

DIN EN 61360-2

ICS 29.100.20; 31.020; 35.040

Einsprüche bis 2008-05-31
Vorgesehen als Ersatz für
DIN EN 61360-2:2004-12

Entwurf**Genormte Datenelementtypen mit Klassifikationsschema für elektrische Bauteile –****Teil 2: EXPRESS-Datenmodell (IEC 3D/159/CDV:2007);****Englische Fassung prEN 61360-2:2007**

Standard data element types with associated classification scheme for electric components –

Part 2: EXPRESS dictionary schema (IEC 3D/159/CDV:2007);

English version prEN 61360-2:2007

Types normalisés d'éléments de données avec plan de classification pour composants électriques –

Partie 2: Schéma d'un dictionnaire EXPRESS (CEI 3D/159/CDV:2007);

Version anglais prEN 61360-2:2007

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2008-03-31 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise als Datei per E-Mail an dke@din.de in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (Hausanschrift: Stresemannallee 15, 60596 Frankfurt am Main).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevante Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 131 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Nationales Vorwort

Die englische Originalfassung des europäischen Dokuments prEN 61360-2:2007 „Genormte Datenelementtypen mit Klassifikationsschema für elektrische Bauteile – Teil 2: EXPRESS-Datenmodell“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 3D/159/CDV:2007 „Standard data element types with associated classification scheme for electric components – Part 2: EXPRESS dictionary schema“ unverändert in den Entwurf prEN 61360-2:2007 übernommen worden.

Das internationale Dokument wurde vom SC 3D „Data sets for libraries“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Es ist vorgesehen, auch bei der entsprechenden zukünftigen Deutschen Norm auf die deutsche Sprachfassung zu verzichten und diese in der englischsprachigen Fassung zu veröffentlichen.

Für diesen Norm-Entwurf ist das nationale Arbeitsgremium K 113 „Produktdatenmodelle, Informationsstrukturen, Dokumentation und graphische Symbole“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Änderungen

Gegenüber DIN EN 61360-2:2004-12 wurden folgende Änderungen vorgenommen:

- a) Ungenauigkeiten und Mehrdeutigkeiten in der Norm wurden beseitigt.
- b) Die Norm wurde mit ISO 13584-42 abgeglichen
- c) Das gemeinsame EXPRESS-Modell wurde bezüglich der in IEC 61360-1 gegebenen elektrotechnischen Spezifikationen eingeschränkt.

CONTENTS

FOREWORD.....	7
INTRODUCTION	
1 General	10
1.1 Scope and object.....	10
1.2 Normative references	10
2 Definitions	12
3 Abbreviations.....	12
4 Overview of the common dictionary schema and compatibility with ISO 13584	13
4.1 Use of the common dictionary schema to exchange IEC 61360-1 compliant data.....	13
4.2 Compatibility with ISO 13584-42	13
4.3 Naming correspondence between IEC 61360-1 and IEC 61360-2	13
4.4 Main structure of the common dictionary schema	14
5 ISO13584_IEC61360_dictionary_schema.....	15
5.1 References to other schemata	15
5.2 Constant definitions	16
5.3 Basic semantic units: defining and using the dictionary	16
5.3.1 Requirements for exchange	16
5.3.2 Three level architecture of the dictionary data	17
5.3.2.1 basic_semantic_unit.....	18
5.3.2.2 dictionary_element	19
5.3.2.3 content_item	20
5.3.3 Overview of basic semantic units and dictionary elements	20
5.3.4 Identification of dictionary elements: three-levels structure	21
5.3.5 Extension possibilities for other types of data	21
5.3.5.1 supplier_related_BSU.....	22
5.3.5.2 class_related_BSU	22
5.3.5.3 supplier_BSU_relationship.....	22
5.3.5.4 class_BSU_relationship.....	22
5.4 Supplier data	23
5.4.1 supplier_BSU.....	23
5.4.2 supplier_element	24
5.5 Class data	24
5.5.1 General	24
5.5.1.1 class_BSU	25
5.5.1.2 class_and_property_elements	27
5.5.1.3 class	27
5.5.2 item_class	30
5.5.3 categorization class	31
5.6 Data element type / properties data.....	32
5.6.1 property_BSU	32
5.6.2 property_DET	33
5.6.3 Condition, dependent and non-dependent data element types	34
5.6.3.1 condition_DET.....	34
5.6.3.2 dependent_P_DET ¹⁸	35
5.6.3.3 non-dependent_P_DET	35

5.6.4	Class-valued properties	35
5.7	Domain data: the type system	36
5.7.1	General	37
5.7.1.1	data_type_BSU	37
5.7.1.2	data_type_element	38
5.7.2	The type system	39
5.7.2.1	data_type	39
5.7.2.2	simple_type	39
5.7.2.3	number_type	39
5.7.2.4	int_type	40
5.7.2.5	int_measure_type	40
5.7.2.6	int_currency_type	41
5.7.2.7	non_quantitative_int_type	42
5.7.2.8	real_type	42
5.7.2.9	real_measure_type	42
5.7.2.10	real_currency_type	44
5.7.2.11	boolean_type	44
5.7.2.12	string_type	44
5.7.2.13	Translatable_string_type	45
5.7.2.14	Non_translatable_string_type	45
5.7.2.15	Remote_http_address	45
5.7.2.16	Date_time_data_type	46
5.7.2.17	Date_data_type	46
5.7.2.18	Time_data_type	47
5.7.2.19	non_quantitative_code_type	47
5.7.2.20	complex_type	48
5.7.2.21	level_type	48
5.7.2.22	level	49
5.7.2.23	class_instance_type	49
5.7.2.24	entity_instance_type	50
5.7.2.25	placement_type	50
5.7.2.26	axis1_placement_type	51
5.7.2.27	axis2_placement_2d_type	51
5.7.2.28	axis2_placement_3d_type	51
5.7.2.29	named_type	52
5.7.3	Values	52
5.7.3.1	value_domain	52
5.7.3.2	value_type	53
5.7.3.3	dic_value	54
5.7.3.4	administrative_data	55
5.7.3.5	Translation_data	56
5.7.4	Extension to ISO 10303-41 unit definitions	56
5.7.4.1	non_si_unit	56
5.7.4.2	assert_ONEOF rule	57
5.7.4.3	dic_unit	57
5.8	Basic type and entity definitions	58
5.8.1	Basic type definitions	58
5.8.1.1	class_code_type	58
5.8.1.2	code_type	58
5.8.1.3	currency_code	58

