes, or product categories.

Note 2 to entry: A product categorization is not a product ontology. It cannot be used for characterizing products.

Note 3 to entry: No property is associated with categorizations.

Note 4 to entry: Several categorizations of the same set of products are possible according to their target usage.

EXAMPLE The UNSPSC classification, defined by the United Nations, is an example of a product categorization that was developed for spend analysis.

Note 5 to entry: Using the *is-case-of* relationship, several product characterization class hierarchies can be connected to a categorization hierarchy to generate a single structure.

3.33**product categorization class**
part categorization class
categorization class

class of products that constitutes an element of a categorization

EXAMPLE *Manufacturing Components and Supplies*, and *Industrial Optics* are examples of a product categorization class defined in the UNSPSC.

Note 1 to entry: No rule is given in this part of IEC 61360 about how to select categorization classes. This concept is introduced (1) to clarify its difference with characterization class, and (2) to explain that the same characterization class can be connected to any number of categorization classes.

Note 2 to entry: There is no property associated with a categorization class.

3.34**product characterization**
part characterization

description of a product by means of a product characterization class to which it belongs and a set of property value pairs

EXAMPLE *Hexagon_head_bolts_ISO_4014* (Product grades = A, thread_type=M, length= 50, Diameter = 8) is an example of a product characterization.

3.35**product characterization class**
part characterization class
characterization class

class of products that fulfil the same function and that share common properties

Note 1 to entry: Product characterization classes can be defined at various levels of details, thus defining a class inclusion hierarchy.

EXAMPLE *Metric threaded bolt/screw* and *hexagon head bolt* are examples of product characterization classes defined in ISO 13584-511. The first characterization class is included in the second one. *Transistor* and *bipolar power transistor* are examples of product characterization classes defined in IEC 61360-DB. The second one is included in the first one.

3.36**product ontology**
part ontology
ontology

model of product knowledge, done by a formal and consensual representation of the concepts of a product domain in terms of identified characterization classes, of class relations and of identified properties

Note 1 to entry: Product ontologies are based on a class-instance model that allows one to recognize and to designate the sets of products, called characterization classes, that have a similar function (e.g., *ball bearing*, *capacitor*), but also to discriminate within a class the various subsets of products, called instances, that are considered as identical. It is advisable that the rules defined in ISO 1087-1 be used for formulating designation and definitions of characterization classes. Instances have no definitions. They are designated by the class to which they belong, and a set of property-value pairs.

Note 2 to entry: Ontologies are not concerned with words but with concepts, independent of any particular language.

Note 3 to entry: "Consensual" means that the conceptualization is agreed upon in some community.

Note 4 to entry: "Formal" means that the ontology is intended to be machine interpretable. Some level of machine reasoning is logically possible over ontology, e.g., consistency checking, making inferences.

Note 5 to entry: "Identified" means that each ontology characterization class and properties are associated with a globally unique identifier allowing one to reference this concept from any context.

Note 6 to entry: The data model for ontology recommended in this part of IEC 61360 is the common ISO13584/IEC61360 dictionary model, whose simplest version is documented in this part of IEC 61360. More complete versions are documented in ISO 13584-25 and IEC 61360-5 (conformance classes 1, 2, 3 and 4 of both documents).

Note 7 to entry: In this part of IEC 61360, each product ontology addressing a particular product domain compliant with the common ISO13584/IEC61360 dictionary model is called a reference dictionary for that domain.

EXAMPLE The reference dictionary for electric components, which is defined in IEC 61360-DB, is a product ontology for electric components compliant with the common ISO13584/IEC61360 dictionary model. It is agreed upon by all member bodies of IEC SC3D. A corporate reference dictionary is agreed upon by experts designated by management on behalf of the company.

3.37

property

defined parameter suitable for the description and differentiation of products

Note 1 to entry: A property describes one aspect of a given object.

Note 2 to entry: A property is defined by the totality of its associated attributes. The types and number of attributes that describe a property with high accuracy are documented in this part of IEC 61360.

Note 3 to entry: This part of IEC 61360 has identified three different kinds of properties: product characteristics, context parameters and context-dependent product characteristics.

Note 4 to entry: This definition of property supersedes the previous definition of the previous edition of this part of IEC 61360 that was the following: “an information that can be represented by a data element type”.

Note 5 to entry: In the ISO 13584 standard series, a property value is represented as a data element type.

3.38

property data type

allowed set of values of a property

3.39

property definition class

product characterization class in the context of which a product property is defined

Note 1 to entry: In the common ISO13584/IEC61360 dictionary model, each product property has one property definition class that defines its domain of application. The property is only meaningful for this class, and all its subclasses, and it is said to be *visible* over this domain.

EXAMPLE In ISO 13584-511, *wrenching height* has *nut* as its property definition class and *major diameter of external thread* has *metric external thread* as its property definition class.

3.40

quantitative data element type

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

3.41

reference dictionary

product ontology compliant with the common ISO13584/IEC61360 dictionary model

Note 1 to entry: In the ISO 13584 standard series, a product ontology that addresses a particular product domain, based on the common ISO13584/IEC61360 dictionary model, is called a reference dictionary for that domain.

3.42

resource construct

collection of EXPRESS language entities, types, functions, rules and references that together define a valid description of data

Note 2 to entry: This definition is adapted from the definition of resource construct in ISO 10303-1:1994, i.e., “the collection of EXPRESS language entities, types, functions, rules and references that together define a valid description of product data”.

[SOURCE: ISO 13584-1:2001, definition 3.1.21, modified]

3.43

subclass

class that is one step below another class in a class inclusion hierarchy

Note 1 to entry: In the common ISO13584/IEC61360 dictionary model, class inclusion hierarchies are defined by the *is-a* relationship. They can also be established by the *case-of* relationships.

3.44

superclass

class that is one step above another class in a class inclusion hierarchy

Note 1 to entry: In the common ISO13584/IEC61360 dictionary model, class inclusion hierarchies are defined by the *is-a* relationship. They can also be established by the *case-of* relationships.

Note 2 to entry: In the common ISO13584/IEC61360 dictionary model, a class has at most one superclass specified by means of an *is-a* relationship.

3.45

supplier library

parts library of which the information supplier is different from the library user

Note 1 to entry: This definition supersedes the definition given in ISO 13584-1:2001, that was the following: “set of data, and possibly of programs, for which the supplier is defined and that describes in the standard format defined in this International Standard a set of products and/or a set of representation of products”.

3.46

visible property

property that has a definition meaningful in the scope of a given characterization class, but that does not necessarily apply to the various products belonging to this class

Note 1 to entry: Meaningful in the scope of a given characterization class means that a human observer is able to determine, for any product of the characterization class, whether the property applies, and, if it applies, to which product aspect it corresponds.

Note 2 to entry: The concept of a visible property allows sharing the definition of a property among product characterization classes where this property does not necessarily apply.

EXAMPLE The *non-threaded length* property is meaningful for any class of *screw* but it applies only to those screws that have a non-threaded part. It can be defined as visible at the *screw* level, while becoming applicable only in some subclasses.

Note 3 to entry: All the visible properties of a superclass that is a product characterization class are also visible properties for its subclasses.

Note 4 to entry: To facilitate integration of component libraries and electronic catalogues based on ISO 13584-24:2003 and ISO 13584-25, these parts of ISO 13584 request that only properties that are applicable to a class be used to characterize their instances in component libraries and electronic catalogues.

