

INTERNATIONAL STANDARD

NORME INTERNATIONALE



HORIZONTAL STANDARD
NORME HORIZONTALE

**Standard data element types with associated classification scheme –
Part 1: Definitions – Principles and methods**

**Types normalisés d'éléments de données avec plan de classification –
Partie 1: Définitions – Principes et méthodes**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2017 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient 20 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

65 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



HORIZONTAL STANDARD

NORME HORIZONTALE

**Standard data element types with associated classification scheme –
Part 1: Definitions – Principles and methods**

**Types normalisés d'éléments de données avec plan de classification –
Partie 1: Définitions – Principes et méthodes**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 31.020

ISBN 978-2-8322-4581-1

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	10
INTRODUCTION.....	12
1 Scope.....	13
2 Normative references	13
3 Terms, definitions and abbreviated terms	14
3.1 Terms and definitions.....	14
3.2 Abbreviated terms.....	20
4 Foundation for the concepts of IEC 61360 standard dictionaries.....	20
5 Dictionary identification	22
5.1 General.....	22
5.2 Dictionary_supplier	22
5.3 Code.....	22
5.4 Version_number.....	23
5.5 Date_of_current_version	23
5.6 Revision_number	23
6 Property	24
6.1 Overview.....	24
6.2 Specification of information object	27
6.3 Data_element_type_class	28
6.4 Depends_on	29
6.5 Formula	29
6.6 Preferred_letter_symbol.....	30
6.7 Synonymous_letter_symbol	30
6.8 Attributes internal to Property.....	30
6.9 Code_of_alternative_unit.....	31
6.10 Code_of_list_of_units	31
6.11 Code_of_unit	32
6.12 Definition_class.....	32
6.13 Drawing_reference.....	32
7 Identifying_attributes	32
7.1 Overview.....	32
7.2 Specification of information object	33
7.3 Code.....	34
7.4 Preferred_name	35
7.5 Revision_number	35
7.6 Short_name	36
7.7 Synonymous_name.....	36
7.8 Version_number.....	36
8 Semantic_attributes	37
8.1 Overview.....	37
8.2 Specification of information object	37
8.3 Definition	37
8.4 Note.....	39
8.5 Remark	39
8.6 Source_document_of_definition	39
9 Administrative_attributes	40

9.1	Overview.....	40
9.2	Specification of information object	40
9.3	Obsolete_date	40
9.4	Proposal_date.....	40
9.5	Published_in	41
9.6	Published_by	41
9.7	Responsible_committee	41
9.8	Revision_released_on.....	42
9.9	Status_level	42
9.10	Version_initiated_on	43
9.11	Version_released_on	43
10	Value_attributes	43
10.1	Overview.....	43
10.2	Specification of information object	44
10.3	Alternative_units_of_measure	44
10.4	Data_type	45
10.4.1	General	45
10.4.2	Simple type	46
10.4.3	Enumeration type	51
10.4.4	Class instance type	52
10.4.5	Class reference type.....	53
10.4.6	Aggregate type	53
10.4.7	Level type.....	55
10.4.8	Large object type.....	56
10.4.9	Placement type.....	56
10.4.10	Data type dependencies	57
10.5	Number_of_significant_digits	59
10.6	Referenced_class_identifier	60
10.7	Unit_of_measure.....	60
10.8	Value_format	61
11	Condition_property.....	63
11.1	Specification of information object	63
11.2	Property_data_element_type	64
12	Dependent_condition_property	64
12.1	Specification of information object	64
12.2	Depends on	65
12.3	Property_data_element_type	65
13	Dependent_property	66
13.1	Specification of information object	66
13.2	Depends on	66
13.3	Property_data_element_type	67
14	Non_dependent_property	67
14.1	Specification of information object	67
14.2	Property_data_element_type	67
15	Translation_information	68
15.1	Specification of information object	68
15.2	Date_of_current_translation_revision	68
15.3	Language_code	68

15.4	Responsible_translator	69
15.5	Responsible_translator_coded	69
15.6	Translation_revision	69
16	Value_list	70
16.1	Specification of information object	70
16.2	Attributes internal to Value_list	71
16.3	Definition_class	71
16.4	Enumerated_list_of_terms	72
17	Term	72
17.1	Specification of information object	72
17.2	Preferred_letter_symbol_in_text	73
17.3	Attributes internal to Term	73
17.4	Definition_class	73
18	Drawing	73
18.1	Information model	73
18.2	Specification of information object	74
18.3	Code	75
18.4	Descriptive_designator	75
18.5	Drawing_title	76
18.6	File_format	76
18.7	File_name	77
18.8	Revision_number	77
18.9	Version_number	78
18.10	Attributes internal to Drawing	78
19	Unit of measure	78
19.1	Overview	78
19.2	Specification of information object	79
19.3	Primary_unit	80
19.4	Unit_in_text	80
19.5	Unit_XML	80
19.6	Attributes internal to Unit_of_measure	81
19.7	Drawing_reference	81
20	Creation of language variants	81
20.1	Overview	81
20.2	Language-dependent attributes of a Property	81
20.3	Language-dependent attributes of an Item_class	82
20.4	Language-dependent attributes of a Drawing	82
20.5	Language-dependent attributes of a Unit_of_measure	82
20.6	Language-dependent attributes of a Relation	82
21	Item_class	82
21.1	Classification tree	82
21.2	Composition tree	85
21.3	Use of auxiliary schemes for classification and coding of values	86
21.4	Information model	87
21.5	Specification of information object	87
21.6	Class_type	88
21.7	Coded_name	89
21.8	Attributes internal to Item_class	89

21.9	Applicable_data_element_type	89
21.10	Applicable_relation	90
21.11	Classifying_data_element_type.....	90
21.12	Drawing_reference.....	91
21.13	Is_case_of	91
21.14	Superclass.....	91
22	Relation.....	92
22.1	Overview.....	92
22.2	Specification of information object	93
22.3	External_solver_for_the_formula.....	93
22.4	Formula	94
22.5	Language_for_formula_interpretation.....	95
22.6	Relation type.....	95
22.7	Role_of_the_relation.....	95
22.8	Attributes internal to Relation	97
22.9	Definition_class	98
22.10	Codomain_of_function	98
22.11	Domain_of_function	98
22.12	Domain_of_relation.....	99
22.13	Drawing_reference.....	99
22.14	Super_relation	99
23	Dictionary_element.....	99
24	Advanced concepts	100
24.1	Overview.....	100
24.2	Condition	100
24.3	Reuse of properties	104
24.4	Classes and properties for common use	106
24.5	Block	107
24.6	Cardinality	109
24.7	Polymorphism	111
24.7.1	Overview	111
24.7.2	Polymorphic choices directly assigned to the specified Item_class	112
24.7.3	Polymorphic choices assigned to the specified Item_class through a Value_list	114
24.8	Relation	116
24.8.1	Overview	116
24.8.2	Restricted enumeration.....	116
24.8.3	Grouping	118
25	Qualifiers.....	119
26	Characters and character sets	120
26.1	Overview.....	120
26.2	Recommended character sets.....	120
26.3	Line feed.....	120
26.4	Subscript	120
26.5	Superscript	121
26.6	Greek characters	121
Annex A (informative)	Data model.....	123
Annex B (normative)	Type classification codes of properties	128

B.1	Property classification	128
B.1.1	Overview	128
B.1.2	Principles	128
B.1.3	Quantitative Property information objects	128
B.1.4	Non-quantitative Property information objects	130
B.2	Survey of type classification codes for non-quantitative properties	130
B.3	Survey of type classification codes for quantitative properties	131
Annex C (informative)	Preparation of new classes and properties	141
C.1	Responsibilities	141
C.2	Recommended elements of data set	141
Annex D (informative)	Rules for defining new versions and revisions of dictionary elements	142
D.1	Changes in the attributes of Property information objects	142
D.2	Changes in the attributes of class information objects	144
Annex E (informative)	Classifying_data_element_type attribute	145
Annex F (informative)	Conventions for names and definitions	146
F.1	Conventions for writing definitions	146
F.1.1	General	146
F.1.2	ISO/IEC 11179-4	146
F.1.3	ISO 704	146
F.1.4	Additional conventions	147
F.2	Conventions for writing names	147
F.2.1	Requirements	147
F.2.2	Recommendations	147
F.2.3	Mechanical quantitative property names	147
F.2.4	Electrical quantitative property names	147
F.2.5	Non-quantitative property names	148
Annex G (normative)	Value format specification	149
G.1	General	149
G.2	Notation	149
G.3	Data value format types	151
G.4	Meta-identifier used to define the formats	151
G.5	Quantitative value formats	151
G.5.1	General	151
G.5.2	NR1-value format	151
G.5.3	NR2-value format	152
G.5.4	NR3-value format	152
G.5.5	NR4-value format	153
G.6	Non-quantitative value formats	154
G.6.1	General	154
G.6.2	Alphabetic value format	154
G.6.3	Mixed characters value format	155
G.6.4	Number value format	155
G.6.5	Mixed alphabetic or numeric characters value format	156
G.6.6	Binary value format	156
G.7	HTML5 format	156
G.8	Value examples	157
Annex H (informative)	Modelling notation	158

H.1	General.....	158
H.2	UML Class	158
H.3	Generalization	158
H.4	Simple association.....	158
H.5	Modularization with UML package.....	159
	Bibliography.....	161
	Figure 1 – Simplified model of IEC 61360-1 (UML class diagram)	21
	Figure 2 – Overview model for Characteristic (UML class diagram).....	27
	Figure 3 – Property (UML class diagram).....	28
	Figure 4 – Identifying attributes (UML class diagram).....	34
	Figure 5 – Semantic attributes (UML class diagram)	37
	Figure 6 – Administrative_attributes (UML class diagram).....	40
	Figure 7 – Value_attributes (UML class diagram).....	44
	Figure 8 – Examples for technical data associated with connecting lines	46
	Figure 9 – Condition_property (UML class diagram).....	64
	Figure 10 – Dependent_condition_property (UML class diagram).....	65
	Figure 11 – Dependent_property (UML class diagram).....	66
	Figure 12 – Non_dependent_property (UML class diagram).....	67
	Figure 13 – Translation_information (UML class diagram).....	68
	Figure 14 – Value_list (UML class diagram).....	71
	Figure 15 – Term (UML class diagram)	72
	Figure 16 – Overview model for Drawing concept (UML class diagram)	74
	Figure 17 – Drawing (UML class diagram)	75
	Figure 18 – Overview model for Measure concept (UML class diagram).....	79
	Figure 19 – Unit_of_measure (UML class diagram).....	80
	Figure 20 – Classification tree	83
	Figure 21 – Composition tree	86
	Figure 22 – Overview model for Concept_data (UML class diagram).....	87
	Figure 23 – Item_class (UML class diagram).....	88
	Figure 24 – Overview model for Relation concept (UML class diagram)	92
	Figure 25 – Relation (UML class diagram)	93
	Figure 26 – Dictionary_element (UML class diagram)	100
	Figure 27 – Working point of a fuse	101
	Figure 28 – Implementation in IEC 62656-1 spreadsheet format of the example in Figure 27	102
	Figure 29 – (a) Dynamic gain and noise figure measurement setup, and (b) measurement with a saturation wavelength of 1550 nm	103
	Figure 30 – Implementation in IEC 62656-1 spreadsheet format of the example in Figure 29	104
	Figure 31 – Reuse of properties (IEC 62656-1 spreadsheet format)	105
	Figure 32 – Use of class information objects to associate tolerated capacitance information to a fixed capacitor (IEC 62656-1 spreadsheet format)	107
	Figure 33 – Interpretation of Class and Property information objects forming a block	108
	Figure 34 – Example of a block (IEC 62656-1 spreadsheet format)	109

Figure 35 – Interpretation of Class and Property information objects forming cardinality	110
Figure 36 – Example for a block whose number of occurrences is limited through PAA506 (IEC 62656-1 spreadsheet format).....	111
Figure 37 – Interpretation of Class and Property information objects forming a polymorphism	112
Figure 38 – Polymorphic choices directly assigned to the specified Item_class (IEC 62656-1 spreadsheet format)	113
Figure 39 – Polymorphic choices assigned to the specified class through a value list (IEC 62656-1 spreadsheet format)	115
Figure 40 – Example – Restricted enumeration (IEC 62656-1 spreadsheet format)	117
Figure 41 – Example – List of units for measuring the gap width of a labyrinth seal	118
Figure 42 – Example – Information objects of Figure 41 (IEC 62656-1 spreadsheet format).....	119
Figure A.1 – Characteristic (UML class diagram)	124
Figure A.2 – Drawing concept (UML class diagram)	125
Figure A.3 – Category (UML class diagram).....	125
Figure A.4 – Measure concept (UML class diagram)	126
Figure A.5 – Relation concept (UML class diagram).....	126
Figure A.6 – Value list concept (UML class diagram)	127
Figure H.1 – Example for generalization	158
Figure H.2 – Example of an association	159
Figure H.3 – Example of an association to a UML class owned by another UML package.....	159
Figure H.4 – Division of the IEC 61360-1 model into modules	160
Table 1 – Examples of generic concepts for individual quantities	19
Table 2 – List of attributes of Property information objects as defined in IEC 61360-1 and their equivalent in IEC 61360-2	25
Table 3 – Globally unique identification.....	33
Table 4 – Data type dependencies	59
Table 5 – List of attributes of Drawing.....	74
Table 6 – List of attributes of Item_class as defined in IEC 61360-1 and their equivalent in IEC 61360-2.....	85
Table 7 – Transliteration of Greek characters to Latin characters.....	122
Table B.1 – Survey of main classes of Property information objects.....	129
Table B.2 – Classification codes of non-quantitative data element types.....	130
Table B.3 – Classification codes for quantities of physical chemistry and molecular physics	131
Table B.4 – Classification codes for quantities of electricity and magnetism.....	133
Table B.5 – Classification codes for quantities of periodic and related phenomena	134
Table B.6 – Classification codes for quantities of acoustics	134
Table B.7 – Classification codes for quantities of heat	135
Table B.8 – Classification codes for quantities of information.....	135
Table B.9 – Classification codes for quantities of mechanics.....	136

Table B.10 – Classification codes for quantities of light and related electromagnetic radiations	137
Table B.11 – Classification codes for amounts	137
Table B.12 – Classification codes for prices and tariffs	138
Table B.13 – Classification codes for dimensionless business quantities and counts	138
Table B.14 – Classification codes for business ratios and percentages	138
Table B.15 – Classification codes for quantities of space and time	139
Table B.16 – Classification codes for quantities of nuclear reactions and ionizing radiations	140
Table D.1 – Overview of configuration management in Property updating operations	143
Table D.2 – Overview of configuration management in class updating operations	144
Table F.1 – Example of the name structure for electrical quantitative properties	148
Table G.1 – ISO/IEC 14977 EBNF syntactic meta-language	150
Table G.2 – Transposing European style digits into Arabic digits	155
Table G.3 – Number value examples	157

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**STANDARD DATA ELEMENT TYPES
WITH ASSOCIATED CLASSIFICATION SCHEME –****Part 1: Definitions – Principles and methods****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of 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, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). 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. 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 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 IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61360-1 has been prepared by subcommittee 3D: Product properties and classes and their identification, of IEC technical committee 3: Information structures and elements, identification and marking principles, documentation and graphical symbols.

It has the status of a horizontal standard in accordance with IEC Guide 108.

This fourth edition cancels and replaces the third edition published in 2009. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- support of advanced constructs such as
 - conditions and constraints,
 - blocks,

- cardinality,
 - polymorphism,
 - generic and restricted enumerations, and
 - mapping;
- extended list of data types;
 - harmonization with IEC 62656-1;
 - support of IEC TS 62720 and of coded units;
 - harmonization of semantic and administrative data among the various information objects;
 - use of UML for data modelling;
 - enhanced definitions and descriptions;
 - introduction of examples of higher level constructs such as block, cardinality, or polymorphism as guidance for the user of the IEC 61360 series.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
3D/295/FDIS	3D/298/RVD

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

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 61360 series, published under the general title *Standard data element types with associated classification scheme*, can be found on the IEC website.

Future standards in this series will carry the new general title as cited above. Titles of existing standards in this series will be updated at the time of the next edition.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific document. At this date, the document will be

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

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The IEC 61360 series as a whole specifies a general purpose dictionary of technical terms covering the field of electrotechnology, electronics and related domains. The dictionary is specified in a computer-sensible form as a reference dictionary. By using the dictionary, applications can interact and share data in an unambiguous way free from semantic uncertainties.

This document addresses domain engineers and provides a detailed introduction to the structure of the dictionary and its uses from the viewpoint of the dictionary provider as well as from the viewpoint of the user of the dictionary. IEC 61360-2 specifies the detailed dictionary data model and IEC 61360-6 stipulates quality criteria for the content of the dictionary.

Referencing to a common dictionary is advantageous in all cases where product information has to be transferred in an unambiguous way. Use cases include catalogues, ordering processes, product information contained in specifications or contracts.

The International Electrotechnical Commission has set up a technical dictionary for the use in the electrotechnical and electronic domain which is maintained by SC 3D. This dictionary is called IEC Common Data Dictionary (IEC CDD) and can be accessed on the following IEC web page: <http://std.iec.ch/iec61360>.

Dictionaries should not be confused with catalogues or master data collections; these make use of dictionary objects. In catalogues or master data collections, values are assigned to instances of dictionary objects. Thus they build upon dictionaries. Consequently, dictionaries are normally exchanged in advance of any catalogue or master data.

Closely associated with this document is IEC 61360-2, which contains the information model using the EXPRESS modelling language. In this model, the definition and structure of IEC 61360-1 is formalized and presented in a computer-sensible form.

This document is largely compliant with IEC 61360-2:2012 and ISO 13584-42:2010. However, practical use has shown the necessity to selectively extend the data model. Thus, this fourth edition of IEC 61360-1 extends IEC 61360-2 in both semantics and syntax and introduces additional constructs available from IEC 62656-1 for practical benefits. For example, constructs such as enumerations of real numbers or relation objects do not exist in IEC 61360-2, but are needed in many fields of engineering.

IEC 62656-1 provides interfaces to IEC CDD. Thus, change requests to IEC CDD need to be formed according to the interface specification available from the latter standard.

In some cases it can be difficult to see whether words represent names of information objects or if just everyday items are addressed, e.g. by the term "property".

For this reason, the following typographic rules are used in this document:

- bold, upper case first letter: name of a UML class, e.g. "**Property**";
- bold, lower case first letter: name of a UML attribute or UML association, e.g. "**revision_number**";
- normal font (i.e. not bold), lower case first letter: floating text, e.g. "property".

UML information object names should be treated as constants and should not be translated in other languages.

STANDARD DATA ELEMENT TYPES WITH ASSOCIATED CLASSIFICATION SCHEME –

Part 1: Definitions – Principles and methods

1 Scope

This part of IEC 61360 specifies principles for the definition of the properties and associated attributes and explains the methods for representing verbally defined concepts with appropriate data constructs available from IEC 61360-2. It also specifies principles for establishing a hierarchy of classification from a collection of classes, each of which represents a technical concept in the electrotechnical domain or a domain related to electrotechnology.

The use of this document facilitates the exchange of technical data through a defined structure for the information to be exchanged in a computer-sensible form. Each property to be exchanged has an unambiguously defined meaning and consistent naming, where relevant a defined value list, a prescribed format and defined units of measure for all quantitative values. There is also provision for:

- a) control of changes to definitions of the properties through version and revision numbers;
- b) inclusion of notes and remarks to clarify and help in the application of the definitions;
- c) indication of the sources of definitions and value lists;
- d) associated figures and formulae.

NOTE IEC TCs and SCs, or other organizations can take this document as a basis for the development of their own dictionaries.

Out of scope of this document are subjects concerning the information technology infrastructure such as:

- security;
- database locking mechanisms;
- access rights management.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 62656-1:2014, *Standardized product ontology register and transfer by spreadsheets – Part 1: Logical structure for data parcels*

IEC TS 62720, *Identification of units of measurement for computer-based processing*

IEC 80000 (all parts), *Quantities and units*

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange*

ISO/IEC 10646, *Information technology – Universal Multiple-Octet Coded Character set (UCS)*

ISO/IEC 11179-3, *Information technology – Metadata registries (MDR) – Part 3: Registry metamodel and basic attributes*

ISO/IEC 19505-1, *Information technology – Object Management Group Unified Modeling Language (OMG UML) – Part 1: Infrastructure*

ISO 639-1, *Codes for the representation of names of languages – Part 1: Alpha-2 code*

ISO 2382 (all parts), *Information processing systems – Vocabulary*

ISO 3166 (all parts), *Codes for the representation of names of countries and their subdivisions*

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

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

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 part families*

ISO/TS 29002-5, *Industrial automation systems and integration – Exchange of characteristic data – Part 5: Identification scheme*

ISO 80000 (all parts), *Quantities and units*

IETF RFC 2045: N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* [online], November 1996 [viewed 2016-07-01]. Available at <http://www.ietf.org/rfc/rfc2045.txt>

IETF RFC 2046: N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types* [online], November 1996 [viewed 2016-07-01]. Available at <http://www.ietf.org/rfc/rfc2046.txt>

IETF RFC 2047: K. Moore, *MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text* [online], November 1996 [viewed 2016-07-01]. Available at <http://www.ietf.org/rfc/rfc2047.txt>

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1.1

applicable property

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 can 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.

[SOURCE: IEC 61360-2:2012, 3.2]

3.1.2

attribute

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

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

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

[SOURCE: ISO/IEC Guide 77-2:2008, 2.2]

3.1.3

cardinality

pattern defining the number of times a concept reoccurs within a description

[SOURCE: IEC 61987-10:2009, 3.1.5, modified — Notes 1 to 5 are omitted.]

3.1.4

characteristic

distinguishing feature

Note 1 to entry: A characteristic can be inherent or assigned.

Note 2 to entry: A characteristic can be qualitative or quantitative.

[SOURCE: ISO 22274:2013, 3.3, modified — Note 4 and the Example are omitted.]

3.1.5

classification

systematic division of a set of items into subsets according to their difference in some predetermined characteristics

3.1.6

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.

[SOURCE: IEC 61360-2:2012, 3.6]

3.1.7 classifying property

property applicable for a particular class, having a value list whose values define the subclasses of the class

3.1.8 concept

unit of knowledge created by a unique combination of characteristics

[SOURCE: ISO 22274:2013, 3.7]

3.1.9 component

industrial product which serves a specific function or functions, which is intended for use in a higher order assembled product

Note 1 to entry: A component is in its context considered not to be decomposable or physically divisible. In another context, such as a service shop, a component can be decomposable.

3.1.10 computer-sensible form

specific representation of information allowing processing of the information content by means of an electronic computer

3.1.11 drawing

illustration used for explanatory purposes

3.1.12 entity

class of information defined by common properties

[SOURCE: ISO 10303-11:2004, 3.3.6]

3.1.13 enumeration

list of named constants called enumerators, each enumerator name in the enumeration being unambiguous

3.1.14 feature

aspect of an item that can be captured by a class structure and set of properties and that cannot exist independently of the item

[SOURCE: ISO 13584-24:2003, 3.41, modified – Examples 1 and 2 are omitted.]

3.1.15**information object**

well-defined piece of information, definition, or specification which requires a name in order to identify its use in communication

3.1.16**item**

thing that can be captured by a structure of class or by a structure of property

Note 1 to entry: Items are considered to reach from a component to assemblies up to systems.

[SOURCE: IEC 62656-1:2014, 3.23]

3.1.17**limiting value**

greatest or smallest admissible value of a quantity in a specification of a component, device, equipment, or system beyond which it incurs damage resulting in permanent unwanted changes of functional or physical characteristics influencing its performance

3.1.18**minimum value****min**

lower bound of a range of values in which the said value is meaningful

EXAMPLE Lowest value specified of a quantity, established for a specified set of operating conditions at which a component, device, equipment, or system can operate and perform according to specified requirements.

Note 1 to entry: Additional information about the nature of the value can be obtained from the definition of the **Property** information object to which the value belongs.

3.1.19**maximum value****max**

upper bound of a range of values in which the said value is meaningful

EXAMPLE Highest value specified of a quantity, established for a specified set of operating conditions at which a component, device, equipment, or system can operate and perform according to specified requirements

Note 1 to entry: Additional information about the nature of the value can be obtained from the definition of the **Property** information object to which the value belongs.

3.1.20**nominal value****nom**

value of a quantity used to designate or identify an item with its value, and not necessarily corresponding to the real value of the property

Note 1 to entry: Additional information about the nature of the value can be obtained from the definition of the **Property** information object to which the value belongs.

3.1.21**non-quantitative property**

property that identifies or describes an object by means of codes, abbreviations, names, references or descriptions

EXAMPLE Typical information content of non-quantitative properties is items such as codes, abbreviations, names, references, or descriptions.

[SOURCE: IEC 61360-2:2012, 3.28, modified – "data element type" is replaced by "property" in the term and definition.]

3.1.22

polymorphism

pattern that allows substitution of a single concept in the same context by a different more specific (specialized) concept

Note 1 to entry: A specialized polymorphic block can replace a more generic one in the same context.

Note 2 to entry: A polymorphic operator (control property) can act in selecting between various specializations.

[SOURCE: IEC 61987-10:2009, 3.1.21]

3.1.23

product

result of labour or of a natural or industrial process

3.1.24

property

data element type

defined parameter suitable for the description and differentiation of objects

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

Note 2 to entry: A property can have attributes such as code, version, and revision.

Note 3 to entry: The specification of a property can include predefined choices of values.

[SOURCE: ISO/IEC Guide 77-2:2008, 2.18, modified — In the definition, "products" is replaced by "objects". Note 4 is omitted]

3.1.25

quantitative property

property with a numerical value representing a physical quantity, a quantity of information or a count of objects

[SOURCE: IEC 61360-2:2012, 3.40, modified – "data element type" is replaced by "property" in the term and definition.]

3.1.26

quantity

property of a phenomenon, body, or substance, where the property has a magnitude that can be expressed by means of a number and a reference

Note 1 to entry: The generic concept "quantity" can be divided into several levels of specific concepts, as shown in Table 1. The left hand side of the table shows specific concepts under "quantity". These are generic concepts for the individual quantities in the right hand column.

Table 1 – Examples of generic concepts for individual quantities

Generic concepts for individual quantities		Individual quantities
length, l	radius, r	radius of circle A, r_A or $r(A)$
	wavelength, λ	wavelength of the sodium D radiation, λ_D or $\lambda(D; Na)$
energy, E	kinetic energy, T	kinetic energy of particle i in a given system, T_i
	heat, Q	heat of vaporization of sample i of water, Q_i
electric charge, Q		electric charge of the proton, e
electric resistance, R		electric resistance of resistor i in a given circuit, R_i
amount-of-substance concentration of substance B, c_B		amount-of-substance concentration of ethanol in wine sample i , $c_i(C_2H_5OH)$
number concentration of substance B, C_B		number concentration of erythrocytes in blood sample i , $C(Erys; B_i)$
Rockwell C hardness (150 kg load), HRC(150 kg)		Rockwell C hardness of steel sample i , HRC _i (150 kg)

Note 2 to entry: A reference can be a measurement unit, a measurement procedure, a reference material, or a combination of them.

Note 3 to entry: Symbols for quantities are given in the IEC 80000 and ISO 80000 series. The symbols for quantities are written in italics. A given symbol can indicate different quantities.

Note 4 to entry: A quantity as defined here is a scalar. However, a vector or a tensor, the components of which are quantities, is also considered to be a quantity.

Note 5 to entry: The concept “quantity” can be generically divided into, for example, “physical quantity”, “chemical quantity”, and “biological quantity”, or “base quantity” and “derived quantity”.

Note 6 to entry: Adapted from ISO/IEC Guide 99:2007, definition 1.1, in which there is an additional note.

[SOURCE: ISO 80000-1:2009, 3.1]

3.1.27

relation

aspect or quality that connects two or more things or parts as being or belonging together

3.1.28

subclass

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

Note 1 to entry: In the common ISO 13584/IEC 61360 dictionary model, class inclusion hierarchies are defined by the is-a relationship. They can also be established by the case-of relationships.

[SOURCE: IEC 61360-2:2012, 3.43]

3.1.29

superclass

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

Note 1 to entry: In the common ISO 13584/IEC 61360 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 ISO 13584/IEC 61360 dictionary model, a class has at most one superclass specified by means of an is-a relationship.

[SOURCE: IEC 61360-2:2012, 3.44]

3.1.30

typical value

typ

value of a quantity used as a representative value or commonly observable value

EXAMPLE Typical value specified of a quantity, established for a specified set of operating conditions at which an item can operate and perform according to specified requirements.

Note 1 to entry: Additional information about the nature of the value can be obtained from the definition of the **Property** information object to which the value belongs.

3.1.31

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".

[SOURCE: IEC 61360-2:2012, 3.46]

3.2 Abbreviated terms

DI	Data Identifier
HTML	Hypertext Mark-up Language
ICID	International Concept Identifier
IEC CDD	IEC Common Data Dictionary
IRDI	International Registration Data Identifier
RAI	Registration Authority Identifier
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

4 Foundation for the concepts of IEC 61360 standard dictionaries

In the IEC 61360 series, the basic concepts underlying the full data model are those of "dictionary", "class", and "property".

The concept of a product is captured by entities called "classes" and its characteristics are captured by entities called "properties". A hierarchy of classes in this document is an ordering of classes according to the commonalities of the properties in the classes.

The following clauses provide details on these concepts and their interrelations.

To show the concepts Unified Modelling Language (UML, see ISO/IEC 19505-1) class diagrams are used in Clauses 6 to 23. Clause 24 uses a spreadsheet-based presentation introduced by IEC 62656-1:2014 for representing structures containing instances.

NOTE 1 The UML models are for explanatory purposes; a detailed data model for implementation is given in IEC 61360-2.

Figure 1 shows a simplified overview model of the IEC 61360-1 information objects and their interrelations.

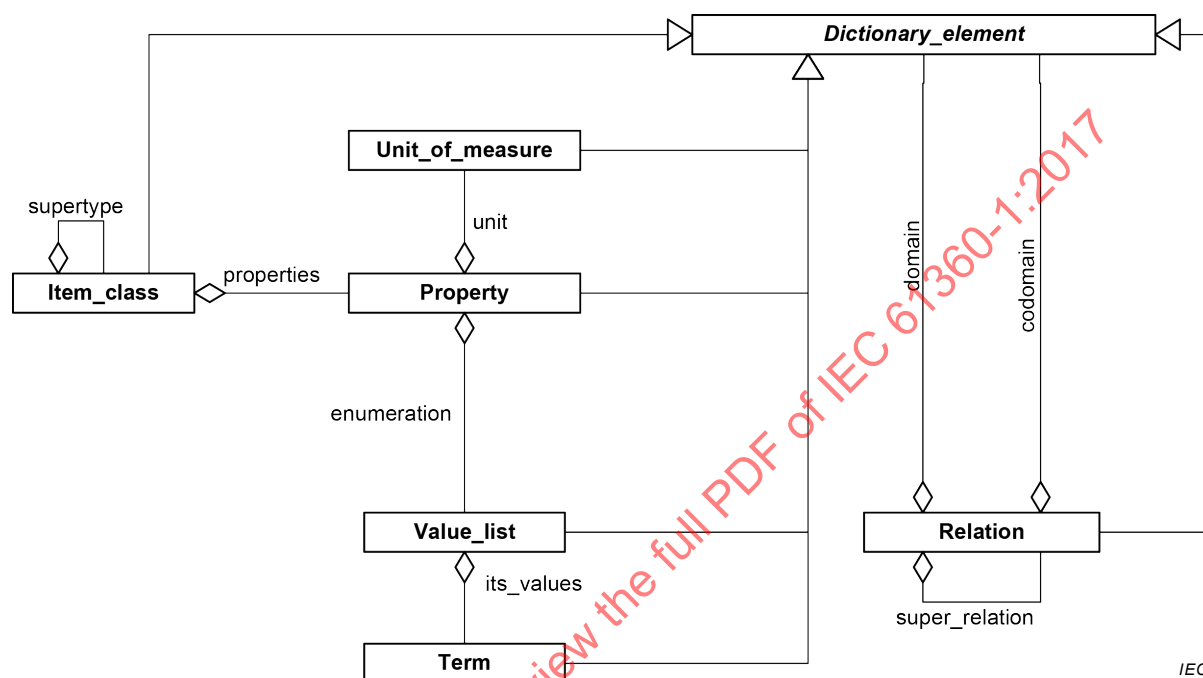


Figure 1 – Simplified model of IEC 61360-1 (UML class diagram)

NOTE 2 The association enumeration of Figure 1 is implemented via the enumeration data types.

The IEC 61360 dictionary model has a single inheritance model. However, from time to time, some properties defined in another class which resides in other branches of specialization hierarchy become necessary and pertinent.

In this case, the model offers the case-of mechanism to import properties of a fixed version (see 24.3). Once imported into a class, such properties become applicable in the class and inherited into its subclasses as applicable properties. The importation of properties is achieved by listing the properties to be imported in the “imported_properties” attribute of the case-of class.

Note that once a case-of relationship is installed, the version of the imported property is fixed, and even if the definition of the property is changed in the source class, the imported property does not change.

See Annex A for the detailed UML class diagrams of IEC 61360-1. Annex H gives a short introduction of the used UML constructs.

Annex C provides advice on preparation of new classes and properties.

5 Dictionary identification

5.1 General

The IEC Common Data Dictionary (IEC CDD) contains a collection of classes and properties. All information objects in a dictionary are uniquely identified including a version identifier as a record of significant changes to a dictionary object. This enables a version control for each dictionary object. And the identification of a dictionary object is therefore not dependent on a publication of the whole dictionary at a particular point in time.

For use of the dictionary content in communication it can be helpful to identify the complete set of dictionary items with a reference. For this reason the IEC CDD as a whole is identified by the following attributes given in 5.2 to 5.6.

5.2 Dictionary_supplier

Attribute name: **dictionary_supplier**

Attribute definition: identification of any organization authorized to register dictionary items

Description: The identifier is specified in ISO 13584-26:2000. The source identification for the IEC 61360-4 reference dictionary is: "0112/2///61360_4".

Other sources may require different identifications.

NOTE 1 The source identification of the standard does not include an edition number contrary to what ISO 13584-26:2000 has defined.

NOTE 2 The hyphen "-" in "61360-4" is replaced by an underscore "_".

EXAMPLE "0112/2///61987" is the source identification for data of IEC 61987 series.

Obligation: mandatory

Character type of values: String constant as specified

NOTE 3 The slash "/" separator is used in a physical file in accordance with ISO 10303-21. In other environments such as XML other separators can be used.

5.3 Code

Attribute name: **code**

Attribute definition: unique identifier of the dictionary item

Description: The content of **code** for the IEC Common Data Dictionary shall be: "IEC CDD".

Obligation: mandatory

Character type of values: String constant as specified

5.4 Version_number

Attribute name: **version_number**

Attribute definition: number used to control the versions of the IEC Common Data Dictionary

Description: The **version_number** of the dictionary should consist of 6 digits. Although the common ISO/IEC EXPRESS information model allows more digits, for the dictionary to be registered into IEC CDD or to be compatible with IEC 61360 series only 6 digits should be used. For specific purposes, version numbers with more digits may be used. Consecutive version numbers starting from "000001" shall be issued in ascending order.

A new version of the dictionary may be generated if at least one class or property is added or when changes are made affecting their version number. When a new version of the dictionary is issued, this data set contains all changes that happened since publication of the last version.

NOTE Intermediate changes of the dictionary can be traced by checking the attribute **date_of_current_version**.

Obligation: mandatory

Character type of values: Digits "0" through "9"

5.5 Date_of_current_version

Attribute name: **date_of_current_version**

Attribute definition: date associated with the present content of the IEC Common Data Dictionary

Description: The date information of the dictionary shall consist of 10 digits and shall follow the format "YYYY-MM-DD" as specified in ISO 8601:2004.

Whenever the content of the dictionary is modified, the value of the attribute shall be updated.

EXAMPLE "1994-03-21".

Obligation: mandatory

Character type of values: Digits "0" through "9" and "-".

5.6 Revision_number

Attribute name: **revision_number**

Attribute definition: number used for administrative control of the IEC Common Data Dictionary

Description: The **revision_number** of the dictionary should consist of 3 digits. Although the common ISO/IEC EXPRESS information model allows more digits, for the dictionary to be registered into IEC CDD or to be compatible with IEC 61360 series only 3 digits should be used. For specific purposes, revision numbers with more digits may be used. Consecutive revision numbers starting from "000" shall be issued in ascending order. Only one **revision_number** of the dictionary, unique by its identifier, is current at any moment.

A new **revision_number** of the dictionary shall be generated if an attribute of a dictionary object is changed which neither affects the use, nor the meaning of the dictionary objects.

EXAMPLE Editorial changes such as correction of typing errors.

The content of **revision_number** shall be reset to the starting value "1" when the **version_number** is incremented

Obligation: mandatory

Character type of values: Digits "0" through "9".

6 Property

6.1 Overview

A property is a characteristic of an object such as a component (see 3.1.24, Note 1). A property, when used, always has an associated value, such as a numerical value together with a unit, or a qualitative value taken from a predefined list.

In Clause 6, the various attributes of properties as encountered in the specifications are explained. For a list of these attributes, see Table 2. These attributes are related to identification, description, value of characteristics and to its relationships.

When specifying properties and their associated data type care, should be taken that the anticipated value is meaningful. In particular, the use of BOOLEAN_TYPE values may lead to deficient specifications if the meaning of the values "TRUE" and "FALSE" is not properly specified. In many cases the use of a value list instead of Boolean values provides more precise information and should be preferred.