5.8.1.4	Data_type_code_type	59
5.8.1.5	date_type	59
5.8.1.6	definition_type	60
5.8.1.7	DET_classification_type	60
5.8.1.8	note_type	60
5.8.1.9	pref_name_type	60
5.8.1.10	property_code_type	61
5.8.1.11	remark_type	61
5.8.1.12	revision_type	61
5.8.1.13	short_name_type	62
5.8.1.14	supplier_code_type	62
5.8.1.15	syn_name_type	62
5.8.1.16	dic_unit_identifier	63
5.8.1.17	value_code_type	63
5.8.1.18	value_format_type	64
5.8.1.19	version_type	64
5.8.1.20	Status_type	64
5.8.2	Basic entity definitions	65
5.8.2.1	dates	65
5.8.2.2	document	65
5.8.2.3	graphics	65
5.8.2.4	External_graphics	66
5.8.2.5	Graphic_files	66
5.8.2.6	Identified_document	66
5.8.2.7	item_names	67
5.8.2.8	label_with_language	68
5.8.2.9	mathematical_string	68
5.9	Function definitions	69
5.9.1	acyclic_superclass_relationship function	69
5.9.2	check_syn_length function	70
5.9.3	codes_are_unique function	70
5.9.4	definition_available_implies function	71
5.9.5	is_subclass function	71
5.9.6	string_for_derived_unit function	72
5.9.7	string_for_named_unit function	74
5.9.8	string_for_SI_unit function	74
5.9.9	string_for_unit function	76
5.9.10	all_class_descriptions_reachable function	76
5.9.11	compute_known_visible_properties function	76
5.9.12	compute_known_visible_data_types function	77
5.9.13	compute_known_applicable_properties function	78
5.9.14	compute_known_applicable_data_types function	78
5.9.15	list_to_set	79
5.9.16	Check_properties_applicability	80
5.9.17	Check_datatypes_applicability	80
5.9.17.1	one_language_per_translation function	81
5.9.17.2	allowed_values_code_types function	81
5.9.17.3	allowed_values_integer_types function	82
6	ISO13584_IEC61360_language_resource_schema	82
6.1	ISO13584_IEC61360_language_resource_schema type and entity definitions	83

6.1.1	language_code	83
6.1.2	global_language_assignment	84
6.1.3	present_translations	84
6.1.4	translatable_label	85
6.1.5	translated_label	85
6.1.6	translatable_text	85
6.1.7	translated_text	86
6.2	ISO13584_IEC61360_language_resource_schema function definitions	86
6.2.1	check_label_length function	86
6.2.2	check_text_length function	87
6.3	ISO13584_IEC61360_language_resource_schema rule definition	87
7	ISO13584_IEC61360_class_constraint_schema	88
7.1	Introduction to the ISO13584_IEC61360_class_constraint_schema	89
7.2	ISO13584_IEC61360_class_constraint_schema entity definitions	90
7.2.1	constraint	90
7.2.2	property_constraint	90
7.2.3	configuration_control_constraint	90
7.2.4	filter	91
7.2.5	integrity_constraint	92
7.2.6	context_restriction_constraint	93
7.2.7	domain_constraint	93
7.2.8	subclass_constraint	94
7.2.9	entity_subtype_constraint	94
7.2.10	subset_constraint	94
7.2.11	range_constraint	95
7.2.12	string_size_constraint	96
7.2.13	string_pattern_constraint	96
7.2.14	cardinality_constraint	98
7.3	ISO13584_IEC61360_class_constraint_schema function definitions	98
7.3.1	allowed_enumeration_values function	98
7.3.2	Integer_values_in_range function	99
7.3.3	correct_precondition function	99
7.3.4	correct_constraint_type function	100
7.3.5	Compatible_data_type_and_value function	103
8	ISO13584_IEC61360_item_class_case_of_schema	106
8.1	Introduction to the ISO13584_IEC61360_item_class_case_of_schema	107
8.2	ISO13584_IEC61360_item_class_case_of_schema entity definitions	107
8.2.1	a priori semantic relationship	107
8.2.2	Item_class_case_of	109
8.3	ISO13584_IEC61360_item_class_case_of_schema function definitions	110
8.3.1	Compute_known_property_constraints function	110
8.3.2	Constraints_over function	111
8.4	ISO13584_IEC61360_item_class_case_of_schema rule definition	112
8.4.1	Imported_properties_are_visible_or_applicable_rule rule	112
8.5	Some example data	113
Annex A	(informative) Example Physical File	114
A.1	File header	114
A.2	Supplier data	114
A.3	Root class data	114

A.4	Material data	115
A.5	Component data	116
A.6	Electric / Electronic component data	117
Annex B (informative)	EXPRESS-G diagram.....	119
Annex C (informative)	Partial dictionaries	130
Annex D (informative)	Bibliography.....	131

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**STANDARD DATA ELEMENT TYPES WITH ASSOCIATED
CLASSIFICATION SCHEME FOR ELECTRIC COMPONENTS****Part 2: EXPRESS dictionary schema**

FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
- 6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61360-2 has been prepared by subcommittee 3D; Data sets for libraries, of IEC technical committee 3: information structures, documentation and graphical symbols.

The text of this standard is based on the following documents:

FDIS	Report on voting
3D/xxx/FDIS	3D/XXX/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 3.

The committee has decided that the contents of this publication will remain unchanged until 20xx. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

IEC 61360 consists of the following parts, under the general title Standard data element types with associated classification scheme for electric components:

- Part 1 : Definitions - Principles and methods
- Part 2 : EXPRESS dictionary schema
- Part 3 : Maintenance and validation procedures
- Part 4 : IEC reference collection of standard data element types, component classes and terms.
- Part 5 : Extensions to the EXPRESS dictionary schema¹.

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 [3].

Annexes A, B and C are for information only.

¹ To be published

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:

“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:

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

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 General

1.1 Scope and object

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

This standard provides a formal model for data according to the scope as given in the publications as cited above, 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.

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: STEP part 41: "Fundamentals of Product Description and Support", and can be seen as an extension to that. It allows for the usage of one specific language throughout an exchange context (Physical File) without the overhead introduced when multiple languages are used.

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.

1.2 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: 2004, *Standard data element types with associated classification scheme for electric components - Part 1: Definitions - Principles and methods*.