Note 5 to entry: This definition of a visible property supersedes the previous definition from ISO 13584-24:2003 that was the following: “a property that is defined for some class of products and that does not necessarily apply to the different products of this class of products”.

4 Overview of the common dictionary schema and compatibility with ISO13584_IEC61360_dictionary_schema

4.1 General

In the following subclauses the architecture of the common dictionary schema will be presented and it will be explained how the same information model has to be used in the International Standards to ensure their compatibility.

The common dictionary schema combines the requirements of IEC 61360 series and ISO 13584 series. Therefore, it contains resources to accommodate the specific requirements of both series of International Standards. These resources are provided either as optional capabilities or as subtypes of the types defined to fulfil the common requirements.

4.2 Use of the common dictionary schema to exchange IEC 61360-1 compliant data

The common dictionary schema to exchange IEC 61360-1 compliant data shall be used as follows:

- a) the ISO 13584 series specific extensions to support multilingual capability are not required for the exchange of dictionary elements defined according to IEC 61360-1. However these extensions that are `present_translations`, `translated_label` and `translated_text` shall be used in the exchange structure for compatibility reasons;
- b) if a component class has a superclass, the `coded_name` shall be defined as a `value_code` in the domain of the classifying data element type of the superclass;
- c) if a classifying data element type exists within a specific component class, for each value in its domain a subclass and a term shall be defined;
- d) a classifying data element type, optional in conformance class 2 in the common dictionary schema, shall always be provided for the component classes defined according to IEC 61360-1;
- e) only SI units shall be used although the common dictionary schema enables the use of many kind of system units. When using this schema however for the exchange of IEC 61360 series compliant data, only SI shall be used for quantitative data element types.

4.3 Compatibility with ISO 13584-42

An implementation compliant with this part of IEC 61360 shall support all the entities, types and associated constraints that belong to the conformance class it claims to support. Therefore conformance to conformance class 1 of this part of IEC 61360 requires that all the entities, types and associated constraints defined in the common dictionary schema be supported. ISO 13584 data conforming to the common dictionary schema may thus be processed by an IEC 61360 implementation that conforms to conformance class 1 that includes all the features of conformance class 1. In ISO 13584, a specific conformance class 3 is intended to contain all the entities, types and associated constraints defined in the common dictionary schema. An ISO 13584 compliant implementation conforms to this conformance class shall therefore be able to support IEC data that belongs to conformance class 1 of this part of IEC 61360.

4.4 Naming correspondence between IEC 61360-1 and IEC 61360-2

Due to specific application restrictions – for example the EXPRESS language allows no spaces in entity names –, a number of similar 'EXPRESS names' are created by replacing the blank in a name by an underscore (for example preferred name is presented as `preferred_name`).

At other places names are used in the EXPRESS model that deviate from those used in IEC 61360-1. This is a consequence of the effort to reach one common EXPRESS information model together with Part Libraries.

The table below presents a help for matching the names used in the two parts of IEC 61360.

Table 1 – Cross reference table

naming in 61360-2	naming in 61360-1
<code>component_class</code>	Component class
<code>condition_DET</code>	Condition data element type
<code>dependent_P_DET</code>	Data element type
<code>det_classification</code>	Data element type class
<code>(DER)dic_identifier</code>	Identifier
<code>dic_value</code>	Value
<code>material_class</code>	Material class
<code>meaning</code>	Value meaning
<code>non_dependent_P_DET</code>	Data element type
<code>preferred_symbol</code>	Preferred letter symbol

naming in 61360-2	naming in 61360-1
revision	Revision number
source_doc_of_definition	Source document of data element type definition
source_doc_of_definition	Source document of component class definition
synonymous_symbols	Synonymous letter symbol
unit	Unit of measure
value_code	Value code
version	Version number

4.5 Main structure of the common dictionary schema

This subclause explains the main resource constructs provided by the common dictionary schema:

- **dictionary_element** is any element defined in the dictionary;
- **supplier_element** captures the data of suppliers of dictionary elements (classes, properties, data types); class models the dictionary element of classes (families) which 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 clause 5: **ISO13584_IEC61360_dictionary_schema**.

In the presentation of the common dictionary schema, some overview diagrams are provided as planning models (see Figure 1 to Figure 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. Figure 1 below outlines as a planning model the main structure of the common dictionary schema. Most of these figures contain overview models (or planning models) but show only that level of detail which is appropriate at a certain place.

For clarification of the diagrams, some of the relationships that are defined in the EXPRESS model are omitted. Figure F.1 below outlines as a planning model the main structure of the common ISO13584/IEC61360 dictionary model.

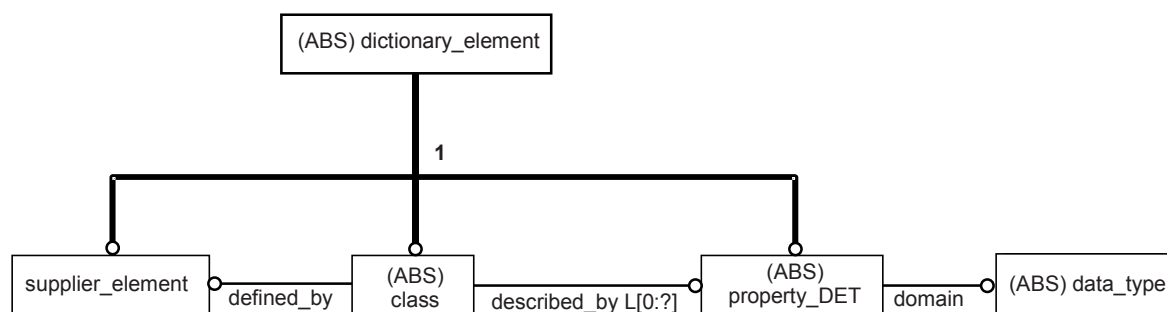


Figure 1 – Overview of the dictionary schema

5 ISO13584_IEC61360_dictionary_schema

5.1 General

This clause, which constitutes the main part of the common information model of ISO 13584-42 and IEC 61360 series, 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.

5.2 Dictionary schema

In the first place, the schema needs to be declared.

EXPRESS specification:

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

5.3 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);

REFERENCE FROM person_organization_schema(organization, address);

REFERENCE FROM measure_schema;

REFERENCE FROM ISO13584_IEC61360_language_resource_schema;

REFERENCE FROM ISO13584_IEC61360_class_constraint_schema;

REFERENCE FROM ISO13584_IEC61360_item_class_case_of_schema;

REFERENCE FROM ISO13584_external_file_schema
    (external_item,
     external_file_protocol,
     external_content,
     not_translatable_external_content,
     not_translated_external_content,
     translated_external_content,
     language_specific_content,
     http_file,
     http_class_directory,
     http_protocol);
(*
```

NOTE The schemata referenced above can be found in the following documents:

support_resource_schema	ISO 10303-41
person_organization_schema	ISO 10303-41
measure_schema	ISO 10303-41

ISO13584_IEC61360_language_resource_schema (which is duplicated for convenience in this document)	IEC 61360-2
ISO13584_IEC61360_class_constraint_schema (which is duplicated for convenience in this document)	IEC 61360-2
ISO13584_IEC61360_item_class_case_of_schema (which is duplicated for convenience in this document)	IEC 61360-2
ISO13584_external_file_schema	ISO 13584-24:2003

5.4 Constant definitions

This subclause contains constant definitions used later in type definitions (see 5.11).

EXPRESS specification:

```
*)
CONSTANT
    dictionary_code_len: INTEGER := 131;
    property_code_len: INTEGER := 35;
    class_code_len: INTEGER := 35;
    data_type_code_len: INTEGER := 35;
    supplier_code_len: INTEGER := 149;
    version_len: INTEGER := 10;
    revision_len: INTEGER := 3;
    value_code_len: INTEGER := 35;
    pref_name_len: INTEGER := 255;
    short_name_len: INTEGER := 30;
    syn_name_len: INTEGER := pref_name_len;
    DET_classification_len: INTEGER := 3;
    source_doc_len: INTEGER := 255;
    value_format_len: INTEGER := 80;
    sep_cv: STRING := '#';
    sep_id: STRING := '#';
END_CONSTANT;
(*
```

5.5 Identification of a dictionary

A **dictionary_identification** entity allows to identify unambiguously a particular version of a particular dictionary of a particular information supplier, standard or not. It contains a **code** defined by the dictionary supplier that identifies the dictionary, a **version** number and **revision** number that characterize a particular state of this dictionary.

NOTE 1 The case where dictionary version and revision should be incremented is defined in IEC 61360-1.

EXPRESS specification:

```
*)
ENTITY dictionary_identification;
    code: dictionary_code_type;
    version: version_type;
    revision: revision_type;
    defined_by: supplier_bsu;
DERIVE
    absolute_id: identifier :=
        defined_by.absolute_id + sep_id + code + sep_cv + version;
```

```

UNIQUE
    UR1: absolute_id;
END_ENTITY; -- dictionary_identification
(*)

```

Attribute definitions:

code: the code that characterizes the dictionary.

version: the version number that characterizes the version of the dictionary.

revision: the revision number that characterizes the revision of the dictionary.

defined_by: the supplier who defines the dictionary.

absolute_id: the unique identification of the dictionary.

Formal propositions:

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

Informal propositions:

IP1: when a dictionary is defined by a standard document that contains only one dictionary, the dictionary **code** shall be the standard number of the document that describes this dictionary if this document only defines one dictionary. It shall be the name defined for the pertinent dictionary in the document that describes it if this document defines several dictionaries. Unless otherwise specified, version shall be set to 1 and revision numbers shall be set to 0 for dictionaries defined by standard documents.

NOTE 2 Representation of the standard numbers of standard documents is specified in 5.1 and 5.2 of ISO 13584-26:2000.

5.6 Basic Semantic Units: defining and using the dictionary

5.6.1 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 shall 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 2.

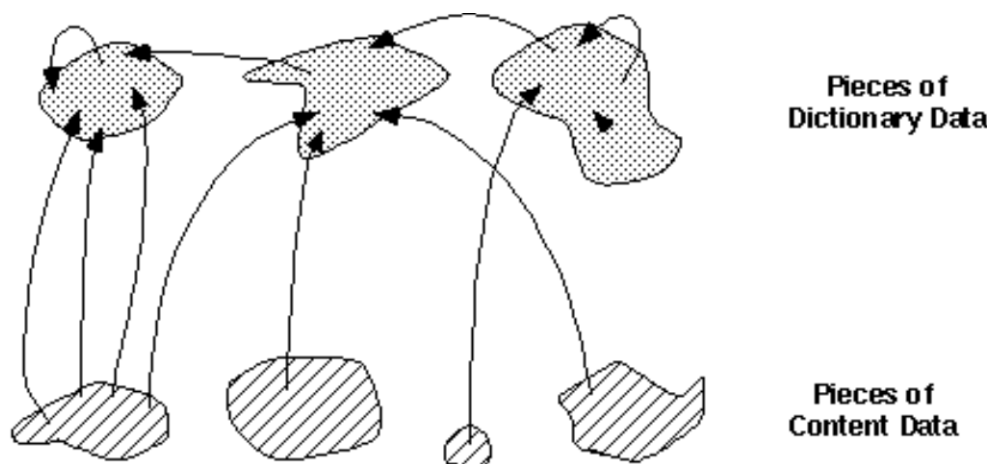


Figure 2 – Pieces of data with relationships

Every one of these pieces corresponds to a physical file (according to ISO 10303-21). EXPRESS (ISO 10303-11:2004) 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.

5.6.2 Three levels architecture of the dictionary data

5.6.2.1 General

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

Assume some piece of content wants to refer a certain dictionary description.

EXAMPLE 1 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.

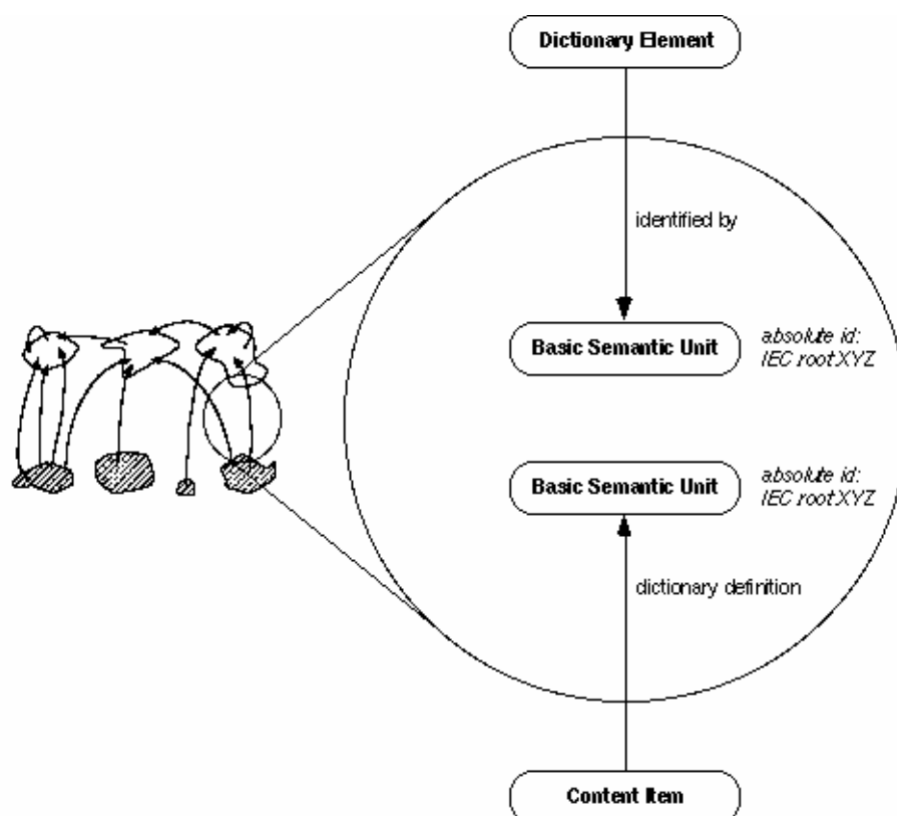


Figure 3 – Implementation of "inter-piece" relationships using basic semantic units

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

EXAMPLE 2 Between a class of components and the associated **property_DETs**.

A BSU provides a reference to a dictionary description in any place where this is needed.

EXAMPLE 3 Dictionary delivery, update delivery, library delivery, component data exchange.

The data associated with a property could be exchanged as a pair (**property_BSU**, <value>).

Figure 3 outlines the implementation of this general mechanism.

5.6.2.2 Basic_semantic_unit

A **basic_semantic_unit** is a 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.

5.6.2.3 Dictionary_element

A **dictionary_element** is a full definition of the data required to be captured in the semantic dictionary for some concepts. 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;
        administration: OPTIONAL administrative_data;
        is_deprecated: OPTIONAL BOOLEAN;
        is_deprecated_interpretation: OPTIONAL note_type;
    WHERE
        WR1: NOT EXISTS (SELF.is_deprecated)
            OR EXISTS (SELF.is_deprecated_interpretation);
    END_ENTITY; -- dictionary_element
    (*

```

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 redefined 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 and classes 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 5.11.3.2).