Table 2 – List of attributes of Property information objects as defined in IEC 61360-1 and their equivalent in IEC 61360-2

Attribute names in IEC 61360-1	Information objects in IEC 61360-2	Subclause number
code	code	7.3
code (of value)	value_code	7.3
code (of value list)	value_code	7.3
code_of_alternative_unit	a	6.9
code_of_unit	a	6.11
code_of_list_of_units	a	6.10
condition_property	condition_DET	11.2
data_element_type_class	det_classification	6.3
data_type	data_type	10.4
date_of_current_translation_revision	a	15.2
definition	definition	8.3
definition (of value)	preferred_name of meaning	8.3
definition (of value list)	preferred_name of meaning	8.3
dependent_condition_property	a	12
dependent_property	dependent_P_DET	13
drawing_reference	figure	6.13
formula	formula	6.5
language_code	a	15.3
non_dependent_property	non_dependent_P_DET	14
note	note	8.4
obsolete_date	is_deprecated/ date_of_current_version	9.3
preferred_letter_symbol	preferred_symbol	6.6
preferred_name	preferred_name	7.4
proposal_date	a	9.4
published_by	a	9.6
published_in	a	9.5
referenced_class_identifier	domain of class_instance_type	10.6
remark	remark	8.5
responsible_committee	a	9.7
responsible_translator	a	15.4
responsible_translator_coded	a	15.5
revision_number	revision	7.5
revision_released_on	date_of_current_revision	9.8
short_name	short_name	7.6
source_document_of_definition	source_doc_of_definition	8.6
source_document_of_definition (of value list)	source_doc_of_value_domain	8.6
status_level	a	9.9
synonymous_letter_symbol	synonymous_symbol	6.7
synonymous_name	synonymous_name	7.7
translation_revision	a	15.6
unit_of_measure	unit	10.7
value_format	value_format	10.8
value_list	value_domain	16

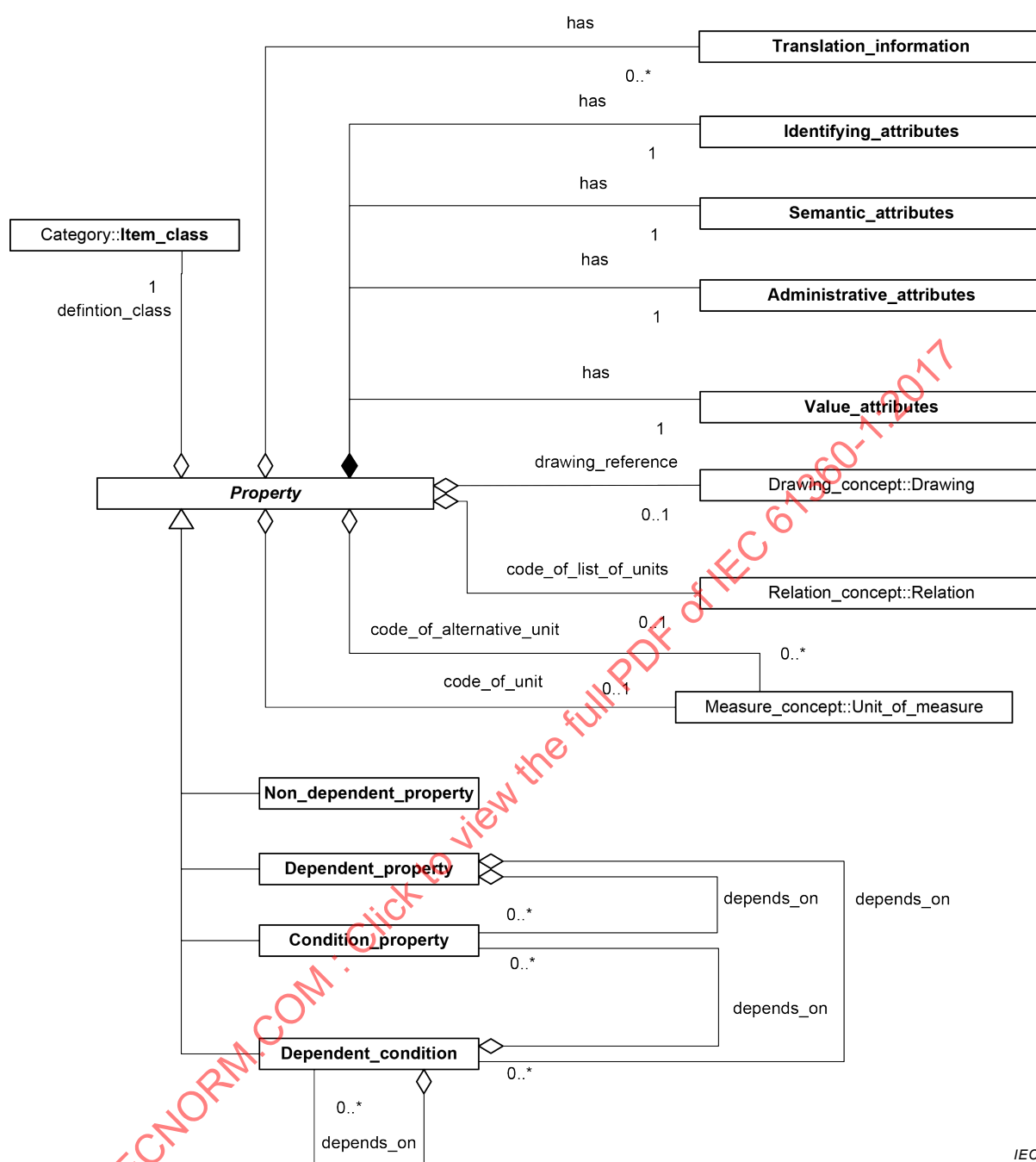
Attribute names in IEC 61360-1	Information objects in IEC 61360-2	Subclause number
version_initiated_on	date_of_original_definition	9.10
version_number	version	7.8
version_released_on	date_of_current_version	9.11
^a Not defined in IEC 61360-2:2012.		

Based on the principles as described in ISO/IEC 11179-3, the attributes are divided into four main groups:

- identification related attributes;
- semantics related attributes;
- value related attributes;
- relationship attributes.

Figure 2 shows the overview model of the module "Characteristic" using UML as presentation technique.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017



**Figure 2 – Overview model for Characteristic
(UML class diagram)**

6.2 Specification of information object

Name: **Property**

Definition: information object that characterizes the **Item_class(es)** to which it is associated

Description:

Figure 3 shows the detailed model of the information object "**Property**" using UML as presentation technique.

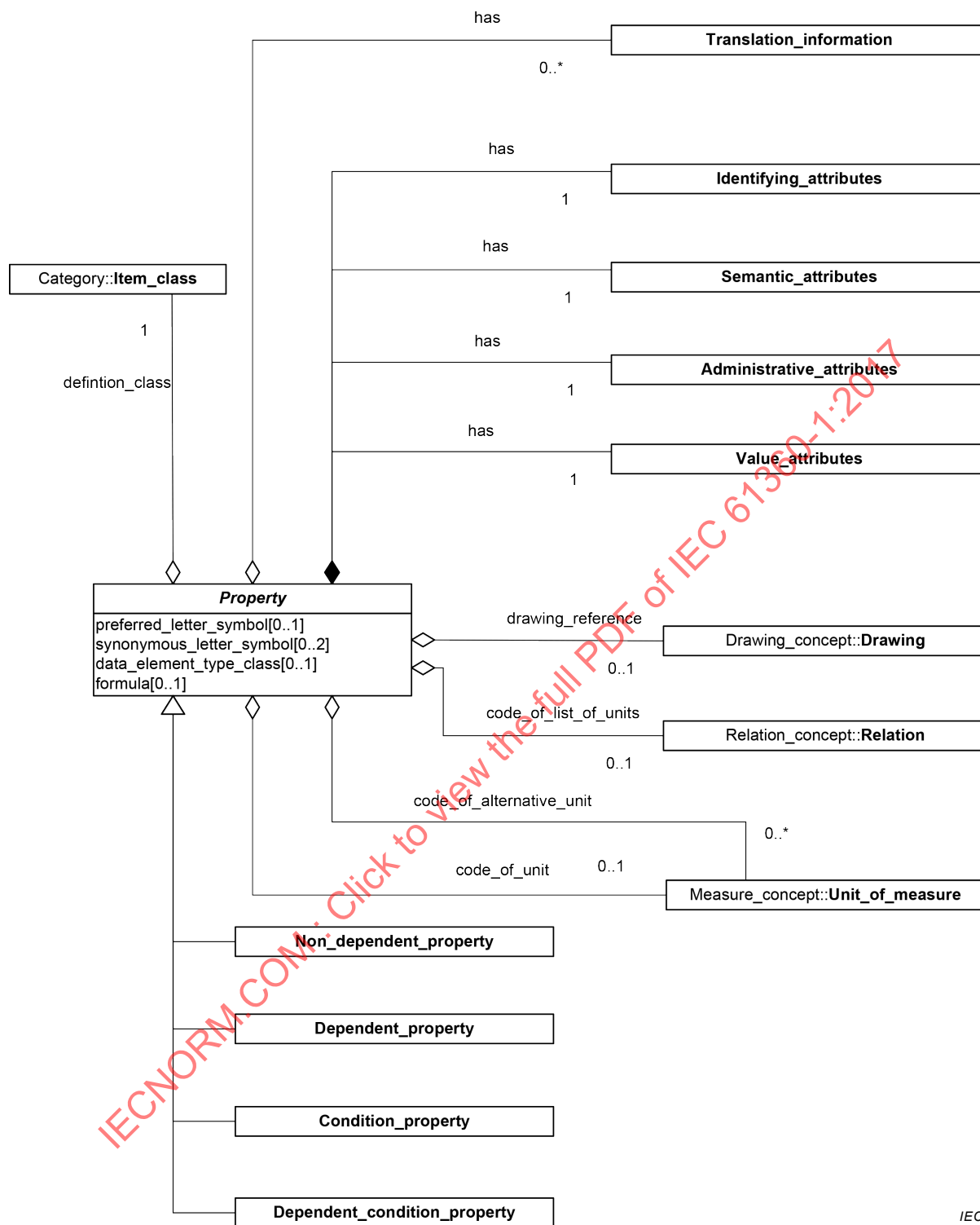


Figure 3 – Property (UML class diagram)

6.3 Data_element_type_class

Attribute name: **data_element_type_class**

Attribute definition: classification of similar property types

Description: Property classification is described in detail in Annex B.

Obligation: optional

Character type of values: See Clause 26.

6.4 Depends_on

Attribute name: **depends_on**

Attribute definition: set of one or more code(s) of **Property** information objects (see 7.3) acting as conditions on which the **Property** in question depends

Description: See also 24.2.

Obligation: conditional

Condition: shall have a value if **data_element_type_class** has one of the values

DEPENDENT_P_DET

DEPENDENT_C_DET

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

6.5 Formula

Attribute name: **formula**

Attribute definition: rule or statement in mathematical form expressing semantics of a **Property**

Description: The content of **formula** shall not change any essential information of the meaning of the definition. The graphical representation of a formula shall be stored as a file in a general available format.

NOTE 1 MathML (see W3C Recommendation 10:2014) is recommended as language to present mathematical formulas.

Obligation: optional

Character type of values: See Clause 26.

6.6 Preferred_letter_symbol

Attribute name: **preferred_letter_symbol**

Attribute definition: mark or characters used as a sign for expressing an information object

EXAMPLE 1 The characters standing for chemical elements in the periodic system (Ag = silver) or the characters standing for electrical concepts (f = frequency, T_{amb} = ambient temperature).

Description: The content of **preferred_letter_symbol** of a **Property** shall be identical to the letter symbol in the relevant standards.

EXAMPLE 2 IEC 60027 series, IEC 60747 series, IEC 60748 series, or IEC 80000 and ISO 80000 series.

For additional letter symbols, the standards relevant for the item in question should be consulted. The first character of the preferred letter symbol of a **Condition** should be a "@".

Obligation: optional

Character type of values: See Clause 26.

6.7 Synonymous_letter_symbol

Attribute name: **synonymous_letter_symbol**

Attribute definition: alternate mark or string of characters used as a sign for describing an information object that differs from the content of **preferred_letter_symbol** but represents the same concept

Description: Synonymous letter symbols are sometimes inevitable for historical reasons.

The number of synonymous letter symbols shall be limited to two.

The first character of the synonymous letter symbol of a **Condition** should be a "@".

Obligation: optional

Character type of values: See Clause 26.

6.8 Attributes internal to Property

Property has attributes that are grouped for better readability in the information objects

- **Identifying_attributes;**
- **Semantic_attributes;**
- **Administrative_attributes;**
- **Translation_information;**

– **Value_attributes.**

These information objects are related to **Property** by the relationship "**has**". The information objects shall be kept together with the referencing class, i.e. they are internal to the instances of **Property**.

6.9 Code_of_alternative_unit

Attribute name: **code_of_alternative_unit**

Attribute definition: one or more class identifier(s) of **Unit_of_measure** information objects that may be used instead of the unit specified by **code_of_unit**

Description: The unit of measure shall be specified by using the codes specified in IEC TS 62720 or compliant unit dictionaries.

NOTE 1 The unit dictionaries can be supplied in computer-sensible form or as paper.

NOTE 2 The unit identifier used in **code_of_alternative_unit** attribute can resolve to a unit provided by a server for units of measure.

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

NOTE 3 In principle the content of **code_of_alternative_unit** has the form RAI#DI##VI where RAI stands for the registration authority identifier, DI for the data identifier, and VI for the version identifier as described in ISO/IEC 11179-5.

6.10 Code_of_list_of_units

Attribute name: **code_of_list_of_units**

Attribute definition: class identifier specifying a **List_of_units** which offers a collection of possible units of measure

Description: The unit of measure may be specified by using the codes specified in IEC TS 62720 or compliant unit dictionaries.

NOTE 1 The unit dictionaries can be supplied in computer-sensible form or as paper.

NOTE 2 The unit identifier used in **code_of_list_of_units** attribute can resolve to a unit provided by a server for units of measure.

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

NOTE 3 In principle the content of **code_of_list_of_units** has the form RAI#DI##VI where RAI stands for the registration authority identifier, DI for the data identifier, and VI for the version identifier as described in ISO/IEC 11179-5.

6.11 Code_of_unit

Attribute name: **code_of_unit**

Attribute definition: class identifier of the preferred unit of measure

Description: The unit of measure may be specified by using the codes specified in IEC TS 62720 or compliant unit dictionaries.

NOTE 1 The unit dictionaries can be supplied in computer-sensible form or as paper.

NOTE 2 The unit identifier used in **code_of_unit** attribute can resolve to a unit provided by a server for units of measure.

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

NOTE 3 In principle the content of **code_of_unit** has the form RAI#DI##VI where RAI stands for the registration authority identifier, DI for the data identifier, and VI for the version identifier as described in ISO/IEC 11179-5.

6.12 Definition_class

Attribute name: **definition_class**

Attribute definition: class in which the information object is defined

Description: The **class** referenced by **definition_class** specifies the scope. Outside of this scope the instances are invalid.

Obligation: mandatory

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

6.13 Drawing_reference

Attribute name: **drawing_reference**

Attribute definition: reference to a **Drawing** illustrating the meaning of a **Property**

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

7 Identifying_attributes

7.1 Overview

The **Identifying_attributes** information object collects the attributes specifying the identity of an information object.

To identify an information object uniquely a language independent combination of characters shall be used. The identifier shall consist of the six-character code, followed by a hyphen followed by the three-digit version number of the information object.

For ensuring the global uniqueness of the identifiers in the reference dictionary the identification of the registration authority should be added to the identifier. The registration authority identification concept is defined in ISO/IEC 6523-1. At first, an organization identification scheme is selected from the registered schemes using the International Code Designator (ICD). Then, within this scheme, the organization is identified with the Organization Identifier (OI) (see Table 3).

For the IEC Common Data Dictionary, the ICD code "0112" shall be used standing for "Registration of Standardization Organization". For the Organization Identifier (OI), number "2" shall be used standing for "IEC".

A source identification for the data in IEC Common Data Dictionary is: "0112/2///61360_4". The added "61360_4" at the end is the Standards Identifier (SI) which is described in ISO 13584-26 as an extension to ISO/IEC 6523-1.

The IEC Common Data Dictionary may comprise multiple collections of dictionary items. Such collections do not create new identification spaces, i.e. identifiers shall be unique within the whole address space.

EXAMPLE 1 "0112/2///61987" for IEC 61987 series or "0112/2///62683" for IEC 62683.

EXAMPLE 2 The globally unique identification of a particular information object can be "0112/2///61340_4#AAF307#005".

Table 3 – Globally unique identification

Name	Description	Obligation	Data type	Maximum length
International Code Designator (ICD)	Identification of an organization identification scheme	Mandatory	Integer	4
Organization Identifier (OI)	Identification of an organization within an identification scheme	Mandatory	String	35

7.2 Specification of information object

Name: **Identifying_attributes**

Definition: collection of attributes identifying an information object

Description:

Figure 4 shows the detailed model of the information object "Identifying_attributes" using UML as presentation technique.

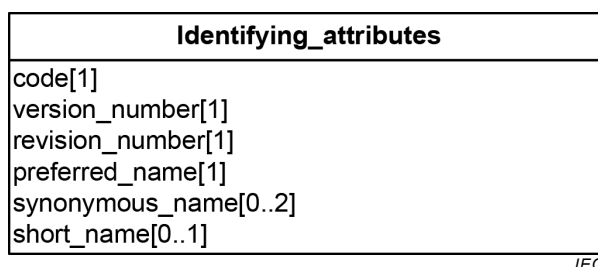


Figure 4 – Identifying attributes (UML class diagram)

7.3 Code

Attribute name: **code**

Attribute definition: unique six-character identifier of an information object

NOTE IEC 61360-2:2012 specifies 14 characters as the maximum length of the attribute "code". The difference is caused by information appended to the code such as the International Registration Data Identifier (see ISO/IEC 11179-6:2005).

Description:

The first three characters shall be alphabetic, the last three numeric (format AAANN). The character "X"¹ shall not be used as first character. The codes are issued sequentially and shall not have any relationship with the meaning of the information object.

In case of a modification of at least one attribute of the information object which affects one or both of the meaning and communication of the information object, a new (other) information object, having a new identifier, shall be defined.

Whether the identifier should be changed, so that a new information object is generated, or whether the version number of the information object should be changed, shall be determined for each case separately.

Such attributes are:

definition²;

unit_of_measure;

depends_on;

data_type.

Removing the relation to a **Condition_property**, i.e. changing the content of **depends_on**, shall result in a change of the **code**.

¹ For local use within private environments, codes starting with XAA up to and including codes starting with XZZ may be used and are therefore excluded from the IEC Common Data Dictionary.

² Extending the semantic context of an information object which does not violate the originally defined meaning by including another **Item_class** in the definition, results in a version change.

Obligation: mandatory

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

7.4 Preferred_name

Attribute name: **preferred_name**

Attribute definition: single-word or multi-word designation assigned to an item

Description: The content of **preferred_name** shall be identical to the name as used in international standards, if available. It is advised to keep the length of the name limited within 35 characters although its length may be extended up to 255 characters.

See Annex F for conventions for names.

Obligation: mandatory

Character type of values: See Clause 26.

7.5 Revision_number

Attribute name: **revision_number**

Attribute definition: number used for administrative control of an information object

Description: The **revision_number** of an information object should consist of 3 digits. Although the common ISO/IEC EXPRESS information model allows more digits, for the dictionary to be registered into IEC CDD or to be compatible with IEC 61360 series only 3 digits should be used. For specific purposes, revision numbers with more digits may be used. Consecutive revision numbers starting from "000" shall be issued in ascending order. Per information object, unique by its identifier, only one **revision_number** is current at any moment.

A new revision of an information object shall be generated if an attribute of the information object is changed which neither affects its use nor its meaning, or when corrections of typing or spelling errors have been implemented.

The **revision_number** shall be reset to its starting value "000" when the **version_number** is changed.

See Annex D for rules for defining new versions or revisions of dictionary elements.

Obligation: mandatory

Character type of values: Digits "0" through "9".

7.6 Short_name

Attribute name:	short_name
Attribute definition:	shortened representation of the content of preferred_name
Description:	The first character of the content of short_name shall be a letter: – the short name should be preceded by a "@" character for a Condition_property; – the short name shall be a short representation of the content in preferred_name. The length of the short name should be limited to 18 characters.
Obligation:	optional
Character type of values:	See Clause 26.

7.7 Synonymous_name

Attribute name:	synonymous_name
Attribute definition:	single-word or multi-word designation that differs from the given preferred name but represents the same concept
Description:	The number of synonymous names shall be limited to two. For pragmatic reasons, the length of the synonymous_name should be limited to 70 characters.
Obligation:	optional
Character type of values:	See Clause 26.

7.8 Version_number

Attribute name:	version_number
Attribute definition:	number used to control the versions of an information object

Description: The **version_number** of an information object should consist of 3 digits. Although the common ISO/IEC EXPRESS information model allows more digits, for the information object to be registered into IEC CDD or to be compatible with IEC 61360 series only 3 digits should be used. For specific purposes, version numbers with more digits may be used. Consecutive version numbers starting from "001" shall be issued in ascending order. Per information object, unique by its identifier, only one version number is current at any time.

A new version of an information object shall be generated if at least one attribute of the information object has changed which can affect its use but which does not affect its meaning.

See Annex D for rules for defining new versions or revisions of dictionary elements.

Obligation: mandatory

Character type of values: Digits "0" through "9".

8 Semantic_attributes

8.1 Overview

The **Semantic_attributes** information object collects the attributes specifying the meaning of an information object.

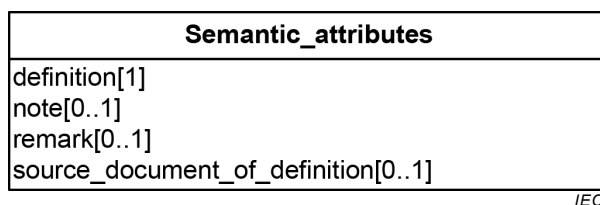
8.2 Specification of information object

Name: **Semantic_attributes**

Definition: collection of attributes specifying the semantic content of an information object

Description:

Figure 5 shows the detailed model of the information object "**Semantic_attributes**" using UML as presentation technique.



IEC

Figure 5 – Semantic attributes (UML class diagram)

8.3 Definition

Attribute name: **definition**

Attribute definition: statement that describes the meaning of an information object in an unambiguous and unique manner to permit its differentiation from other information objects

EXAMPLE 1 Definition of "arcing distance":

arcing distance

value of the shortest distance in air external to the insulator between metallic parts normally having the operating voltage between them

Description: Conventions and requirements:

NOTE An information object standardized under former editions of IEC 61360-1 can be updated only by initiating a change request.

- a) The content of **definition** of an information object shall be derived from the original definition as appearing in the latest corresponding IEC or ISO standards, if available.
- b) Where possible, definitions of property information objects shall be independent from classes. Thus, reuse of properties as part of other class information objects is supported.
- c) ISO 704 should be used as a basis for the writing of the definition statement.
- d) The unit of measure shall not be included in the definition statement.
- e) The level information should not occur in the definition statement.

NOTE Level information can be specified with the data type **level_type**.

- f) The semantic context should be included in the definition statement, if this is essential for the understanding of its meaning.
- g) If the concept requires a limitation of its applicability, this shall be explicitly expressed in the definition statement.

EXAMPLE 2 There exist different semantics of the term "rated voltage"; within products $\geq 1,5$ kV, the term rated voltage expresses the maximum voltage for which a product is being designed and can be operated. This is currently not applicable to products less than 1,5 kV.

- h) If dependency relations are an inherent part of the concept, these shall be included in the definition statement.

EXAMPLE 3 Definition statement of the Property "reverse recovery time":

"value of the time required for the reverse current of a diode to recover to a specified value, when switched from a specified forward current to a specified reverse voltage, at specified conditions"

- i) In the case conditions are specified, the definition statement should end with the wording "at specified condition(s)".
- j) If the concept represents an average value, the method of calculating the average shall be identified, by using a term that designates the method, such as "arithmetic mean", "geometric mean", "median", or "mode", either in the **preferred_name** or in the **definition**.

See Annex F for conventions for definitions.

Obligation: mandatory

Character type of values: See Clause 26.

8.4 Note

Attribute name: **note**

Attribute definition: statement providing further information on the **definition**, which is considered to be essential for its understanding

Description: EXAMPLE The **Property** "reverse recovery time" is further clarified by a **note**: "The reverse recovery time is measured as the time interval between t_0 , the point where the forward current crosses the zero current axis, and the instant when for decreasing values of i_R a line through the points for $0,9 I_{RM}$ and $0,25 I_{RM}$ crosses the zero current axis."

Obligation: optional

Character type of values: See Clause 26.

8.5 Remark

Attribute name: **remark**

Attribute definition: additional information in text

Description: The content of **remark** shall not change the meaning of the **definition**.

Obligation: optional

Character type of values: See Clause 26.

8.6 Source_document_of_definition

Attribute name: **source_document_of_definition**

Attribute definition: reference to the source document from which the content of **definition** was derived

Description: The document shall be recognized by the ISO or IEC committee concerned as having wide acceptance and authoritative status as well as being publicly available.

Obligation: optional.

Character type of values: See Clause 26.

9 Administrative_attributes

9.1 Overview

The **Administrative_attributes** information object collects the attributes used to administer an information object.

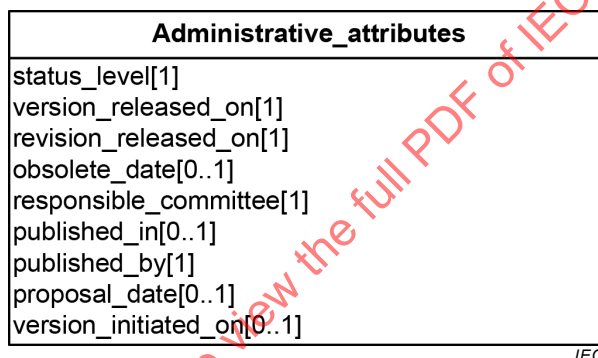
9.2 Specification of information object

Name: **Administrative_attributes**

Definition: collection of attributes specifying the information used to manage an information object

Description:

Figure 6 shows the detailed model of the information object "Administrative_attributes" using UML as presentation technique.



IEC

**Figure 6 – Administrative_attributes
(UML class diagram)**

9.3 Obsolete_date

Attribute name: **obsolete_date**

Attribute definition: date when the status level (see 9.9) received the value "Superseded"

EXAMPLE "2010-06-28"

Obligation: conditional

Character type of values: date format according to "YYYY-MM-DD" as specified by ISO 8601

9.4 Proposal_date

Attribute name: **proposal_date**

Attribute definition: date when the information object was uploaded to the database

EXAMPLE "2003-05-14"

Obligation: optional

Character type of values: date format according to "YYYY-MM-DD" as specified by ISO 8601

9.5 Published_in

Attribute name: **published_in**

Attribute definition: identity number of the publicly available document in which the item and its definition were first published

One or both of edition number and year should be included as appropriate.

EXAMPLE "IEC 61175:2009"

Obligation: optional

Character type of values: See Clause 26.

9.6 Published_by

Attribute name: **published_by**

Attribute definition: organization responsible for the publication as defined by "published in"

EXAMPLE "IEC"

Obligation: optional

Character type of values: See Clause 26.

9.7 Responsible_committee

Attribute name: **responsible_committee**

Attribute definition: Technical Committee or Subcommittee of IEC or ISO in charge of maintaining the information object

EXAMPLE "SC3D"

Obligation: mandatory

Character type of values: See Clause 26.

9.8 Revision_released_on

Attribute name:	revision_released_on
Attribute definition:	date when the information object with the current revision number was authorized
	EXAMPLE "2004-05-20"
Obligation:	mandatory
Character type of values:	date format according to "YYYY-MM-DD" as specified by ISO 8601

9.9 Status_level

Attribute name:	status_level
Attribute definition:	name of the stage of the standardization workflow the information object is in
Description:	The following status levels may be assigned:
Proposal	status level of a new information object from its registration and identification in the database, until it has been accepted for work and a resolution has been reached on how to proceed following the preliminaries
Draft	status level of a new information object that has been accepted for work following the preliminaries with either the normal or extended database procedure, until the moment a decision has been taken on whether or not it is to be part of the standard
	NOTE 1 The transition to status_level "draft" is from status_level "proposal".
Standard	status level of a new information object that has been released for use as part of the standard
	NOTE 2 The transition to status_level "standard" is from status_level "draft".
Superseded	status level of an information object that is no longer part of the standard, irrespective of reason
	NOTE 3 The transition to status_level "superseded" is from status_level "standard". On the information object page a note or a reference to a replacing information object further indicates the reason for obsolescence.

Rejected status of an information object that has been entered into the database as part of a change request, but has not been approved to be part of the standard

NOTE 4 The transition to "rejected" is either from **status_level** "proposal" or from **status_level** "draft".

Obligation: mandatory

Character type of values: See Clause 26.

9.10 **Version_initiated_on**

Attribute name: **version_initiated_on**

Attribute definition: date when the information object with the current version number was initially issued

EXAMPLE "2004-05-15"

Obligation: optional

Character type of values: date format according to "YYYY-MM-DD" as specified by ISO 8601

9.11 **Version_released_on**

Attribute name: **version_released_on**

Attribute definition: date when the information object with the current version number was authorized

EXAMPLE "2004-05-15"

Obligation: mandatory

Character type of values: date format according to "YYYY-MM-DD" as specified by ISO 8601

10 **Value_attributes**

10.1 **Overview**

The **Value_attributes** information object collects the attributes that specify the numerical or non-numerical value of a **Property**.

Every **Property** information object may have a value domain that covers the range of permissible values being either implicitly specified by the **value_format** or explicitly specified by listing the allowed values.

For quantitative **Property** information objects, the value domain may be expressed either as a value range providing the minimum and maximum values or as an explicit enumeration of discrete possible values.

For non-quantitative **Property** information objects, the value domain can be expressed by enumerating the permissible values in the form of text strings.

NOTE The attribute **value_format** (see 10.8) can be used as an indicator for the intended values that can be assigned to a **Property** during its usage. Additionally, it can be used for performing a basic quality control on the data provided.

For **Property** information objects with predefined value sets, a mechanism using values, represented by value codes and associated value meanings, is described in Clause 16.

If a **Property** serves as classifying property, a value list shall be specified.

10.2 Specification of information object

Name:	Value_attributes
Definition:	collection of attributes specifying the value(s) and unit(s) of an information object
Description:	

Figure 7 shows the detailed model of the information object "Value_attributes" using UML as presentation technique.

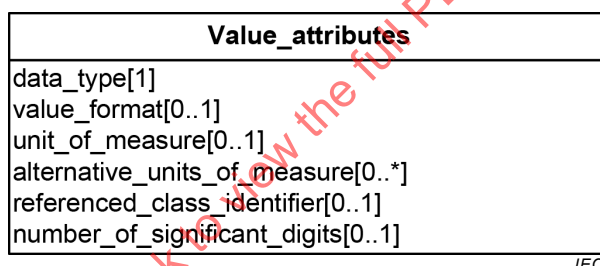


Figure 7 – Value_attributes (UML class diagram)

10.3 Alternative_units_of_measure

Attribute name:	alternative_units_of_measure
Attribute definition:	units which may be used as an alternative to specify the value of a property

NOTE 1 Alternative units allow the recommendation of further units in addition to the preferred unit (see 10.7). Reasons to choose an alternative unit are causes such as specific measurement principles, particular application domains, or special legislative systems.

NOTE 2 Alternative units can be made available in purpose-related groups of units (see 24.8.3).

Description:

Conventions and requirements:

- a) Alternative units may be supplied in addition to a preferred unit. Such alternative units can, but do not have to be, SI units.

EXAMPLE bar, psi, or Torr instead of Pa.

- b) Alternative units may be specified by using the code of the unit as specified in IEC TS 62720 or compliant unit dictionaries.

NOTE 3 The unit dictionaries can be supplied in computer-sensible form or as paper.

- c) The conversion to a different unit is outside the scope of this document.
- d) It is out of scope of this document to prescribe the use of a specific unit for transaction purposes including possible prefixes.

NOTE 4 **Property** information objects with **data_type** "CLASS_REFERENCE_TYPE" do not carry a unit.

- e) It is strongly recommended not to specify the same concept in different properties if only the prefixes of the related units are different. Exceptions may occur if certain units are very common in particular business domains, such as "nm" (nanometre) in semiconductor manufacturing or "kV" (kilovolt) in the power distribution domain.
- f) Each unit contained in the `alternative_units_of_measure` attribute is required to be associated with a string representation, whose text representation can be used for characterizing the unit at the instance level.

Obligation:

optional

Character type of values: See Clause 26.

10.4 Data_type

10.4.1 General

Attribute name:

data_type

Attribute definition:

identifies a specific quality of the value of a **Property**

Description:

The content of **data_type** is divided into categories:

- simple type;
- enumeration type;
- aggregate type;
- class instance type;
- class reference type;
- large object data;
- placement type;
- level type.

Obligation: mandatory

Character type of value: See Clause 26.

10.4.2 Simple type

A data type specifying that the value of a **Property** is a single numeric or a non-numeric value. The value format shall be defined according to 10.8.

In each designation of type where “_TYPE” is found, “_TYPE” may be omitted.

For a simple type, the valid data types are:

- **BOOLEAN_TYPE**

The data type **BOOLEAN_TYPE** may be used for values representing truth values of logic or Boolean algebra or pattern of truth values.

The use of "TRUE" as Boolean value indicates a true or accurate result. "FALSE" as Boolean value expresses a wrong or inaccurate result. Other values are deprecated.

EXAMPLE 1 Deprecated Boolean values are values such as "YES" or "NO" and "Y" or "N".

If a value contains a single truth value the data format shall contain "B1".

NOTE 1 The use of **BOOLEAN_TYPE** values can lead to deficient specifications if the meaning of the values is not properly specified. In many cases the use of a value list instead of **BOOLEAN_TYPE** values provides more precise information.

Additionally, some types of presentations can become meaningless if **BOOLEAN_TYPE** values are used.

EXAMPLE 2 In circuit diagrams, lead texts are often omitted because of space restrictions or avoidance of the need for translation, and only the values are presented. Figure 8 shows examples for technical data associated with connecting lines presented in an abbreviated form (see IEC 61082:2014, Figure 38). In such cases, the use of values of **BOOLEAN_TYPE** could lead to presentations that are difficult to interpret, as such values are meaningless without lead text.

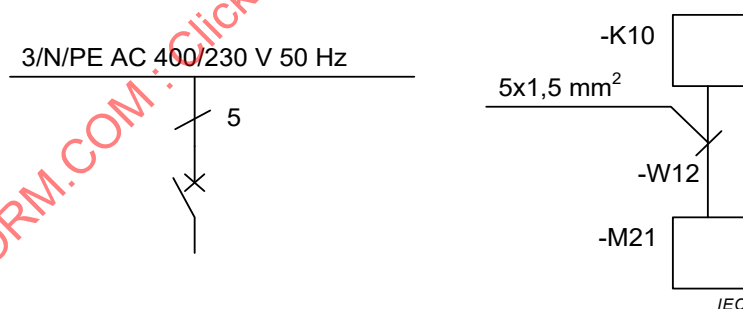


Figure 8 – Examples for technical data associated with connecting lines

- **BINARY_TYPE**

The data type **BINARY_TYPE** may be used for values containing a sequence of bits, each bit being represented by "0" or "1". The value format shall be "B n " with $n > 1$. The textual part of the property, i.e. content of the attributes "definition", "note" or "remark", should give guidance about the interpretation of the bit pattern.

NOTE 1 There is no upper limit specified for the value of " n ". Current implementations (in 2014) support sequences of bits ("bitstrings") of a size up to 2 gigabytes.

- **STRING_TYPE**

The data type **STRING_TYPE** may be used for values containing any sequence of elements, typically characters, using some character encoding.

Instead of the `STRING_TYPE` one of the more specific data types may be used:

- `TRANSLATABLE_STRING_TYPE`;
- `NON_TRANSLATABLE_STRING_TYPE`;
- `NON_QUANTITATIVE_CODE_TYPE` (deprecated);
- `DATE_TIME_TYPE`;
- `DATE_TYPE`;
- `TIME_TYPE`;
- `IRDI_STRING`;
- `URI_TYPE`.
- `HTML5_TYPE`.

- `TRANSLATABLE_STRING_TYPE`

The data type `TRANSLATABLE_STRING_TYPE` may be used for values containing strings, but that are supposed to be represented as different strings in different languages.

- `NON_TRANSLATABLE_STRING_TYPE`

The data type `NON_TRANSLATABLE_STRING_TYPE` may be used for values containing strings but that shall be represented in the same way in any language, i.e. such strings shall not be translated.

EXAMPLE 3 The MAC address of a computer, e.g. "00-80-41-ae-fd-7e".

- `DATE_TYPE`

The data type `DATE_TYPE` may be used for values containing a calendar date specified according to a representation compliant with ISO 8601.

A value whose data type is `DATE_TYPE` shall comply with the following lexical representation, which is a subset of the lexical representations allowed by ISO 8601. The lexical representation for `DATE_TYPE` is the reduced (right truncated) lexical representation for `DATE_TIME_TYPE`: "YYYY-MM-DD". No left truncation is allowed. An optional following time zone qualifier is allowed as for `DATE_TIME_TYPE`. To accommodate year values outside the range from "0001" to "9999", additional digits may be added to the left of this representation and a preceding "-" sign is allowed.

EXAMPLE 4 "1999-05-31" is the `DATE_TYPE` representation of: "31 May 1999".

Attribute **format** should contain the value "M..10".

- `TIME_TYPE`

The data type `TIME_TYPE` may be used for values defining a time specified according to a representation compliant with ISO 8601. A value of `TIME_TYPE` represents an instant of time that recurs every day.