² All the names that stand for items, formally defined within the EXPRESS model, are presented in **bold face**.

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

ISO 31 (all parts), *Quantities and Units*

ISO 639: 1988, *Code for the representation of names of languages*

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

ISO 4217: 1995, *Codes for the representation of currencies and funds.*

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

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

ISO 8601: 2000, *Data elements and interchange formats - Information interchange - Representation of dates and times.*

ISO/IEC 8824-1 2002: —, *Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of Basic Notation.*

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

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

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

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

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

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

ISO 10303-42: 2000, *Industrial automation systems and integration - Product data representation and exchange - Part 42: Integrated generic resources: Geometric and topological representation.*

ISO 12083: 1994, *Information and documentation - Electronic manuscript preparation and markup.*

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

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

ISO/IEC, *International Classification of Standard (ICS).*

2 Definitions

For the purpose of this part of IEC 61360 the following definitions apply:

2.1

basic semantic unit (BSU)

entity that provides an absolute and universal identification of certain objects of the application domain (for example classes, data element types).

2.2

dictionary element

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

2.3

common dictionary schema

information model for a dictionary, using the information modelling language EXPRESS.

2.4

data type

set of allowed values of a data element type.

NOTE - Within IEC the **data_type** that is either a unit of measure or a value domain is defined separately for each data element type.

2.5

IEC root class

class that is the superclass of all the classes defined in IEC 61360-4; its class code is 'AAA000' and its version is '001'.

2.6

applicable data element type

data element type that is defined for some component class and that applies to any component that belongs to this component class.

2.7

visible data element type

data element type that is defined for some component class and that may or may not apply to the different components of this component class.

NOTE 1 - The code of the class where a data element type is defined as visible is part of the identification of this data element type.

NOTE 2 - Within IEC all data element types are defined as visible at the level of the root class, that is the superclass of both the component class and the material class.

2.8

item

a thing whose description can be captured by a class structure and a set of properties.

3 Abbreviations

In this part of IEC 61360 the following abbreviations are used:

– BSU: basic semantic unit;

- DET: data element type;
- ICS: international classification of standards;
- MathML: The XML schema for mathematical notations;
- SI: international system of units;
- UNSPSC: The classification of products and services defined by the United Nations.

4 Overview of the common dictionary schema and compatibility with ISO 13584

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

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

- a) the ISO 13584 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 is **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 compliant data, only SI shall be used for quantitative data element types.

4.2 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 class³ is intended to contain all the entities, types and associated constraints defined in the common dictionary schema. An ISO 13584 compliant implementation conform to this conformance class shall therefore be able to support IEC data that belongs to conformance class 1 of this part of IEC 61360.

4.3 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**).

³ This conformance class is defined as conformance class 0 in ISO 13584-24.

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 - X-REFERENCE table

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

4.4 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.

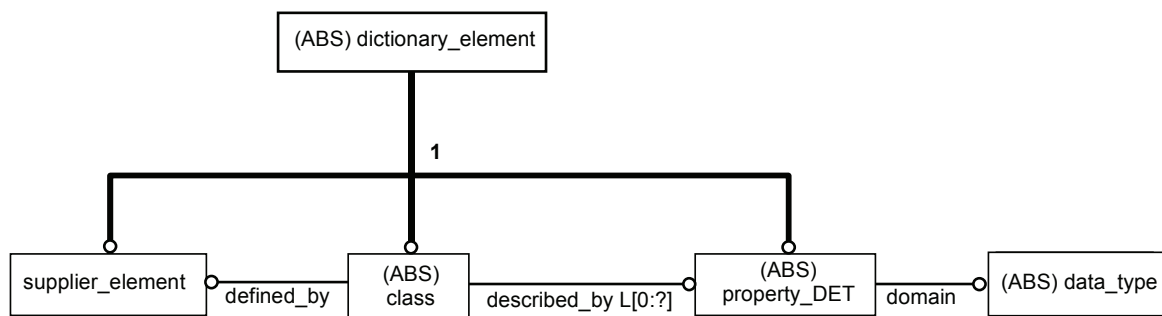


Figure 1 - Overview of the Dictionary Schema

5 ISO13584_IEC61360_dictionary_schema

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

EXPRESS specification:

```

*)
SCHEMA ISO13584_IEC61360_dictionary_schema;
(*)

```

5.1 References to other schemata

This subclause contains references to other EXPRESS schemata which are used in the Dictionary Schema. Their source is indicated in the respective comment.

EXPRESS specification:

```

*)

REFERENCE FROM support_resource_schema(identifier, label, text);
  (* from ISO 10303-41: STEP Part 41: "Fundamentals of Product
    Description and Support" *)
REFERENCE FROM person_organization_schema(organization, address);
  (* from ISO 10303-41: STEP Part 41: "Fundamentals of Product
    Description and Support" *)
REFERENCE FROM measure_schema;
  (* from ISO 10303-41: STEP Part 41: "Fundamentals of Product
    Description and Support" *)
REFERENCE FROM ISO13584_IEC61360_language_resource_schema;
  (* see clause 6 "ISO13584_IEC61360_language_resource_schema" *)
REFERENCE FROM ISO13584_IEC61360_class_constraint_schema;
  (* see clause 7 " ISO13584_IEC61360_class_constraint_schema " *)
REFERENCE FROM ISO13584_IEC61360_item_class_case_of_schema;
  (* see clause 8 " 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);
    (* from ISO 13584-24:2003, PLIB Part 24: "Logical model of supplier
       library" *)
    (*

```

5.2 Constant definitions

This subclause contains constant definitions used later in clause 5.8 (Basic type and entity definitions).