NOTE 3 The **revision** attribute will be used to encode the property and class attribute "Revision Number".

administration: optional information on the life cycle of the **dictionary_element**.

NOTE 4 The **administration** attribute will be used to represent the information related to the configuration management and translation history.

is_deprecated: an optional Boolean. When true, it specifies that the **dictionary_element** shall no longer be used.

is_deprecated_interpretation: specifies the deprecation rationale and how instance values of the deprecated element, and of its corresponding **BSU**, should be interpreted.

Formal propositions:

WR1: when **is_deprecated** exists, **is_deprecated_interpretation** shall exist.

Informal propositions:

IP1: instance values of **is_deprecated_interpretation** element shall be defined at the time where deprecation decision was taken.

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

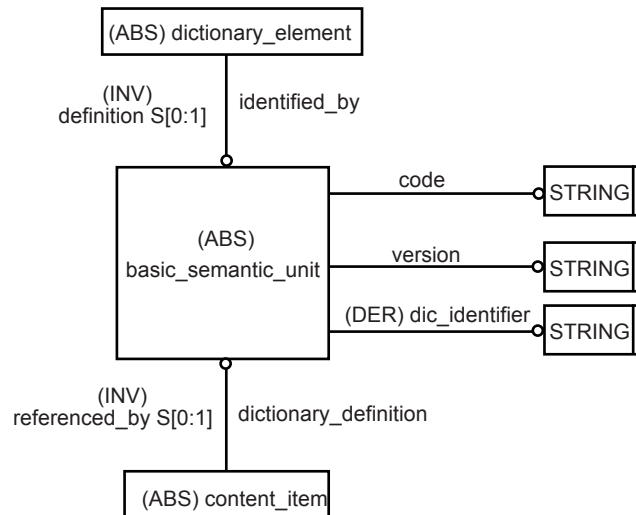


Figure 4 – Relationship between basic semantic unit and dictionary element

5.6.2.4 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; -- content_item
(*

```

Attribute definitions:

dictionary_definition: the basic semantic unit to be used for referring to the definition in the dictionary.

5.6.3 Overview of basic semantic units and dictionary elements

For every kind of dictionary data, a pair of **basic_semantic_unit** and **dictionary_element** subtypes shall be defined. Figure 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 5.7;
- for classes see 5.8;
- for properties / data element types see 5.9;
- for data types see 5.10.

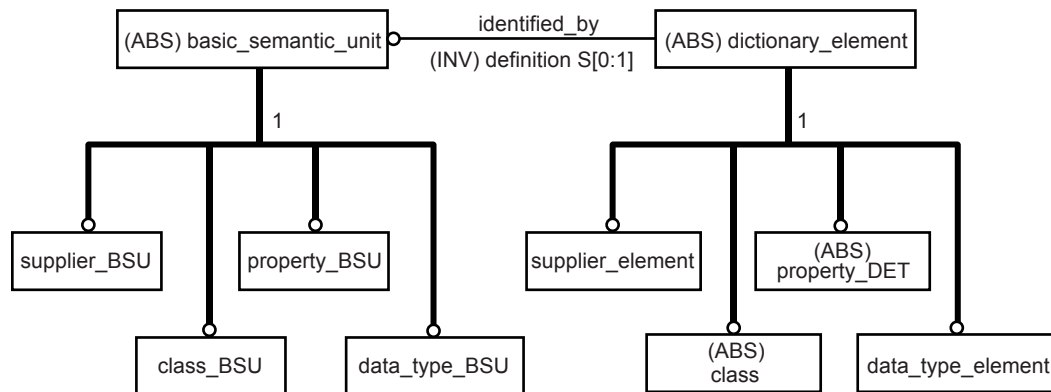


Figure 5 – Current BSUs and dictionary elements

5.6.4 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);
- supplier-defined dictionary element (any supplier-defined dictionary element defined in the model; in this document supplier-defined dictionary element are **property_DET** and **data_type_element**, but there are provisions to extend this mechanism to other items);
- version of the supplier-defined dictionary element.

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

NOTE The structure on this absolute identification is different from the structure defined in edition 1 of IEC 61360-2 (duplicated for convenience in ISO 13584-42:1998). In the previous edition, the absolute identification of any **dictionary_element** associated with a **name_scope** (including **property_DET** and **data_type_element**) consisted of: supplier code + class code (corresponding to the **name_scope** class) + dictionary element code + dictionary element version. In this edition, the class code has been removed. Thus, the dictionary element code is unique, for the same type of dictionary element, over all the classes defined by the same supplier. For existing reference dictionaries, registration authorities, maintenance authorities or standardization groups in charge of standard dictionaries should ensure this unicity, possibly by defining new codes prefixed by **name_scope** class codes.

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 shall be present, to avoid clashes of identifiers.

This identification scheme is described formally in the **absolute_id** attribute of the xxx_BSU entities defined from 5.7 through 5.12.

5.6.5 Extension possibilities for other types of data

5.6.5.1 General

The BSU – dictionary element mechanism is very general and not limited to the four kinds of data used here (see Figure 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 **xxx_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 5.7.3 through 5.10 or the current kinds of data).

5.6.5.2 Supplier_related_BSU

The **supplier_related_BSU** provides for the dictionary elements to be associated with suppliers.

EXAMPLE For ISO 13584 series: program libraries.

EXPRESS specification:

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

5.6.5.3 Class_related_BSU

The **class_related_BSU** provides for the dictionary elements to be associated with classes.

EXAMPLE For ISO 13584 tables, documents, etc.

EXPRESS specification:

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

5.6.5.4 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; -- supplier_BSU_relationship  
(*
```

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.