NOTE 2 Only a subset of the lexical representations allowed by ISO 8601 is allowed for values of `TIME_TYPE`.

NOTE 3 The above restriction of ISO 8601 representations is the same as the one defined by XML Schema.

NOTE 4 Since the lexical representation allows an optional time zone indicator, `TIME_TYPE` values are partially ordered because it is not possible to determine the order of two values one of which has a time zone and the other does not.

A value whose data type is `TIME_TYPE` shall comply with the following lexical representation, which is a subset of the lexical representations allowed by ISO 8601. The lexical representation for `TIME_TYPE` is the left truncated lexical representation for `DATE_TIME_TYPE` "hh:mm:ss.sss" with optional following time zone indicator.

EXAMPLE 5 "13:20:00-05:00" is the `TIME_TYPE` representation of: 1.20 p.m. for Eastern Standard Time, which is 5 hours behind Coordinated Universal Time (UTC).

Attribute **format** should contain the value "M..14".

- `DATE_TIME_TYPE`

The data type `DATE_TIME_TYPE` may be used for values defining an instant of time specified according to a particular representation compliant with ISO 8601.

A value whose data type is `DATE_TIME_TYPE` shall comply with the following lexical representation, which is a subset of the lexical representations allowed by ISO 8601. This lexical representation is the ISO 8601 extended format "YYYY-MM-DDThh:mm:ss" where "YYYY" represents the year, "MM" the month and "DD" the day, preceded by an optional leading "-" sign. The letter "T" is the date/time separator and "hh", "mm", "ss" represent hour, minute and second, respectively. Additional digits can be used to increase the precision of fractional seconds if desired, i.e. the format "ss.ss..." with any number of digits after the decimal point is supported. The fractional second part is optional; other parts of the lexical form are not optional. To accommodate year values greater than "9999" additional digits can be added to the left of this representation. Leading zeros are required if the year value would otherwise have fewer than four digits; otherwise they are forbidden. The year "0000" is prohibited. The "YYYY" field shall have at least four digits, the "MM", "DD", "SS", "hh", "mm" and "ss" fields exactly two digits each (not counting fractional seconds); leading zeroes shall be used if the field would otherwise have too few digits. This representation may be immediately followed by a "Z" to indicate "Coordinated Universal Time (UTC)" or, to indicate the time zone, i.e. the difference between the local time and Coordinated Universal Time, immediately followed by a sign, "+" or "-", followed by the difference from UTC represented as "hh:mm". If the time zone is included, both hours and minutes shall be present.

ISO 8601 provides details about legal values in the various fields. If the time zone is included, both hours and minutes shall be present.

EXAMPLE 6 To indicate 1:20 p.m. on 31 May 1999 for Eastern Standard Time which is 5 hours behind Coordinated Universal Time (UTC), one would write: "1999-05-31T13:20:00-05:00".

NOTE 5 The above restriction of ISO 8601 representations is the same as the one defined by XML Schema.

Attribute **format** should contain the value "M-25".

- **IRDI_STRING_TYPE**

The data type `IRDI_STRING_TYPE` may be used for values conforming to ISO/IEC 11179 series global identifier sequences.

IRDI, DI and **code** have the following relationship:

`IRDI::= RAI'#DI'#VI`

NOTE 6 RAI stands for the registration authority identifier, DI for the data identifier, and VI for the version identifier. **Code** attribute is used as DI in this document.

NOTE 7 In implementation, a data validation can be applied to check whether an information object exists for the referenced IRDI code.

`IRDI_STRING_TYPE` may be used instead of one of the more specific data types:

- `ICID_STRING_TYPE`;
- `ISO_29002_IRDI`.

- **ICID_STRING_TYPE**

The data type `ICID_STRING_TYPE` may be used for values conforming to `IRDI_STRING_TYPE`, where the delimiter between RAI and DI is "#" while the delimiter between DI and VI is confined to "##".

ICID, DI and **code** have the following relationship:

`ICID::= RAI'#DI'##VI`

NOTE 8 RAI stands for the registration authority identifier, DI for the data identifier, and VI for the version identifier. **Code** attribute is used as DI in this document.

NOTE 9 In implementation, a data validation can be applied to check whether an information object exists for the referenced ICID code.

- **ISO_29002_IRDI_TYPE**

The data type `ISO_29002_IRDI_TYPE` may be used for values containing a global identifier that identifies an administrated item in a registry. The structure of this identifier complies with identifier syntax defined in ISO/TS 29002-5. The identifier shall fulfill the requirements specified in ISO/TS 29002-5 for an "international registration data identifier" (IRDI).

IRDI, DI and **code** have the following relationship:

`IRDI ::= RAI '#' DI '#' VI`

NOTE 10 RAI stands for the registration authority identifier, DI for the data identifier, and VI for the version identifier. **Code** attribute is used as DI in this document.

As specified in ISO/TS 29002-5, the length of the identifier string shall not be greater than 290 characters.

NOTE 11 According to ISO/TS 29002-5 an IRDI consists either of a string that does not contain the '#' character, to identify an organization, or of three sub-strings that do not contain the '#' character and that are separated by the '#' character to identify any other administrated item.

NOTE 12 If IRDI is not used for identifying an organization:

- the first sub-string, called the "Registration Authority Identifier (RAI)", identifies the organization which is responsible for the administration of the administrated item;
- the second sub-string, called the "Data Identifier (DI)", contains both a categorization of the administrated item, represented by two characters followed by the minus ("-") sign as defined in ISO/TS 29002-5 (e.g. "class", "property", "unit"), and the identifier assigned to the administrated item by the RAI;
- the third sub-string corresponds to the "Version Identifier (VI)" of the IRDI.

• `URI_TYPE`

The data type `URI_TYPE` may be used for values containing any sequence of characters that is a URI.

NOTE 13 A `URI_type` allows in particular to express a URL.

• `HTML5_TYPE`

The data type `HTML5_TYPE` may be used for values containing strings with any sequence of characters, using the syntax of HTML5 (see W3C Recommendation 28:2014). This content provides for a pictorial presentation that can be impossible to achieve by solely using characters.

NOTE 14 As HTML5 supports MathML (see W3C Recommendation 10:2014), mathematical formulas can be presented.

EXAMPLE 7 The value $\frac{1}{\sqrt{3}}420\text{ kV}$ (242,48711305964282109384248781082 kV) can be expressed with `HTML5_TYPE` as

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
  </head>
  <body>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mrow>
        <mrow>
          <mfrac>
            <mn>1</mn>
            <msqrt>
              <mn>3</mn>
            </msqrt>
          </mfrac>
        </mrow>
        <mo> · </mo>
        <mn>420</mn>
        <mi>kV</mi>
      </mrow>
    </math>
```