EXPRESS specification:

```

    *)
    CONSTANT
        property_code_len: INTEGER := 14;
        class_code_len: INTEGER := 14;
        data_type_code_len: INTEGER := 14;
        supplier_code_len: INTEGER := 70;
        version_len: INTEGER := 9;
        revision_len: INTEGER := 3;
        value_code_len: INTEGER := 18;
        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;
    (*

```

5.3 Basic semantic units: defining and using the dictionary

5.3.1 Requirements for exchange

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

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

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

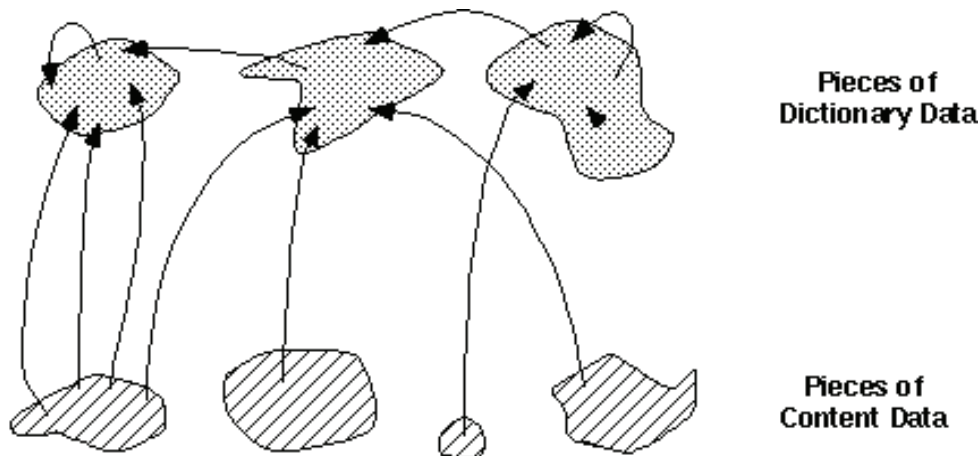


Figure 2 - Pieces of data with relationships

5.3.2 Three level architecture of the dictionary data

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

Assume some piece of content (**content_item**) wants to refer a certain dictionary description, for example to convey the value of a property of a component. It does this by referring to a basic semantic unit through the attribute **dictionary_definition**.

A dictionary description (**dictionary_element**) refers to a basic semantic unit through the attribute **identified_by**. From the correspondence of the absolute identifiers of the basic semantic units this indirect relation is established.

Note that:

- both dictionary element and content item can be present in the same Physical File, but need not be;
- the dictionary element does not need to be present for the exchange of some content item referring to it. In this case it is assumed to be present in the dictionary of the target system already. Conversely, dictionary data can be exchanged without any content data;
- the basic semantic unit can be one single instance in the case where both dictionary element and content item instances are in the same Physical file;
- the same mechanism applies also to references between various dictionary elements (for example between a component class and the associated **property_DETs**).

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

Figure 3 outlines the implementation of this general mechanism.

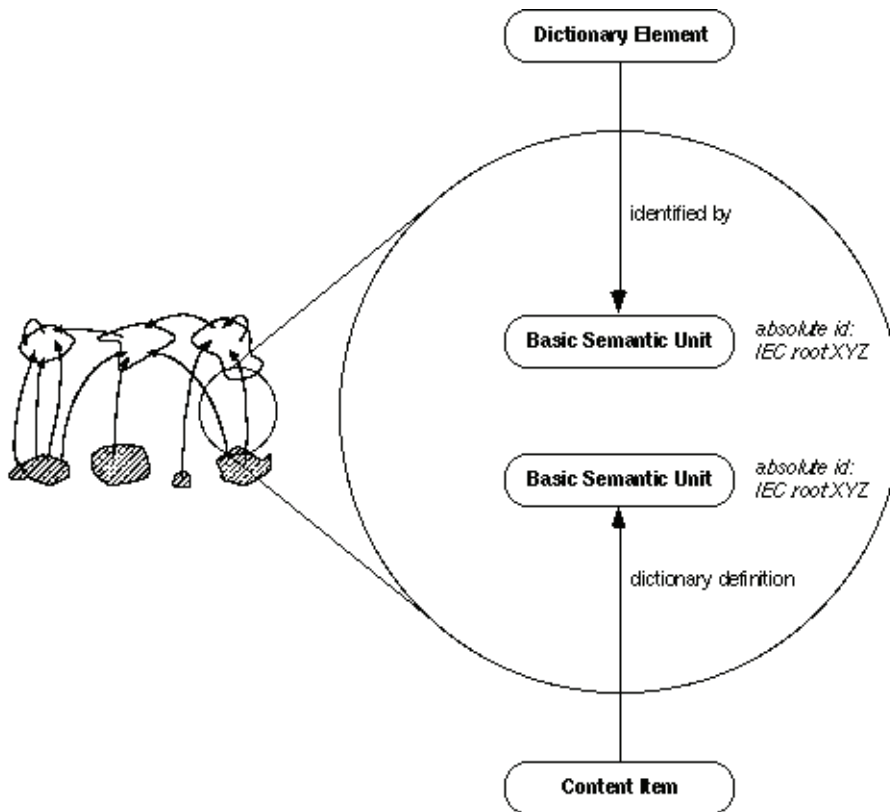


Figure 3 - Implementation of "inter-piece" relationships using basic semantic units

5.3.2.1 basic_semantic_unit

A **basic_semantic_unit** is a unique identification of a **dictionary_element**.

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:

code⁴: the code assigned to identify a certain dictionary element.

version⁴: the version number of a certain dictionary element.

dic_identifier⁴: 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.

5.3.2.2 dictionary_element

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

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

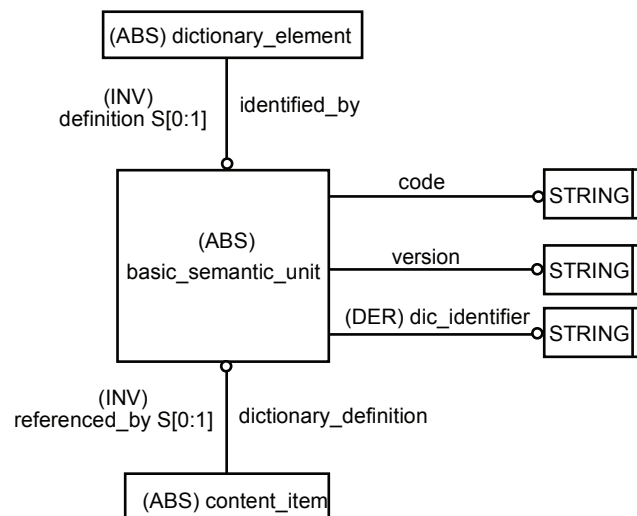


Figure 4 - Relationship between basic semantic unit and dictionary element

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,

```

⁴ See relevant clause in IEC 61360-1

```

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 ;
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.

NOTE 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 5.8.2).

revision⁵: the revision number of this dictionary element.

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

NOTE 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.

5.3.2.3 content_item

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

EXPRESS specification:

```

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

```

Attribute definitions:

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

5.3.3 Overview of basic semantic units and dictionary elements

For every kind of dictionary data a pair of **basic_semantic_unit** and **dictionary_element** subtypes shall be defined. Figure 5 outlines, as a planning model, the basic semantic units 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.