5.6.5.5 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; -- class_BSU_relationship
(*
```

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.

5.7 Supplier data

5.7.1 General

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 6 presents a planning model of the data associated with suppliers, followed by the EXPRESS definition.

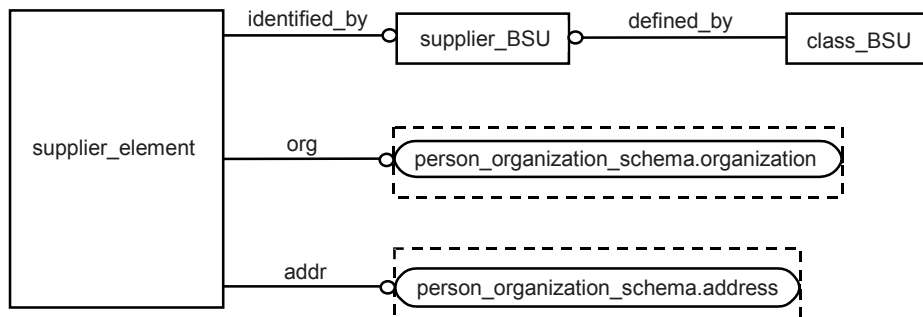


Figure 6 – Overview of supplier data and relationships

5.7.2 Supplier_BSU

The **supplier_BSU** entity provides for unique identification of information suppliers.

EXPRESS specification:

```
*)
ENTITY supplier_BSU
SUBTYPE OF(basic_semantic_unit);
    SELF\basic_semantic_unit.code: supplier_code_type;
DERIVE
    SELF\basic_semantic_unit.version: version_type := '1';
    absolute_id: identifier := SELF\basic_semantic_unit.code;
UNIQUE
    UR1: absolute_id;
```

```
END_ENTITY; -- supplier_BSU
(*
```

Attribute definitions:

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

version: the version number of a supplier code shall be equal to 1.

absolute_id: the absolute identification of the supplier.

Formal propositions:

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

5.7.3 Supplier_element

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

EXPRESS specification:

```
*)
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; -- supplier_element
(*
```

Attribute definitions:

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.

EXAMPLE Program library in ISO 13584-24:2003.

5.8 Class data

5.8.1 General

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

Figure 7 outlines, as a planning model, the data associated with classes and their relationship to other dictionary elements.



The dictionary data for classes (as shown in Figure 7) is spread over three inheritance levels:

- EXAMPLE Other subtypes of **classes**, in particular **functional_view_class**, **functional_model_class** and **fm_class_view_of** are specified in ISO13584-24:2003. They do not characterize products, but they provide for exchanging particular representations of product, e. g., geometrical representations.

- NOTE 1 Two subtypes of **item_class**, named **component_class** and **material_class**, were defined within the dictionary model of the first edition of this IEC 61360. These subtypes are deprecated, and they are removed from this part of IEC 61360.

NOTE 2 The following changes ensure that the class definitions of a dictionary conforming with the first edition of IEC 61360-2 conforms to this edition: (1) replace **component_class** and **material_class** by **item_class** throughout the reference dictionary; (2) add to each new **item_class** class the **instance_sharable** attribute, the value of which being true; (3) add for each new **item_class** class the optional **hierarchical_position** attribute without setting any value; (4) add for each new **item_class** class the **keywords** attribute, the value of which being an empty collection.

NOTE 3 Another subtype of **item_class**, named **feature_class**, was provided in ISO 13584-24:2003. This subtype is also deprecated and its usage is not allowed in new implementations of this part of IEC 61360.

NOTE 4 The following changes ensure that the class definitions of a dictionary conforming with ISO 13584-25 conforms to this part of IEC 61360: (1) replace **feature_class** by **item_class** throughout the reference dictionary; (2) add to each new **item_class** class the **instance_sharable** attribute, the value of which being false; (3) add for each new **item_class** class the optional **hierarchical_position** attribute without setting any value; (4) add for each new **item_class** class the **keywords** attribute, the value of which being an empty collection.

5.8.2 Structural detail

5.8.2.1 Class_BSU

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

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_BSUs** 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 subclass).

NOTE 1 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_definitions** of the BSUs are required to be available. Therefore, on the receiving system, the **known_visible_properties** attribute contains all properties visible for the class.

known_visible_data_types: the set of **data_type_BSUs** 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 subclass).

NOTE 2 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_types** defined as visible by this super-class do not belong to the **known_visible_data_types** attribute. Only on the receiving system all the **dictionary_definitions** of the BSUs are required to be available. Therefore, on the receiving system, the **known_visible_data_types** attribute contains all **data_types** visible for the class.

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

added_visible_properties: the set of **property_BSUs** that refer to the class as their **name_scope** and that are therefore visible for the class (and any of its subclass).

NOTE 3 Only the **property_BSUs** that belongs to the same exchange context are referenced by this inverse attribute. On the receiving system they may already exist other **property_BSUs** that refer to this class (a PLIB exchange context is never assumed to be complete).

NOTE 4 The **added_visible_properties** attribute will be used to encode the class attribute "Visible properties".

added_visible_data_types: the set of **data_type_BSUs** that refer to the class as their **name_scope** and that are therefore visible for the class (and any of its subclass).

NOTE 5 Only the **data_type_BSUs** that belongs to the same exchange context are referenced by this inverse attribute. On the receiving system they may already exist other **data_type_BSUs** that refer to this class (a PLIB exchange context is never assumed to be complete).

NOTE 6 The **added_visible_data_types** attribute will be used to encode the class attribute "Visible types".

Formal propositions:

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

5.8.2.2 Class_and_property_elements

The **class_and_property_elements** entity captures the attributes that are common to both **classes** and **property_DETs**.

EXPRESS specification:

```
*)
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; -- class_and_property_elements
(*)
```

Attribute definitions:

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.

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".

NOTE 2 The **definition** attribute will be used to encode the property attribute "Definition" and the class attribute "Definition".

NOTE 3 The **source_of_doc_definition** attribute will be used to encode the property attribute "Source document of definition" and the class attribute "Source document of definition".

NOTE 4 The **note** attribute will be used to encode the property and class attribute "Note".

NOTE 5 The **remark** attribute will be used to encode the property and class attribute "Remark".

5.8.2.3 Class

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

EXPRESS specification:

```
*)
ENTITY class
ABSTRACT SUPERTYPE OF( ONEOF (item_class, categorization_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;
    constraints: SET [0:?] OF constraint_or_constraint_id;
    hierarchical_position: OPTIONAL hierarchical_position_type;
    keywords: SET [0:?] OF keyword_type;
    sub_class_properties: SET [0:?] OF property_BSU;
    class_constant_values: SET [0:?] OF class_value_assignment;
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);
    known_property_constraints: SET [0:?] OF property_constraint
        := compute_known_property_constraints(
            [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);
WR5: NOT all_class_descriptions_reachable(
        SELF\dictionary_element.identified_by)
        OR (QUERY (cdp <* described_by
        | (SIZEOF (cdp\basic_semantic_unit.definition)=1)
        AND (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
        +'.DEPENDENT_P_DET') IN TYPEOF
        (cdp\basic_semantic_unit.definition[1]))
        AND NOT
        (cdp\basic_semantic_unit.definition[1].depends_on
        <= known_applicable_properties))=[]);
WR6: check_datatypes_applicability(SELF);
WR7: QUERY (cons <* constraints
        | ('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
        +'.INTEGRITY_CONSTRAINT' IN TYPEOF (cons))
        AND (SIZEOF (cons\property_constraint.constrained_property
        .definition) =1)
        AND NOT correct_constraint_type(
        cons\integrity_constraint.redefined_domain,
        cons\property_constraint.constrained_property
        .definition[1].domain)) = []);
WR8: QUERY (cons <* constraints
        | (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
        +'.CONFIGURATION_CONTROL_CONSTRAINT') IN TYPEOF (cons))
        AND NOT correct_precondition (cons, SELF)) = []);
WR9: NOT all_class_descriptions_reachable(
        SELF\dictionary_element.identified_by)
        OR (QUERY (cons <* constraints
        | (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
        +'.PROPERTY_CONSTRAINT') IN TYPEOF (cons))
        AND NOT
        ((cons\property_constraint.constrained_property
        IN SELF\dictionary_element.identified_by
        \class_BSU.known_visible_properties)
        OR (cons\property_constraint.constrained_property
        IN known_applicable_properties)))=[]);
WR10: (SIZEOF( QUERY (lab <* keywords
        | ('ISO13584_IEC61360_DICTIONARY_SCHEMA'
        +'.LABEL_WITH_LANGUAGE') IN TYPEOF (lab)))
        = SIZEOF( keywords))
        OR (SIZEOF (QUERY (lab <* keywords
        | ('ISO13584_IEC61360_DICTIONARY_SCHEMA'
        +'.LABEL_WITH_LANGUAGE') IN TYPEOF (lab)))
        = SIZEOF( keywords));
WR11: (('ISO13584_IEC61360_ITEM_CLASS_CASE_OF_SCHEMA'
        + '.A_PRIORI_SEMANTIC_RELATIONSHIP')
        IN TYPEOF (SELF)) OR
        ( QUERY(p <* sub_class_properties

```