```
</body>
</html>
```

Cases exist where, in addition to the presentation of the value, its exact numeric value is of interest too. In such cases reference to an **Item_class** is recommended that provides a **Property** for the numeric value (data type INTEGER, REAL, etc.) and a **Property** for the HTML presentation (data type HTML5).

NOTE 15 The class is referenced by using CLASS_INSTANCE_REFERENCE.

- NUMBER_TYPE

The data type NUMBER_TYPE may be used for values containing numbers that can be written as numeric values. This data type may be used when a more specific numeric representation is not important or unwanted.

Instead of the NUMBER_TYPE one of the more specific data types may be used:

- INTEGER;
- RATIONAL;
- REAL.

- INTEGER, INT_TYPE

The data type INT_TYPE may be used for values containing numbers that can be written as positive or negative whole numbers or as zero.

Instead of the INT_TYPE one of the more specific data types may be used:

- INT_MEASURE_TYPE;
- INT_CURRENCY_TYPE.

- INT_MEASURE_TYPE

The data type INT_MEASURE_TYPE may be used for values containing values that are measures of type INTEGER.

In addition, the data type specifies a unit or a unit identifier, in which the values are expressed. It may also specify alternative units or alternative unit identifiers that are valid if the value is explicitly associated with one of these units.

Either a unit or a unit identifier is mandatory. In cases where both are provided, the unit takes precedence.

NOTE 16 The unit identifier used in **code_of_unit**, **code_of_alternative_unit**, or **code_of_list_of_units** attributes can resolve to a unit provided by a server for units of measure.

- INT_CURRENCY_TYPE

The data type INT_CURRENCY_TYPE may be used for values containing values that are integer currencies.

- RATIONAL_TYPE

The data type RATIONAL_TYPE may be used for values containing values that are of type rational.

EXAMPLE 8 $\frac{1}{2}$, $\frac{3}{4}$ or $\frac{7}{2}$ are values that can be represented with RATIONAL_TYPE.

Instead of the RATIONAL_TYPE the more specific data type

- RATIONAL_MEASURE_TYPE
- may be used.

- RATIONAL_MEASURE_TYPE

The data type RATIONAL_MEASURE_TYPE may be used for values containing values that are measures of type RATIONAL.

EXAMPLE 9 A very common diameter for water pipes is "¾ inch".

In addition, the data type specifies a unit or a unit identifier, in which the values are expressed. It may also specify alternative units or alternative unit identifiers that are valid if the value is explicitly associated with one of these units.

Either a unit or a unit identifier is mandatory. In cases where both are provided, the unit takes precedence.

NOTE 17 The unit identifier used in **code_of_unit**, **code_of_alternative_unit**, or **code_of_list_of_units** attributes can resolve to a unit provided by a server for units of measure.

- **REAL_TYPE**

The data type **REAL_TYPE** may be used for values containing numbers that can be written as a terminating or non-terminating decimal; a rational or irrational number.

Instead of the **REAL_TYPE**, one of the more specific data types may be used:

- **REAL_MEASURE_TYPE**;
- **REAL_CURRENCY_TYPE**.

- **REAL_MEASURE_TYPE**

The data type **REAL_MEASURE_TYPE** may be used for values containing values that are measures of type **REAL**.

In addition, the data type specifies a unit or a unit identifier in which the values are expressed. It may also specify alternative units or alternative unit identifiers that are valid if the value is explicitly associated with one of these units.

Either a unit or a unit identifier is mandatory. In cases where both are provided, the unit takes precedence.

NOTE 18 The unit identifier used in **code_of_unit**, **code_of_alternative_unit**, or **code_of_list_of_units** attributes can resolve to a unit provided by a server for units of measure.

- **REAL_CURRENCY_TYPE**

Data type containing real values of currencies.

10.4.3 Enumeration type

The enumeration types may be used for specifying that the content of a value shall be taken from a predefined set of values.

In each designation of type where “_TYPE” is found, “_TYPE” may be omitted.

For an enumeration type, the valid data types are:

- **ENUM_CODE_TYPE**(enum_id), **ENUM_CODE_TYPE**(enum_id(code1, code2, ...))

The data type **ENUM_CODE_TYPE** may be used for values referencing a set of strings or numerals, as values, each representing assigned meanings.

NOTE 1 enum_id is a global identifier for the enumeration list.

NOTE 2 code1, code2, ... are the value codes of enumerated values.

- **ENUM_INT_TYPE**(enum_id), **ENUM_INT_TYPE**(enum_id(code1, code2, ...))

The data type **ENUM_INT_TYPE** may be used for values referencing a set of integer values as values.

NOTE 3 enum_id is a global identifier for the enumeration list.

NOTE 4 code1, code2, ... are the value codes of enumerated values.

- **ENUM_REAL_TYPE**(enum_id), **ENUM_REAL_TYPE**(enum_id(code1, code2, ...))

The data type **ENUM_REAL_TYPE** may be used for values referencing a set of real values as values.

NOTE 5 enum_id is a global identifier for the enumeration list.

NOTE 6 code1, code2, ... are the value codes of enumerated values.

NOTE 7 "ENUM_REAL_TYPE" is not available in ISO 13584-42/ IEC 61360-2, instead a constraint mechanism named "enumeration constraint" can be used for the purpose. However, the specification of the enumeration constraint is much more complex than ENUM_REAL, because the former needs to define first the data type, and then constrain the selectable subset, with an implicit understanding that only one of them is selected.

- ENUM_STRING_TYPE(enum_id), ENUM_STRING_TYPE(enum_id(code1, code2, ...))

The data type ENUM_STRING_TYPE may be used for values referencing a set of string values, as values, each representing assigned meanings.

NOTE 8 "enum_id" is the global identifier of the enumeration list.

NOTE 9 "code1", "code2", ... are the codes of the enumerated values.

- ENUM_RATIONAL_TYPE(enum_id),
ENUM_RATIONAL_TYPE(enum_id(code1, code2, ...))

The data type ENUM_RATIONAL_TYPE may be used for values referencing a set of string values, as values, each representing a rational number.

NOTE 10 "enum_id" is the global identifier of the enumeration list.

NOTE 11 "code1", "code2", ... are the codes of the enumerated values.

- ENUM_REFERENCE_TYPE(enum_id), ENUM_INSTANCE_TYPE(enum_id),
ENUM_REFERENCE_TYPE(enum_id(code1, code2, ...)),
ENUM_INSTANCE_TYPE(enum_id(code1, code2, ...))

The data type ENUM_REFERENCE_TYPE/ ENUM_INSTANCE_TYPE may be used for values referencing a set of string values, as values, each representing the global identifier of a class.

NOTE 12 "enum_id" is the global identifier of the enumeration list.

NOTE 13 "code1", "code2", ... are the codes of the enumerated values.

- ENUM_BOOLEAN_TYPE(enum_id),
ENUM_BOOLEAN_TYPE(enum_id(code1, code2, ...))

The data type ENUM_BOOLEAN_TYPE may be used for values referencing a set of Boolean values.

NOTE 14 "enum_id" is the global identifier of the enumeration list.

NOTE 15 "code1", "code2", ... are the codes of the enumerated values.

10.4.4 Class instance type

- CLASS_INSTANCE_TYPE(iclass)

The data type CLASS_INSTANCE_TYPE may be used for values containing a reference to an **Item_class**. The referenced **Item_class** may reside in any part of the classification hierarchy and contributes a collection of zero, one or more **Property** information objects. The value of the referencing **Property** shall contain the **code** of the referenced **Item_class**.

A CLASS_INSTANCE_TYPE references an **Item_class** in a way similar to CLASS_REFERENCE_TYPE (see 10.4.5), but unlike the CLASS_REFERENCE_TYPE, instances conforming to the referenced **Item_class** shall be kept together with the referencing class, i.e. are internal to the instances of the referencing class. This means if there is an **Item_class** A with instances A₁, A₂, ..., A_n, where A₁ references instance B₁ of **Item_class** B, then the instances A₂, ..., A_n of A reference to different instances of B.

In other words, CLASS_INSTANCE_TYPE assumes that each referencing instance A₁, A₂, ..., A_n of **Item_class** A is linked to its "personal" instance of B. Thus, if A₁ references B₁ and in the referenced **Item_class** instance B₁, an instance value is changed, the change is not visible from all (other) referencing **Item_class** instances A₂, ..., A_n.

NOTE 1 The "icid" in CLASS_REFERENCE_TYPE can omit "VI" (version identifier), in this case the class having the specified identifier with the latest version can be referenced.

NOTE 2 See IEC 62656-1 for detailed information on the structure of the global identifier ICID.

10.4.5 Class reference type

- CLASS_REFERENCE_TYPE(icid)

The data type CLASS_REFERENCE_TYPE may be used for values containing a reference to an **Item_class**. The referenced **Item_class** may reside in any part of the classification hierarchy and contributes a collection of zero, one or more **Property** information objects. The value of the referencing **Property** shall contain the **code** of the referenced **Item_class**.

If there are classes $A_1, A_2, \dots, A_n$ that reference one and the same external **Item_class** information object B, then the instances of $A_1, A_2, \dots, A_n$ reference to the same instance B_1 of B. In other words, CLASS_REFERENCE_TYPE assumes that the instance stored at the side of the referenced **Item_class** B, is shared among all the referencing classes $A_1, A_2, \dots, A_n$. Thus, if in the referenced **Item_class** an instance value is changed, the change is visible from all referencing **Item_class** information objects.

NOTE 1 The "icid" in CLASS_REFERENCE_TYPE can omit "VI" (version identifier). In this case the reference to the latest version of the referenced class is made.

NOTE 2 See IEC 62656-1 for detailed information on the structure of the global identifier ICID.

10.4.6 Aggregate type

Data type containing collections of two or more values. Individual values within the aggregate can be identified by a numeric index. One-dimensional sets of values may be represented as bag, list, or set. Multi-dimensional sets of values can be expressed as arrays.

The format applicable to each value of the aggregate shall be defined according to 10.4.2.

NOTE 1 An aggregate connects to a simple type by using the keyword "OF".

- BAG($b1, b2$)

A BAG data type has as its domain unordered collections of like constituents. The lower and upper bounds, which are integer-valued, define the minimum and maximum number of constituents that can be held in the collection defined by a BAG data type.

Rules and restrictions:

- The $b1$ expression shall be an integer value greater than or equal to zero. It gives the lower bound, which is the minimum number of constituents that can be in a bag. $b1$ shall not be the indeterminate value ("?").
- The $b2$ expression shall be an integer value greater than or equal to $b1$, or an indeterminate value ("?"). It gives the upper bound, which is the maximum number of constituents that can be in a bag. If this value is indeterminate ("?"), the number of constituents in a bag is not being bounded from above.
- If the boundary specification is omitted, the limits are (0,?).

- LIST($b1, b2$)

A LIST data type has as its domain sequences of like constituents. The lower and upper bounds, which are integer-valued, define the minimum and maximum number of constituents that can be held in the collection defined by a LIST data type.

Rules and restrictions:

- The $b1$ expression shall be an integer value greater than or equal to zero. It gives the lower bound, which is the minimum number of constituents that can be in a list. $b1$ shall not be the indeterminate value ("?").
- The $b2$ expression shall be an integer value greater than or equal to $b1$, or an indeterminate value ("?"). It gives the upper bound, which is the maximum number of

constituents that can be in a list. If this value is indeterminate ("?"), the number of constituents in a list is not being bounded from above.

c) If the boundary specification is omitted, the limits are (0,?).

- **UNIQUE_LIST(*b1*, *b2*)**

The value has the form of a LIST in which duplicate values shall not occur.

Rules and restrictions:

- The rules and restrictions of the LIST data type apply.
- Each constituent in an occurrence of a UNIQUE_LIST data type shall be different from (that is, not instance equal to) every other constituent in the same list.

- **SET(*b1*, *b2*)**

A SET data type has as its domain unordered collections of like constituents. The lower and upper bounds, which are integer-valued, define the minimum and maximum number of constituents that can be held in the collection defined by a SET data type.

Rules and restrictions:

- The *b1* expression shall be an integer value greater than or equal to zero. It gives the lower bound, which is the minimum number of constituents that can be in a set. *b1* shall not be the indeterminate value ("?").
- The *b2* expression shall be an integer value greater than or equal to *b1*, or an indeterminate value ("?"). It gives the upper bound, which is the maximum number of constituents that can be in a set value of this data type. If this value is indeterminate ("?"), the number of constituents in a set is not being bounded from above.
- If the boundary specification is omitted, the limits are (0,?).
- Each constituent in an occurrence of a SET data type shall be different from (that is, not instance equal to) every other constituent in the same set.

- **CONSTRAINED_SET(*b1*, *b2*, *cmn*, *cmx*)**

The CONSTRAINED_SET data type has as its domain unordered collections of constituents that may be expressed as SET of values of which subsets may be extracted. The sizes of allowed subsets are defined by their minimal and maximal values. If these sizes do not exist, any subset is allowed.

Rules and restrictions:

- The rules and restrictions of the SET data type apply.
- The *cmn* shall be an integer value greater than or equal to zero. It gives the lower bound, which is the minimum number of constituents that may be extracted. *cmn* shall not be the indeterminate value ("?").
- The *cmx* shall be an integer value greater than or equal to *cmn*, or an indeterminate value ("?"). It gives the upper bound, which is the maximum number of constituents that may be extracted.

- **ARRAY(*b1*, *b2*)**

An ARRAY data type has as its domain indexed fixed-size collections of like constituents. The lower and upper bounds, which are integer-valued numbers, define the range of index values, and thus the size of each array collection.

Given that *m* is the lower bound and *n* is the upper bound, there are exactly $n - m + 1$ constituents in the array. These constituents are indexed by subscripts from *m* to *n*, inclusive. The bounds may be positive, negative or zero, but shall not be indeterminate ("?")

Rules and restrictions:

- Both values in the bound specification, *b1* and *b2*, shall be integer values. Neither shall contain the indeterminate (?) value.

- b) *b1* gives the lower bound of the array. This shall be the lowest index which is valid for an array value of this data type.
- c) *b2* gives the upper bound of the array. This shall be the highest index which is valid for an array value of this data type.
- d) *b1* shall be less than or equal to *b2*.
- **OPTIONAL_ARRAY(*b1*, *b2*)**
The value has the form of an ARRAY in which indeterminate values ("?.") may occur.
Rules and restrictions:
 - a) The rules and restrictions of the ARRAY data type apply.
 - b) An array value of this data type may have the indeterminate value ("?.") at one or more index positions.
- **UNIQUE_ARRAY(*b1*, *b2*)**
The value has the form of an ARRAY in which duplicate values shall not occur.
Rules and restrictions:
 - a) The rules and restrictions of the ARRAY data type apply.
 - b) Each constituent in the array shall be different from (that is, not instance equal to) every other constituent in the same array.
- **UNIQUE_OPTIONAL_ARRAY(*b1*, *b2*)**
The value has the form of an ARRAY in which duplicate values shall not occur. Indeterminate values ("?.") may occur.
Rules and restrictions:
 - a) The rules and restrictions of the ARRAY data type apply.
 - b) An array value of this data type may have the indeterminate value ("?.") at one or more index positions.
 - c) Each constituent in an array shall be different from (that is, not instance equal to) every other constituent in the same array value.

10.4.7 Level type

- **LEVEL_TYPE(MIN,NOM,TYP,MAX)**
The data type LEVEL_TYPE may be used for values representing up to four variants of a numeric value which may be given as
 - INT_TYPE;
 - INT_MEASURE_TYPE;
 - REAL_TYPE;
 - REAL_MEASURE_TYPE.
 Each variant of this value expresses the number value in a specific role:
 - MIN minimum of the value;
 - NOM value as designated;
 - TYP value as typically present;
 - MAX maximum of the value.

As the interpretation of the level type values relies on the fixed sequence of the levels, all levels specified for a property have to be represented. There is no obligation to fill all levels with values although it is advised to do so for data consistency reasons.

EXAMPLE 1 In the case of having a property which is of the LEVEL_TYPE min/nom/typ/max and

- only the "MIN" value shall be transmitted, each of the other values should contain a "NIL" value;
- only the "MIN" and "MAX" values shall be transmitted, each of the other values should contain a "NIL" value.

EXAMPLE 2 In the case of having a property which is of the LEVEL_TYPE min/max – expressing a range – only those two values need be provided.

NOTE The separator sign used to separate the values of the defined levels depends on the implementation.

The use of the level type should be restricted to those **Property** information objects that are applicable in domains where the reporting of multiple values on a single characteristic is recognized as common practice.

EXAMPLE 3 Values used in industries such as the electronic component industry.

Limiting values can be expressed by defining a **Property** specifying the levels "MIN" or "MAX" or both.

EXAMPLE 4 A 12 V (nominal) car battery has 6 cells with a typical voltage of about 2,2 V each, giving a typical battery voltage of about 13,5 V. On charge, the voltage can reach a maximum of about 14,5 V but it is considered fully discharged when the voltage falls below a minimum of 12,5 V.

A specific use of LEVEL_TYPE is the concept of limiting values. When the qualification "limiting" is added to a **Property**, which may be expressed in its **definition**, the property (together with its value) specifies a threshold beyond which a device incurs damage resulting in permanent unwanted changes of functional or physical characteristics influencing the performance of the device.

EXAMPLE 5 In the case of the car battery (see EXAMPLE 4), the limiting maximum value might be specified at a value of 16,5 V.

In the context of the limiting values, the levels "NOM" and "TYP" do not apply.

Aggregates may contain LEVEL_TYPE values.

EXAMPLE 6 LIST(1,?) OF LEVEL(MIN,NOM,TYP,MAX) OF REAL_MEASURE_TYPE

10.4.8 Large object type

- LOB(*location*, *media_type*, *media_subtype*, *content_transfer_encoding*)

A large object (LOB) is an information object that is logically stored as a value of a **Property** but may be physically stored independent (mostly separated) of the **Property** information object. Large objects typically store large amounts of data.

EXAMPLE Video data, audio data, images, xml data, archives.

A value of type LOB consists of a list of four values:

- *location*: Specification of the place where the value of the large object may be accessed. This storage location should be specified in a way that all parties, i.e. the writer and the reader(s) of the LOB, can access the data.
- *media_type*: Specification of the type of the LOB value specified in accordance with the multipurpose internet mail extensions (MIME), Part 1. See IETF RFC 2045:1996.
- *media_subtype*: Specification of the subtype of the LOB value specified in accordance with the multipurpose internet mail extensions (MIME), Part 2. See IETF RFC 2046:1996.
- *content_transfer_encoding*: Specification of the character encoding of the LOB value specified in accordance with the multipurpose internet mail extensions (MIME), Part 3. See IETF RFC 2047:1996.

10.4.9 Placement type

Data type containing direction information for locating an item with respect to the coordinate system of its geometric context. Placement type values provide the information for setting up the local coordinate system required for locating the item.

NOTE 1 A detailed specification of the mathematical methods used by the placement types can be found in IEC 62656-1 and ISO 10303-42.

NOTE 2 In the case of ISO 10303-42, AXIS1_PLACEMENT, AXIS2_PLACEMENT_2D, AXIS2_PLACEMENT_3D inherit the attribute "location" that is of CARTESIAN_POINT type in 2D or 3D space, from the generic geometric entity "placement". In the case of IEC 61360-1, this is achieved by associating a condition type **Property** of the

placement type, i.e. CARTESIAN_POINT, to the geometric shape that uses either one of AXIS1_PLACEMENT_2D, AXIS1_PLACEMENT_3D, AXIS2_PLACEMENT_2D, or AXIS2_PLACEMENT_3D data types.

- **AXIS1_PLACEMENT_2D**(*ref_dir_X*, *ref_dir_Y*)
The values specify a reference direction in two-dimensional space for use as line of symmetry for bodies that are line symmetric.
 - *ref_dir_X*: Component of the direction vector of the reference direction in X direction of the geometric context coordinate system.
 - *ref_dir_Y*: Component of the direction vector of the reference direction in Y direction of the geometric context coordinate system.
- **AXIS1_PLACEMENT_3D**(*ref_dir_X*, *ref_dir_Y*, *ref_dir_Z*)
The values specify a reference direction in three-dimensional space for use as line of symmetry for bodies that are line symmetric.
 - *ref_dir_X*: Component of the direction vector of the reference direction in X direction of the geometric context coordinate system.
 - *ref_dir_Y*: Component of the direction vector of the reference direction in Y direction of the geometric context coordinate system.
 - *ref_dir_Z*: Component of the direction vector of the reference direction in Z direction of the geometric context coordinate system.
- **AXIS2_PLACEMENT_2D**(*ref_dir_X*, *ref_dir_Y*)
The values specify the local X axis of the local coordinate system used for locating an item.
 - *ref_dir_X*: Component of the direction vector of the local X axis in X direction of the geometric context coordinate system.
 - *ref_dir_Y*: Component of the direction vector of the local X axis in Y direction of the geometric context coordinate system.
- **AXIS2_PLACEMENT_3D**((*ref1_dir_X*, *ref1_dir_Y*, *ref1_dir_Z*), (*ref2_dir_X*, *ref2_dir_Y*, *ref2_dir_Z*))
The values specify the local Z axis and the direction used to determine the direction of the X axis of the local coordinate system used for locating an item.
 - *ref1_dir_X*: Component of the direction vector of the local Z axis in X direction of the geometric context coordinate system.
 - *ref1_dir_Y*: Component of the direction vector of the local Z axis in Y direction of the geometric context coordinate system.
 - *ref1_dir_Z*: Component of the direction vector of the local Z axis in Z direction of the geometric context coordinate system.
 - *ref2_dir_X*: Component of the direction vector of the direction used to determine the direction of the local X axis in X direction of the geometric context coordinate system.
 - *ref2_dir_Y*: Component of the direction vector of the direction used to determine the direction of the local X axis in Y direction of the geometric context coordinate system.
 - *ref2_dir_Z*: Component of the direction vector of the direction used to determine the direction of the local X axis in Z direction of the geometric context coordinate system.

10.4.10 Data type dependencies

As specified in 10.1, **Property** information objects can have numerical or non-numerical values according to the specifications in the attributes **data_type** and **value_format**. Nested values such as aggregates containing other aggregates may occur.

Table 4 gives an overview of the allowed combinations of **data_type** values, (see 10.4.2 to 10.4.9):

- an allowed combination is marked with: “ok”;
- a combination that is not allowed is marked with a minus sign: “–”.

Possible content of **value_format** (see 10.8) is listed, too.

EXAMPLE 1 A **Property** can have **data_type** SET (category: Aggregate type) of REAL_MEASURE_TYPE (category: Real), which is marked with "ok" in Table 4.

EXAMPLE 2 A **Property** with **data_type** LIST (category: Aggregate type) of ENUM_INT_TYPE (category: Enumeration type) is deprecated and marked with "-" in Table 4.

When specifying a new **Property**, the following sequential steps can be taken to define the complete data type by process of elimination in the given sequence:

- choose property;
- choose data type;
- choose data value range;
- choose value format.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

Table 4 – Data type dependencies

	Category of data_type (see 10.4.1)								
	Simple type	Aggregate type	Enumeration	Class instance type	Class reference type	Large object data	Placement type	Level type	
Category of content:									Content of attribute value_format (see 10.8):
Simple type	–	ok	–	–	–	–	–	^d	A, M, N, X, B, NR1, NR2, NR3, NR4, HTML5
Aggregate type	–	ok	–	–	–	^a	–	–	
Enumeration type	–	–	–	–	–	–	–	–	
Class instance type	–	^b	–	–	–	–	–	–	
Class reference type	–	^b	–	–	–	–	–	–	
Large object data	–	–	–	–	–	–	–	–	
Placement type	–	ok	–	–	–	–	–	–	
Level type	ok	–	–	–	–	–	–	–	
Boolean/ Binary	ok ^c	ok	–	–	–	–	–	–	B
Integer	ok	ok	ok	–	–	–	–	ok	NR1
Real	ok	ok	–	–	–	–	–	ok	NR2, NR3
Rational	ok	ok	–	–	–	–	–	ok	NR4
String	ok	ok	^c	^b	^b	–	ok	–	A, M, N, X, HTML5
^a Array containing the pointer to the LOB information object. ^b Pointer to Item_class information object. ^c Pointer to Value_list information object. ^d Only numeric values of type number.									

10.5 Number_of_significant_digitsAttribute name: **number_of_significant_digits**

Attribute definition: prescription of the precision that shall be obtained when assigning a numerical value to a property

Description:

Conventions and requirements:

- a) Trailing zeros without a decimal mark indicate lack of significant digits unless a decimal mark is appended.

EXAMPLE 1 "1000" has one significant digit and "1000," has four significant digits.

- b) All digits (following a non-zero digit) are significant once a decimal mark is used, so trailing zeros located to the right of the decimal mark are to be considered significant. Zeros with significant digits on each side of the decimal mark are also significant.

EXAMPLE 2 "30,4", "34,0", and "3,40" each have three significant digits but "340" has only two significant digits.

- c) Leading zeros are not treated as significant.

EXAMPLE 3 One or more leading zeroes are not treated as significant, e.g. 00257,7 has 4 significant digits, and 0,00492 has 6 significant digits.

- d) The sign of the number is independent and is not counted within the significant digits.

EXAMPLE 4 +30,4", "-34,0", and "3,40" each have three significant digits.

Obligation:

optional

Character type of values:

See Clause 26.

10.6 Referenced_class_identifier

Attribute name:

referenced_class_identifier

Attribute definition:

link to an **Item_class** information object

Description:

The class referenced contains a set of **Property** information objects. The **Property** information object from which the reference is made shall have the **value_format** "CLASS_INSTANCE_TYPE" or "CLASS_REFERENCE_TYPE" (see 10.4.4 and 10.4.5).

Obligation:

optional

Character type of values:

Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

10.7 Unit_of_measure

Attribute name:

unit_of_measure

Attribute definition:

prescription of the unit in which the value of a property should be expressed

Description:

Conventions and requirements:

- a) Preference shall be given to SI units without decimal multiplier prefixes as defined in the IEC 80000 and ISO 80000 series.
- b) Alternative units may be specified by using the code of the unit as specified in IEC TS 62720 or compliant unit dictionaries.

NOTE 1 The unit dictionaries can be supplied in computer-sensible form or as paper.

- c) The conversion to a different unit is outside the scope of this document.
- d) It is out of scope of this document to prescribe the use of a specific unit for transaction purposes including possible prefixes.
- e) It is strongly recommended not to specify the same concept with different **Property** information objects if only the prefixes of the related units are different. Exceptions may occur if certain units are common in particular business domains, such as "nm" (nanometre) in semiconductor manufacturing or "kV" (kilovolt) in power distribution area.
- f) Each unit contained in the **unit_of_measure** attribute is required to be associated with a string representation, whose text representation can be used for characterizing the unit at the instance level.

NOTE 2 **Property** information objects with **data_type** "CLASS_REFERENCE_TYPE" do not carry a unit.

EXAMPLE The IRDI of unit "%" is "0112/2///62720#UAA000#01"³.

Obligation:

conditional

Condition:

unit_of_measure shall be specified for quantitative **Property** information objects

Character type of values:

See Clause 26.

10.8 Value_format

Attribute name:

value_format

Attribute definition:

specification of type and length of the representation of the value of a **Property**

Obligation:

optional

Character type of values:

See Clause 26.

³ Use of the identifiers of the units is not restricted to IEC CDD but may be used outside IEC CDD.

Comments

The **value_format** is specified as part of a **Property** and can be used at implementation level for purposes of consistency and error checking in electronic exchange.

At implementation level, e.g. in the set-up of product information databases or electronic exchange, it can be treated as a default format that can be used as-is, re-defined (restricted or extended), or ignored entirely depending on the requirements and context in which the information is used.

For **Property** information objects of which the **data_type** has the value "class_reference_type", no **value_format** is requested.

See Annex G for the specification of syntax and content of **value_format**.

Below an overview of the content of **value_format** is provided:

a) Non-quantitative **value_format** types:

- A = alphabetic, letters only;
- M = mixed, all characters allowed;
- N = numeric, digits only;
- X = alphanumeric;
- B = binary, 0 or 1.

b) Quantitative **value_format** types:

- NR1 = integers;
- NR2 = rational numbers with decimal-mark (real numbers);
- NR3 = rational numbers with decimal-mark and exponent-mark (floating point numbers);
- S = signed (positive or negative numbers);
- . = decimal-mark;
- E = exponent-mark, base 10: (A)E(B) represents the value $A \times 10^B$.

c) Field length

The field length of a non-quantitative value shall be indicated by a number (e.g. 17). A variable field length shall start with two dots. The following preferred formats have been defined:

A..3	N..3	X..3	M..3	B 1
A..8	N..8	X..8	M..8	
A..17	N..17	X..17	M..17	
A..35	N..35	X..35	M..35	
A..($n \times 35$)	N..($n \times 35$)	X..($n \times 35$)	M..($n \times 35$)	

A fixed field length starts with one space. In such formats, no special characters shall occur.

EXAMPLE 1 "A 3", "N 8", "X 17", "M 35"

The field length of a quantitative value shall be indicated by a combination of digits and characters (e.g. "3.3ES2"). The following preferred standard formats have been defined:

NR1..4

NR1 S..4

NR2..3.3

NR2 S..3.3

NR3..3.3ES2

NR3 S..3.3ES2

A fixed field length starts with one space. In such formats no special characters shall occur.

EXAMPLE 2 "NR1 4", "NR1 S 4"

11 Condition_property

11.1 Specification of information object

Name: **Condition_property**

Definition: property information object that affects the value of another property.

Description: Any instance of **Condition_property** shall contain the value "CONDITION_DET" in the (inherited) attribute **product_data_element_type**.

Many properties have values which depend on the value(s) of one or more additional **Property** information objects.

When the value of a condition is given as a range, it shall be specified by two **Condition_property** information objects, representing the upper and lower bound of this range.

NOTE **Condition_property** is a subtype of **Property**. Thus it inherits all relationships from the attribute classes of the supertype, i.e. identifying attributes, semantic attributes, administrative attributes, value attributes, and translation data.

Figure 9 shows the detailed model of the information object "**Condition_property**" using UML as presentation technique.

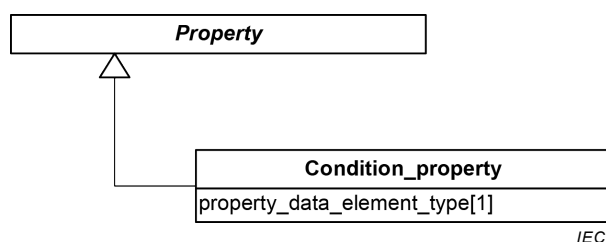


Figure 9 – Condition_property (UML class diagram)

11.2 Property_data_element_type

Attribute name:	property_data_element_type
Attribute definition:	role of the Property
Description:	One of the following roles shall be assigned: CONDITION_DET Property acting as condition
Obligation:	mandatory
Character type of values:	See Clause 26.

12 Dependent_condition_property

12.1 Specification of information object

Name:	Dependent_condition_property
Definition:	condition that further depends on one or more other conditions
Description:	Any instance of Dependent_condition_property shall contain the value "DEPENDENT_C_DET" in the (inherited) attribute product_data_element_type. Many Condition_property information objects have values which again depend on the value(s) of one or more additional properties. When the value of a Dependent_condition_property information object is given as a range, it shall be specified by two Dependent_condition_property information objects, representing the upper and lower bound of this range.

NOTE **Dependent_condition_property** is a subtype of **Condition_property** and **Property**. Thus it inherits all relationships from the attribute classes of its supertypes, i.e. identifying attributes, semantic attributes, administrative attributes, value attributes, and translation data.

Figure 10 shows the detailed model of the information object "**Dependent_condition_property**" using UML as presentation technique.

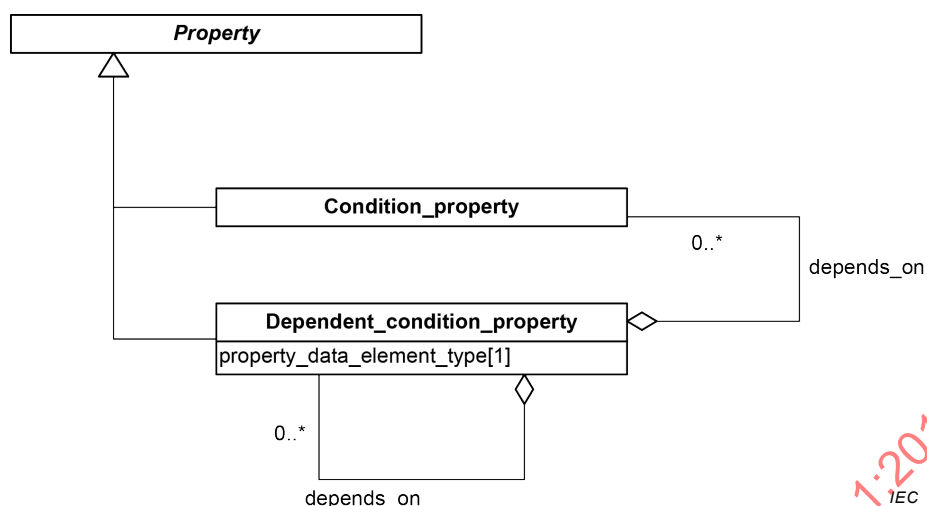


Figure 10 – Dependent_condition_property (UML class diagram)

12.2 Depends on

Attribute name: **depends_on**

Attribute definition: set of one or more identifier(s) of **Condition_property** information objects acting as conditions on which the characteristic depends that is expressed by the **Dependent_condition_property** information object.

Description: Conditions are specified in detail in Clauses 11 and 12.

Obligation: mandatory

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

12.3 Property_data_element_type

Attribute name: **property_data_element_type**

Attribute definition: role of the **Property**

Description: One of the following roles shall be assigned:

DEPENDENT_C_DET	Property acting as dependent condition
-----------------	---

Obligation: mandatory

Character type of values: See Clause 26.

13 Dependent_property

13.1 Specification of information object

Name: **Dependent_property**

Definition: property information object that is subject to restrictions expressed by one or more condition properties or dependent condition properties

Description: any instance of **Dependent_property** shall contain the value "DEPENDENT_P_DET" in the (inherited) attribute **product_data_element_type**.

NOTE **Dependent_property** is a subtype of **Property**. Thus it inherits all relationships from the attribute classes of its supertypes, i.e. identifying attributes, semantic attributes, administrative attributes, value attributes, and translation data.

Figure 11 shows the detailed model of the information object "**Dependent_property**" using UML as presentation technique.

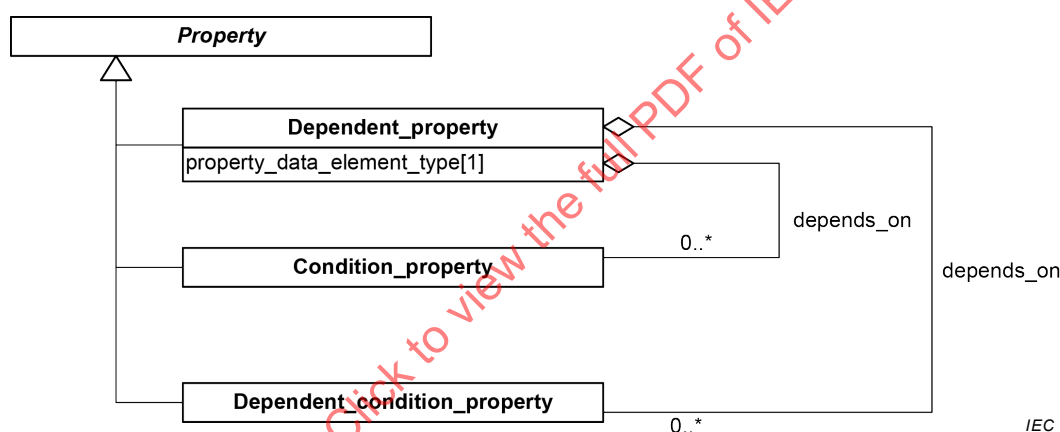


Figure 11 – Dependent_property (UML class diagram)

13.2 Depends on

Attribute name: **depends_on**

Attribute definition: set of one or more identifiers of conditions on which the characteristics of a dependent property or dependent condition depends

Description: Conditions are specified in detail in Clauses 11 and 12.

Obligation: mandatory

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

13.3 Property_data_element_type

Attribute name:	property_data_element_type	
Attribute definition:	role of the Property	
Description:	One of the following roles shall be assigned:	
	DEPENDENT_P_DET	Property acting as dependent property
Obligation:	mandatory	
Character type of values:	See Clause 26.	

14 Non_dependent_property

14.1 Specification of information object

Name:	Non_dependent_property
Definition:	property information object that is not subject to any restrictions within its defined scope
Description:	any instance of Non_dependent_property shall contain the value "NON_DEPENDENT_P_DET" in the (inherited) attribute product_data_element_type .

NOTE **Non_dependent_property** is a subtype of **Property**. Thus it inherits all relationships from the attribute classes of the supertype, i.e. identifying attributes, semantic attributes, administrative attributes, value attributes, and translation data.

Figure 12 shows the detailed model of the information object "**Non_dependent_property**" using UML as presentation technique.

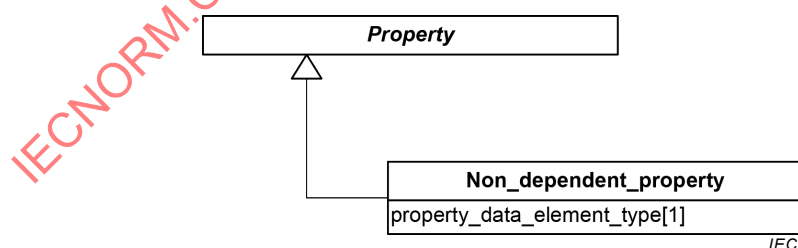


Figure 12 – Non_dependent_property (UML class diagram)

14.2 Property_data_element_type

Attribute name:	property_data_element_type
Attribute definition:	role of the Property
Description:	One of the following roles shall be assigned:

NON_DEPENDENT_P_DET **Property** acting as non-dependent property

Obligation: mandatory

Character type of values: See Clause 26.

15 Translation_information

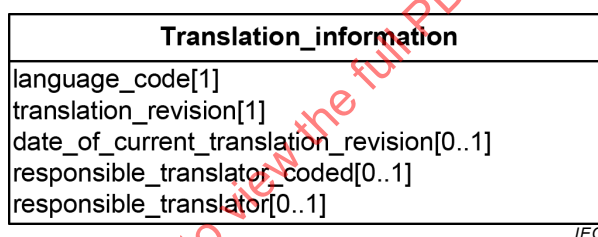
15.1 Specification of information object

Name: **Translation_information**

Definition: collection of attributes to manage localized content of information objects such as **Property** and **Item_class**

Description:

Figure 13 shows the detailed model of the information object "**Translation_information**" using UML as presentation technique.



**Figure 13 – Translation_information
(UML class diagram)**

15.2 Date_of_current_translation_revision

Attribute name: **date_of_current_translation_revision**

Attribute definition: date of the latest revision of the translation

EXAMPLE "2005-06-28"

Obligation: optional

Character type of values: date format according to YYYY-MM-DD as specified by ISO 8601

15.3 Language_code

Attribute name: **language_code**

Attribute definition: code of the language in which the translated text is written

EXAMPLE "en GB", "en US", "de DE", "de AT"

Description: The value of **language_code** shall be taken from the two-letter language code according to ISO 639-1 followed by an intermediate blank and extended with the alpha-2 country code according to ISO 3166-1 or the country subdivision code according to ISO 3166-2 to enable the possibility to distinguish between particular variants within one language.

Obligation: mandatory

Character type of values: See Clause 26.

15.4 Responsible_translator

Attribute name: **responsible_translator**

Attribute definition: the name of the organization responsible for the translation identified by its name and its two-letter country code.

EXAMPLE "DKE DE"

Description:

Obligation: optional

Character type of values: String constant as specified

15.5 Responsible_translator_coded

Attribute name: **responsible_translator_coded**

Attribute definition: coded identification of the organization responsible for the translation

EXAMPLE "0112/2"

Description: The identifier is specified in ISO 13584-26.

Obligation: optional

Character type of values: String constant as specified

15.6 Translation_revision

Attribute name: **translation_revision**

Attribute definition: revision number of the corresponding translation to administer changes

EXAMPLE "01"

Obligation: mandatory

Character type of values: Digits "0" through "9".

16 Value_list

16.1 Specification of information object

Name:	Value_list
Definition:	set of representations of permissible values of a Property
Obligation:	conditional
Condition:	Value_list shall be specified for classifying property information objects. For non-classifying property information objects a Value_list is optional.
Description:	An enumeration data type always references a Value_list for specifying the set of possible values (see 10.4.3).

Value lists that form a polymorphism (see 24.7) are considered to be restrictive, because these are exhaustive and no additional or modified value is possible.

All other value lists may be specified as being extendable by the user. For explicitly indicating the fact of having an extended value list, the option "others" shall be added to the value list in question to indicate an extended value list. The options specifying the meaning of the "others" shall be added as a condition of the property referencing the value list. This condition shall contain the additional options as an enumeration data type. In cases when the additional options are not yet specified, this enumeration remains empty.

NOTE 1 In this case the (original) enumeration type property is specified as **DEPENDENT_P_DET**. Such entries can be incorporated into the value list for a future release of the dictionary.

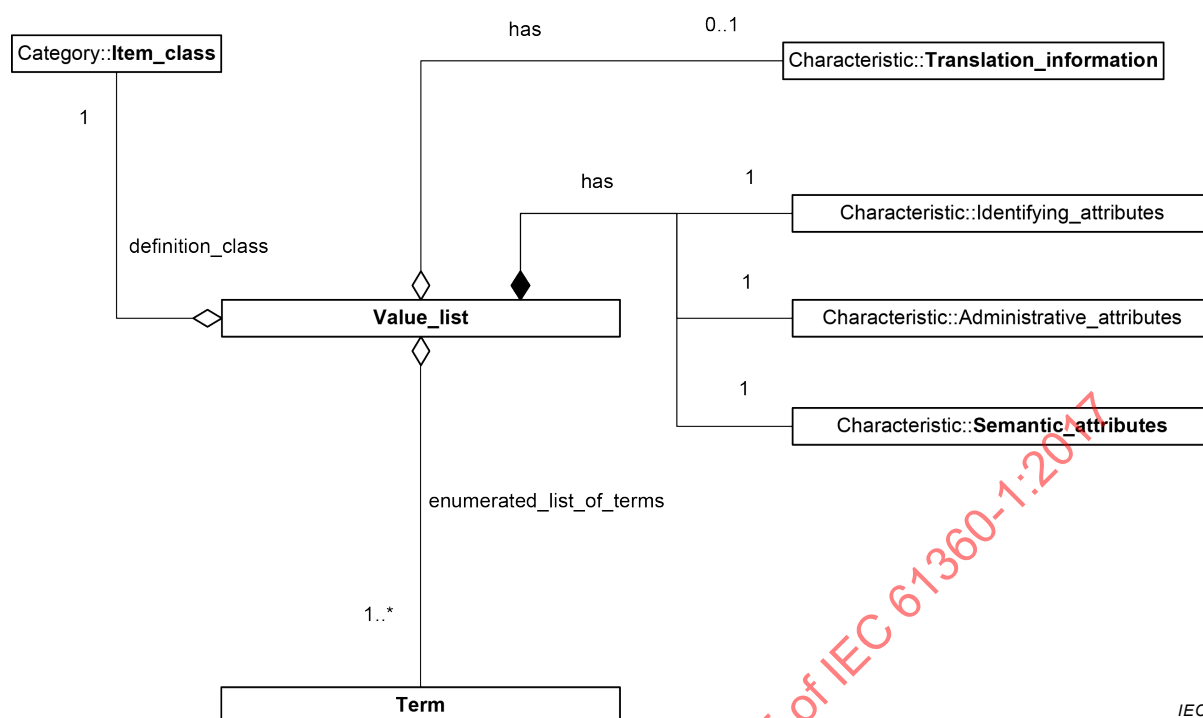
Property specialization as introduced in IEC 62656-1:2014, 5.4.5 is another technique for specifying properties with value lists differing from those contained in the ontology supplied by the original provider of the dictionary. This technique avoids any interference with the content delivered by the original provider.

NOTE 2 In cases where value lists have to be narrowed, **Relation** information objects can be used (see 24.8.2).

NOTE 3 In cases where a value list should be permanently extended, the value range can be extended by revising the enumeration information object in question.

NOTE **Value_list** can be assigned to a **Property** via the attribute **data_type**. In case of an assigned value list this attribute contains one of the enumeration data types which include the reference to the **Value_list** (see 10.4.3).

Figure 14 shows the detailed model of the information object "**Value_list**" using UML as presentation technique.



IEC

Figure 14 – Value_list (UML class diagram)

16.2 Attributes internal to Value_list

Value_list has attributes that are grouped for better readability in the information objects:

- **Identifying_attributes**;
- **Semantic_attributes**;
- **Administrative_attributes**;
- **Translation_information**.

These attribute containers are related to **Value_list** by the relationships "has". The information objects shall be kept together with the referencing class, i.e. they are internal to the instances of **Value_list**.

16.3 Definition_class

Attribute name:	definition_class
Attribute definition:	class in which the information object is defined
Description:	The class referenced by definition_class specifies the scope for the Value_list information objects. Outside of this scope the instances are invalid.
Obligation:	mandatory
Character type of values:	Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

16.4 Enumerated_list_of_terms

Attribute name: **enumerated_list_of_terms**

Attribute definition: set of references to information objects that make up in their totality the content of the value list

Description:

Obligation: mandatory

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

17 Term

17.1 Specification of information object

Name: **Term**

Definition: representation of a permissible value as an element of a **value_list**

Description:

Figure 15 shows the detailed model of the information object "**Term**" using UML as presentation technique.

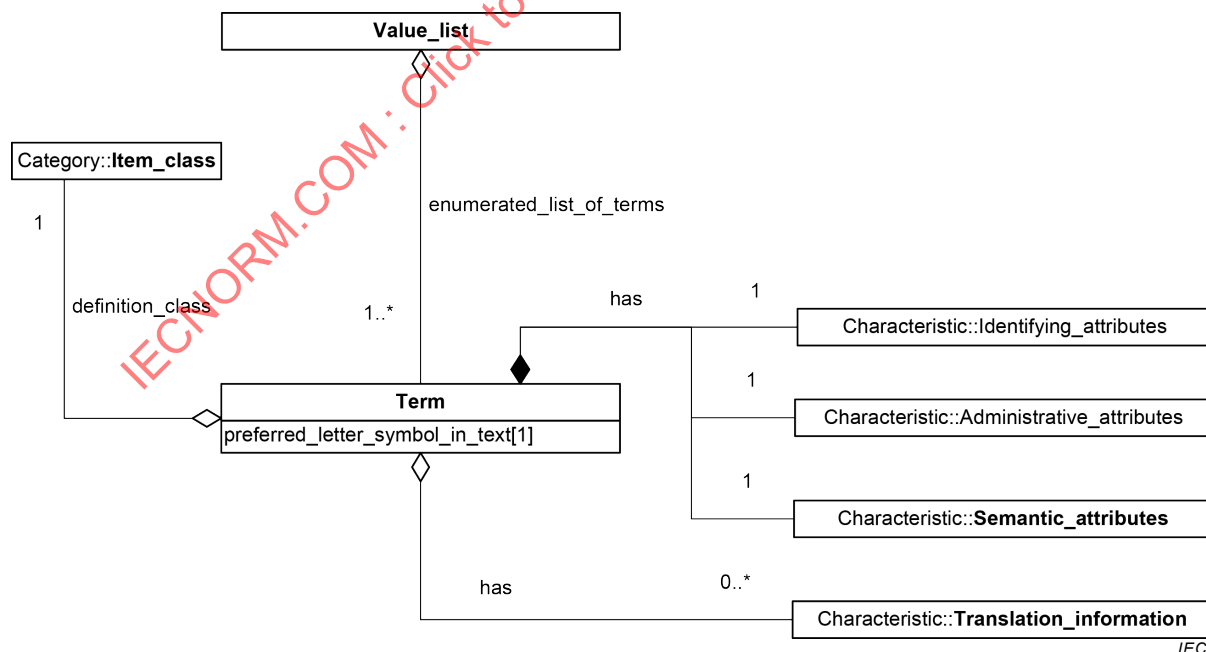


Figure 15 – Term (UML class diagram)

17.2 Preferred_letter_symbol_in_text

Attribute name:	preferred_letter_symbol_in_text
Attribute definition:	representation of a permissible value within a Value_list
Description:	If used as value code of a classifying property this value code shall be the same as the coded name of the Item_class which is created by that classifying property. The value code of a Classifying_property shall have a maximum length of 18 alphanumeric characters.
Obligation:	mandatory
Character type of values:	See Clause 26.

17.3 Attributes internal to Term

Term has attributes that are grouped for better readability in the information objects:

- **Identifying_attributes;**
- **Semantic_attributes;**
- **Administrative_attributes;**
- **Translation_information.**

These attribute containers are related to **Term** by the relationship "**has**". The information objects shall be kept together with the referencing class, i.e. they are internal to the instances of **Term**.

17.4 Definition_class

Attribute name:	definition_class
Attribute definition:	class in which the item is defined
Description:	The class referenced by definition_class specifies the scope for the Term information object. Outside of this scope the instances are invalid.
Obligation:	mandatory
Character type of values:	Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

18 Drawing

18.1 Information model

In Clause 18, the various attributes of a **Drawing** are explained. For a list of these attributes, see Table 5. These attributes are related to the identification of drawings.

Table 5 – List of attributes of Drawing

Attribute names in IEC 61360-1	Attribute names in IEC 61360-2	Subclause number
code	Code	18.3
date_of_current_translation_revision	a	15.2
descriptive_designator	a	18.4
drawing_title	a	18.5
file_format	a	18.6
file_name	a	18.7
language_code	a	15.3
obsolete_date	a	9.3
proposal_date	date_of_original_definition	9.4
published_by	a	9.6
published_in	a	9.5
responsible_translator	a	15.4
responsible_translator_coded	a	15.5
revision_number	Revision	18.8
status_level	a	9.9
translation_revision	a	15.6
version_initiated_on	a	9.10
version_number	Version	18.9
version_released_on	date_of_current_version	9.11
^a Not defined in IEC 61360-2.		

Figure 16 shows the overview model of the module "Drawing concept" using UML as presentation technique.

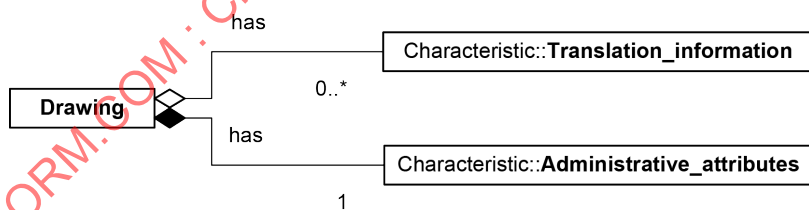


Figure 16 – Overview model for Drawing concept (UML class diagram)

18.2 Specification of information object

Name: **Drawing**

Definition: collection of identifying and administrative attributes to allow management and use of data containing pictorial information

Description:

Figure 17 shows the detailed model of the information object "**Drawing**" using UML as presentation technique.

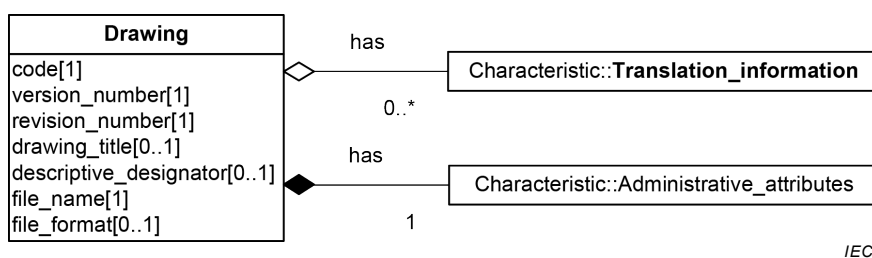


Figure 17 – Drawing (UML class diagram)

18.3 Code

Attribute name: **code**

Attribute definition: unique six-character code for a **Drawing**

NOTE IEC 61360-2:2012 specifies 14 characters as the maximum length of the attribute **code**. The difference is caused by information appended to the code such as the International Registration Data Identifier (see ISO/IEC 11179-6:2005).

Description: The first three characters shall be alphabetic, the last three numeric (format AAANNN). The character "X"⁴ shall not be used as the first character. The codes are issued sequentially and should not have any relationship with the meaning of the drawing.

Obligation: mandatory

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

18.4 Descriptive_designator

Attribute name: **descriptive_designator**

Attribute definition: coded representation of the **drawing_title**

⁴ For local use, all codes starting with XAA up to and including codes starting with XZZ may be used and are therefore excluded from the set of standard drawings.

Description: The descriptive designator is used to identify the content of a particular **Drawing**.

To ensure consistency and comprehensibility of **descriptive_designator**, coding systems may be used. The example demonstrates the use of a coding system to specify the content of drawings for semiconductor device packages.

EXAMPLE The descriptive designator has the form **XYX-Z-Tnnn-Bmmm**, where the characters have the following meanings:

- X indicates the disposition of the terminals, defined by the relevant classifying property;
- YY indicates the semiconductor device package style of the component, defined by the relevant classifying property;
- Z indicates the shape of the terminals, defined by the relevant classifying property;
- Tnnn is the terminal variant code, defined by the relevant classifying property;
- Bmmm is the optional body variant code, defined by the relevant classifying property.

The **descriptive_designator** of a **Drawing** shall have a maximum length of 17 alphanumeric characters.

Obligation: optional

Character type of values: See Clause 26.

18.5 Drawing_title

Attribute name: **drawing_title**

Attribute definition: single-word or multi-word descriptive caption applied to a drawing

Description: The **drawing_title** shall have a maximum of 70 characters

Obligation: mandatory

Character type of values: See Clause 26.

18.6 File_format

Attribute name: **file_format**

Attribute definition: the data format in which the drawing is represented

Description: More than one data format for a drawing is allowed ⁵

If the file format specified by the attribute **file_format** contradicts the file format information derived from attribute **file_name**, the format given in the attribute **file_format** shall have priority.

Obligation: mandatory

18.7 File_name

Attribute name: **file_name**

Attribute definition: the name of the file on a carrier which contains the **Drawing** data

Description: The **file_name** shall be complete, including the **file_name** extension.

If the file is stored on a remote server, the complete path of the URL (including the **file_name**) shall be provided.

Obligation: mandatory

18.8 Revision_number

Attribute name: **revision_number**

Attribute definition: number used for administrative control of a **Drawing**

Description: The **revision_number** of a **Drawing** should consist of 3 digits. Although the common ISO/IEC EXPRESS information model allows more digits, for the dictionary to be registered into IEC CDD or to be compatible with IEC 61360 series only 3 digits should be used. For specific purposes, revision numbers with more digits may be used. Consecutive revision numbers starting from "000" shall be issued in ascending order. Per **Drawing**, unique by its identifier, only one revision number is current at any moment.

A new revision of a **Drawing** shall be generated if the drawing is changed without affecting either its use or its meaning, or when corrections of typing or spelling errors have been implemented.

The **revision_number** shall be reset to its starting value "000" when the **version_number** changes.

Obligation: mandatory

⁵ It is the responsibility of the sender of a drawing to clarify the data format used. Possible formats include bit-map formats such as BMP, TIFF, GIF and JPEG, vector formats such as CGM, DXF and HP-GL and document formats such as PDF.

Character type of values: Digits "0" through "9".

18.9 Version_number

Attribute name: **version_number**

Attribute definition: number used to indicate the versions of a **Drawing** during its life cycle

Description: The **version_number** of a drawing should consist of 3 digits. Although the common ISO/IEC EXPRESS information model allows more digits, for the drawing to be registered into IEC CDD or to be compatible with IEC 61360 series only 3 digits should be used. For specific purposes, version numbers with more digits may be used. Consecutive version numbers starting from "000" shall be issued in ascending order. Per **Drawing**, unique by its identifier, only one **version_number** is current at any time.

The **version_number** of a **Drawing** shall be changed when there is a major change such as the addition of a new dimension parameter. A change to a **Drawing** version should result in a change to the version of the information object that references the **Drawing**.

Obligation: mandatory

Character type of values: Digits "0" through "9".

18.10 Attributes internal to Drawing

Drawing has attributes that are grouped for better readability in the information objects:

- **Administrative_attributes**; and
- **Translation_information**.

These attribute containers are related to **Drawing** by the relationship "**has**". The information objects shall be kept together with the referencing class, i.e. they are internal to the instances of **Drawing**.

19 Unit of measure

19.1 Overview

Quantitative **Property** information objects require information about their physical quantity and the applicable unit(s). Without the unit of measure, any assigned values are meaningless and serious errors can occur.

Attribute **data_element_type_class** (see 6.3) specifies the physical quantity of the **Property** information object. Thus preselecting suitable units is supported and the identification and assignment of the unit of measure is much simplified.

NOTE 1 Annex B lists use and available type classification codes of properties.

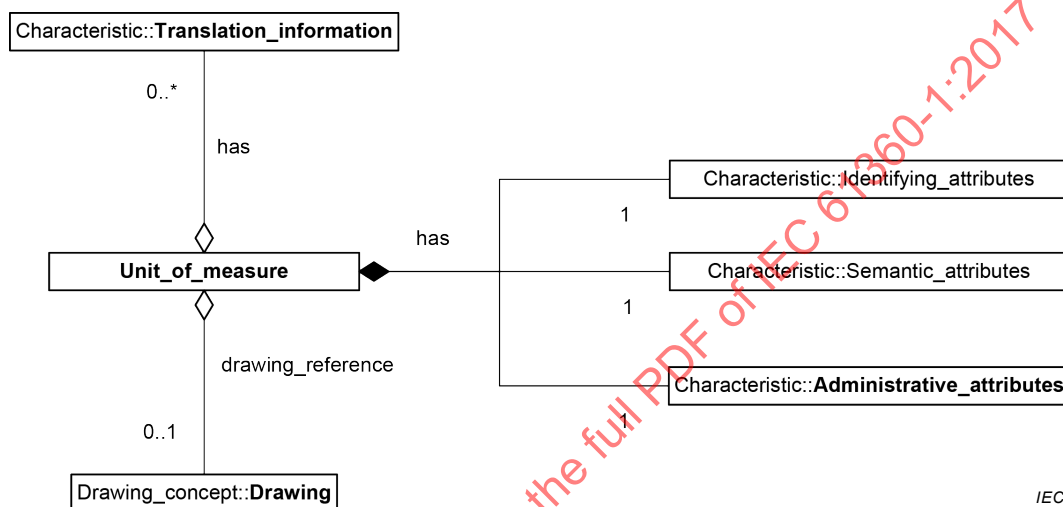
The unit of measure may be specified by using the codes specified in IEC TS 62720 or compliant unit dictionaries.

NOTE 2 IEC maintains in IEC CDD an IEC TS 62720 compliant collection of some 1 500 units grouped as physical quantities readily available for use with **Property** information objects. The **Property** attributes **code_of_unit** and **code_of_alternative_unit** allow referencing these information objects.

Additionally, system of quantities is available through **Relation** and its attribute **role_of_the_relation** = QUANTITY (see 22.7). Thus, the existing systems of units may be implemented.

NOTE 3 Currently the system(s) of units are implemented via the IEC CDD information object "List of units" which is a predecessor of the above referenced **Relation** information object.

NOTE 4 Even though being designed as integral part of IEC 61360 series compliant environments, instantiated **Unit_of_measure** information objects can also be referenced via ICID from environments other than IEC 61360 series for providing unambiguous information about unit of measure. Figure 18 shows the overview model of the module "Measure concept" using UML as presentation technique.



IEC

**Figure 18 – Overview model for Measure concept
(UML class diagram)**

19.2 Specification of information object

Name: **Unit_of_measure**

Definition: definite magnitude of a physical quantity, defined and adopted by convention or by law, that is used as a standard for measurement of the same physical quantity

EXAMPLE metre, second, degree Celsius, etc.

Description:

Figure 19 shows the detailed model of the information object "**Unit_of_measure**" using UML as presentation technique.

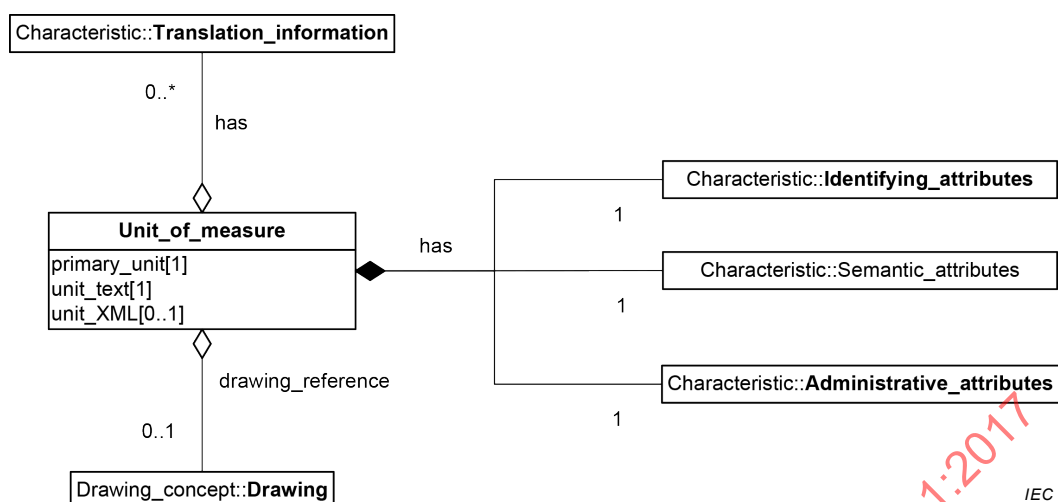


Figure 19 – Unit_of_measure (UML class diagram)

19.3 Primary_unit

Attribute name: **primary_unit**

Attribute definition: unit expressed with SI base units

Description:

Obligation: mandatory

Character type of values: See Clause 26.

19.4 Unit_in_text

Attribute name: **unit_in_text**

Attribute definition: textual presentation of the unit

Description: Units may comprise sophisticated mathematical formulae, thus the expression presented here may deviate from the mathematical correct presentation of the unit

Obligation: mandatory

Character type of values: See Clause 26.

19.5 Unit_XML

Attribute name: **unit_XML**

Attribute definition: presentation of the unit expressed with xml code

Description:

Obligation: optional

Character type of values: characters in accordance with the xml specification

19.6 Attributes internal to Unit_of_measure

Unit_of_measure has attributes that are grouped for better readability in the information objects:

- **Administrative_attributes;**
- **Identifying_attributes;**
- **Semantic_attributes;**
- **Translation_information.**

These attribute containers are related to **Unit_of_measure** by the relationships with the name "has". The information objects shall be kept together with the referencing class, i.e. they are internal to the instances of **Unit_of_measure**.

19.7 Drawing_reference

Attribute name: **drawing_reference**

Attribute definition: reference to the identifier of a drawing which contains a pictorial representation of the concept

Comments

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

20 Creation of language variants

20.1 Overview

Figure 13 shows the attributes that contain additional administrative information about localized properties or classes. These attributes shall be maintained for each localized information object.

An information object in a language other than English shall only be prepared if previously the released English reference version is available.

NOTE Due to delays in the preparation of national language versions, users can face inconsistencies between the English reference version and additional language variants.

20.2 Language-dependent attributes of a Property

The following attributes of a **Property** are language-dependent and may be translated:

- **definition;**
- **note;**
- **preferred_name;**
- **remark;**

- **short_name**;
- **synonymous_name**.

20.3 Language-dependent attributes of an **Item_class**

The following attributes of an **Item_class** are language-dependent and may be translated:

- **definition**;
- **note**;
- **preferred_name**;
- **remark**;
- **short_name**;
- **synonymous_name**.

20.4 Language-dependent attributes of a **Drawing**

The following attributes of a **Drawing** are language-dependent and may be translated:

- **drawing_title**.

20.5 Language-dependent attributes of a **Unit_of_measure**

The following attributes of a **Unit_of_measure** are language-dependent and may be translated:

- **definition**;
- **note**;
- **preferred_name**;
- **remark**;
- **short_name**;
- **synonymous_name**.

20.6 Language-dependent attributes of a **Relation**

The following attributes of a **Relation** are language-dependent and may be translated:

- **definition**;
- **note**;
- **preferred_name**;
- **remark**;
- **short_name**;
- **synonymous_name**.

21 **Item_class**

21.1 **Classification tree**

For the classification of items, the principle of dividing a set of items into subsets may be applied repeatedly, thereby creating a hierarchical tree of classes using so-called "is-a" structures. The tree shall start with the root class and consists further of superclasses and subclasses.

Classification principles are:

- a superclass should have two or more subclasses;

- a superclass should have one classifying property defining these subclasses;
- a subclass shall have exactly one superclass;
- a subclass becomes a superclass when it gets subclasses.

The topmost class (root class) of a classification tree has no superclass. This class has only subclass information objects. This topmost class shall carry the value "UNIVERSE" as **code** and shall neither be involved in any relationship nor reference any other information object.

NOTE 1 The union of two universe nodes is still a universe.

NOTE 2 Universe is a virtual class signifying the Universal Class in mathematical logic: the class of all classes.

The goal of a classification is to split the confusing set of items in the examined domain into smaller, more manageable sets that may be considered separately.

EXAMPLE One of the most successful classifications, still in use, is the classification of the kingdom of animals set up by Carl Linnaeus in the year 1735.

A property which is applicable to a class shall also be applicable to all subclasses of this class. Properties which are only applicable for a limited number of subclasses have to be specified in the relevant subclass(es).

In this way, a tree structure is created with superclasses and subclasses as shown in Figure 20. In this figure, each class shows, as dots, the collection of properties which are applicable for this class. However, the set of applicable properties in a class includes also the properties inherited from higher level classes (superclasses). So where a class is shown as empty, it nevertheless includes all inherited property information objects.

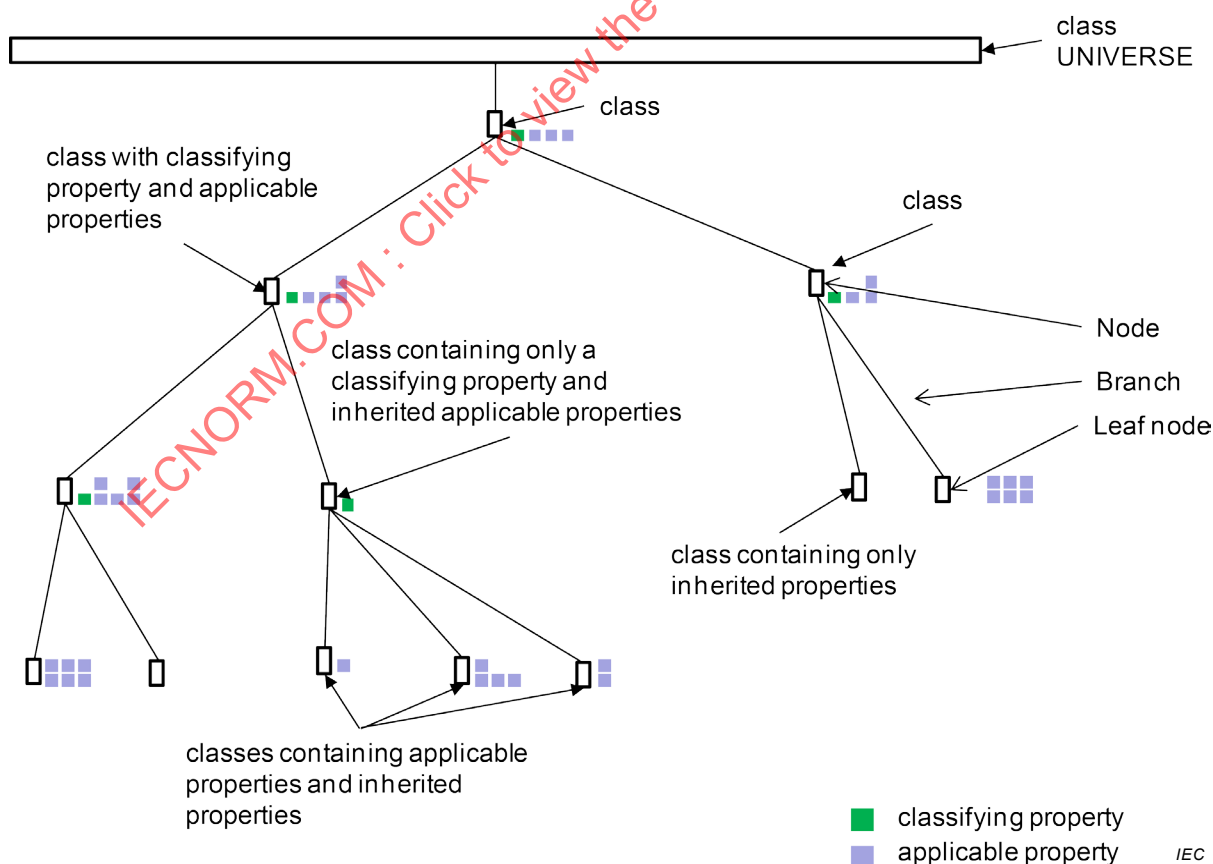


Figure 20 – Classification tree

The classification of the items described here is set up along the following lines.

- a) The hierarchical order of the classes is determined by specialization, meaning that a class always comprises its next lower level subclass(es). The classification ends at the level where no further division of items shall be made.
- b) From the collection of properties applicable at a particular node, one property is the classifying one.
- c) The classifying properties, applicable to all classes with subclasses in the tree, shall have the following characteristics:
 - each one comprises a single elementary attribute and not a combination of different attributes, i.e. they have homogeneous value lists;
 - the subclasses designated by the values of a classifying property are complementary; they fully cover in their totality the whole field specified by their superclass without overlap. So-called multiple value fields, where two or more values apply at the same time, are not allowed;
 - the value codes of classifying properties shall be unique within the whole classification scheme;
 - no garbage classes are allowed, such as "miscellaneous", "remaining", "various";
 - any relevant attributes may be used for classification; their choice and order are determined by whether or not they lead to specific properties which are relevant to the users;
 - the classification criteria used are mostly related to functionality, but other criteria may occur such as technology, application, material, or geometry; the choice may vary within a classification tree between items of different type;
 - classifying properties have always the data type value: "ENUM_CODE_TYPE".
- d) For all classifying properties, visibility and applicability is ensured:
 - a property is specified as applicable for a class if it is explicitly associated to that class. It shall satisfy an essential characteristic of the items represented by that class;
 - a condition becomes applicable to a class when it is referenced by one of the properties relating to the class.
- e) The terms which relate to the branches of the tree, being a value of the appropriate classifying property, shall have the following characteristics:
 - they are precise – no vague or ambiguous terms such as "general purpose", "economical", "high speed", etc. are allowed;
 - they have a clearly defined meaning;
 - they conform to an international standard where available;
 - they are free from synonymy – where synonyms do occur, one term shall be selected as the preferred name and the others refer to it;
 - the specific meaning of homonyms shall be explained by context indications.

When the content of IEC CDD is extended, new sets of classes and properties have to be added to IEC CDD. Four approaches may be applied.

- 1) Adding individual classes and properties at various points within the existing class tree of IEC CDD.
- 2) Adding complete branches to the IEC CDD class tree(s). Within the branches the new classes shall have domain specific structures suitable for the concepts characterized.
- 3) Adding complete tree(s) to the IEC CDD data structure that completely cover the domain in question. In many cases domains (re-)use concepts from neighbouring domains; for example, the domain of process automation can embrace capacitors, which are contained in the domain of electronic components. To avoid duplicated classes or properties, the class "capacitors" should be imported using the case-of relationship.
- 4) Adding complete tree(s) without any specific relation to already existing classes. This case is not desirable because duplicates of classes or properties in the IEC CDD data can

occur. Such trees should be subjected to revision and migrated to approach 3) in a mid-term timeframe.

When extending the dictionary with content of a new domain the UNIVERSE class plays an important role. The UNIVERSE class has no restrictions by definition. Thus, the UNIVERSE class embraces all other classes by definition and any new trees shall have this class as topmost class.

Table 6 – List of attributes of Item_class as defined in IEC 61360-1 and their equivalent in IEC 61360-2

Attribute names in IEC 61360-1	Attribute names in IEC 61360-2	Subclause number
applicable_data_element_type	described_by	21.9
applicable_relation	a	21.10
classifying_data_element_type	a	21.11
class_type	a	21.6
code	code	7.3
coded_name	short_name	21.7
definition	definition	8.3
drawing_reference	simplified_drawing	21.12
note	note	8.4
obsolete_date	is_deprecated/ date_of_current_version	9.3
preferred_name	preferred_name	7.4
proposal_date	date_of_original_definition	9.4
published_in	a	9.5
published_by	a	9.6
remark	remark	8.5
responsible_committee	a	9.7
revision_number	revision	7.5
revision_released_on		9.8
source_document_of_definition	source_doc_of_definition	8.6
status_level	a	9.9
superclass	its_superclass	21.14
synonymous_name	synonymous_name	7.7
version_initiated_on	a	9.10
version_number	version	7.8
version_released_on	date_of_current_version	9.11
^a Not defined in IEC 61360-2:2012.		

21.2 Composition tree

A composition tree expresses the part-whole structure between an item and its members such as between a product and its parts or between an information collection and its constituents.

EXAMPLE 1 The bill of material of a part represents a composition tree.

These relationships between an item and its members are called "has-a" relationships. The "has-a" relationship is implemented by a **Property** information object of type "CLASS_INSTANCE_TYPE" or "CLASS_REFERENCE_TYPE" (see 10.4.4 and 10.4.5).

In Figure 21 A100 is made up of two constituents B110 and B120, whereas B120 is again made from C111 and C112. There is no inheritance within composition trees.

EXAMPLE 2 In Figure 21 A100 can be a flow meter consisting of two modules, an input interface and an output interface, the output interface offering two configurations: 20 mA interface (C111) and 0-24 V interface (C112).

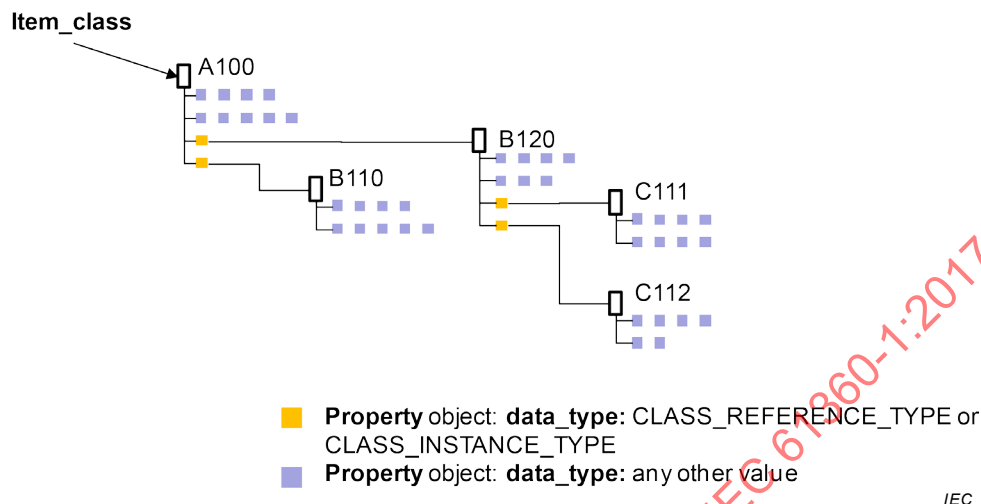


Figure 21 – Composition tree

Table 6 lists the various attributes of classes as encountered in the specifications. These attributes are related to identification, description and to relationships.

21.3 Use of auxiliary schemes for classification and coding of values

To ensure consistency and comprehensibility of properties, the use of coding systems and classification schemes for definitional parts and associated values is recommended. The example demonstrates the use of a classification scheme to specify the shapes of semiconductor device package outlines.

NOTE When used, auxiliary schemes are referenced in the "note" attribute.

EXAMPLE The classification scheme for the semiconductor device package outlines is based on the following three characteristics of the semiconductor device package according to the codes defined in IEC 60191-4:

- outline style;
- terminal position;
- terminal form.

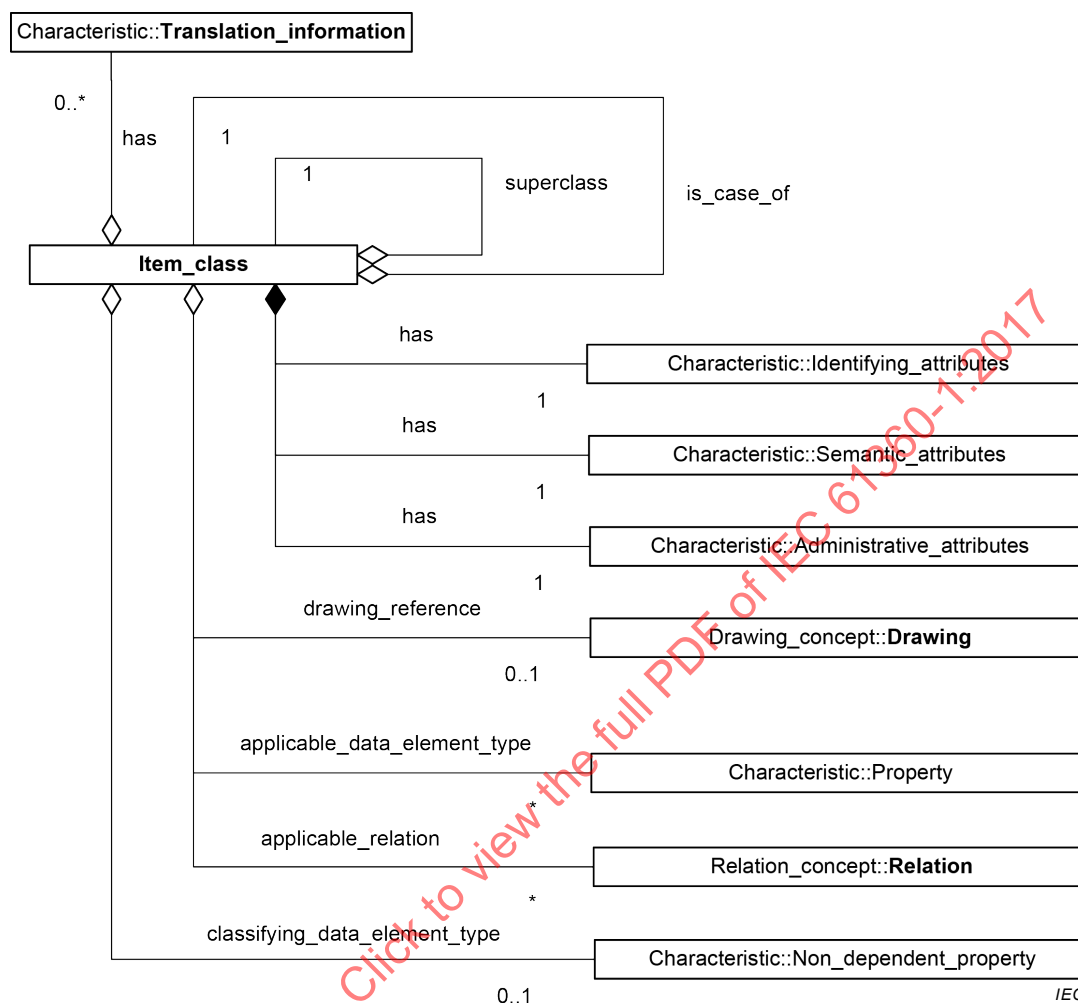
Based on these characteristics, five levels of classification are defined as follows:

- level 1: outline style;
- level 2: terminal position;
- level 3: terminal form;
- level 4: terminal variant;
- level 5: body variant (optional).

The specification of each class at the lowest level of the classification tree of the semiconductor device package contains an attribute which makes a reference to a unique drawing graphically specifying the dimension parameters. See IEC 60191-4:2013, Clause 4 for the description of the parameters.

21.4 Information model

Figure 22 shows the model of the information object "**Concept_data**".



**Figure 22 – Overview model for Concept_data
(UML class diagram)**

21.5 Specification of information object

Name: **Item_class**

Definition: description of a set of information objects that share the same characteristics

Description:

Figure 23 shows the detailed model of the information object "**Item_class**" using UML as presentation technique.

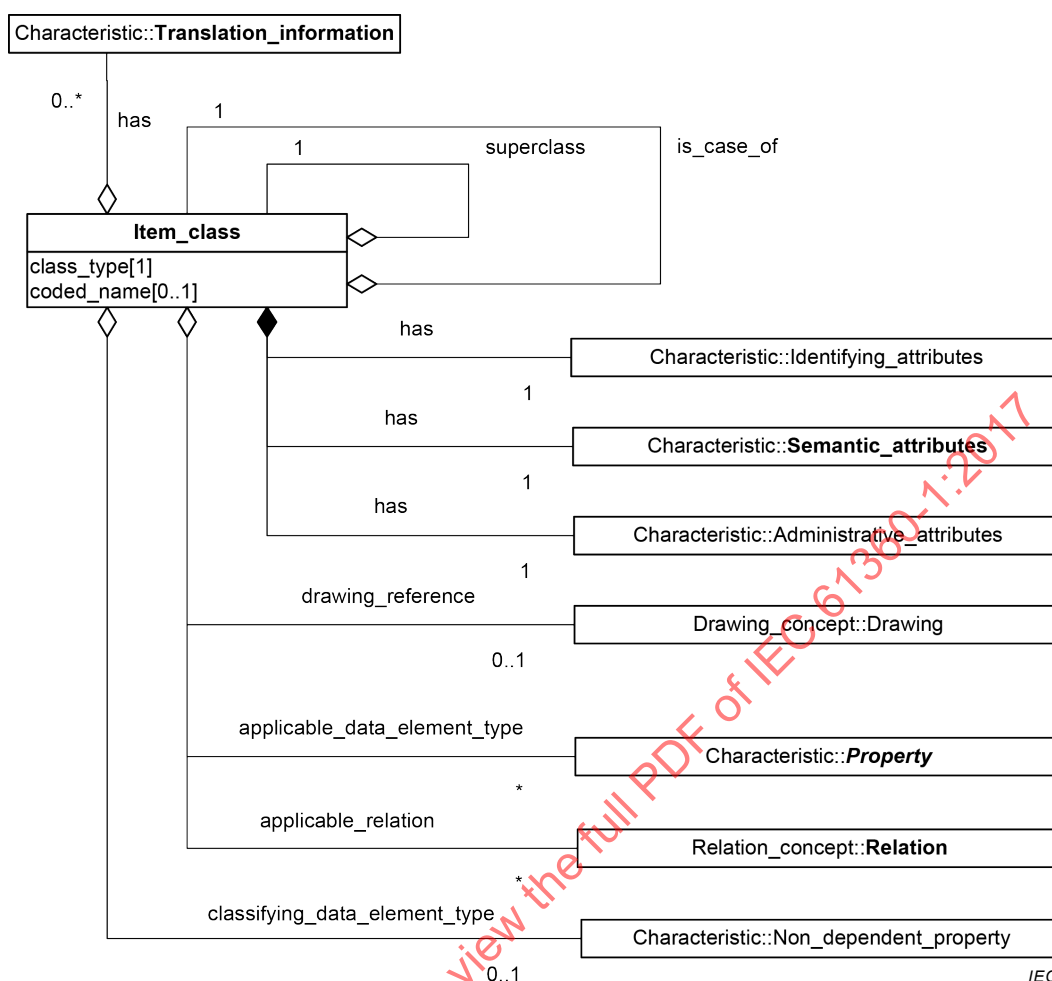


Figure 23 – Item_class (UML class diagram)

21.6 Class_type

Attribute name: **class_type**

Attribute definition: additional information about the type of hierarchy containing the class

Description: Allowed values:

ITEM_CLASS class is part of an is-a hierarchy and related to its superclass by inheritance (see IEC 61360-2:2012, 5.8)

CATEGORICAL_CLASS class is part of a categorization hierarchy and related to its superclass by **is_case_of** (see IEC 61360-2:2012, 5.8)

FEATURE_CLASS All classes below class "AAA233 – Features" represent commonly used concepts that may be used independently or be referenced from other classes.

CLASS_INSTANCE_REFERENCE should be used if referencing is applied.

Obligation: mandatory

Character type of See Clause 26.
values:

NOTE **COMPONENT_CLASS** and **MATERIAL_CLASS** were defined within the dictionary model of IEC 61360-1:1995. These types are deprecated, and they are removed from this document. (See also IEC 61360-2:2012, 5.8.1.)

21.7 Coded_name

Attribute name: **coded_name**

Attribute definition: coded representation of the **preferred_name** which shall be equal to the corresponding value code (see 21.7) of the classifying property

Description: The first character of the **coded_name** shall be a letter. The **coded_name** of an **Item_class** shall have a maximum length of 18 alphanumeric characters

Obligation: mandatory

Character type of values: See Clause 26.

21.8 Attributes internal to Item_class

Item_class has attributes that are grouped for better readability in the information objects:

- **Administrative_attributes;**
- **Identifying_attributes;**
- **Semantic_attributes;**
- **Translation_information.**

These attribute containers are related to **Item_class** by the relationship "**has**". The information objects shall be kept together with the referencing class, i.e. they are internal to the instances of **Item_class**.

21.9 Applicable_data_element_type

Attribute name: **applicable_data_element_type**

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

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

NOTE 2 The above definition is conceptual; there is no requirement for using all applicable properties of a class for describing each part of this class at the data model level.

NOTE 3 All the applicable properties of a superclass are also applicable properties for the subclasses of this superclass.

NOTE 4 Only properties defined or inherited as visible and imported properties of a class can be applicable properties.

[The definition and Notes are taken from IEC 61360-2:2012, 3.2]

Obligation: conditional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

Being listed as **applicable_data_element_type** or being inherited from a supertype does not imply that a specific **Property** is optional or mandatory. There is no obligation for the user to provide values for any **Property** information object.

NOTE 5 Contracts between the owner of a database and the user(s) can stipulate the existence of mandatory content within the database in question.

21.10 Applicable_relation

Attribute name: **applicable_relation**

Attribute definition: **Relation** defined for an **Item_class**

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

Being listed as **applicable_relation** does not imply that a specific **Relation** is optional or mandatory.

NOTE Contracts between the owner of a database and the user(s) can stipulate the existence of mandatory content within the database in question.

This attribute is not subject to inheritance.

21.11 Classifying_data_element_type

Attribute name: **classifying_data_element_type**

Attribute definition: **Value_list** applicable for a particular **Item_class**, addressing a single elementary property having a value list, whose values define the subclasses

Description: The value codes (i.e. **preferred_letter_symbol_in_text**, see 17.2) of the values shall be the same as the **coded_names** (see 21.7) of the subclasses. It is also desirable that the **preferred_name** of a subclass is the same as the **preferred_name** associated with the value code.

See Annex E for additional information.

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

21.12 Drawing_reference

Attribute name: **drawing_reference**

Attribute definition: reference to the identifier of a **Drawing** which illustrates the meaning of an **Item_class**

Description: The identifier of a **Drawing** shall consist of the combination of the six-character drawing code, followed by a hyphen followed by the three digit version number of the drawing references.

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

21.13 Is_case_of

Attribute name: **is_case_of**

Attribute definition: (other) class that partially or fully conforms to the specification of the class

Description: **is_case_of** allows the reuse of already defined classes.

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

21.14 Superclass

Attribute name: **superclass**

Attribute definition: class that is designated as the canonical parent class of the class

Description: If no higher level **Item_class** information object exists the term "UNIVERSE" shall be used.

Obligation: mandatory

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

22 Relation

22.1 Overview