⁵ See relevant clause in IEC 61360-1

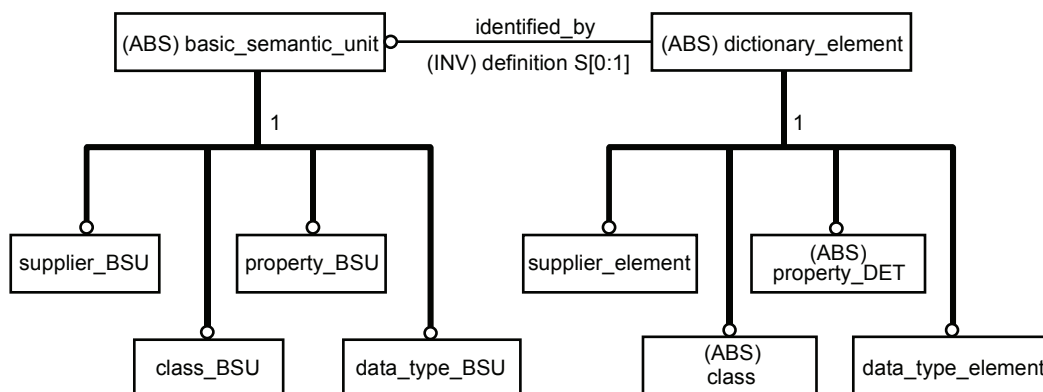


Figure 5 - Current BSUs and dictionary elements

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

- for suppliers see 5.4 "Supplier data";
- for classes see 5.5 "Class data";
- for properties / data element types see 5.6 "data element type / properties data";
- for data types see 5.7 "Domain data: the type system".

5.3.4 Identification of dictionary elements: three-levels structure

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

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

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

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

This identification scheme is described formally in the ..._BSU entities in 5.3 through 5.6, attribute **absolute_id**.

5.3.5 Extension possibilities for other types of data

The BSU - dictionary element mechanism is very general and not limited to the four kinds of data used here (see figure 5). This subclause specifies some facilities which allow for extensions for other kinds. Depending on whether the scope of the identifier is given by a class or a supplier, the corresponding _related_BSU entity has to be subtyped. It is necessary to redefine the **identified_by** attribute of the entity **dictionary_element** (as is done in 5.4 through 5.7 for the current kinds of data).

5.3.5.1 supplier_related_BSU

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

EXPRESS specification:

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

5.3.5.2 class_related_BSU

The **class_related_BSU** provides for the dictionary elements to be associated with classes, for 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
(*
```

5.3.5.3 supplier_BSU_relationship

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

EXPRESS specification:

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

Attribute definitions:

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

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

5.3.5.4 class_BSU_relationship

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

EXPRESS specification:

```
*)
ENTITY class_BSU_relationship
ABSTRACT SUPERTYPE;
    relating_class: class;
```