```
        | NOT(p IN SELF.described_by)) = []);  
WR12: NOT all_class_descriptions_reachable(SELF.identified_by) OR  
      (QUERY(va <* class_constant_values |  
            NOT is_class_valued_property(  
              va.super_class_defined_property, SELF.identified_by)) = []);  
WR13: QUERY(val <* SELF.class_constant_values  
          | QUERY (v <* class_value_assigned (  
              val.super_class_defined_property, SELF.identified_by)  
              | val.assigned_value <> v) <>[]) = [];  
END_ENTITY; -- class  
(*
```

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 products within the class, and any of its subclasses.

NOTE 1 A property may also be applicable to a class when this property is imported from another class through an **a_priori_semantic_relationship** as defined in 8.5 of this part of IEC 61360. Therefore the properties referenced by the **described_by** attribute do not define all the applicable properties for a class.

NOTE 2 The list order is the presentation order of the properties suggested by the supplier.

NOTE 3 A property that is a context dependent property (**context_dependent_P_DET**) may become applicable to a class only if all the context parameters (**condition_DET**) on which its value depends are also applicable to this class. This is stated in where rule 5 (WR5).

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

NOTE 4 A **data_type** may also be applicable to a class when this **data_type** is imported from another class through an **a_priori_semantic_relationship** as defined in 8.5 of this part of IEC 61360. Therefore the data types referenced by the **defined_types** attribute do not define all the applicable data types for a class.

constraints: the set of constraints that restrict the target domains of values of some properties of the class to some subsets of their inherited domains of values.

NOTE 5 Each constraint in the **constraints** attribute should be fulfilled by class instances. Thus the **constraints** attribute is a conjunction of constraints.

hierarchical_position: the coded representation of the class position in a class inclusion hierarchy to which it belongs; a **hierarchical_position** of a class changes when the class structure of an ontology is changed. Thus it cannot be used as a stable identifier for classes.

NOTE 6 This kind of coded name is used in particular in product categorization hierarchies for representing the class inclusion structure through some coding conventions.

EXAMPLE 1 In UNSPSC, *Manufacturing Components and Supplies* has the hierarchical position 31000000, *Hardware* has the hierarchical position 31160000 and *Bolt* the hierarchical position 31161600. By convention, this representation of the hierarchical position allows to represent that *Manufacturing Components and Supplies* is at the first level of the hierarchy, that *Hardware* is at the second level of the hierarchy and is included in *Manufacturing Components and Supplies* and that *Bolt* is at the third level of the hierarchy and is included in *Hardware*.

keywords: a set of keywords, possibly in several languages, allowing to search the class.

sub_class_properties: declares properties as class-valued, i.e. in subclasses one single value will be assigned per class. See 5.9.6.

class_constant_values: assignments in the current class for class-valued properties declared in superclasses. See 5.9.6.

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(es) by their **described_by** attribute and that are therefore applicable to this class (and to any of its subclasses).

NOTE 7 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 system all the **dictionary_definitions** of the **BSUs** are required to be available. Therefore, on the receiving system, 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(es) by their **defined_types** attribute and that are therefore applicable to this class (and to any of its subclasses).

NOTE 8 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_types** defined as applicable by this super-class do not belong to the **known_applicable_data_types** attribute. Only on the receiving system all the **dictionary_definitions** of the **BSUs** are required to be available. Therefore, on the receiving system, the **known_applicable_data_types** attribute contains all the **data_types** that are applicable to a class by virtue of being referenced by a **defined_types** attribute.

known_property_constraints: the **constraints** over a property that are referenced by the class or any of its known is-a superclass by their **constraints** attribute, or, in case of a class that is a subtype of **a_priori_semantic_relationship**, by its **referenced_constraints** 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.

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 context dependent properties (**dependent_P_DET**) whose all context parameters (**condition_DET**) are applicable in to class may become applicable for this class by virtue of being referenced by its **described_by** attribute.

WR6: 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 9 The **its_superclass** attribute will be used to encode the class attribute "Superclass".

NOTE 10 The **described_by** attribute provides the encoding for the "Applicable Properties" of a class.

NOTE 11 The **defined_types** attribute is used to encode the "Applicable Types" attribute of a class.

WR7: the set of constraints that are property constraints shall define restrictions that are compatible with the domain of values of the properties to which they apply.

WR8: all the properties referenced in the precondition of a **configuration_control_constraint** shall be applicable to the class.

WR9: all the properties referenced in the **constraint** attribute shall be either visible or applicable to the class.

WR10: either all **keywords** are represented as **label_with_languages** or all are represented as **labels**.

WR11: if the **class** is not an **a_priori_semantic_relationship**, the **sub_class_properties** shall belong to the **described_by** list.

NOTE 12 Through an **a_priori_semantic_relationship**, **sub_class_properties** may also be imported.

WR12: the properties referenced in **class_constant_values** are declared as class-valued in some superclass of the current class, or in the current class itself.

NOTE 13 The **sub_class_properties** attribute of the **class** entity is used to encode the "Class valued properties" attribute for classes.

NOTE 14 The **class_constant_values** attribute of the **class** entity is used to encode the "Class constant values" for classes.

WR13: if a property referenced in **class_constant_values** was already assigned a value in a superclass, the value assigned in the current class should be the same.

Informal propositions:

IP1: if all the is-a superclasses of the class are available, then the **known_property_constraints** are all the constraints that apply to the properties that are connected to the class either as visible or as applicable properties.

5.8.3 Item_class

The entity **item_class** enables the modeling of any type of entity of the application domain that may be captured by a characterization class defined by a class structure and a set of properties. In particular, both instances of products and instances of particular aspects of products represented as features, are mapped onto **item_class**.

The **item_class** entity includes an **instance_sharable** attribute that specifies the conceptual status of an item. If this attribute is true, then each instance represents an independent item, otherwise it is a feature, i.e., a dependent item that has to be component of another item. This does not prescribe any specific implementation at the data representation level.

EXAMPLE The *head of a screw* is a feature described by a number of properties but that may only exist when referenced by a screw. It is represented as an **item_class** with the **instance_sharable** attribute equal to *false*.

NOTE 1 Two subtypes of **item_class**, named **component_class** and **material_class**, were defined within the dictionary model of the first edition of ISO 13584-42 and in IEC 61360-2. These subtypes are deprecated, and they are removed from this edition of IEC 61360.

NOTE 2 Another subtype of **item_class**, named **feature_class**, was provided in ISO 13584-24:2003. This subtype is also deprecated and its usage is not recommended in new implementations of this part of IEC 61360.

EXPRESS specification:

```
*)  
ENTITY item_class  
SUBTYPE OF(class);  
    simplified_drawing: OPTIONAL graphics;  
    coded_name: OPTIONAL value_code_type;
```

```

        instance_sharable: OPTIONAL BOOLEAN;
    END_ENTITY; -- item_class
    (*

```

Attribute definitions:

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

NOTE 3 The **simplified_drawing** attribute of the **item_class** entity is used to encode the "Simplified Drawing" attribute for classes.

coded_name: may be used as a class constant value to characterize the class in the value domain of a **sub_class_properties** of its superclass.

NOTE 4 This attribute is not used in ISO 13584 series. It is only used in IEC 61360 series.

instance_sharable: when false, it specifies that instances of the **item_class** are features; when not provided or true it specifies that instances of the **item_class** are stand-alone items.

NOTE 5 In the common ISO13584/IEC61360 dictionary model, it is implementation dependent to decide whether several real world instances of features modeled by the same set of property-values pairs are represented by several EXPRESS pieces of data or by the same piece of data in the data exchange file. Thus, an instance of an **item_class** whose **instance_sharable** equals *false* and that is referenced by several instances of **item_classes** at the data model level is interpreted as several real world instances of the same feature.

5.8.4 Categorization_class

The **categorization_class** entity enables the modeling of a grouping of a set of objects that constitutes an element of a categorization.

EXAMPLE 1 Manufacturing components and supplies, industrial optics, are example of product categorization class defined in UNSPSC.

Neither properties nor datatypes, nor constraints are associated, as visible or applicable, with such a class. Moreover, **categorization_classes** may not be related to each other by the is-a inheritance relationship, but they may only be related to each other through the is-case-of class relationship. A specific attribute, called **categorization_class_superclasses**, allows to record the **categorization_classes** that are superclasses of a **categorization_class** in a case-of hierarchy.

NOTE Using the case-of resource constructs, **item_classes** may also be connected to **categorization_classes**.

EXAMPLE 2 The following example shows how characterization classes and categorization classes may be connected to achieve some particular goals. A ball bearing supplier wants to design its own ontology and to make it easy to retrieve and easy to use. To achieve these goals, he/she wants to use standard properties and to be connected to standard classifications. The supplier provides only ball bearings, but some bearings are sealed, some others are not. Particular properties may be associated with sealed bearings and with not sealed bearings, but these categories do not exist as classes in standard bearing ontologies. Thus, the bearing supplier processes as follows. (1) He/she designs a proprietary ontology consisting of three characterization classes: *my_bearing*, *my_sealed_bearings*, *my_non_sealed_bearing*. The two latter are connected to the former by the is-a inheritance relationship, and all the properties assigned to the former are inherited by the latter. (2) To use some of the properties defined in the ISO/TS 23768-1 (to be published), the bearing supplier specifies that his/her class *my_bearings* is case-of the standard bearing class *ball bearing* defined in ISO/TS 23768-1. Through this case-of relationship, he/she may import in his/her class *my_bearings* the standard-defined properties: *bore diameter*, *outside diameter*, *ISO tolerance class*. Moreover, he/she creates those needed properties that are not defined in the standard. (3) To facilitate the retrieval of the server that display the supplier's catalogue, he/she represents a small fragment of the UNSPSC classification, and a case-of relationship between the UNSPSC class *ball_bearings* and its own class *my_bearing*. The result is presented in Figure 8 below:

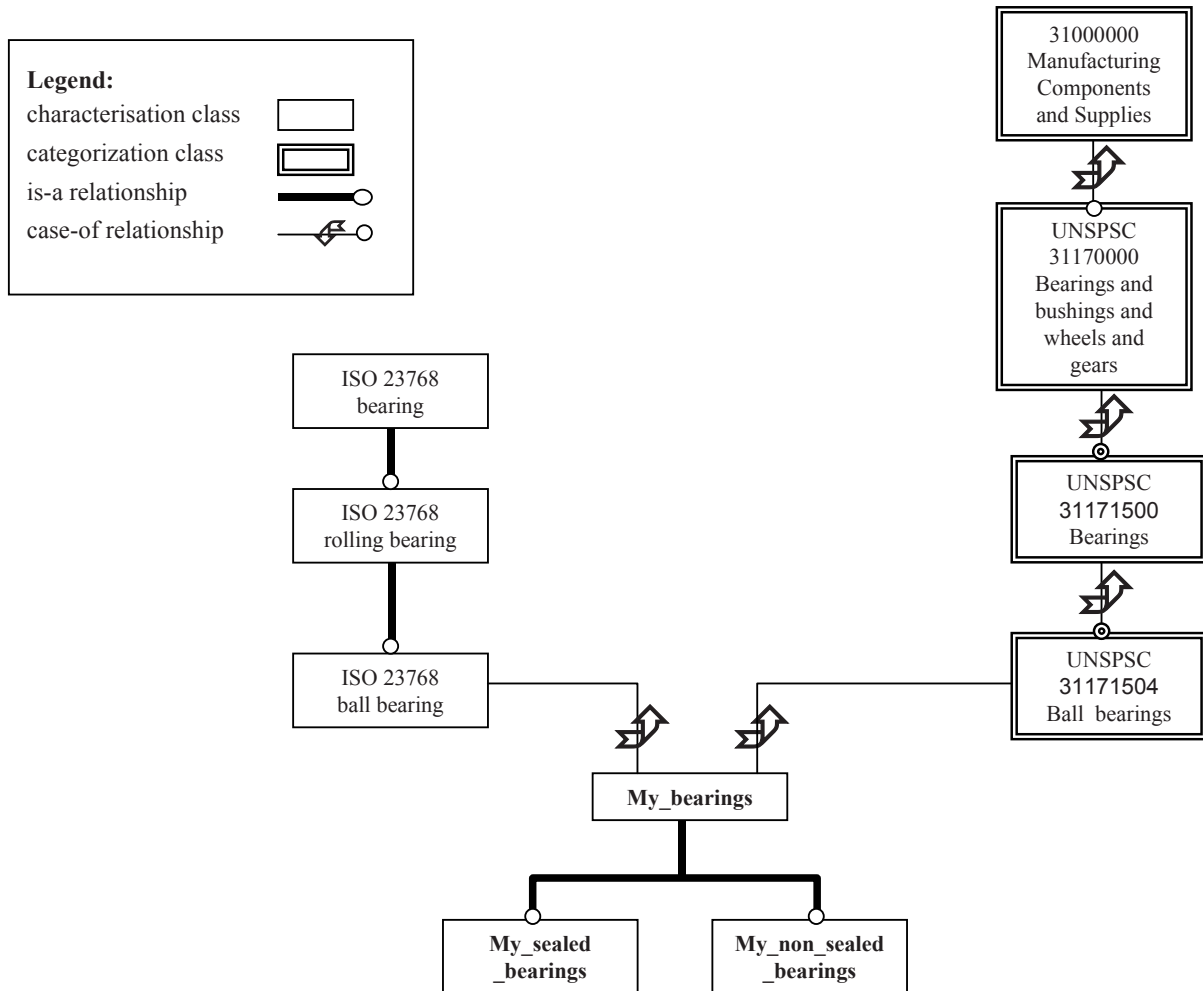


Figure 8 – Example of a supplier ontology

EXPRESS specification:

```

*)
ENTITY categorization_class
SUBTYPE OF(class);
    categorization_class_superclasses: SET [0:?] of class_BSU;
WHERE
    WR1: QUERY (cl <* SELF. categorization_class_superclasses
        | NOT (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
        +'.CATEGORIZATION_CLASS') IN TYPEOF(cl.definition[1])))
        = [];
    WR2: NOT EXISTS(SELF\class.its_superclass);
    WR3: SIZEOF(SELF\class.described_by) = 0;
    WR4: SIZEOF(SELF\class.defined_types) = 0;
    WR5: SIZEOF(SELF\class.constraints) = 0;
    WR6: SIZEOF(compute_known_visible_properties
        (SELF\dictionary_element.identified_by)) = 0;
    WR7: SIZEOF(SELF\class.sub_class_properties) = 0;
    WR8: SIZEOF(SELF\class.class_constant_values) = 0;
    WR9: SIZEOF(SELF\class.identified_by.known_visible_properties)
        = 0;
    WR10: SIZEOF(SELF\class.identified_by.known_visible_data_types)
        = 0;
  
```

```
END_ENTITY; -- categorization_class
(*
```

Attribute definitions:

categorization_class_superclasses: the **categorization_classes** that are one step above the categorization class in a case-of class hierarchy.

Formal propositions:

WR1: only **categorization_classes** may appear as superclasses of a **categorization_class**.

WR2: a **categorization_class** shall not have is-a superclass.

WR3: no property shall be associated with a **categorization_class**.

WR4: no datatype shall be associated with a **categorization_class**.

WR5: no constraint shall be associated with a **categorization_class**.

WR6: a **categorization_class** shall not be the property definition class of any property.

WR7: no subclass property shall be associated with a **categorization_class**.

WR8: no class constant value shall be associated with a **categorization_class**.

WR9: no visible property shall be associated with a **categorization_class**.

WR10: no visible datatype shall be associated with a **categorization_class**.

5.9 Data element type / properties data

5.9.1 General

This clause contains definitions for the dictionary data for properties.

5.9.2 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
      + sep_id + dic_identifier;
  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]))= []);
END_ENTITY; -- property_BSU
(*
```

Attribute definitions:

code: to allow for unique identification of the property over all ontologies defined by the same **name_scope.defined_by** supplier.

name_scope: 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 product.

Formal propositions:

WR1: any class referenced by the **describes_classes** attribute of a **property_BSU** either is the class referenced by its **name_scope** attribute, or is a subclass of this class.

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.

5.9.3 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:?] 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; -- property_DET
(*
```

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 80000/IEC 80000 (formerly 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 product.

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

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

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

NOTE 4 The **domain** attribute is used as a starting point for the encoding of the property attribute "Data Type". 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.

Figure 9 presents a planning model of the data associated with **property_DETs**.

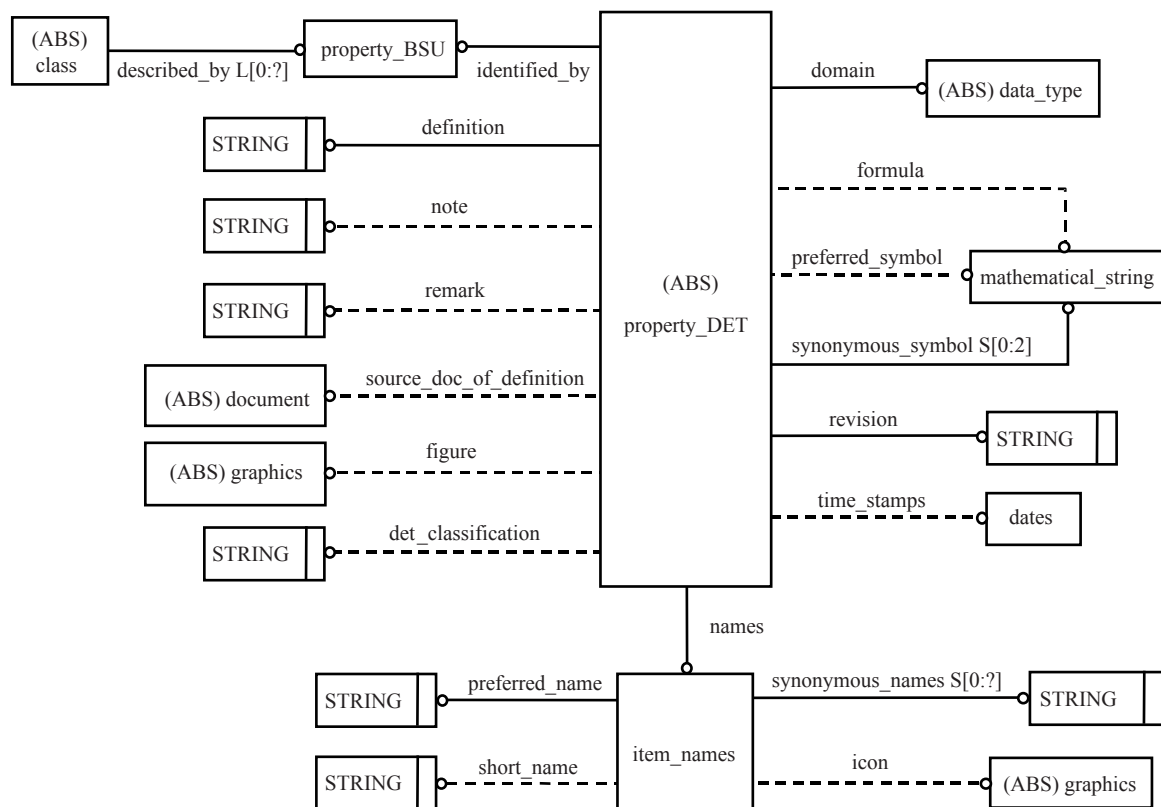


Figure 9 – Overview of property data element type data and relationships

5.9.4 Condition, dependent and non-dependent Data Element Types

Figure 10 depicts the various kinds of Data Element Types in the format of a planning model.

Note that Figure 10 is simplified: the "**depends_on**" relation essentially is implemented with a BSU reference, but a constraint is specified that the referred-to **property_DET** shall be a **condition_DET** (see entity **dependent_P_DET**).

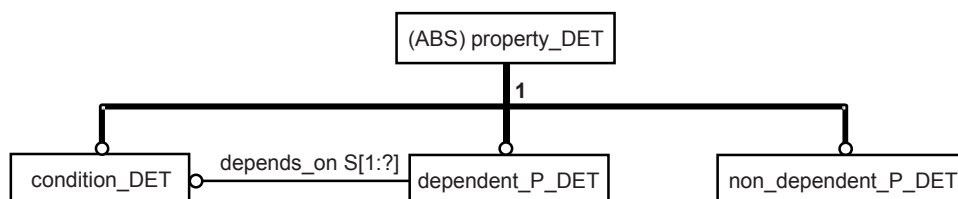


Figure 10 – Kinds of data element types