Relation is used to describe a named relation with a globally unique identifier. **Relation** embraces two categories; one is called PREDICATION, and the other is called FUNCTION. The difference between them is that the PREDICATION has a domain but no codomain, whilst the FUNCTION shall have domain and codomain.

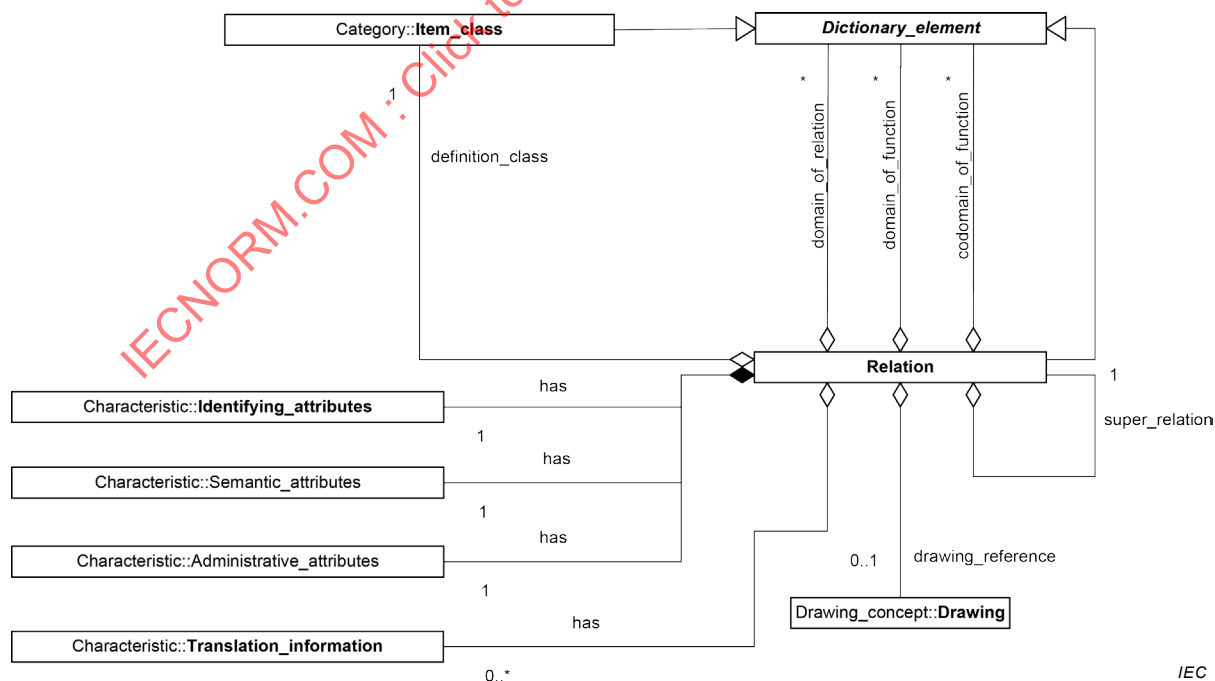
NOTE When a relation R exists among $S_1, S_2, \dots, S_n$ then R can be mathematically equated as a set R that is a subset of the Cartesian product of the n sets, i.e. $R \subseteq S_1 \times S_2 \times \dots \times S_n$. Of course, from a mathematical point of view, a function as well as a predication is a kind of relation, and thus the basic characteristics of the relation are maintained in both the function and predication.

Different relationships among information objects may be modelled as instances of relation:

- constraint on one or more properties, as an extension of a single property constraint;
- grouping mechanism among information objects, in particular among heterogeneous ones;
- mapping, correspondence, or transition among information objects.

Each role is designated by the attribute **role_of_the_relation**.

Figure 24 shows the overview model of the module "Relation_concept" using UML as presentation technique.



IEC

**Figure 24 – Overview model for Relation concept
(UML class diagram)**

22.2 Specification of information object

Name: **Relation**

Definition: information object connecting two or more instances of type class or property for the purpose of applying specific side conditions

Description:

Figure 25 shows the detailed model of the information object **"Relation"** using UML as presentation technique.

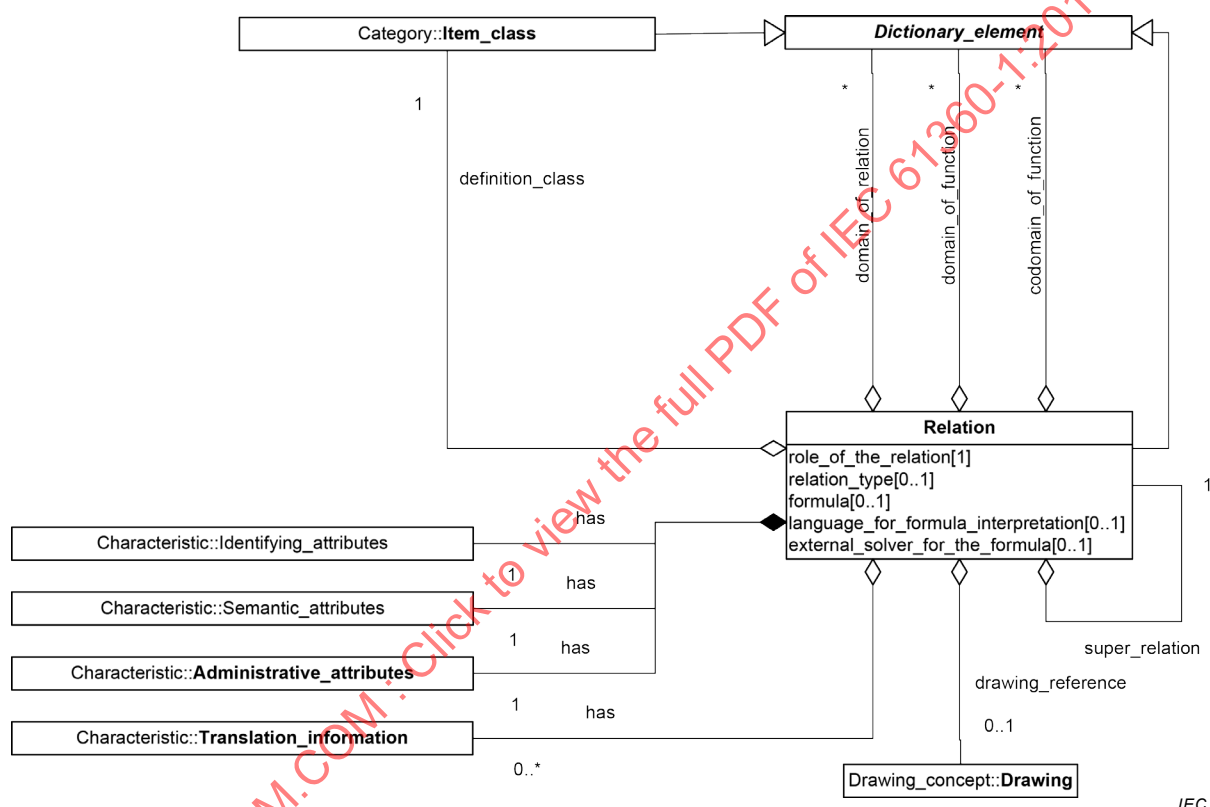


Figure 25 – Relation (UML class diagram)

22.3 External_solver_for_the_formula

Attribute name: **external_solver_for_the_formula**

Attribute definition: reference to an external solver of content of **formula** that is passed to the environment of the parcel system and from which the value of the function is returned

Description:

Obligation: optional

Character type of values: See Clause 26.

22.4 Formula

Attribute name: **formula**

Attribute definition: logical or mathematical formula by which the constraining effect on the domain or on the value of the codomain, in case of a functional relation, or on both may be computed

<code>enum_constraint (value1, value2, ...)</code> ^a	ENUMERATION
<code>string_pattern_constraint (pattern)</code> ^b	STRING_TYPE
<code>string_range_constraint (min, max)</code> ^c	STRING_TYPE
<code>range_constraint (</code> <code>min,</code> <code>max,</code> <code>OPTIONAL min_inclusive,</code> <code>OPTIONAL max_inclusive</code> ^d , <code>OPTIONAL complement</code> ^e <code>)</code>	INT, REAL, RATIONAL
<code>cardinality_constraint (bound1, bound2)</code> ^f	Aggregate types
<code>subclass_constraint (icid1, icid2,...)</code> ^g	CLASS_REFERENCE, or CLASS_INSTANCE
<code>subtype_constraint (type1, type2,...)</code> ^h	ENTITY INSTANCE

^a *value1, value2*: one of the elements of an enumeration

^b *pattern*: a regular expression

^c *min, max*: INTEGER

^d *min_inclusive, max_inclusive*: BOOLEAN, its default value is TRUE

^e *Complement*: BOOLEAN, its default value is FALSE

^f *bound1, bound2*: cardinality boundary

^g *icid1, icid2, ...*: a list of ICIDs of classes which the CLASS_REFERENCE or CLASS_INSTANCE type property is allowed to reference

^h *type1, type2,...* data types as defined in ISO 10303-11:2004

Description:

Obligation: optional

Character type of values: See Clause 26.

22.5 Language_for_formula_interpretation

Attribute name: **language_for_formula_interpretation**

Attribute definition: computer language or logical description system by which the content of **formula** can be interpreted

Description:

Obligation: optional

Character type of values: See Clause 26.

22.6 Relation type

Attribute name: **relation_type**

Attribute definition: type of the relation

allowed values:

FUNCTION

relation r such that for any entity a there is exactly one entity b to which a is related by r . [SOURCE: IEC 60050-103:2009, 103-01-01, modified – Notes omitted]

PREDICATION

Boolean-valued function $P: X \rightarrow \{\text{true}, \text{false}\}$ that amounts to the characteristic function or the indicator function of the relation

Description:

Obligation: mandatory

Character type of values: See Clause 26.

22.7 Role_of_the_relation

Attribute name: **role_of_the_relation**

Attribute definition: specific use of the relation

Allowed values:

ARROW

Relation represents a transition from one collection of information objects to another collection of (different) information objects.

The source shall be specified in **domain_of_the_function** and the destination shall be designated in the **codomain_of_the_function**.

relation_type shall be set to "FUNCTION".

NOTE 1 Attribute **codomain_of_the_function** can be empty.

PACKAGE

The information objects listed in **domain_of_the_function** are information objects that are considered to be of the same kind and belonging together.

relation_type shall be set to "FUNCTION". Input is **domain_of_function**, the package is always referenced in **codomain_of_function**.

EXAMPLE 1 A "package" used by the object-oriented computer language "Ada".

EXAMPLE 2 "Module" in "Modula-2" and "C++".

CONSTRAINT

IEC 61360-2/ISO 13584-42 compliant value constraint. The relation is used to model a numeric constraint on a property or on a set of properties.

relation_type shall be set to "PREDICATION".

NOTE 2 In the CONSTRAINT case, relation is assigned to the **Item_class** via the attribute **applicable_relation**.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

QUANTITY

Relation is used to model the concept of a physical quantity comprising a set of units of measurement.

relation_type shall be set to "PREDICATION" or "FUNCTION".

The attribute **domain_of_the_relation** collects the IDs of units of measurement.

EXAMPLE 3 Units "millimetre", "micrometre", and "nanometre", etc., including "metre", to cover the quantity "SI length measure".

The attribute **codomain_of_the_function** contains the ID of the coherent SI unit.

NOTE 3 **codomain_of_the_function** can be empty.

EXAMPLE 4 "metre" as coherent SI unit of length measure.

The **role_of_the_relation** as a quantity may be used to model a hierarchy among quantities. The attribute **super_relation** then contains the link to the next higher level relation.

EXAMPLE 5 One **Relation** information object can comprise all length measures of SI units such as "metre", "millimetre", "micrometre", or "nanometre". The other relation information object comprises Imperial length measures with Imperial units, such as "mile", "yard", "foot", or "inch". A third relation information object which is referenced by the content of the "super relation" attributes of the "SI relation" and the "Imperial relation" represents the generic quantity "length measure" comprising SI length measure and Imperial length measure units.

NOTE 4 Future editions of IEC 61360-1 might extend this list of keywords.

NOTE 5 Other standards building upon this document might extend this list of keywords in a way which conforms to the semantic of the relation.

Description:

Obligation: mandatory

Character type of values: See Clause 26.

22.8 Attributes internal to Relation

Relation has attributes that are grouped for better readability in the information objects:

- **Administrative_attributes;**
- **Identifying_attributes;**
- **Semantic_attributes;**
- **Translation_information.**

These attribute containers are related to **Relation** by the relationship "**has**". The information objects shall be kept together with the referencing class, i.e. they are internal to the instances of **Relation**.

22.9 Definition_class

Attribute name:	definition_class
Attribute definition:	Item_class that has a scope in which the relation is meaningful
Obligation:	mandatory
Character type of values:	Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

22.10 Codomain_of_function

Attribute name:	codomain_of_function
Attribute definition:	ID of the property or class information object(s) that serve as codomain (value range) of a function

NOTE 1 What possibly can come out of a function is called the codomain. What actually comes out of a function is called the range.

NOTE 2 **codomain_of_function** can be empty.

Description:	
Obligation:	conditional
Character type of values:	Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

22.11 Domain_of_function

Attribute name:	domain_of_function
Attribute definition:	list of IDs of property or class information object(s) that serve as domain for the functional relation identified by Relation

NOTE What can go into a function is called the domain.

Description:	
Obligation:	conditional
	Either this or domain_of_relation shall be used for function
Character type of values:	Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

22.12 Domain_of_relation

Attribute name: **domain_of_relation**

Attribute definition: list of IDs of property or class information object(s) that serve as the domain for the **Relation** identified by the relation IDs

NOTE What can go into a relation is called the domain.

Description:

Obligation: conditional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

22.13 Drawing_reference

Attribute name: **drawing_reference**

Attribute definition: reference to a **Drawing** illustrating the meaning of the **Relation**

Comments

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

22.14 Super_relation

Attribute name: **super_relation**

Attribute definition: generic type of the **Relation**

Description: If no higher level **Relation** information object exists the term "UNIVERSE" shall be used.

Obligation: optional

Character type of values: Upper-case Latin letters "A" through "Z", except the upper-case Latin letters "O" and "I" plus digits "0" through "9".

23 Dictionary_element

Name: **Dictionary_element**

Definition: generalization of **Item_class**, **Property**, **Unit_of_measure**, and **Relation** for the purpose of being able to create relations among them

Description:

Figure 26 shows the detailed model of the information object "**Dictionary_element**" using UML as presentation technique.

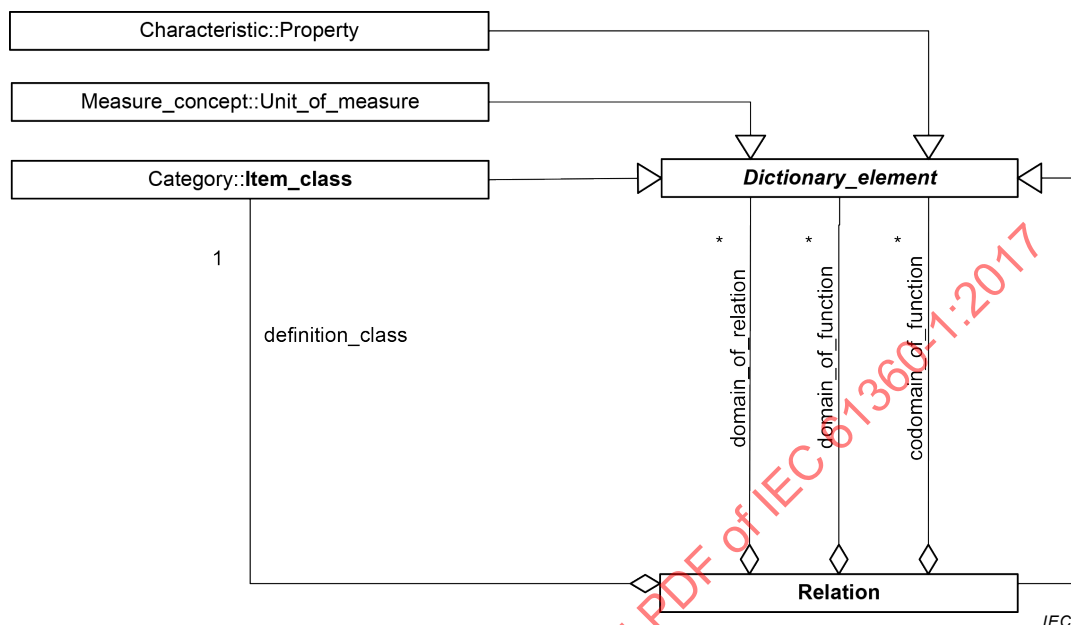


Figure 26 – Dictionary_element (UML class diagram)

24 Advanced concepts

24.1 Overview

To allow effective and error free specification or use of dictionary elements, concepts are provided that support constraining or reuse of already specified properties, grouping, or description of configurable items.

24.2 Condition

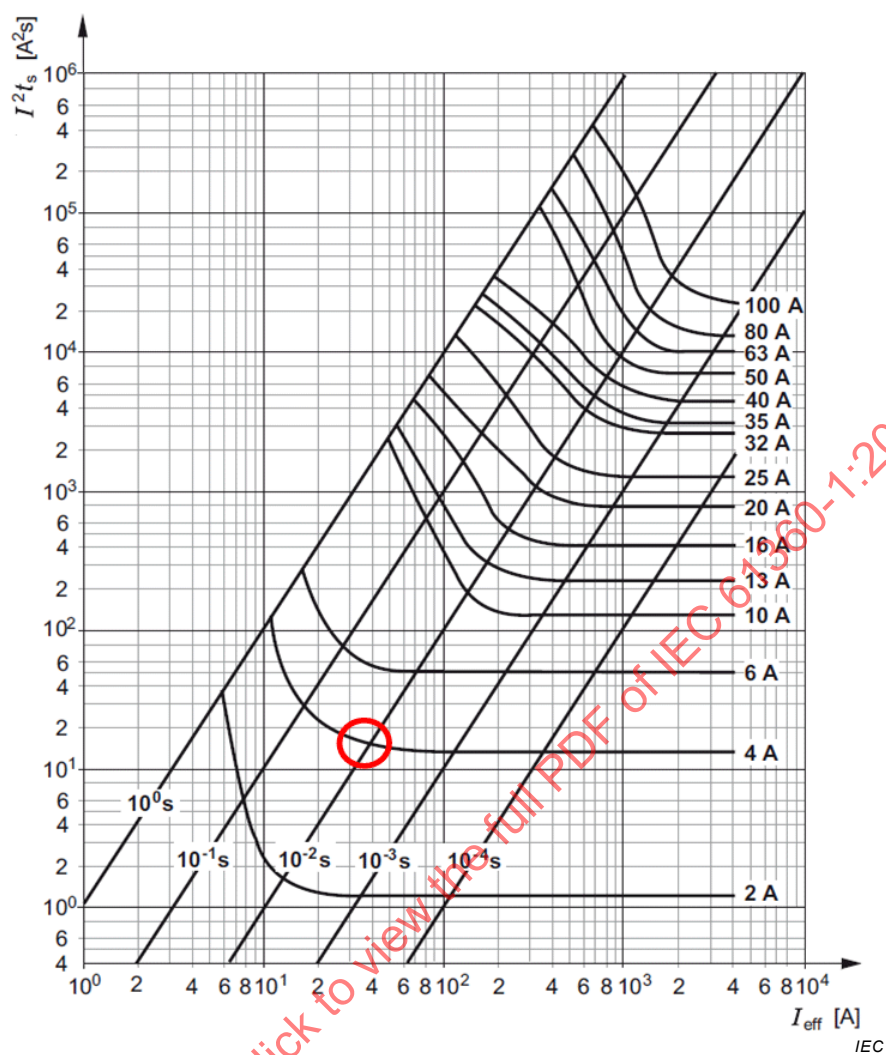
In many cases statements or propositions are valid only if additional information is available such as side conditions or initial states. Conditions allow specifying information that has to be known for interpretation of a given value without ambiguities.

A **Property** may refer to any number of conditions. Conditions represent always additional information. Conditions may again be subject to other conditions.

NOTE This document does not provide concepts such as optional conditions or mandatory conditions. Nevertheless preconditions can be expressed. In the example below, the condition "Response time" acts as precondition for the conditions "Melting time" and "Arc quenching time".

Empty conditions, i.e. condition **Property** information objects without associated value, may exist.

EXAMPLE 1 When specifying the I^2t_s value of a fuse, several side conditions have to be considered (see Figure 27 and Figure 28).

**Key** I^2t_s melting I^2t value I_{eff} r.m.s. value of the prospective short circuit current**Figure 27 – Working point of a fuse**

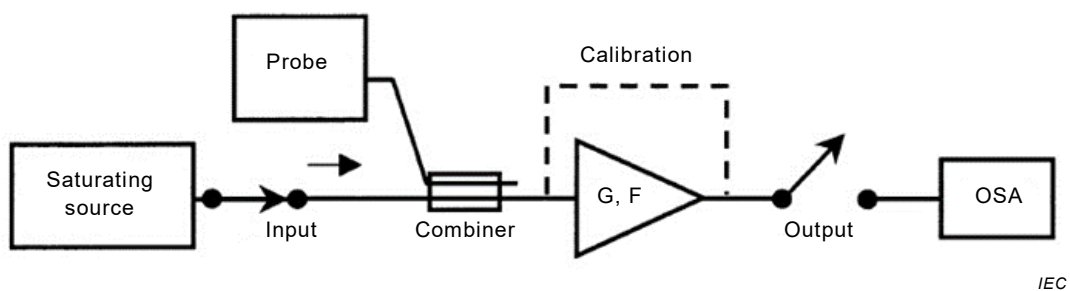
#CLASS_ID:= MDC_C003						
#CLASS_NAME.en:= Property meta-class						
#PROPERTY_NAME.en	code	preferred_name	formula	property_data_element_type	condition	unit_of_measure
#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en	MDC_P024	MDC_P020	MDC_P028	MDC_P023
	PAA300	melting i^2t value		DEPENDENT_P_DET	PAA301,PAA302,PAA305	A²s
	PAA301	r.m.s. value of the protective short circuit current		CONDITION_DET		A
	PAA302	rated current		CONDITION_DET		A
	PAA303	response time	$T=f(t_s, t_i)$	DEPENDENT_C_DET	PAA303,PAA304	s
	PAA303	melting time		CONDITION_DET		s
	PAA304	arc quenching time		CONDITION_DET		s

IEC

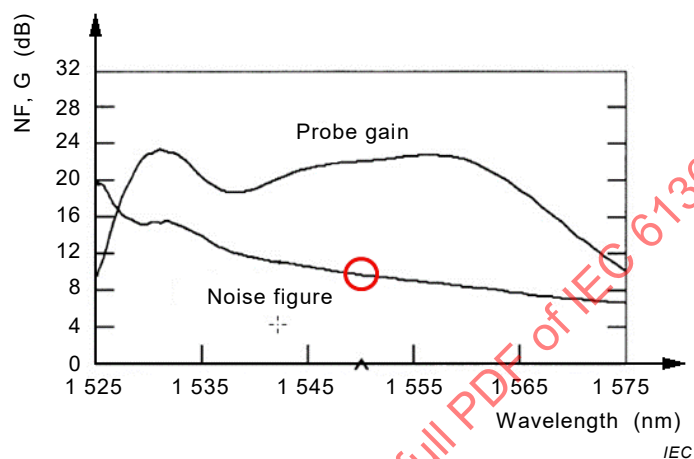
NOTE To keep the figure simple, all attributes that are not relevant are omitted.

Figure 28 – Implementation in IEC 62656-1 spreadsheet format of the example in Figure 27

EXAMPLE 2 For measurement of the noise figure of optical amplifiers, various measurement methods exist. Condition can be used for providing the information about the measuring environment. See Figure 29 and Figure 30 for an example [SOURCE: Douglas M. Baney, et al, *Theory and Measurement Techniques for the Noise Figure of Optical Amplifiers*].



a) Dynamic gain and noise figure measurement setup



b) measurement with a saturation wavelength of 1550 nm

Key

DUT	device under test
F	noise factor
G	gain
NF	noise figure
OSA	optical spectrum analyzer

Figure 29 – (a) Dynamic gain and noise figure measurement setup, and (b) measurement with a saturation wavelength of 1550 nm

#CLASS_ID:= MDC_C003							
#CLASS_ NAME.en:= Property meta- class							
#PROPERTY_ NAME.en	code	preferred_name	formula	property_data_ element_type	condition	unit_of_ measure	drawing_reference
#PROPERTY_ ID	MDC_ P001_6	MDC_P004_1.en	MDC_P024	MDC_P020	MDC_P028	MDC_ P023	MDC_P008_2
	PAA320	noise figure	$10\log_{10}(F)$	DEPENDENT_ P_DET	PAA321, PAA322	dB	
	PAA321	wavelength		CONDITION_DET		nm	
	PAA322	noise figure measurement setup		CONDITION_DET			PAA330
#CLASS_ID:= MDC_C007							
#CLASS_ NAME.en:= Document meta-class							
#PROPERTY_ NAME.en	code	drawing_title	file_name	file_format			
	MDC_ P001_8						
	PAA330	noise figure measurement setup picture	figure001	pdf			

IEC

**Figure 30 – Implementation in IEC 62656-1 spreadsheet format
of the example in Figure 29**

24.3 Reuse of properties

Situations occur where **Property** information objects should be referenced to avoid duplicated concepts.

The attribute **is_case_of** provides an import of one or more "foreign" properties from another class. It is insignificant whether this class is part of the same class tree or if it resides in another class tree. The imported relations are inherited to subclasses in the same way as relations that are not assigned by **is_case_of**.

EXAMPLE A class "personal computer" can exist in a class tree "computer" and a class "hard drive" can reside in a tree "memory". To avoid duplication of the **Property** information objects "disk speed" and "storage capacity", the attribute **is_case_of** is used. Figure 31 shows the related information objects.

#CLASS_ID:= MDC_C002						
#CLASS_ NAME.en:= Class meta- class						
#PROPERTY_ NAME.en	code	preferred_name	superclass	applicable_ properties	is_case_of	
#PROPERTY_ ID	MDC_ P001_5	MDC_P004_1.en	MDC_P010	MDC_P014	MDC_P013	
	CAA101	memory	UNIVERSE			
	CAA102	magnetic storage device	CAA101			
	CAA103	hard disk	CAA102	PAA500, PAA501, PAA502, PAA503		

#CLASS_ID:= MDC_C003						
#CLASS_ NAME.en:= Property meta- class						
#PROPERTY_ NAME.en	code	preferred_name				definition_class
#PROPERTY_ ID	MDC_ P001_6	MDC_P004_1.en				MDC_P021
	PAA500	rated current				CAA101
	PAA501	rated voltage				CAA101
	PAA502	disk speed				CAA101
	PAA503	storage capacity				CAA101

#CLASS_ID:= MDC_C002						
#CLASS_ NAME.en:= Class meta- class						
#PROPERTY_ NAME.en	code	preferred_name	superclass	applicable_ properties	is_case_of	imported_ properties
#PROPERTY_ ID	MDC_ P001_5	MDC_P004_1.en	MDC_P010	MDC_P014	MDC_P013	MDC_P040
	CAA201	computer	UNIVERSE			
	CAA202	personal computer	CAA201	PAA511	CAA103	PAA502, PAA503

#CLASS_ID:= MDC_C003						
#CLASS_ NAME.en:= Property meta- class						
#PROPERTY_ NAME.en	code	preferred_name				definition_class
#PROPERTY_ ID	MDC_ P001_6	MDC_P004_1.en				MDC_P021
	PAA511	screen size				CAA201

IEC

NOTE To keep the figure simple, all attributes that are not relevant are omitted.

Figure 31 – Reuse of properties (IEC 62656-1 spreadsheet format)

24.4 Classes and properties for common use

Classes below the node "Feature class" are a specific type of classes that may be used as readily defined building blocks to provide solutions for frequently occurring use cases, such as specification of address data, tolerance data, environmental conditions.

This allows the definition of classes and associated lists of properties which lie outside the classification schema for items, materials, geometry, etc. With this facility, classes can be defined which group together a number of related properties. A link to such classes can then be established by creating a **Property** with **data_type** CLASS_INSTANCE_TYPE (see 10.4.4) or CLASS_REFERENCE_TYPE (see 10.4.5). This may be done in every class throughout the whole classification hierarchy. The link makes all properties of the referenced class available within the class from which the link is made and, by inheritance, within any of its subclasses.

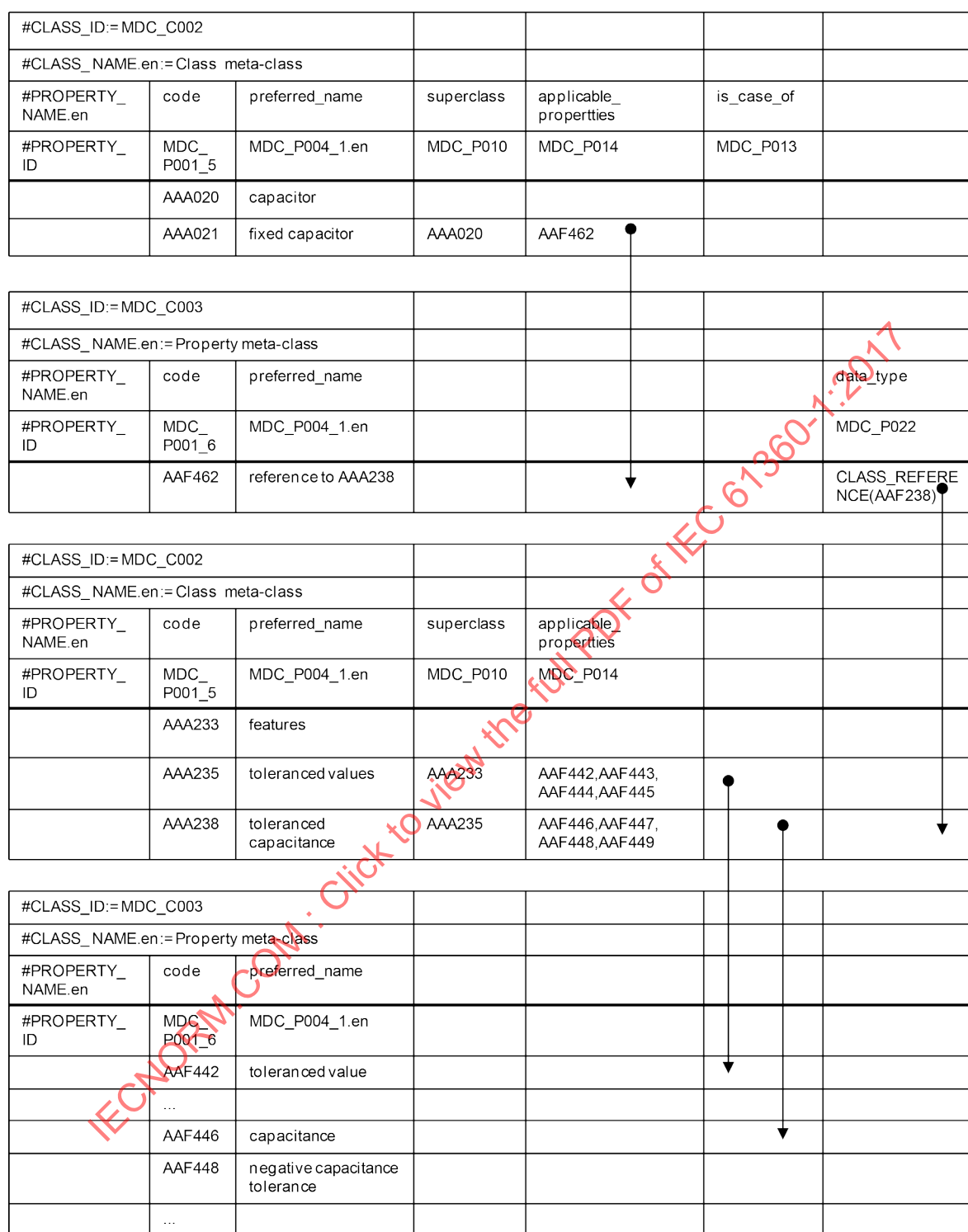
EXAMPLE The example shows how a feature class can be used to create a property of a tolerated capacitance (see Figure 32).

The entry class AAA238 is available for reference from anywhere within the classification hierarchy. As an example, tolerated capacitance could be defined as a property within the subclass for fixed capacitors.

NOTE The reference property can be subject to conditions in the same way as any other property.

The main use of such commonly usable classes is grouping related properties for creation of more complex data types than provided in this document.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017



IEC

NOTE To keep the figure simple, all attributes that are not relevant are omitted.

Figure 32 – Use of class information objects to associate toleranced capacitance information to a fixed capacitor (IEC 62656-1 spreadsheet format)

24.5 Block

Blocks are a means to structure properties in a convenient and task related way. By using blocks the properties relevant for an **Item_class** can be collected and simply reused. Using blocks saves time because blocks can serve as readymade templates and, in addition, blocks help to improve quality by ensuring that an already agreed information set can be reused

without unwanted modification. Figure 33 and Figure 34 show the use of **Item_class** and **Property** for creating a block.

NOTE See IEC 61987-10:2009 for additional information.

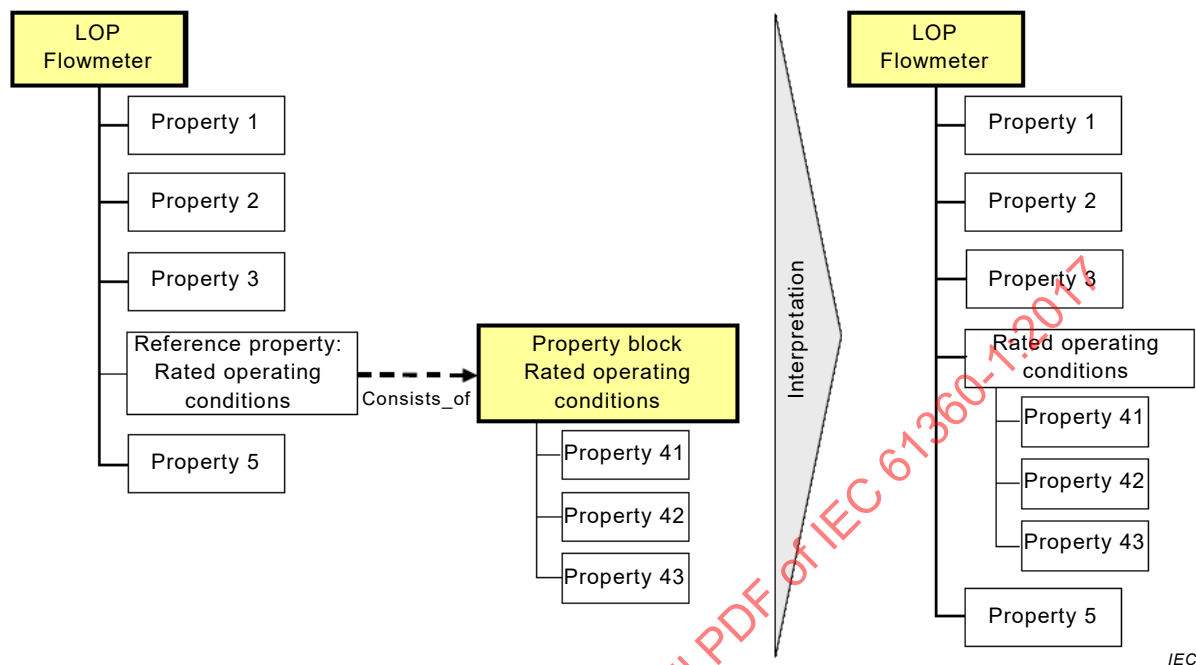
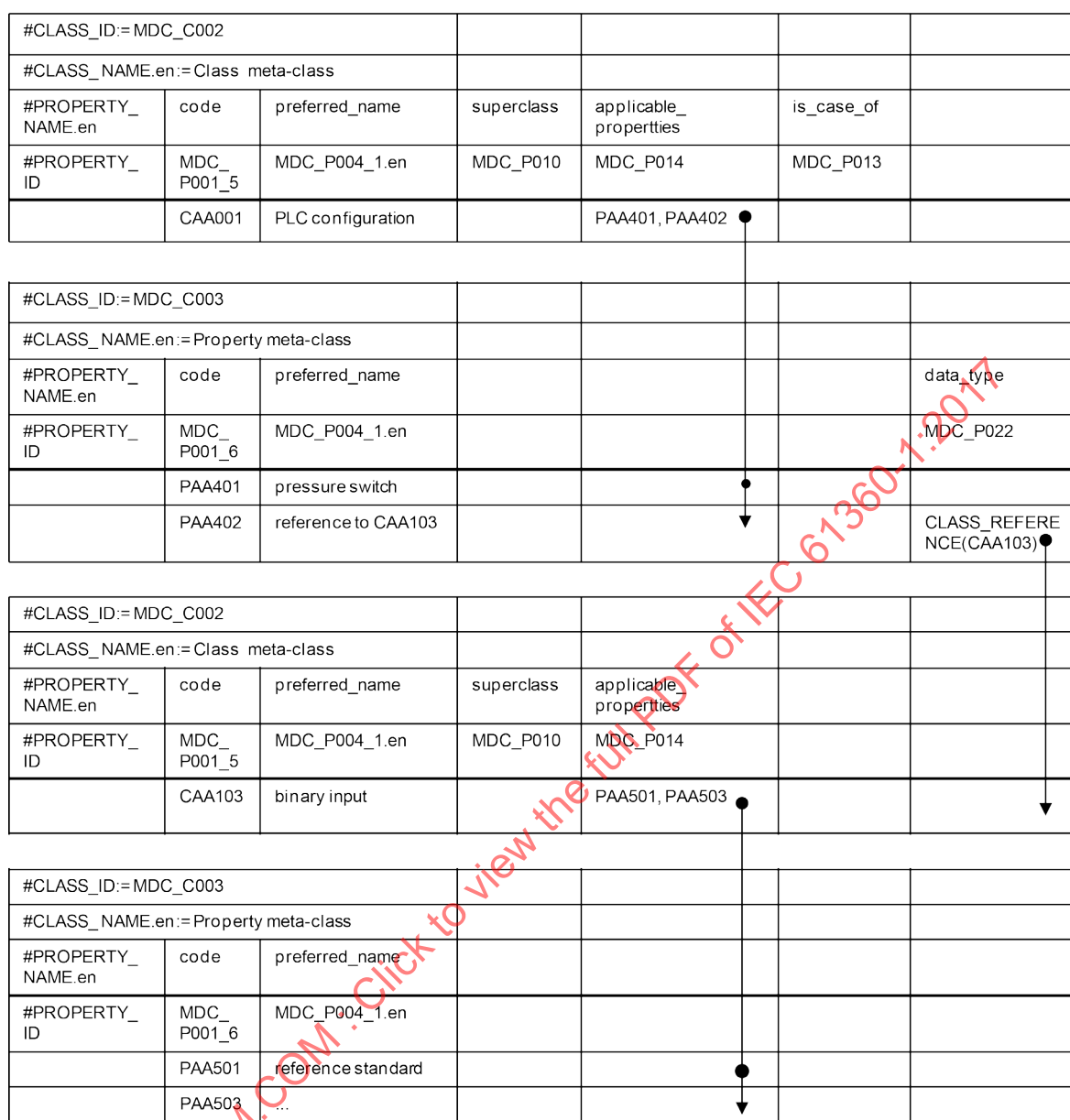


Figure 33 – Interpretation of Class and Property information objects forming a block

A block consists of an **Item_class** serving as handle for the block as a whole plus one or more **Property** information objects that carry the information content of the block. Properties of data type CLASS_REFERENCE_TYPE may be part of the **Property** information objects of a block. Such properties add other blocks to the structure. This nesting of blocks may occur to any depth.



IEC

NOTE To keep the figure simple, all attributes that are not relevant are omitted.

Figure 34 – Example of a block (IEC 62656-1 spreadsheet format)

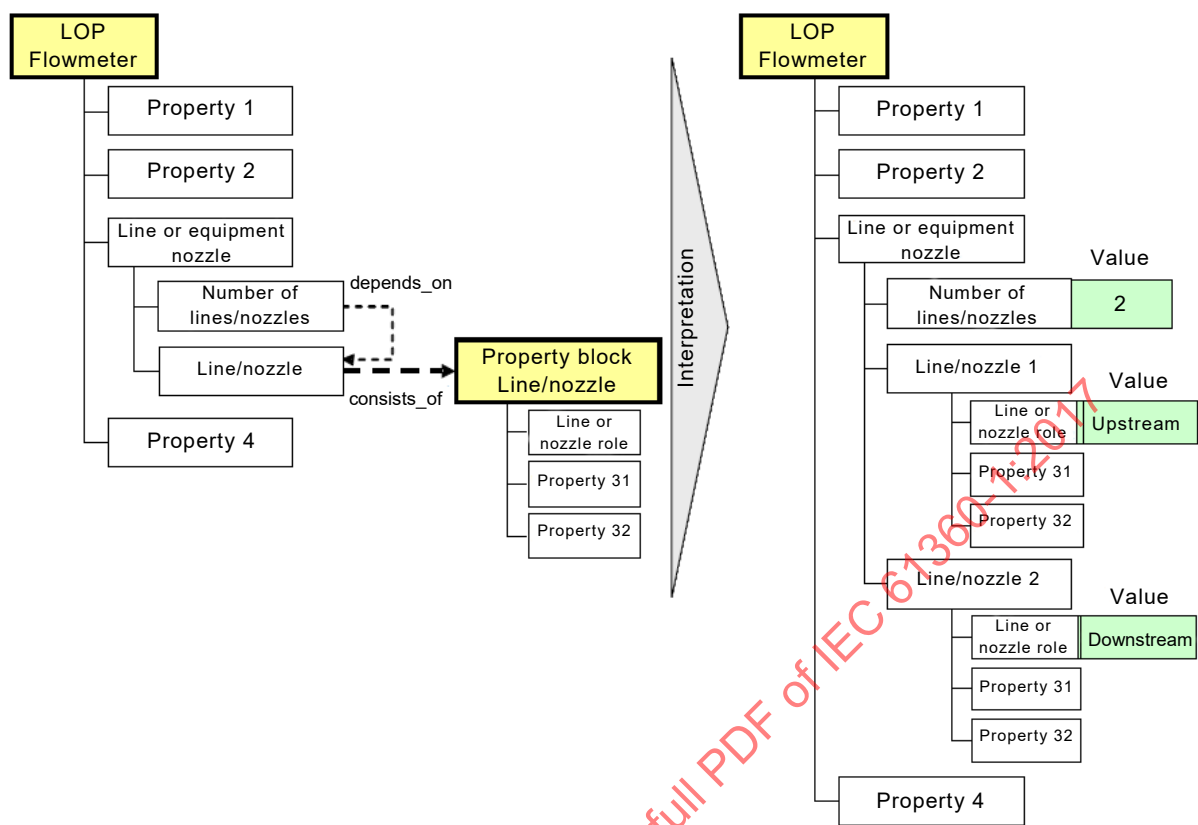
24.6 Cardinality

Recurring structures may occur when specifying data structures. Making use of such structures can greatly enhance compactness and clearness of the resulting dictionary. Cardinality allows specifying limits for the creation of occurrences of such structures.

EXAMPLE 1 A bicycle has two similar wheels. To specify the wheels, the creation of a specification for a wheel is sufficient, which then is used two times.

Figure 35 and Figure 36 show the use of **Item_class** and **Property** information objects for creating cardinality.

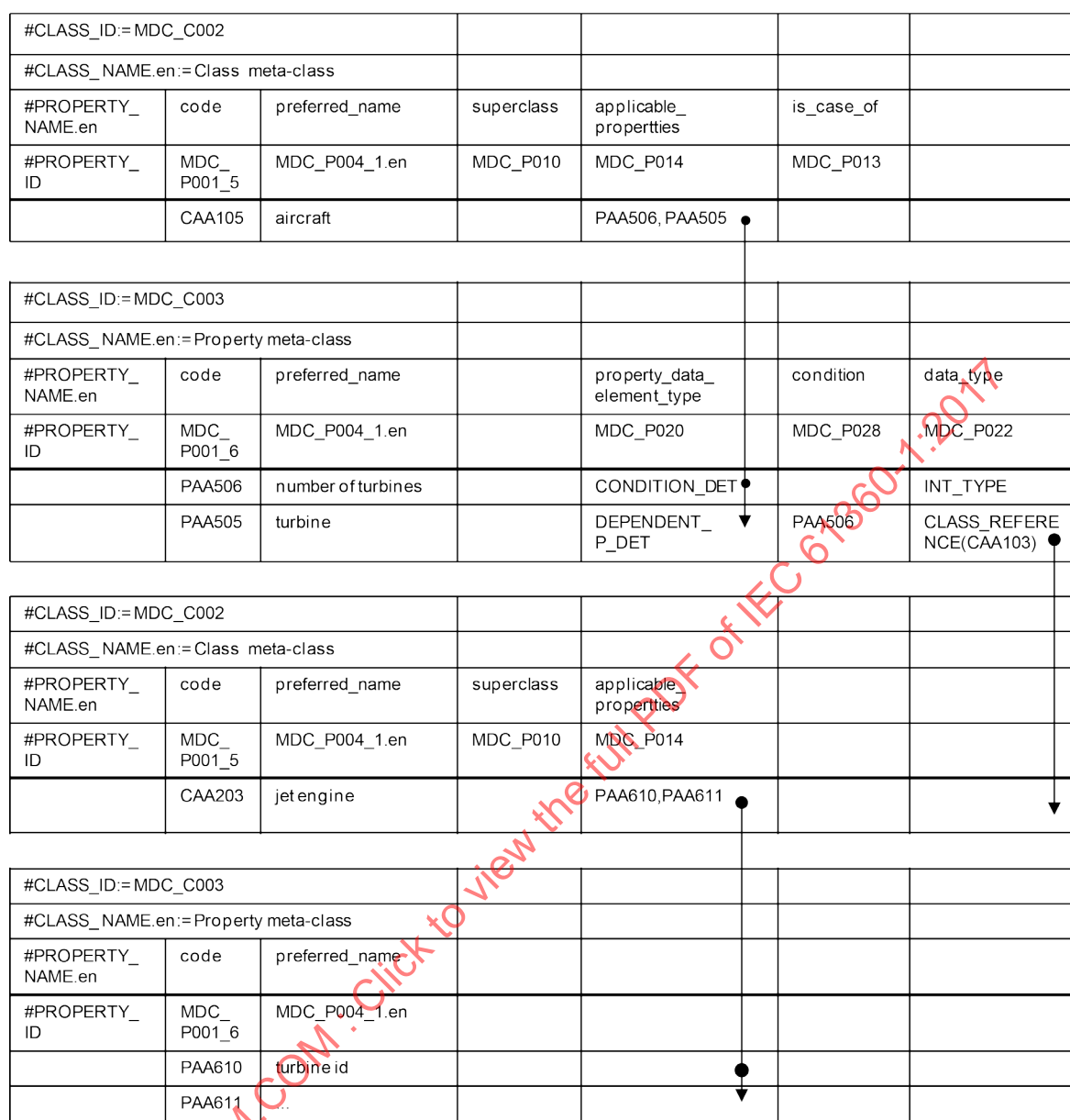
NOTE See IEC 61987-10:2009 for additional information.



IEC

Figure 35 – Interpretation of Class and Property information objects forming cardinality

EXAMPLE 2 Figure 36 shows a situation where an aircraft can have between one and four jet engines. The user can specify in the condition (PAA506) the exact number of engines for a specific case, e.g. three. Depending on this number, an application can then create three occurrences of the jet engine description block (CAA203).



IEC

NOTE To keep the figure simple, all attributes that are not relevant are omitted.

Figure 36 – Example for a block whose number of occurrences is limited through PAA506 (IEC 62656-1 spreadsheet format)

24.7 Polymorphism

24.7.1 Overview

Devices can largely change their function depending on configuration parameters. Especially devices controlled by software can be configured for executing specific functions out of a set of multiple predefined functions, thus requiring specific (predefined) sets of characteristics for specification.

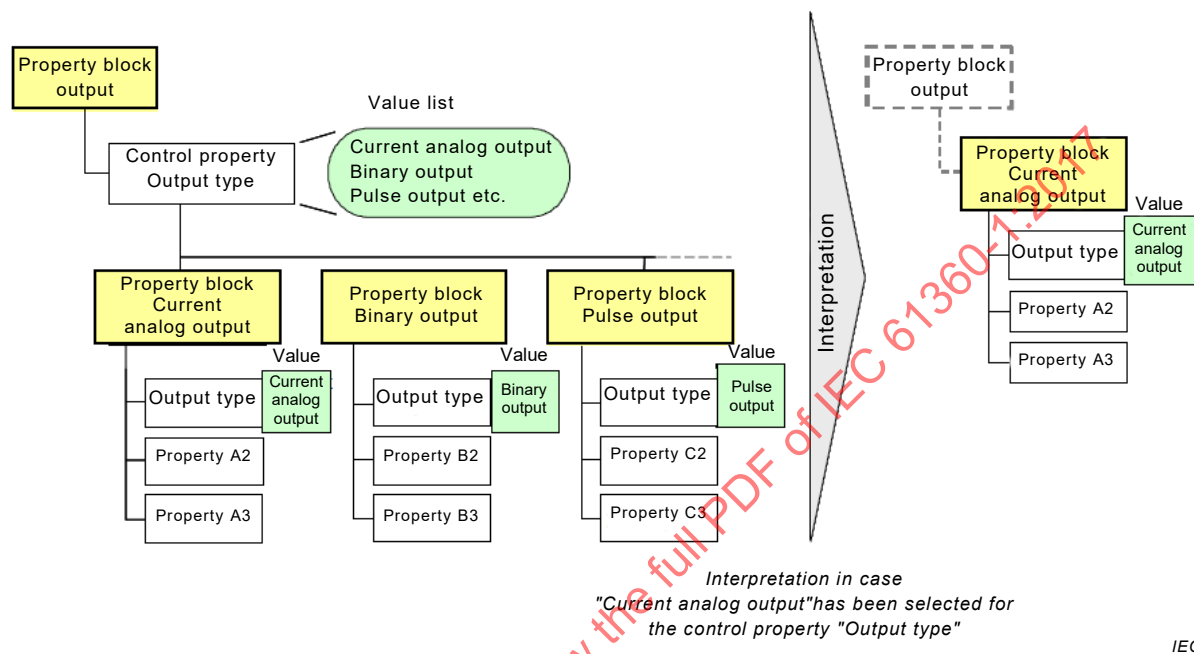
NOTE 1 See IEC 61987-10:2009, 4.3.2 for additional information.

Polymorphism structures provide the ability for selecting a block of properties from a predefined set of blocks for specifying the characteristics of a class according to the intended configuration of its class instances. Thus, it can be ensured that the class is always specified by a set of characteristics specifically tailored to the intended use of its instances.

NOTE 2 Care should be taken not to create inconsistencies by providing polymorphic choices that contradict the definition of the class.

EXAMPLE A measuring device has a communication interface offering the choices of 20 mA communication, wireless communication, or bus communication. By providing a purpose-made block of properties for each communication alternative, the user can select the appropriate specification for his specific configuration.

Figure 37 shows the use of **Item_class** and **Property** information objects for creating a polymorphism.



[SOURCE: IEC 61987-10:2009, Figure 6]

Figure 37 – Interpretation of Class and Property information objects forming a polymorphism

24.7.2 Polymorphic choices directly assigned to the specified **Item_class**

The polymorphic choices to choose from are linked to an **Item_class** using pointers, i.e. **Property** information objects, with **property_data_element_type** = **CLASS_REFERENCE_TYPE**. These pointers are restricted by a condition enumerating the allowed choices. The wording used within the **Value** information objects and the **preferred_name** attributes of the **Item_class** information objects of the polymorphic choices shall be identical (see Figure 38).

These polymorphic choices may carry clear text names, such as “Aqueous phase”.

#CLASS_ID:=MDC_C010						
#CLASS_NAME.en:=Term meta-class						
#PROPERTY_NAME.en	code	preferred_letter_symbol				
#PROPERTY_ID	MDC_P001_11	MDC_P025_1				
	ABU001	aqueous phase	●			
	ABU002	non-aqueous phase	●			
	ABU003	vapor phase	●			
#CLASS_ID:=MDC_C005						
#CLASS_NAME.en:=Enumeration meta-class						
#PROPERTY_NAME.en	code	enumerated_list_of_terms				
#PROPERTY_ID	MDC_P001_12	MDC_P043				
	ABU088	ABU001, ABU002, ABU003	▼			
#CLASS_ID:=MDC_C002						
#CLASS_NAME.en:=Class meta-class						
#PROPERTY_NAME.en	code	preferred_name	superclass	applicable_properties	is_case_of	
#PROPERTY_ID	MDC_P001_5	MDC_P004_1.en	MDC_P010	MDC_P014	MDC_P013	
	ABC510	phase		ABU127, ABU129, ABU130, ABU131	●	
#CLASS_ID:=MDC_C003						
#CLASS_NAME.en:=Property meta-class						
#PROPERTY_NAME.en	code	preferred_name		property_data_element_type	condition	data_type
#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en		MDC_P020	MDC_P028	MDC_P022
	ABU127	phasetype		CONDITION_DET	●	ENUM_STRING_TYPE(ABU088(...))
	ABU129	block ABC130		DEPENDENT_P_DET	●	CLASS_REFERENCE(ABC130)
	ABU130	block ABC447		DEPENDENT_P_DET	●	CLASS_REFERENCE(ABC447)
	ABU131	block ABC680		DEPENDENT_P_DET	▼	CLASS_REFERENCE(ABC680)
#CLASS_ID:=MDC_C002						
#CLASS_NAME.en:=Class meta-class						
#PROPERTY_NAME.en	code	preferred_name	superclass	applicable_properties		
#PROPERTY_ID	MDC_P001_5	MDC_P004_1.en	MDC_P010	MDC_P014		
	ABC130	aqueous phase		...		
	ABC447	non-aqueous phase		...		
	ABC680	vapor phase		...		

IEC

NOTE To keep the figure simple, all attributes that are not relevant are omitted.

Figure 38 – Polymorphic choices directly assigned to the specified Item_class (IEC 62656-1 spreadsheet format)

24.7.3 Polymorphic choices assigned to the specified **Item_class** through a **Value_list**

The polymorphic choices to choose from are linked to the **Item_class** using a value list, i.e. a **Value_list** information object, together with the value of **data_type** as **ENUMERATION_REFERENCE_TYPE**. This value list is restricted by a condition enumerating the codes of the allowed polymorphic choices (see Figure 39).

Unlike the case in 24.7.2, the names of the polymorphic choices are not free. The polymorphic choices shall always carry the content of the **code** attributes of the **Item_class** information objects, such as “BBA600”.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

#CLASS_ID:=MDC_C010						
#CLASS_NAME.en:=Term meta-class						
#PROPERTY_NAME.en	code	preferred_letter_symbol				
#PROPERTY_ID	MDC_P001_11	MDC_P025_1				
	AQA101	BBA500	▲			●
	AQA102	BBA600	●			●
#CLASS_ID:=MDC_C005						
#CLASS_NAME.en:=Enumeration meta-class						
#PROPERTY_NAME.en	code	enumerated_list_of_terms				
#PROPERTY_ID	MDC_P001_12	MDC_P043				
	AFA001	AQA101, AQA102	●			▲
#CLASS_ID:=MDC_C002						
#CLASS_NAME.en:=Class meta-class						
#PROPERTY_NAME.en	code	preferred_name	superclass	applicable_properties	is_case_of	
#PROPERTY_ID	MDC_P001_5	MDC_P004_1.en	MDC_P010	MDC_P014	MDC_P013	
	ABC510	phase		AAE341, AAE342	●	
#CLASS_ID:=MDC_C003						
#CLASS_NAME.en:=Property meta-class						
#PROPERTY_NAME.en	code	preferred_name		property_data_element_type	condition	data_type
#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en		MDC_P020	MDC_P028	MDC_P022
	AAE341	phase type		CONDITION_DET	●	ENUM_STRING_TYPE(AFA001(...))
	AAE342			DEPENDENT_P_DET	▼	ENUM_REFERENCE_TYPE(AFA001(...))
#CLASS_ID:=MDC_C002						
#CLASS_NAME.en:=Class meta-class						
#PROPERTY_NAME.en	code	preferred_name	superclass	applicable_properties		
#PROPERTY_ID	MDC_P001_5	MDC_P004_1.en	MDC_P010	MDC_P014		
	BBA500	aqueous phase		...		▼
	BBA600	non-aqueous phase		...		▼

IEC

NOTE To keep the figure simple, all attributes that are not relevant are omitted.

Figure 39 – Polymorphic choices assigned to the specified class through a value list (IEC 62656-1 spreadsheet format)

24.8 Relation

24.8.1 Overview

Relation is a powerful means to link instances in dictionaries specifying sub-collections of information objects. Depending on the content of the attributes **relation_type** and **role_of_the_relation**, the following cases may be expressed:

- a) constraint on one or more properties, as an extension of a single property constraint;
- b) grouping among ontological entities, in particular among heterogeneous ones;
- c) mapping, correspondence, or transition among ontological entities.

Cases a) and b) are explained in greater detail in 24.8.2 and 24.8.3, respectively.

24.8.2 Restricted enumeration

Predefined enumerations (i.e. value lists) avoid selection of wrong or meaningless choices as values for properties. It is strongly recommended to apply existing standardized enumerations providing in most cases exhaustive lists of values.

However, in some cases there is a requirement for narrowing the choices offered.

EXAMPLE A given enumeration provides all colours that can be delivered as colour of a specific sort of glass. If the intended use of the glass is to be the shield of a traffic light, it can be desirable to restrict the enumeration of deliverable colours to the three colours that are foreseen for traffic lights (red, yellow, and green). Figure 40 shows the example.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

#CLASS_ID:=MDC_C010						
#CLASS_NAME.en:=Term meta-class						
#PROPERTY_NAME.en	code	preferred_letter_symbol				
#PROPERTY_ID	MDC_P001_11	MDC_P025_1				
	VNN001	red	↑			
	VNN002	yellow	●			
	VNN003	green	●			
	VNN004	blue	●			
#CLASS_ID:=MDC_C005						
#CLASS_NAME.en:=Enumeration meta-class						
#PROPERTY_NAME.en	code	enumerated_list_of_terms				
#PROPERTY_ID	MDC_P001_12	MDC_P043				
	ENN001	VNN001, VNN002, VNN003, VNN004	●		↑	↑
#CLASS_ID:=MDC_C011						
#CLASS_NAME.en:=Relation meta-class						
#PROPERTY_NAME.en	code	relation_type	role_of_relation	formula	domain_of_relation	
#PROPERTY_ID	MDC_P001_13	MDC_P200	MDC_P210	MDC_P204	MDC_P201	
	RNN001	PREDICATION	constraint	enum_constraint (VNN001, VNN002, VNN003)	ENN001	●
#CLASS_ID:=MDC_C002						
#CLASS_NAME.en:=Class meta-class						
#PROPERTY_NAME.en	code	preferred_name	superclass	applicable_properties	applicable_relaton	
#PROPERTY_ID	MDC_P001_5	MDC_P004_1.en	MDC_P010	MDC_P014		
	CNN001	glass		PNN001	●	
	CNN002	glass for traffic light	CNN001	PNN001	●	RNN001
#CLASS_ID:=MDC_C003						
#CLASS_NAME.en:=Property meta-class						
#PROPERTY_NAME.en	code	preferred_name		property_data_element_type	condition	data_type
#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en		MDC_P020	MDC_P028	MDC_P022
	PNN001	glass color		NON_DEPENDENT_P_DET		ENUM_STRING_TYPE(ENN001(...))

IEC

NOTE To keep the figure simple, all attributes that are not relevant are omitted.

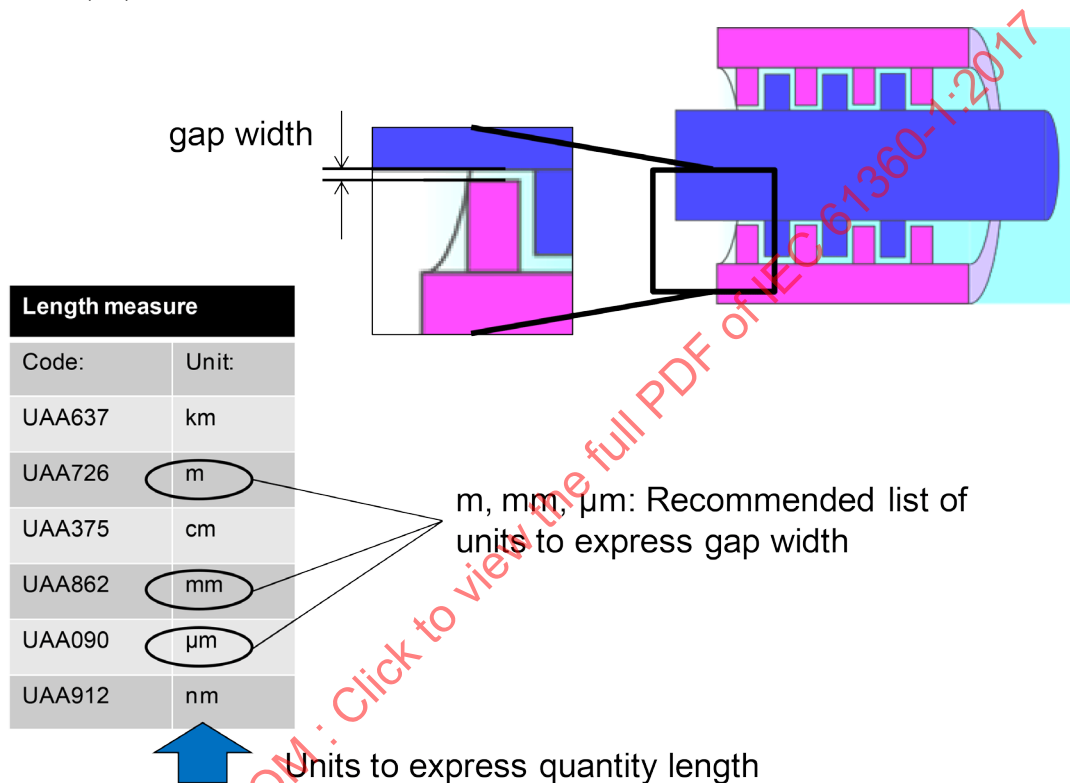
**Figure 40 – Example – Restricted enumeration
(IEC 62656-1 spreadsheet format)**

24.8.3 Grouping

Besides the grouping of objects in hierarchies, another type of grouping can be desirable.

The example below shows the use of **Relation** information objects for creating special predefined lists of units. In general, predefined lists of units offer a selection of units for users that can minimize erroneous choices. In cases where the offered list is too bulky containing units that are not appropriate, the possible selections have to be restricted thus achieving meaningful choices (see Figure 41 and Figure 42).

EXAMPLE To specify the gap of a labyrinth seal, units such as metre (m), millimetre (mm), or micrometre (µm) can apply. By specifying a list of units embracing m, mm, and µm, absurd length measures such as kilometre (km) or nanometre (nm) can be excluded.



IEC

Figure 41 – Example – List of units for measuring the gap width of a labyrinth seal

#CLASS_ID:=MDC_C009						
#CLASS_NAME.en:=UoM meta-class						
#PROPERTY_NAME.en	code	preferred_name				
#PROPERTY_ID	MDC_P001_10	MDC_P004_1.en				
	UAA637	kilometre				
	UAA726	metre				
	UAA375	centimetre				
	UAA862	millimetre				
	UAA090	micrometre				
	UAA912	nanometre				

#CLASS_ID:=MDC_C011						
#CLASS_NAME.en:=Relation meta-class						
#PROPERTY_NAME.en	code	relation_type	role_of_relation	formula	domain_of_function	codomain_of_function
#PROPERTY_ID	MDC_P001_13	MDC_P200	MDC_P210	MDC_P204	MDC_P202	MDC_P203
	NAA001	PREDICATION	quantity		UAA637, UAA726, UAA375, UAA862, UAA090, UAA912	UAA726
	NAA002	PREDICATION	quantity		UAA726, UAA862, UAA090	UAA726

#CLASS_ID:=MDC_C003						
#CLASS_NAME.en:=Property meta-class						
#PROPERTY_NAME.en	code	preferred_name		property_data_element_type	code_of_unit	code_of_alternative_unit
#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en		MDC_P020	MDC_P41	MDC_P042
	NAA003	gap width		NON_DEPENDENT_P_DET	UAA862	NAA002

IEC

NOTE To keep the figure simple, all attributes that are not relevant are omitted.

**Figure 42 – Example – Information objects of Figure 41
(IEC 62656-1 spreadsheet format)**

25 Qualifiers

There are cases where users want to add additional information to the values assigned to a property. This information contains data required for the correct interpretation of the value.

EXAMPLE A property can be used along the life cycle of a product and its value can depend on the life-cycle stage. Thus, assigning the life-cycle stage to the property value is essential for the correct interpretation of the value. The value associated during the stage “as designed” can well differ from the value during the stage “as built”.

The additional information is contained in so-called (value-)qualifiers and may be specified using the rules of this document.

NOTE 1 See IEC 62569-1 for additional information.

Qualifiers shall not be confused with conditions. Conditions apply to all possible values of a **Property** whereas qualifiers are strictly related to identified value(s) of a **Property**.

The same value may be applied with one or more qualifiers. When assigning another value to the **Property** the qualifiers can change.

NOTE 2 Association of qualifiers to values is not an activity performed in a dictionary but is performed by application software. By providing qualifier objects in the dictionary the application software can select from predefined, standardized names for qualifiers. Such qualifiers can provide information such as status data or life-cycle data.

26 Characters and character sets

26.1 Overview

Characters and character sets play an important role in supporting the versatility and usability of a dictionary. Thus, efforts should be made to equally support European and non-European character sets.

26.2 Recommended character sets

Unicode characters that correspond to the G0 set of ISO/IEC 646 or to row 00, columns 002 to 007 of ISO/IEC 10646, UTF-16 shall be used for any textual or numerical information.

For alternative languages or missing characters such as "θ" (GREEK THETA SYMBOL), the full character set as defined by ISO/IEC 10646 may be used.

NOTE 1 The character "θ" (GREEK THETA SYMBOL) can be expressed by using the Unicode coding "U+03D1".

NOTE 2 The Unicode standard is published by The Unicode Consortium, P.O. Box 391476, Mountain View, CA 94039-1476, U.S.A., www.unicode.org.

Characters from ISO/IEC 10646 shall be represented by their 4 digit hexadecimal identifier preceded by a number sign "#". A number sign "#" not followed by a number shall not be interpreted as the start of a Unicode hexadecimal identifier.

EXAMPLE "∅" is coded as "#2205"

26.3 Line feed

Many applications do not support the use of line feed characters as part of a value. The vertical line "|" character may be used instead.

Two consecutive vertical line characters "||" do not mean double line feed but create a single vertical line "|".

26.4 Subscript

Subscripts shall be indicated by Latin letter(s) preceded by an underline "_" character followed by a closing curly bracket "}" after the subscript.

An underline "_" not followed by a closing curly bracket shall not be interpreted as subscript command.

EXAMPLE 1 " λ_{peak} " is coded as "\$\lambda_{\text{peak}}\$".

NOTE Non-character subscript glyphs such as subscript numbers can be presented by using Unicode characters.

EXAMPLE 2 " H_2SO_4 " can be presented by writing "H#2082SO#2084".

26.5 Superscript

Superscripts shall be indicated by Latin letter(s) preceded by two asterisk characters "***" followed by a closing curly bracket "}" after the superscript.

Two asterisks "**" not followed by a closing curly bracket shall not be interpreted as superscript command.

EXAMPLE 1 " α^b " is coded as "\$a**b)".

NOTE Non-character superscript glyphs such as superscript numbers can be presented by using Unicode characters.

EXAMPLE 2 " 10^{-3} " can be presented by writing "10#207B#00B3".

26.6 Greek characters

To express Greek characters the transliteration specified in Table 7 (Table 1 of ISO 843:1997) may be used. For retranslation, the translated Greek letters shall be preceded by a dollar "\$" character.

The character "\$" not followed by one of the characters specified in Table 7 shall not be interpreted as transliteration command.

NOTE 1 Table 7 is only to express quantities, not to transliterate into New Greek writing.

NOTE 2 See 26.2 for recommendations about use of Unicode characters.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

Table 7 – Transliteration of Greek characters to Latin characters

	Greek character		Latin character ^a	
	Upper case	Lower case	Upper case	Lower case
1	A	α	A	a
2	B	β	B	b
3	Γ	γ	G	g
4	Δ	δ	D	d
5	E	ε	E	e
6	Z	ζ	Z	z
7	H	η	I	i
8	Θ	θ	TH	th
9	I	ι	I	i
10	K	κ	K	k
11	Λ	λ	L	l
12	M	μ	M	m
13	N	ν	N	n
14	Ξ	ξ	X	x
15	O	ο	O	o
16	Π	π	P	p
17	P	ρ	R	r
18	Σ	σ, ζ ^b	S	s, s
19	T	τ	T	t
20	Υ	υ	Y	y
21	Φ	φ	F	f
22	X	χ	CH	ch
23	Ψ	ψ	PS	ps
24	Ω	ω	Ö	ö

^a In case the character set does not support the overline character, the apostrophe character may be used instead of the overline character.

EXAMPLE The characters “\$’i” express the “i” (see line 7).

^b The Greek character ζ is translated into Latin s, the same as for the Greek character σ. The character σ is used at the beginning or in the middle of a word, while the character ζ is used at the end of the word. [SOURCE: ISO 843:1997, 3.1]

Annex A

(informative)

Data model

Figure A.1 to Figure A.6 use UML class diagrams for illustrating the IEC 61360-1 data model. The figures correspond to the presentations used in this document.

NOTE References in the model are made through identifiers containing the version information.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

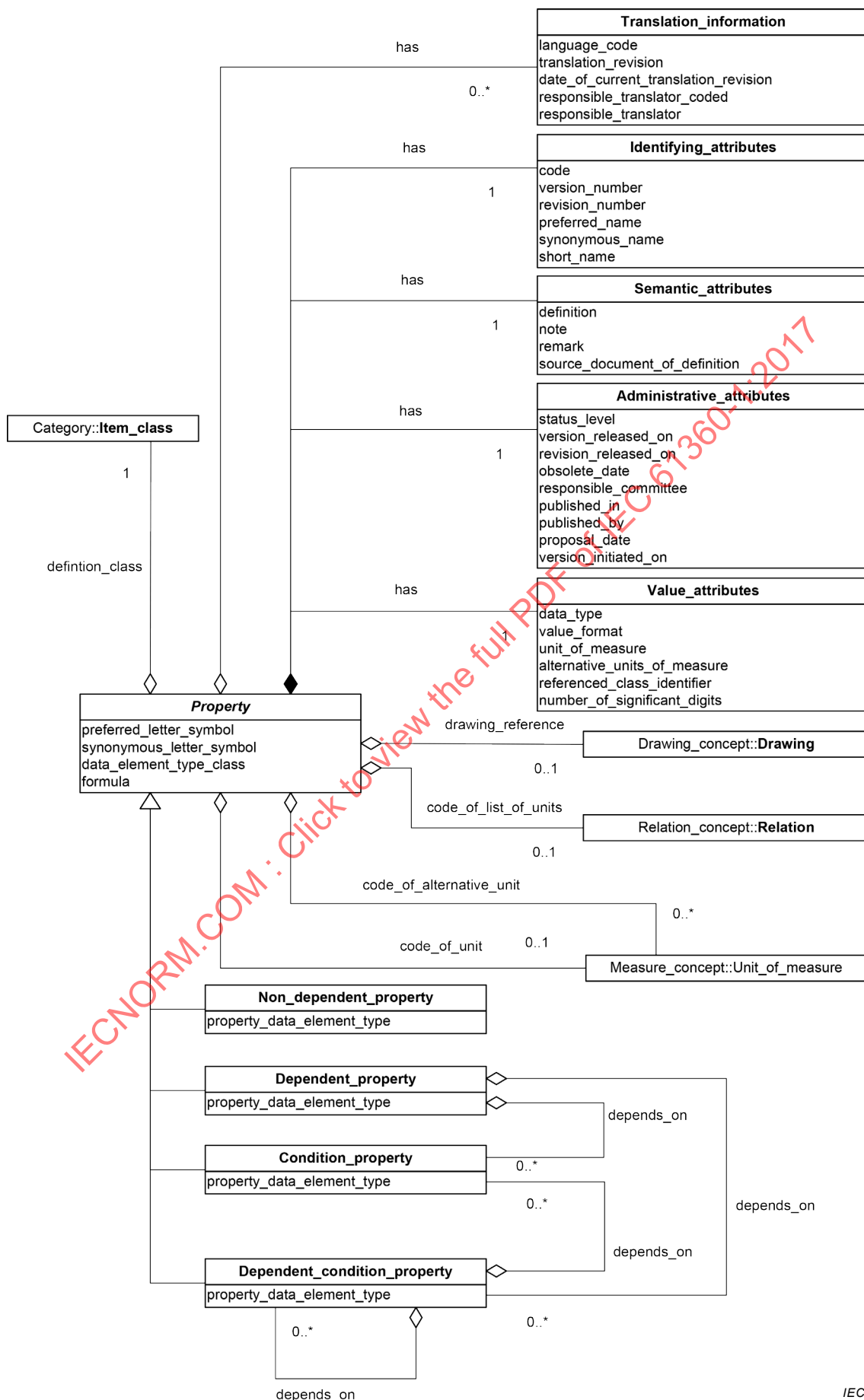


Figure A.1 – Characteristic (UML class diagram)

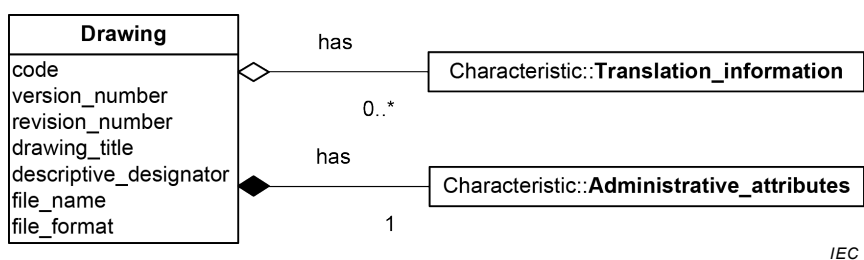


Figure A.2 – Drawing concept (UML class diagram)

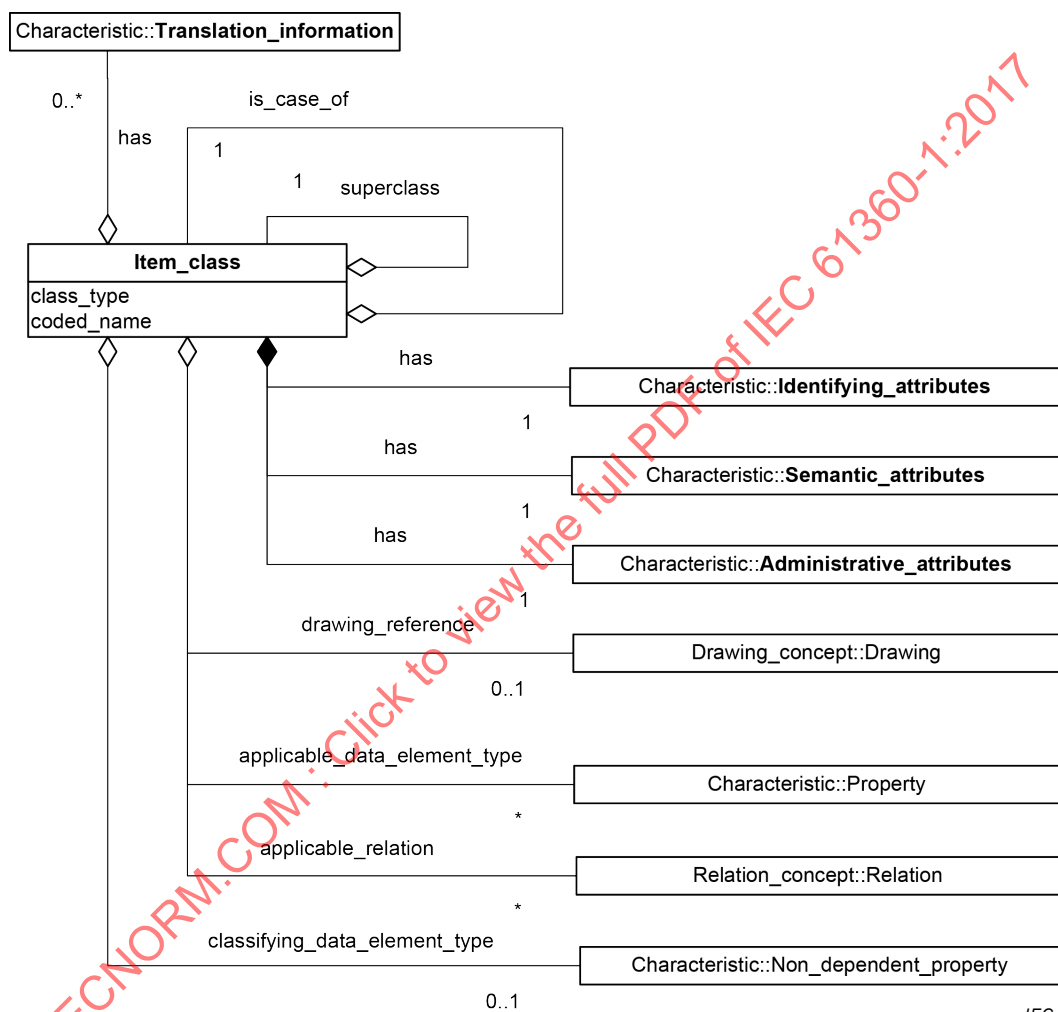
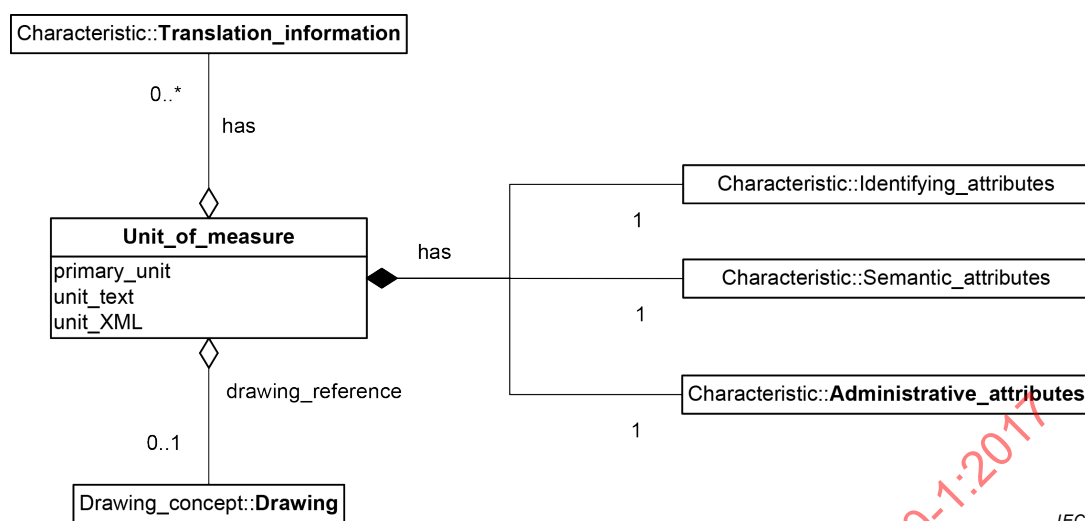
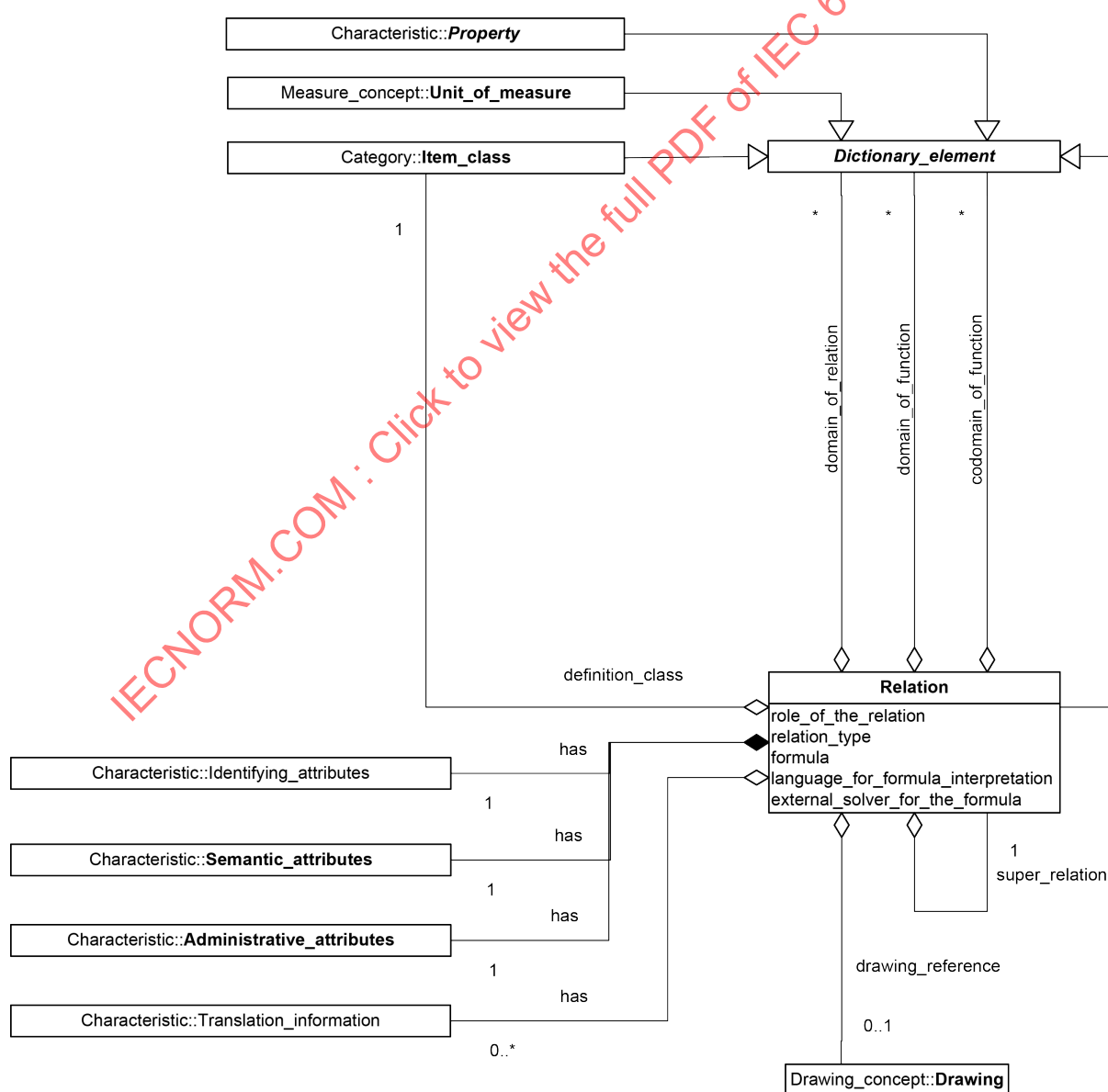


Figure A.3 – Category (UML class diagram)



IEC

Figure A.4 – Measure concept (UML class diagram)



IEC

Figure A.5 – Relation concept (UML class diagram)

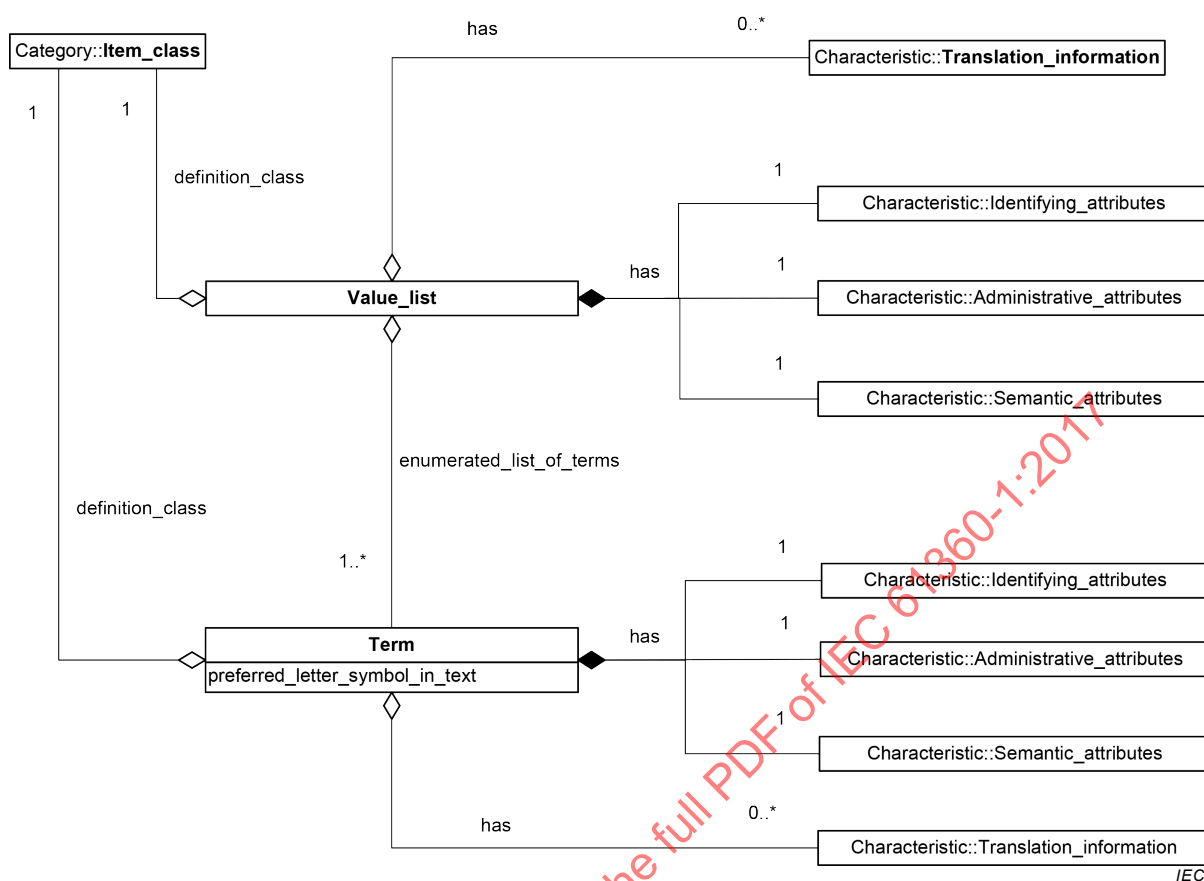


Figure A.6 – Value list concept (UML class diagram)

Annex B (normative)

Type classification codes of properties

B.1 Property classification

B.1.1 Overview

Property information objects should be classified in order to make large collections of properties more manageable. This may be implemented by dividing the subject field from the generic to the specific, bringing related concepts together (e.g. concepts referring to the same physical quantity such as temperature, voltage, capacitance) and thus making orientation easier. The field of interest can be divided into a number of classes which should be complementary, i.e. in the ideal case they cover in their totality the whole field without overlap.

The classification supports tasks such as:

- analysing property collections;
- designating properties;
- managing properties.

B.1.2 Principles

There shall be two general categories of **Property** information objects:

- quantitative properties, also called measures, representing the values of a measurable characteristic of an item. These **Property** information objects shall be part of the main classes "C", "E", etc.;
- non-quantitative properties such as identifiers or indicators labelling objects or designating items. The non-quantitative **Property** information objects shall be implemented by means of codes, names, descriptions, etc. Such **Property** information objects belong to main class "A".

B.1.3 Quantitative Property information objects

The quantitative **Property** information objects shall be classified into main classes according to their measure concepts or quantities in accordance with Table B.1.

Table B.1 – Survey of main classes of Property information objects

Non-quantitative properties	Main class	Description/subjects
	A	Identifications and indicators
Quantitative properties	Main class	Description/subjects
Physical measures:	C	Physical chemistry and molecular physics
	E	Electricity and magnetism
	F	Periodic and related phenomena
	G	Acoustics
	H	Heat
	J	Information
	K	Mechanics
	L	Light and related electromagnetic radiations
	T	Space and time
	U	Atomic and nuclear physics
	V	Nuclear reactions and ionizing radiations
	W	Solid-state physics
Non-physical measures:	M	Financial amounts
	P	Financial rates, prices, tariffs
	Q	Denumerable quantities, counts
	R	Business ratios, percentages

The classification of properties uses the following general rules.

- a) The classification shall have two levels:
 - main classes;
 - classes identified by a single capital letter and two numeric digits.
- b) The main classes, except class J, reflect main areas of physics, such as space and time, mechanics, heat, etc.
- c) The main classes are subdivided into classes of physical quantities such as T10: velocity, K01: mass, H12: thermal resistance.
- d) The main class J shall be subdivided into classes based on definitions from the ISO 2382 series.

In addition the following rules apply.

- e) For quantities occurring in various physical domains, their complete physical context shall be decisive for the classification.

EXAMPLE 1 Wavelengths of light and sound are classified in class L03 or G05, respectively.

- f) ISO classes containing a physical constant shall not be used.

EXAMPLE 2 E32/ L06: speed of light, L18: Stefan-Boltzmann constant.

- g) Dimensionless ratios of identical quantities, expressed as fractions or percentages and not contained in Annex B, shall be classified in the same class as the quantity from which they originate.

EXAMPLE 3 Current gain of a transistor is classified as E01.

- h) Derived quantities shall be classified in the class of the numerator.

EXAMPLE 4 Steepness of a voltage pulse (in V/s) in class E06.

B.1.4 Non-quantitative Property information objects

The values of non-quantitative properties represent identifiers or designators of items such as locations, organizations, persons, messages, documents, etc. As a result, these **Property** information objects belong to the main class "A" and may be further classified according to their subject, identified by two additional numeric digits.

B.2 Survey of type classification codes for non-quantitative properties

In this survey, the complete classification and title per main class are given in alphabetical order of the main class code. See Table B.2.

Table B.2 – Classification codes of non-quantitative data element types

code	Description
A11	Geographical unit (greater than a place)
A12	Geographical location (place or smaller)
A13	Geographical route and network
A21	Organization
A22	Functionary
A23	Organization class
A31	Date and time period
A32	Time of day
A41	Private person
A51	Product
A52	Product class
A53	Product batch and package (type)
A54	Transport mode, means and unit
A55	Manufacturing process and technology
A56	Product function and application
A57	Material
A58	Product geometry, shape, and size
A59	Product quality, performance, and test
A61	Document and message
A62	Information element and information group
A63	Data medium and transmission unit
A71	Measuring unit
A79	Type of measurement
A81	Account
A82	Project, project activity
A83	Procedure
A91	Abstract identification such as language, colour, etc.
A93	Clause

B.3 Survey of type classification codes for quantitative properties

In Table B.3 to Table B.16, the complete classification and title per main class are given in alphabetical order of the main class code. See also 6.3 for the use of the classification codes.

Table B.3 – Classification codes for quantities of physical chemistry and molecular physics (1 of 2)

Code	Description	Code	Description
C01	relative atomic mass, relative molecular mass	C16	molality of solute B
C02	number of molecules or other elementary elements	C17	chemical potential of B
C03	amount of substance	C18	absolute activity of B
C04	not used	C19	partial pressure of B (in a gaseous mixture)
C05	molar mass	C20	fugacity of B (in a gaseous mixture)
C06	molar volume	C21	standard absolute activity of B (in a gaseous mixture)
C07	molar thermodynamic energy	C22	activity coefficient of B (in a liquid or a solid mixture), standard absolute activity of B (in a liquid or a solid mixture)
C08	molar heat capacity	C23	activity of solute B, relative activity of solute B (especially in a dilute liquid solution)
C09	molar entropy	C24	activity coefficient of solute B (especially in a dilute liquid solution), standard absolute activity of solute B (especially in a dilute liquid solution)
C10	volumic number of molecules (or particles), number density of molecules (or particles), molecular concentration of B	C25	activity of solvent A, relative activity of solvent A (especially in a dilute liquid solution), osmotic coefficient of solvent A (especially in a dilute liquid solution) standard absolute activity of solvent A (especially in a dilute liquid solution)
C11	volumic mass, mass density, density, mass concentration of B	C26	osmotic pressure
C12	mass fraction of B	C27	stoichiometric number of B
C13	concentration of B, amount-of-substance concentration of B	C28	affinity (of a chemical reaction)
C14	mole fraction of B, mole ratio of solute B	C29	extent of reaction
C15	volume fraction of B	C30	standard equilibrium constant

Table B.3 (2 of 2)

Code	Description	Code	Description
C31	mass of molecule	C43	elementary charge
C32	electric dipole moment of molecule	C44	charge number of ion
C33	electric polarizability of molecule	C45	not used
C34	microcanonical partition function, canonical partition function, grand-canonical partition function, grand partition function, molecular partition function, partition function of a molecule	C46	ionic strength
C35	statistical weight	C47	degree of dissociation
C36	molar gas constant	C48	electrolytic conductivity
C37	not used	C49	molar conductivity
C38	mean free path	C50	transport number of the ion B, current fraction of the ion B
C39	diffusion coefficient	C51	angle of optical rotation
C40	thermal diffusion ratio, thermal diffusion factor	C52	molar optical rotatory power
C41	thermal diffusion coefficient	C53	massic optical rotatory power, specific optical rotatory power
C42	proton number	C54	pH

Table B.4 – Classification codes for quantities of electricity and magnetism

Code	Description	Code	Description
E01	electric current	E25	relative permeability
E02	electric charge, quantity of electricity	E26	magnetic susceptibility
E03	volumic charge, volume density of charge, charge density	E27	magnetic moment, electromagnetic moment
E04	areic charge, surface density of charge	E28	magnetization
E05	electric field strength	E29	magnetic polarization
E06	electric potential, potential difference, tension, electromotive force	E30	volumic electromagnetic energy electromagnetic energy density
E07	electric flux density	E31	Poynting vector
E08	electric flux	E32	not used
E09	capacitance	E33	resistance (to direct current)
E10	permittivity, electric constant, permittivity of vacuum	E34	conductance (for direct current)
E11	relative permittivity	E35	power (for direct current)
E12	electric susceptibility	E36	resistivity
E13	electric polarization	E37	conductivity
E14	electric dipole moment	E38	reluctance
E15	areic electric current, electric current density	E39	permeance
E16	lineic electric current, linear electric current density	E40	not used
E17	magnetic field strength	E41	frequency, rotational frequency
E18	magnetic potential difference, magnetomotive force, current linkage	E42	angular frequency, pulsance
E19	magnetic flux density, magnetic induction	E43	phase difference
E20	magnetic flux	E44	impedance, modulus of impedance, resistance, reactance
E21	magnetic vector potential	E45	admittance, modulus of admittance, conductance, susceptance
E22	self-inductance, mutual inductance	E46	quality factor
E23	coupling factor, leakage factor	E47	loss factor
E24	permeability, magnetic constant, permeability of vacuum	E48	loss angle
		E49	active power
		E50	apparent power, reactive power
		E51	power factor
		E52	active energy

Table B.5 – Classification codes for quantities of periodic and related phenomena

Code	Description	Code	Description
F01	period, periodic time	F08	phase velocity, group velocity
F02	time constant of an exponentially varying quantity	F09	level of a field quantity
F03	frequency, rotational frequency	F10	level of a power quantity
F04	angular frequency, pulsance	F11	damping coefficient
F05	wavelength	F12	logarithmic decrement
F06	wave number, repetency	F13	attenuation coefficient, phase coefficient, propagation coefficient
F07	angular wave number, angular repetency		

Table B.6 – Classification codes for quantities of acoustics

Code	Description	Code	Description
G01	period, periodic time	G18	acoustic impedance
G02	frequency	G19	mechanical impedance
G03	frequency interval	G20	characteristic impedance of a medium, surface density of mechanical impedance
G04	angular frequency, pulsance	G21	sound-pressure level
G05	wavelength	G22	sound-power level
G06	wave number, repetency	G23	damping coefficient
G07	angular wave number, angular repetency	G24	time constant, relaxation time
G08	volumic mass, mass density, density	G25	logarithmic decrement
G09	static pressure sound pressure	G26	attenuation coefficient, phase coefficient, propagation coefficient
G10	sound particle displacement	G27	dissipation factor, reflection factor, transmission factor, absorption factor
G11	sound particle velocity	G28	sound-reduction index
G12	sound particle acceleration	G29	equivalent absorption area of a surface or item
G13	volume flow rate	G30	reverberation time
G14	velocity of sound (phase velocity), group velocity	G31	loudness level
G15	sound energy density, volumic sound energy	G32	loudness
G16	sound power		
G17	sound intensity		

Table B.7 – Classification codes for quantities of heat

Code	Description	Code	Description
H01	thermodynamic temperature	H17	ratio of massic heat capacities, ratio of the specific heat capacities, isentropic exponent
H02	Celsius temperature	H18	entropy
H03	linear expansion coefficient, cubic expansion coefficient, relative pressure coefficient	H19	massic entropy, specific entropy
H04	pressure coefficient	H20	energy, thermodynamic energy, enthalpy, Helmholtz free energy, Helmholtz function, Gibbs free energy, Gibbs function
H05	isothermal compressibility, isentropic compressibility	H21	massic energy, specific energy, massic thermodynamic energy, specific thermodynamic energy, massic enthalpy, specific enthalpy, massic Helmholtz free energy, specific Helmholtz free energy, massic Gibbs free energy, specific Helmholtz function, specific Gibbs free energy, specific Gibbs function
H06	quantity of heat, heat	H22	Massieu function
H07	heat flow rate	H23	Planck function
H08	areic heat flow rate, density of heat flow rate		
H09	thermal conductivity		
H10	coefficient of heat transfer, surface coefficient of heat transfer		
H11	thermal insulance, coefficient of thermal insulation		
H12	thermal resistance		
H13	thermal conductance		
H14	thermal diffusivity		
H15	heat capacity		
H16	massic heat capacity, specific heat capacity at: – constant pressure, – constant volume, – saturation		

Table B.8 – Classification codes for quantities of information

Code	Description	Code	Description
J01	word length storage capacity register length	J04	volume storage density
J02	linear storage density	J05	transmission rate
J03	surface storage density	J06	error rate, code rate, efficiency

Table B.9 – Classification codes for quantities of mechanics

Code	Description	Code	Description
K01	mass	K17	Poisson ratio, Poisson number
K02	volumic mass, mass density, density	K18	modulus of elasticity, shear modulus, modulus of rigidity, bulk modulus, modulus of compression
K03	relative volumic mass, relative mass density, relative density	K19	compressibility (bulk compressibility)
K04	massic volume, specific volume	K20	second moment of area (second axial moment of area), second polar moment of area
K05	lineic mass, linear density	K21	section modulus
K06	areic mass, surface density	K22	dynamic friction factor, static friction factor
K07	moment of inertia	K23	viscosity (dynamic viscosity)
K08	momentum	K24	kinematic viscosity
K09	force, weight	K25	surface tension
K10	impulse	K26	energy, work, potential energy, kinetic energy
K11	moment of momentum, angular momentum	K27	power
K12	moment of force, moment of a couple, torque	K28	efficiency
K13	angular impulse	K29	mass flow rate
K14	gravitational constant	K30	volume flow rate
K15	pressure, normal stress, shear stress		
K16	linear strain (relative elongation), shear strain, volume strain (bulk strain)		

Table B.10 – Classification codes for quantities of light and related electromagnetic radiations

Code	Description	Code	Description
L01	frequency	L26	photon exitance
L02	angular frequency	L27	photon irradiance
L03	wavelength	L28	photon exposure
L04	repetency, wave number	L29	luminous intensity
L05	angular repetency, angular wave number	L30	luminous flux
L06	not used	L31	quantity of light
L07	radiant energy	L32	luminance
L08	radiant energy density	L33	luminous exitance
L09	spectral concentration of radiant energy density	L34	illuminance
L10	radiant power, radiant energy flux	L35	light exposure
L11	radiant energy fluence	L36	luminous efficacy, spectral luminous efficacy, maximum spectral luminous efficacy
L12	radiant energy fluence rate	L37	luminous efficiency, spectral luminous efficiency
L13	radiant intensity	L38	CIE colorimetric functions
L14	radiance	L39	trichromatic coordinates
L15	radiant exitance	L40	spectral absorption factor, spectral reflection factor, spectral transmission factor, spectral radiance factor
L16	irradiance	L41	optical density
L17	radiant exposure	L42	linear attenuation coefficient, linear absorption coefficient
L18	not used	L43	molar absorption coefficient
L19	not used	L44	refractive index
L20	not used	L45	item distance, image distance, focal distance
L21	emissivity, spectral emissivity, directional spectral emissivity	L46	vergence, lens power
L22	photon number		
L23	photon flux		
L24	photon intensity		
L25	photon luminance, photon radiance		

Table B.11 – Classification codes for amounts

Code	Description	Code	Description
M41	amount of persons	M53	amount of goods in aggregations
M43	amount of goods	M62	amount of information objects in transactions
M51	amount of goods in situations	M82	amount of currency transaction
M52	amount of goods in transactions	M83	amount of money in aggregations

Table B.12 – Classification codes for prices and tariffs

Code	Description	Code	Description
P51	prices of goods in situations	P71	price measuring units
P52	prices of goods in transactions	P81	price currency type

Table B.13 – Classification codes for dimensionless business quantities and counts

Code	Description	Code	Description
Q31	quantity of defined time periods in situations	Q59	quantity of products in quality
Q43	quantity of person aggregation	Q61	quantity of document (part)s
Q33	quantity of aggregated time periods	Q62	quantity of information objects in transactions
Q51	quantity of goods in situations	Q63	quantity of information objects in aggregations
Q52	quantity of goods in transactions	Q71	quantity of measurement unit situations
Q53	quantity of goods in aggregations	Q74	scalar quantity
Q54	quantity of goods in hierarchical structures	Q79	quantity of type measures
Q55	quantity of manufacturing processes	Q82	quantity of projects
Q56	quantity of product functions	Q83	quantity of procedure statistics

Table B.14 – Classification codes for business ratios and percentages

Code	Description	Code	Description
R41	percentages/ratios related to persons in situations	R71	percentages/ratios related to measurement unit situations
R51	percentages/ratios related to goods in situations	R81	percentages/ratios related to rate situations
R52	percentages/ratios related to goods in transactions	R82	percentages/ratios related to rate transactions
R53	percentages/ratios related to goods in aggregations	R91	percentages/ratios related to rate of abstract things
R54	percentages/ratios related to goods in hierarchical structures		

Table B.15 – Classification codes for quantities of space and time

Code	Description	Code	Description
T01	angle	T06	volume
T02	solid angle	T07	time, time interval, duration
T03	length, breadth, height, thickness, radius, diameter, length of path, distance, cartesian coordinates, radius of curvature	T08	angular velocity
T04	curvature	T09	angular acceleration
T05	area	T10	velocity
		T11	acceleration, acceleration of free fall, acceleration due to gravity

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

Table B.16 – Classification codes for quantities of nuclear reactions and ionizing radiations

Code	Description	Code	Description
V01	reaction energy	V32	diffusion coefficient for neutron number density
V02	resonance energy	V33	diffusion coefficient for neutron fluence rate
V03	cross-section, total cross-section	V34	neutron source density
V04	angular cross-section	V35	slowing-down density
V05	spectral cross-section	V36	resonance escape probability
V06	spectral angular cross-section	V37	lethargy
V07	macroscopic cross-section, volumic cross-section	V38	average logarithmic energy decrement
V08	particle fluence	V39	mean free path
V09	particle fluence rate (particle flux density)	V40	slowing-down area, diffusion area, migration area
V10	energy fluence	V41	slowing-down length, diffusion length, migration length
V11	energy fluence rate (energy flux density)	V42	neutron yield per fission, neutron yield per absorption
V12	current density of particles	V43	fast fission factor
V13	linear attenuation coefficient	V44	thermal utilization factor
V14	mass attenuation coefficient	V45	non-leakage probability
V15	molar attenuation coefficient	V46	multiplication factor, infinite medium multiplication factor, effective multiplication factor
V16	atomic attenuation coefficient	V47	reactivity
V17	half-thickness, half value thickness	V48	reactor time constant
V18	total linear stopping power	V49	activity
V19	total atomic stopping power	V50	energy imparted, mean energy imparted
V20	total mass stopping power	V51	specific energy imparted, absorbed dose
V21	mean linear range	V52	dose equivalent
V22	mean mass range	V53	absorbed dose rate
V23	linear ionization by a particle	V54	linear energy transfer
V24	total ionization by a particle	V55	kerma
V25	average energy loss per ion pair formed	V56	kerma rate
V26	mobility	V57	mass energy transfer coefficient
V27	ion number density, ion density	V58	exposure
V28	recombination coefficient	V59	exposure rate
V29	neutron number density		
V30	neutron speed		
V31	neutron fluence rate (neutron flux density)		

Annex C (informative)

Preparation of new classes and properties

C.1 Responsibilities

Classes and properties are used for computer-sensible specification of items, such as services, products, systems or components. The IEC Common Data Dictionary (IEC CDD) is the IEC repository for classes and properties usable for item characterization. All content provided for publication in IEC CDD is subject to the copyright rules of IEC.

NOTE This implies that proposed content is free from third-party rights restricting the exploitation rights of IEC.

C.2 Recommended elements of data set

The IEC 61360 series specifies a number of information objects that may be used for specifying domain technology. Domain specialists new to IEC 61360 dictionaries can feel confused by the number of information objects and can find it difficult finding a starting point. Therefore, the following list enumerates the core information objects for specifying products together with their characteristics.

The following information objects and attributes are relevant for a first version of a product specification. The information objects written in *italics* are indispensable.

- *Item_class*
- *Property*
- *Identifying attributes:*
 - *code*
 - *preferred_name*
 - *synonymous_name*
 - *short_name*
- *Semantic attributes:*
 - *definition*
 - *note*
 - *remark*
 - *source_document_of_definition*
- *Value attributes:*
 - *data_type*
 - *value_format*
 - *unit_of_measure*

Annex D (informative)

Rules for defining new versions and revisions of dictionary elements

D.1 Changes in the attributes of Property information objects

Two concepts are provided to mark updating operations for **Property** information objects. The concept of version number is part of the identification of an information object whereas the concept of revision number is for administrative control. Table D.1 gives an overview of how updating operations (add, modify, delete an attribute) should have effects on the version number (**V**) or the revision number (**R**). There are cases where the denoted operation is meaningless on the conceptual level, and syntactically constrained in the common dictionary schema (–), or forbidden (**X**). The table does not reflect the creation of information objects.

NOTE 1 To modify means to modify the value of an attribute.

NOTE 2 If changes of version or revision numbers are propagated in the classification hierarchy, situations might occur where modifications of properties could entail unwanted version or revision changes of major parts of the classification tree.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

**Table D.1 – Overview of configuration management
in Property updating operations**

Attribute	Add	Modify	Delete
applicable_class ^a	V	V ⁱ	X
code	–	X	X
code_of_unit	R/ V ⁱ	R/ V ⁱ	X
code_of_alternative_unit	R/ V ⁱ	R/ V ⁱ	X
code_of_list_of_units	R/ V ⁱ	R/ V ⁱ	X
data_element_type_class	–	R	X
data_type	–	X/ V ^b	X
definition	–	X/ V/ R ^c	X
depends_on	V ^e	V ⁱ	V
drawing_reference	R	R	R
formula	R	R	R
note	V	V/ R	V ^d
preferred_letter_symbol	V	V	X
preferred_name	–	V	X
referenced_class_identifier	X	V ^{g,i}	X
remark	R	R	R
revision_number	–	R	X
revision_released_on			
short_name	–	V	X
source_document_of_definition	R	R	R
synonymous_letter_symbol	R	R	R
synonymous_name	R	R	R
unit_of_measure	R/ V ⁱ	R/ V ⁱ	X
value_format	–	V ^f	X
version_number	–	V	X
version_released_on	–	X	X

V: version change, R: revision change, X: no change of version and revision.

^a There is no formal link from a property to the class it is made applicable in. However, the addition or modification of an applicable class can result in an update of the definition.

^b The SUBTYPE of the **data_type** that defines the domain cannot be changed (X). The only allowed changes consist of

- 1) adding new allowed values in an enumerated type; or
 - 2) changing the version of the class that constitutes the domain of the property.
- These changes modify the version (V) of the property.

^c Changes to the definition that affect the semantics lead to a change of the code, i.e. constitute a new property. Extending the definition, e.g. to include a new applicable class, results in a version change. Editorial changes to the definition text lead to a revision change.

^d It is only allowed to delete the note when the definition covers sufficiently the originally defined meaning.

^e It is only allowed to add an optional condition to a dependent property with condition(s). It is not allowed to add a condition to a non-dependent property.

^f It is only allowed to extend the defined length of the value format; all other changes result in a change of the code.

^g It is only allowed to change the version of the class that constitutes the domain of the data element type.

^h The addition of a new code and associated meaning, and semantic type changes to the meaning of an existing code result in a version change. In case a code is modified, it should be decided if a new property is issued on a case-by-case basis.

ⁱ Changes of the version number of the referenced properties are not considered to be a modification.

D.2 Changes in the attributes of class information objects

Two concepts are provided to mark updating operations for class information objects. The concept of version number is part of the identification of a class whereas the concept of revision number is one for administrative control of a class.

Table D.2 gives an overview of how updating operations (add, modify, delete some attributes of a class) have effects on the version number (**V**) or the revision number (**R**) for the given class. There are cases where the denoted operation is meaningless, and syntactically constrained in the common dictionary schema (**–**), or forbidden at all (**X**). The table does not reflect the creation of an **Item_class**.

NOTE 1 To modify means to modify the value of an attribute.

NOTE 2 If changes of version or revision numbers are propagated in the classification hierarchy, situations might occur where modifications of classes could entail unwanted version or revision changes of major parts of the classification tree.

**Table D.2 – Overview of configuration management
in class updating operations**

Attribute	Add	Modify	Delete
applicable_data_element_type	V	V ^e	X
code	–	X	X
coded_name	–	V	X
definition	–	R ^c	X
drawing_reference	–	V ^e	V
note	V	V/ R	V ^d
preferred_name	–	V ^b	X
remark	R	R	R
revision_number	–	R	X
source_document_of_definition	R	R	R
superclass	V ^a	V	V
synonymous_name	R	R	R
version_number	–	V	X
V: version change, R: revision change, X: no change of version and revision.			
^a Modification of superclass should not remove any inherited visible or applicable property or type. The modification can consist either in changing the version of the referenced superclass, which defines a new version of the subclass, or, for example, in adding an intermediate class in between one class and its previous superclass.			
^b If the version of a class is changed due to a change in the preferred name, subclasses should also change the version. Editorial changes lead to a revision change.			
^c Changes to the definition that affect the semantics lead to a change of the code, i.e. constitute a new class. Editorial changes lead to a revision change.			
^d It is only allowed to delete the note when the definition covers sufficiently the originally defined meaning.			
^e Changes of the version number of the referenced property information objects are not considered to be a modification.			

Annex E (informative)

Classifying_data_element_type attribute

A **classifying_data_element_type** attribute has a value domain consisting of a value list containing at least two options. Its principal function, when associated with an **Item_class**, is to provide a list of that class' subclasses. Thus, in the IEC dictionary, all classes except for those at the ends of branches of the classification tree have an associated **classifying_data_element_type**, which is one of the attributes applicable to that class.

The use of the **classifying_data_element_type** is supported in the information model of IEC 61360-2 and is a mandatory requirement of IEC 61360-1, though some dictionaries currently under development using the same information model have chosen not to take advantage of the facility.

In this document, there is a requirement that the **coded_name** of a subclass shall be identical to one of the value codes in the value list of its super-class, but this requirement is not included in the information model of IEC 61360-2. It is also desirable that the **preferred_name** of a subclass be the same as the value meaning associated with the corresponding value code, but this not a requirement of either this document or IEC 61360-2.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

Annex F (informative)

Conventions for names and definitions

F.1 Conventions for writing definitions

F.1.1 General

Clause F.1 provides guidance on the wording of definitions. Subclauses F.1.2 and F.1.3 give a high-level overview of the conventions for writing definitions as laid down in ISO/IEC 11179-4 and ISO 704, respectively.

F.1.2 ISO/IEC 11179-4

F.1.2.1 Requirements

A definition shall:

- a) be stated in the singular;
- b) state what the concept is, not what it is not;
- c) be stated as one descriptive phrase or sentence;
- d) contain only commonly understood abbreviations;
- e) be expressed without embedding definitions of other data or underlying concepts.

F.1.2.2 Recommendations

A definition should:

- a) state the essential meaning of the concept;
- b) be precise and unambiguous;
- c) be concise;
- d) be able to stand alone;
- e) be expressed without embedding rationale, functional usage, or procedural information;
- f) avoid circular reasoning;
- g) use the same terminology and consistent logical structure for related definitions;
- h) be appropriate for the type of data being defined.

F.1.3 ISO 704

- a) Define what it is, not what it is not.
- b) Define the concept, do not write a list of examples (extensional definitions, i.e. a list of subordinate concepts, are allowed in highly specialized domains only).
- c) Avoid circular definitions.
- d) Do not define two concepts. A definition shall describe only one concept. It shall not include hidden definitions for any concepts used to identify characteristics. Any characteristic that requires an explanation shall be defined separately as a concept or given in a note.
- e) A definition is valid if it can replace a designation in a text without loss of or change in meaning.
- f) Definitions shall be as brief as possible and as complex as necessary. Complex definitions can contain several dependent clauses, but carefully written definitions contain only that information which makes the concept unique. Any additional descriptive information deemed necessary should be included in a note.

- g) The definition should not contain characteristics that belong logically to superordinate or subordinate concepts.
- h) A definition shall describe the content of the concept precisely. It shall be neither too narrow nor too broad.

F.1.4 Additional conventions

- a) The term “value” shall not be written in plural even if it is a level type (e.g. minimum, typical and maximum value).
- b) Changing attributes to conform to the new conventions of this document is by default a revision change (minor change) unless specified otherwise, and is done along with a change request associated with the relevant classes or properties.

F.2 Conventions for writing names

F.2.1 Requirements

A dictionary item name shall:

- a) be stated in the singular;
- b) be written in lower case with the exception of particular abbreviations and acronyms that are commonly written in upper case and approved for use;
- c) contain only commonly understood abbreviations and acronyms;
- d) be consistent with the concept specified in the definition.

EXAMPLE The name ought not include the term “continuous” in the property name attribute if “continuous” is not addressed in its definition.

F.2.2 Recommendations

A dictionary item name should:

- a) be concise;
- b) use the same terminology and consistent logical structure for related names.

F.2.3 Mechanical quantitative property names

Mechanical quantitative property names should start with the concept or item being specified, followed by the measured aspect such as length, height, or diameter.

F.2.4 Electrical quantitative property names

- a) They should reflect the electrical symbol in words reading in reverse order.
- b) They should start with the concept being specified, followed by the characterized quantity such as voltage, current, capacitance, or temperature.
- c) They should use the term “repetitive” as part of the name, when defined as such in the symbol:
 - “repetitive” in symbol as subscript “R” (e.g. VRRM, Repetitive peak reverse voltage), “non-repetitive” in symbol as subscript “S” (e.g. VRSM, Surge reverse voltage). Do not use the term “single pulse”.
 - Include “repetitive” and “non-repetitive” in primary property name and the definition.“Repetitive” should not be used instead of “pulsed”. Pulsed does not indicate if it is repetitive or non-repetitive.
- d) The concept may be preceded by one or more qualifiers such as maximum, peak, average, or total.
- e) The measured quantity is possibly followed by a non-quantitative condition such as: “from junction to lead”.

- f) For limiting values, names should be chosen that include the term "limiting" at the beginning.

EXAMPLE See Table F.1 for a name structure for electrical quantitative **properties**.

Table F.1 – Example of the name structure for electrical quantitative properties

Qualifier	Concept	Characterized quantity	Non-quantitative condition
maximum	input	voltage	
	ambient	temperature	
initial	junction	temperature	
maximum repetitive peak	off-state	voltage	
transient	thermal	impedance	from junction to lead

F.2.5 Non-quantitative property names

Non-quantitative property names start with the concept being specified followed by a qualifier such as type, code, name, or description.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

Annex G (normative)

Value format specification

G.1 General

ISO 13584-42 and IEC 61360-2 provide a particular syntax to specify the allowed formats for the string and numeric values that may be associated with a property.

NOTE 1 The format description language defined here is based on stipulations given in ISO 6093:1985 and ISO 9735-1:2002.

EXAMPLE 1 The format NR1..3 allows to specify that only integer values consisting of exactly three digits are allowed.

NOTE 2 No value format is defined for any other data_type, including boolean_type.

NOTE 3 In ISO 13584-42, it is not mandatory to define the format of property values.

NOTE 4 The syntax of the allowed formats is defined in Annex G using a subset of the Extended Backus-Naur Form (EBNF) defined in ISO/IEC 14977.

EXAMPLE 2 The syntax of the format NR1 3 are the characters 'NR1' ' ' '3'.

The semantics of the characters used to represent a value cannot be defined using the EBNF. Thus the meaning is specified separately for each part of the format.

EXAMPLE 3 The syntax of the format "NR1 3" has the following meaning: "NR1" means that only an integer value may be represented. The space means that this format specifies a fixed number of characters. "3" means that exactly three digits are required.

G.2 Notation

Table G.1 summarizes the subset of the ISO/IEC 14977 EBNF syntactic meta-language used by ISO 13584-42 / IEC 61360-2 specifying the value format of properties.

Using these notations, the syntax of the subset of the EBNF meta-language used by ISO 13584-42 / IEC 61360-2 to specify the value format of properties is summarized by the following grammar (the meta-identifier character, letter and digit are not detailed):