```

    related_tokens: SET [ 1 : ? ] OF class_related_BSU;
END_ENTITY; -- class_BSU_relationship
(*)

```

Attribute definitions:

relating_class: the class which identifies the dictionary element.

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

5.4 Supplier data

This subclause 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.

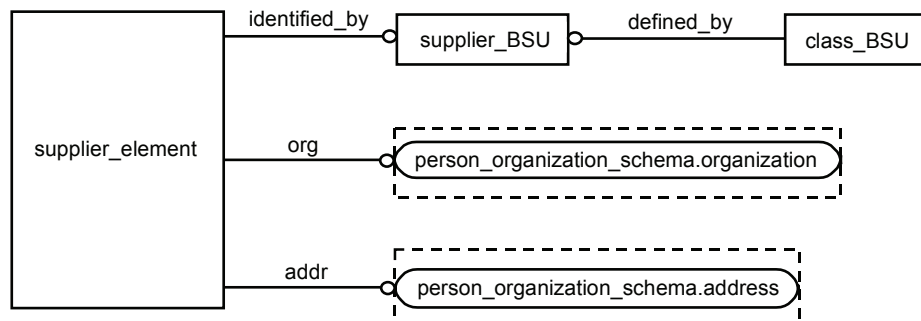


Figure 6 - Overview of supplier data and relationships

5.4.1 supplier_BSU

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

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 := '001';
    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 001.

absolute_id: the absolute identification of the supplier.

Formal propositions:

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

5.4.2 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 (for example program library in ISO 13584).

5.5 Class data

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

5.5.1 General

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

As indicated in figure 7 with the **its_superclass** attribute, classes form an inheritance tree. It is important to note that throughout this standard the terms "inheritance" and "to inherit" stand for this relationship between classes (defined in the dictionary), although EXPRESS has an inheritance concept, too. These shall be clearly distinguished to avoid misunderstandings.

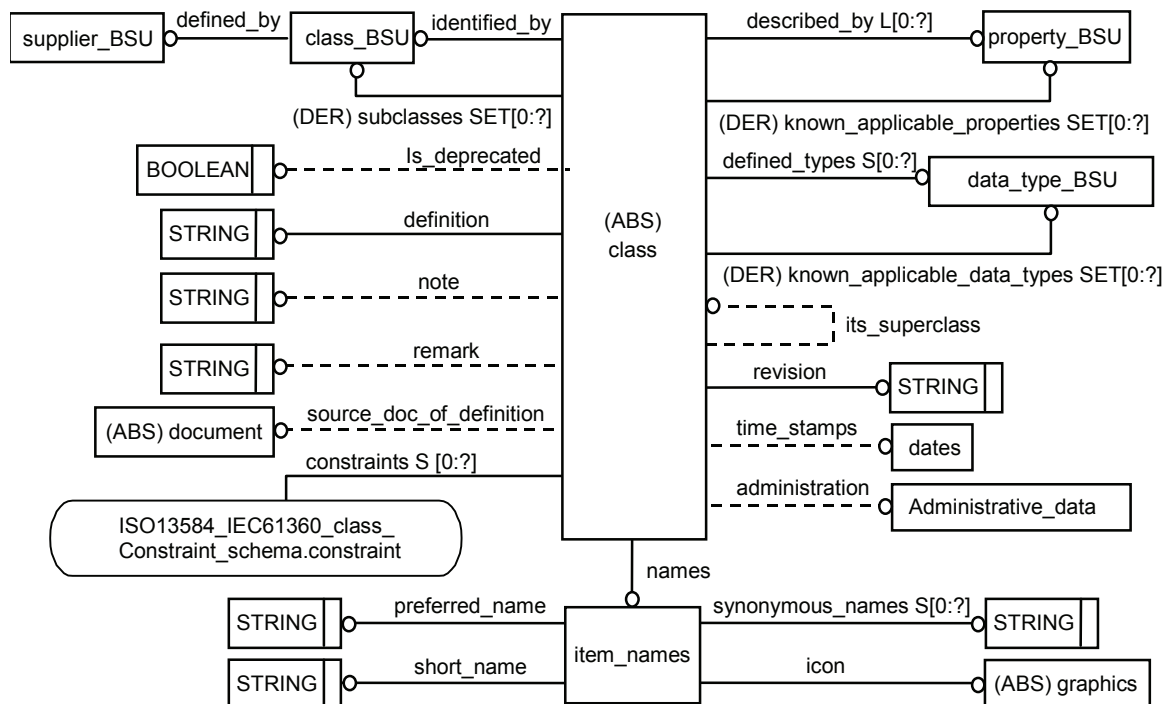


Figure 7 - Overview of class data and relationships

The dictionary data for component classes (as shown in figure 7) is spread over four inheritance levels:

- **class_and_property_elements** defines data common to both classes and **property_DETs**;
- **class** allows for other kinds of classes to be specified later;

EXAMPLE Other subtypes of class, in particular **functional_view_class** and **functional_view_class** are specified in ISO13584-24. They do not characterize parts, but they provide for exchanging particular representations of part, e. g., geometrical representations.

- **item_class** and **categorization_class** are the entities that hold data of different classes of application domain objects.

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 IEC 61360-2. These subtypes are deprecated, and they are removed from Edition 2 of this part of ISO 13584.

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 ISO 13584-42 and IEC 61360-2.

5.5.1.1 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_properties⁶: the set of **property_BSUs** that refer to the class as their **name_scope** attribute or to any known superclass of this class and that are therefore visible for the class (and any of its subclass).

known_visible_data_types⁷: the set of **data_type_BSUs** that refer to the class as their **name_scope** attribute or to any known superclass of this class and that are therefore visible for the class (and any of its subclasses).

subclasses: the set of classes specifying this class as their superclass.
added_visible_properties⁸: 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).

added_visble_properties: the set of **propertyi_BSUs** that refer to the class as their **name_scope** and that are therefore visible for the class (and any of its sub-class)

added_visible_data_types⁹: 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).

Formal propositions:

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

⁶ Within IEC all the data element types refer to the IEC root at their **name_scope** attribute. Therefore all the data element types defined within IEC are visible for every IEC class.

⁷ The capability to define visible data types that may be re-used for different data element types is not used in the current version of IEC 61360. Therefore the **added_visible_data_types** attribute is empty for all classes.

⁸ Within IEC all the data element types refer to the IEC root at their **name_scope** attribute. Therefore the **added_visibility_properties** attribute is empty for all the other classes.

⁹ The capability to define **data_types** that may be re-used for different data element types is not used in the current version of IEC 61360.

5.5.1.2 class_and_property_elements

The **class_and_property_elements** entity captures the attributes which 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:

names¹⁰: the names describing this dictionary element.

definition¹⁰: the text describing this dictionary element.

source_doc_of_definition⁹: the source document of this textual description.

note⁹: further information on any part of the dictionary element, which is essential to the understanding.

remark⁹: explanatory text further clarifying the meaning of this dictionary element.

5.5.1.3 class

The **class**⁹ 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 ;
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);
```

¹⁰ See relevant clause in IEC 61360-1

```

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);
    WR4: check_properties_applicability(SELF);
    WR5: check_datatypes_applicability(SELF);
    WR6: 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)) = []);
    WR7: QUERY (cons <* constraints
        | (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
        + '.CONFIGURATION_CONTROL_CONSTRAINT') IN TYPEOF (cons))
        AND NOT correct_precondition (cons, SELF)) = [];
END_ENTITY; -- class
(*

```

Attribute definitions:

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

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

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

defined_types¹¹: the set of references to the types that can be used for various **property_DETs** throughout the inheritance tree descending from this 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.

¹¹ The capability to define **data_types** that may be re-used for different data element types is not used in the current version of IEC 61360.

subclasses: the set of classes specifying this class as their superclass¹².

known_applicable_properties¹³: the **property_BSUs** that are referenced by the class or any of its known superclass by their **described_by** attribute and that are therefore applicable to this class (and to any of its subclass).

known_applicable_data_types¹⁴: the **data_type_BSUs** that are referenced by the class or any of its known superclass by their **defined_types** attribute and that are therefore applicable to this class (and to any of its subclass).

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_property_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 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.

WR6: 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.

WR7: all the properties referenced in the precondition of a **configuration_control_constraint** shall be applicable to the class.

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.

IP2: If all the is-a superclasses of the class are available, then the **known_data_type_constraints** are all the constraints that apply to the data types that are connected to the class either as visible or as applicable properties.

¹² According to IEC 61360-1 a class shall have 0 or more than 1 subclass .

¹³ According to IEC 61360-1 the data element types that apply to some component class apply also to all component classes at lower levels. This attribute gathers the data element types that apply to a class either by virtue of belonging to the **described_by** attribute or by inheritance.

¹⁴ This attribute gathers the **data_types** that may be referenced from a component class either by virtue of belonging to the **described_by** attribute or by inheritance. Note that the capability to define **data_types** that may be re-used for different data element types is not used in the current version of IEC 61360.

5.5.2 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 parts and instances of particular aspects of parts represented as features, are mapped onto **item_class**.

The **item_class** entity includes an **is_sharable** attribute that specifies the conceptual status of an item. If this attribute is true, then the 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 **is_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 IEC 61360-2. These subtypes are deprecated, and they are removed from Edition 3 of this part 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 the ISO 13584 and IEC 61360-2 series.

EXPRESS specification:

```

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

```

Attribute definitions:

simplified_drawing: optional drawing (**graphic**) that can be associated to the described class.

sub_class_properties¹⁵: declares properties as class-valued, that is in subclasses one single value will be assigned per class. See 5.6.4 "Class-valued properties".

class_constant_values¹⁵: assignments in the current class for class-valued properties declared in superclasses. See 5.6.4 "Class-valued properties".

coded_name¹⁵: to be used in the value domain of the Classifying DET of the superclass.

¹⁵ See relevant clause in IEC 61360-1

is_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 An instance of an **item_class** whose **is_sharable** equals *false* and that is referenced by several instances of **item_classes** at the data model level represents several real world instances of the corresponding feature.

Formal propositions:

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

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

5.5.3 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 Manufacturing Components and Supplies, Industrial optics, are example of product categorization class defined in UNSPSC

Neither properties nor datatypes, or constraints are associated 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 inclusion relationship. A specific attributes, 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**.

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;
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 inclusion 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.

5.6 Data element type / properties data

This clause contains definitions for the dictionary data for properties.

5.6.1 property_BSU

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

EXPRESS specification:

```
*)
ENTITY property_BSU
SUBTYPE OF(basic_semantic_unit);
    SELF\basic_semantic_unit.code: property_code_type;
    name_scope: class_BSU;
DERIVE
    absolute_id: identifier :=
        name_scope.defined_by.absolute_id
        + sep_id + name_scope.dic_identifier
        + 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 within the scope indicated by the **name_scope** attribute.

name_scope¹⁶: 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_classes¹⁷: the classes declaring this property as available for use in the description of a part.

¹⁶ Within IEC 61360 all the data element types refer to the IEC root class by their **name_scope** attribute. The **dic_identifier** of the IEC root class is AAA000-001.

¹⁷ See relevant clause in IEC 61360-1

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 it is a subclass of this class.

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

5.6.2 property_DET

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

EXPRESS specification:

```

*)
ENTITY property_DET
ABSTRACT SUPERTYPE OF (ONEOF (
    condition_DET, dependent_P_DET, non_dependent_P_DET))
SUBTYPE OF (class_and_property_elements);
    SELF\dictionary_element.identified_by: property_BSU;
    preferred_symbol: OPTIONAL mathematical_string;
    synonymous_symbols: SET [0:?] 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_symbol¹⁶: a shorter description of this property.

synonymous_symbols¹⁶: synonymous for the shorter description of the property.

figure¹⁶: an optional **graphic** which describes the property.

det_classification¹⁶: the ISO 31 class for this property.

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

formula¹⁶: a mathematical expression for explaining the property.

describes_classes¹⁷: the classes declaring this property as available for use in the description of a part.

Figure 8 presents a planning model of the data associated with **property_DET**s.

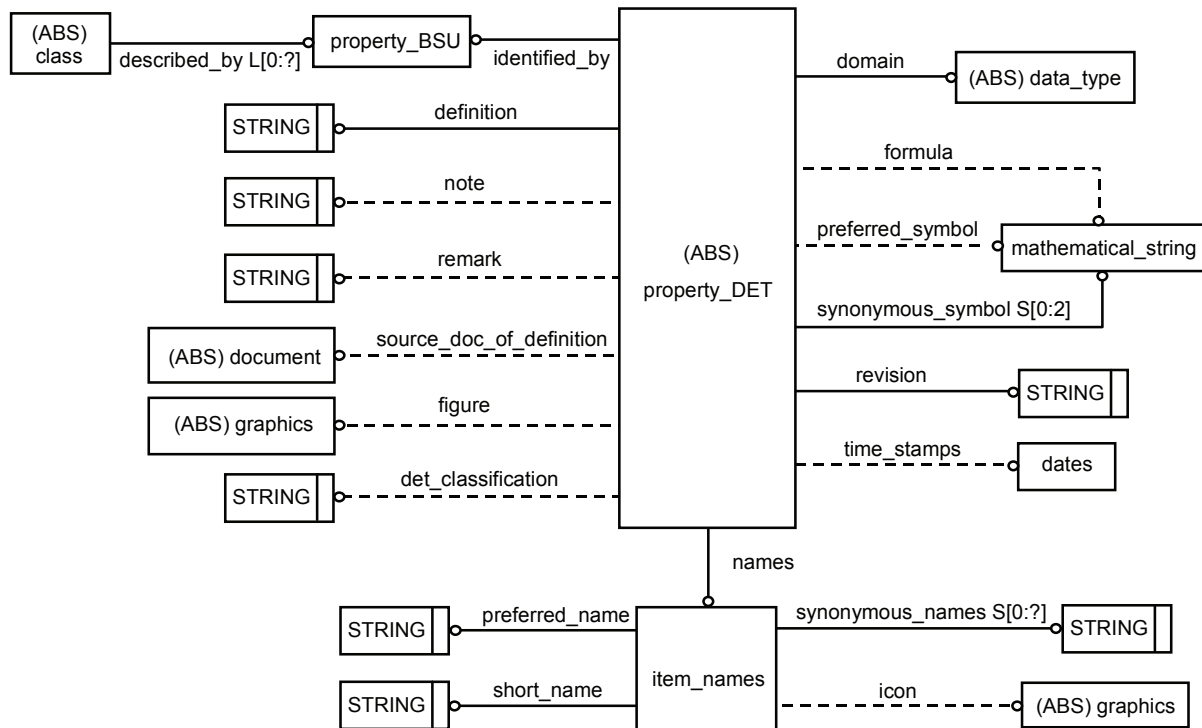


Figure 8 - Overview of property data element type data and relationships

5.6.3 Condition, dependent and non-dependent data element types

Figure 9 depicts the various kinds of data element types in the format of a planning model.

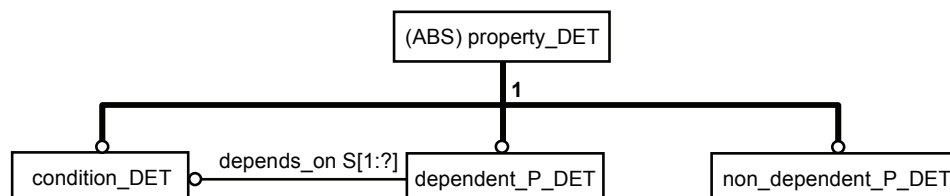


Figure 9 - Kinds of data element types

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

5.6.3.1 condition_DET¹⁸

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

EXPRESS specification:

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

¹⁸ See relevant clause in IEC 61360-1

```
END_ENTITY; -- condition_DET
(*)
```

5.6.3.2 dependent_P_DET¹⁸

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

EXAMPLE Ambient temperature.

EXPRESS specification:

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

Attribute definitions:

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

Formal propositions:

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

5.6.3.3 non-dependent_P_DET

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

EXPRESS specification:

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

5.6.4 Class-valued properties

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

EXPRESS specification:

```
*)
ENTITY class_value_assignment;
    super_class_defined_property: property_BSU;
```

```

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

```

Attribute definitions:

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

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

Formal propositions:

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

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

5.7 Domain data: the type system

The following subclauses contain definitions for the representation of the data types of a **property_DET**. Figure 10 outlines, as a planning model, the entity hierarchy for data types.

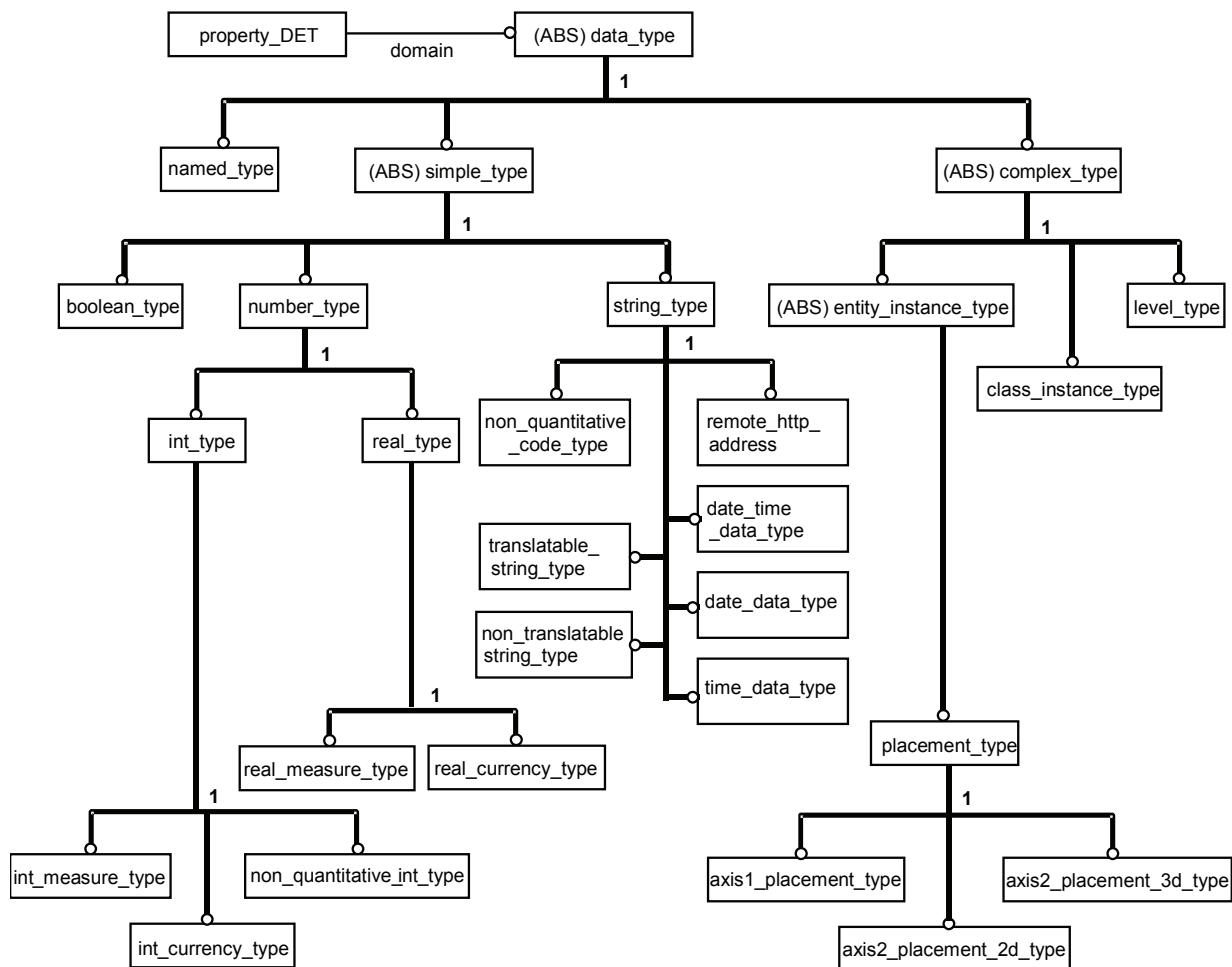


Figure 10 - Entity hierarchy for the type system

5.7.1 General

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

5.7.1.1 data_type_BSU

The **data_type_BSU** entity provides for identification of **data_type_elements**.

EXPRESS specification:

```

*)
ENTITY data_type_BSU
SUBTYPE OF(basic_semantic_unit);
    SELF\basic_semantic_unit.code: data_type_code_type;
    name_scope: class_BSU;
DERIVE
    absolute_id: identifier :=
        name_scope.defined_by.absolute_id      (* Supplier*)
        + sep_id + name_scope.dic_identifier    (* Class*)

```

```

        + sep_id + dic_identifier;          (* Data_type *)
INVERSE
    defining_class: SET OF class FOR defined_types;
UNIQUE
    absolute_id;
WHERE
    WR1: is_subclass(defining_class[1], name_scope.definition[1]);
END_ENTITY; -- data_type_BSU
(*

```

Attribute definitions:

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

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

absolute_id: the unique identification of this property.

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

Formal propositions:

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

5.7.1.2 data_type_element

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

EXPRESS specification:

```

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

END_ENTITY; -- data_type_element
(*

```

Attribute definitions:

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

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

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

5.7.2 The type system

5.7.2.1 data_type

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

EXPRESS specification:

```
*)
ENTITY data_type
ABSTRACT SUPERTYPE OF (ONEOF(
    simple_type,
    complex_type,
    named_type));
constraints: SET [0 : ?] OF domain_constraint ;
WHERE
    WR1: QUERY (cons <* constraints
        |NOT correct_constraint_type(cons, SELF)) = [];
END_ENTITY; -- data_type
(*
```

5.7.2.2 simple_type

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

EXPRESS specification:

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

Attribute definitions:

value_format¹⁹: the optional encoding of the format of values for properties.

NOTE If any **constraint** applies to the value of a simple type, then it takes precedence.

5.7.2.3 number_type

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

EXPRESS specification:

```
*)
ENTITY number_type
SUPERTYPE OF (ONEOF(
    int_type,
    real_type))
```

¹⁹ See relevant clause in IEC 61360-1