```
syntax = syntaxrule , { syntaxrule };
syntaxrule = metaidentifier , '=' , definitionslist , ';';
definitionslist = singledefinition , { '|' , singledefinition };
singledefinition = term , { ',', term };
term = primary , [ '-', primary ];
primary = optionalsequence | repeatedsequence | groupedsequence |
        metaidentifier | terminal | empty;
optionalsequence = '[' definitionslist ']';
repeatedsequence = '{' definitionslist '}';
groupedsequence = '(' definitionslist ')';
metaidentifier = letter , { letter };
terminal = '"', (character - '"'), {character - '"'}, '"'
        | "'", (character - "'"), {character - "'"}, "'";
empty =;
```

The equals sign '=' indicates a syntax rule. The meta-identifier on the left may be re-written by the combination of the elements on the right. Any spaces appearing between the elements are meaningless unless they appear within a terminal. A syntax rule is terminated by a semicolon ';'.

Table G.1 – ISO/IEC 14977 EBNF syntactic meta-language

Representation	ISO/IEC 10646 character names	Meta-language symbol and role
' '	apostrophe	First quote symbol: represents language terminals. Terminal shall not contain apostrophe. EXAMPLE 1 'Hello'
" "	quotation mark	Second quote symbol: represents language terminals. Terminal shall not contain quotation mark. EXAMPLE 2 "John's car"
()	left parenthesis, right parenthesis	Start/end group symbols. The content is considered as a single symbol.
[]	left square bracket, right square bracket	Start/end option symbols. The content may be present or not present.
{ }	left curly bracket, right curly bracket	Start/end repeat symbols. The content may be present 0 to <i>n</i> times.
-	hyphen-minus	Except symbol.
,	comma	concatenate symbol.
=	equals sign	Defining symbol. Syntax rule: defines the symbol of the left by the formula on the right.
	vertical line	Alternative separator symbol.
;	semicolon	Terminator symbol. End of a syntax rule.

The use of a meta-identifier within a definition-list denotes a non-terminal symbol which appears on the left side of another syntax rule. A meta-identifier is composed of letters or digits, the first character being a letter. If a term contains both a `primary` preceding a minus sign, and a `primary` that follows the minus sign, only the sequence of symbols that are represented by the first `primary` and that are not represented by the second `primary` are represented by the term.

EXAMPLE 1 The notation `""`, character `- ""`, `""` means any character but the apostrophe character, inserted between two apostrophe characters.

The `terminal` denotes a symbol which cannot be expanded further by a syntax rule, and which appears in the final result. Two ways to represent a `terminal` are allowed: either a set of characters without apostrophe, inserted between two apostrophes, or a set of characters without quotation marks, inserted between two quotation marks.

EXAMPLE 2 Assume that we want to describe, by such a grammar, the price of a product in euros. Such a price is a positive number with no more than two digits in the cents part. We introduce three meta-identifiers associated with three syntax rules:

```
digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';
cents = [ '.' , digit [ , digit ] ];
euros = digit { , digit } cents;
```

With these syntax rules, "012", "4323.3", "3.56" are examples of permitted representations of euros. "12.", ".10" are examples of not permitted representations of euros.

G.3 Data value format types

The grammar defined in Annex G defines eight different types of value formats: four quantitative and five non-quantitative value formats.

Clause G.4 defines the meta-identifiers that are used to specify these formats. Clause G.5 defines the syntax rule for the four meta-identifiers that represent the four quantitative value formats, together with their meaning at the value level. Clause G.6 defines the meta-identifiers for the five non-quantitative value formats, together with their meaning at the value level.

G.4 Meta-identifier used to define the formats

The meta-identifiers used in the grammar that define the various value formats are the following:

```
dot = '.';
decimalMark = '.';
exponentIndicator = 'E';
numeratorIndicator = 'N';
denominatorIndicator = 'D';
leadingDigit = '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';
lengthOfExponent = leadingDigit, {trailingDigit};
lengthOfIntegerPart = (leadingDigit, {trailingDigit});
lengthOfNumerator = leadingDigit, {trailingDigit};
lengthOfDenominator = leadingDigit, {trailingDigit};
lengthOfFractionalPart = (leadingDigit, {trailingDigit}) | '0';
lengthOfIntegralPart = (leadingDigit, {trailingDigit}) | '0';
lengthOfNumber = leadingDigit, {trailingDigit};
trailingDigit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';
signedExponent = 'S';
signedNumber = space, 'S';
space = ' ';
variableLengthIndicator = '..';
decimalMark: separator between integral and fractional part of numbers of format NR2 or NR3.
leadingDigit: first cipher of a number comprising one or more ciphers.
trailingDigit: one of the ciphers that combines to form numbers, except the first one.
```

If a number comprises only one digit, no trailingDigit is present.

G.5 Quantitative value formats

G.5.1 General

The four quantitative value format syntax rules and their meanings for value representation are defined in G.5.2 to G.5.5.

G.5.2 NR1-value format

The NR1-value syntax specifies the format of an integer property value.

Syntax rule:

```
NR1Value = 'NR1', ((signedNumber, variableLengthIndicator) |
(signedNumber, space) | variableLengthIndicator | space),
lengthOfNumber;
```

The meaning of NR1-value format components for value representation is as follows:

- 'NR1': the value shall be an integer. NR1-values shall not contain any spaces.
- `lengthOfNumber`: number of digits of the value. If preceded by a `variableLengthIndicator` the actual number of digits may be less.
- `signedNumber`: if `signedNumber` is present, the related number shall have either a positive, negative, or zero value. In case of positive values, a '+' sign may be present. Negative values shall be preceded by a '-' sign. The value zero shall not be preceded by a '-' sign.
- `variableLengthIndicator`: if `variableLengthIndicator` is present, the related number shall contain a number of digits that is less than or equal to its length specification, i.e. less than or equal to `lengthOfNumber`.

G.5.3 NR2-value format

The NR2-value syntax specifies the format of a real property value that does not need an exponent.

Syntax rule:

```
NR2Value = 'NR2', ((signedNumber, variableLengthIndicator) |
(signedNumber, space) | variableLengthIndicator | space),
lengthOfIntegralPart, decimalMark, lengthOfFractionalPart;
```

The meaning of NR2-value format components for value representation is as follows:

- 'NR2': the value shall be a real number. NR2-values shall not contain spaces.
 - `lengthOfFractionalPart`: number of digits of the fractional part of the number. If preceded by a `variableLengthIndicator` the actual number of digits of the fractional part may be less.
- NOTE `lengthOfFractionalPart` implicitly specifies the recommended accuracy of the value. The actual accuracy of the number from which this value was derived can have been greater than the value expressed here.
- `lengthOfIntegralPart`: number of digits of the integral part of the number. If preceded by a `variableLengthIndicator` the actual number of digits of the integral part may be less.
 - `signedNumber`: if `signedNumber` is present, the related number shall have either a positive, negative, or zero value. In case of positive values a '+' sign may be present. Negative values shall be preceded by a '-' sign. The value zero shall not be preceded by a '-' sign.
 - `variableLengthIndicator`: if `variableLengthIndicator` is present, either integral part or fractional part of the number or both parts shall contain a number of digits that is less than or equal to its related length specification, i.e. less than or equal to `lengthOfIntegralPart` or `lengthOfFractionalPart`. At least one cipher shall be present in the number.

G.5.4 NR3-value format

The NR3-value syntax specifies the format of a real property value that is represented with an exponent.

Syntax rule:

```
NR3Value = 'NR3', ((signedNumber, variableLengthIndicator) |
(signedNumber, space) | variableLengthIndicator | space),
```

`lengthOfIntegralPart`, `decimalMark`, `lengthOfFractionalPart`,
`exponentIndicator`, [`signedExponent`], `lengthOfExponent`;

The meaning of NR3-value format components for value representation is as follows:

- 'NR3': the value shall be a real number with an exponent of base 10. There shall be at least one digit and the decimal mark in the mantissa. The exponent shall contain at least one digit, too. NR3-values shall not contain spaces.
- `exponentIndicator`: separator between mantissa and exponent in numbers of format NR3.
- `lengthOfExponent`: number of digits of the exponent. If preceded by a `variableLengthIndicator` the actual number of digits of the exponent may be less.

NOTE 1 Any signs or decimal marks present are not counted by `lengthOfNumber`, `lengthOfIntegralPart`, `lengthOfFractionalPart` or `lengthOfExponent`.

- `lengthOfFractionalPart`: number of digits of the fractional part of the mantissa. If preceded by a `variableLengthIndicator` the actual number of digits of the fractional part may be less.

NOTE 2 `lengthOfFractionalPart` implicitly specifies the recommended accuracy of the value. The actual accuracy of the number from which this value was derived can have been greater than the value expressed here.

- `lengthOfIntegralPart`: number of digits of the integral part of the mantissa. If preceded by a `variableLengthIndicator` the actual number of digits of the integral part may be less.
- `signedExponent`: if `signedExponent` is present, the related exponent shall have either a positive, negative, or zero value. In case of positive values, a '+' sign may be present. Negative values shall be preceded by a '-' sign. The value zero shall not be preceded by a '-' sign.
- `variableLengthIndicator`: if `variableLengthIndicator` is present, the related number or exponent shall contain a number of digits that is less than or equal to its related length specification, i.e. less than or equal to `lengthOfIntegralPart`, `lengthOfFractionalPart`, or `lengthOfExponent`. At least one cipher shall be present in the mantissa and in the exponent.

G.5.5 NR4-value format

The NR4-value syntax specifies the format of a rational property value that is presented with an integer part or a fraction part with a denominator and a numerator.

Syntax rule:

```
NR4Value = 'NR4', ((signedNumber, variableLengthIndicator) |
(signedNumber, space) | variableLengthIndicator | space),
lengthOfIntegerPart, numeratorIndicator, lengthOfNumerator,
denominatorIndicator, lengthOfDenominator;
```

The meaning of NR4-value format components for value representation is as follows:

- 'NR4': the value shall be a rational number represented either as an integer, or as a fraction consisting of a numerator and a denominator. There shall be at least one digit either in the integer part, or both in the numerator and in the denominator part. If one part of the fraction contains a digit, the other part shall also contain some digits. NR4 number values shall not contain any spaces.

EXAMPLE 12 and $\frac{3}{4}$ are valid numbers that can be presented in the NR4 format. $12\frac{3}{4}$ is not a valid presentation.

- `numeratorIndicator`: separator between the integer part description and the fraction part description in NR4 formats.

- `lengthOfNumerator`: number of digits of the numerator. If the value of the rational number is completely represented by its integer part, neither the numerator of the fraction nor its denominator shall be represented. If preceded by a `variableLengthIndicator` the actual number of digits of the numerator may be less.
- `denominatorIndicator`: separator between the numerator part description and the denominator part description in NR4 formats.
- `lengthOfDenominator`: number of digits of the denominator. If the value of the rational number is completely represented by its integer part, neither the numerator of the fraction nor its denominator shall be represented. If preceded by a `variableLengthIndicator` the actual number of digits of the denominator may be less.
- `lengthOfIntegerPart`: number of digits of the integer part of the rational number. If preceded by a `variableLengthIndicator` the actual number of digits of the denominator may be less.
- `variableLengthIndicator`: if `variableLengthIndicator` is present, the three parts of the rational number shall contain a number of digits that is less than or equal to its related length specification, i.e. less than or equal to `lengthOfIntegralPart`, `lengthOfNumerator`, or `lengthOfDenominator`. At least one cipher shall be present either in the integral part, or in the two parts of the fraction.

G.6 Non-quantitative value formats

G.6.1 General

The five non-quantitative value format syntax rules and their meanings are defined in G.6.2 to G.6.6.

NOTE 1 For `translatable_string_type` the value format applies to any language-specific representation of the string.

NOTE 2 For `non_quantitative_code_type`, the value format applies to the code.

Non-quantitative values are represented by strings which comprise characters. The length of a string may be either specified by directly specifying the upper limit of the number of contained characters or by specifying that the total number of characters may be any integral multiple of the length specified:

Syntax rule:

```
factor = leadingDigit, {trailingDigit};
numberOfCharacters = (leadingDigit, {trailingDigit}) | ( ' (nx',
factor, ' ) ' )
```

The meaning of the factor components is as follows:

- `factor`: when `factor` is present, then `numberOfCharacters` shall be any integral multiple of the value given in `factor`. `factor` shall not contain the value zero.
- `numberOfCharacters`: determines the maximum number of characters contained in the string.

G.6.2 Alphabetic value format

An “Alphabetic value format (A)” defines the value format of a string that contains alphabetic letters. Thus, the content shall be taken from the characters of row 00, cell 20, cells 40 to 7E, or cells C0 to FF, of the Basic Multilingual Plane (BMP) (Plane 00 of Group 00) of ISO/IEC 10646. Due to potential interpretation problems of value content within components of one system or of multiple systems, where possible, the characters used should be restricted to the G0 set of ISO/IEC 646 or to row 00, columns 002 to 007 of ISO/IEC 10646.

NOTE For alternative languages, as supported by translated data, the relevant characters or ideographs of the related language-specific character set are available as defined by the Unicode Standard. In some cases, there is no 1:1 relation between the characters of the source language and the characters of the target language.

EXAMPLE CJK (Chinese-Japanese-Korean) ideographs.

Syntax rule:

```
AValue = 'A', (space | variableLengthIndicator), numberOfCharacters;
```

The meaning of A-value format components for value representation is as follows:

- 'A': the value shall be a string, or several substrings, of alphabetic letters.
- variableLengthIndicator: if variableLengthIndicator is present, the string may contain fewer characters than indicated by numberOfCharacters. The string shall contain at least one character.

G.6.3 Mixed characters value format

A “Mixed characters value format (M)” format defines the value format of a string that may contain any character specified in Clause 26.

NOTE For alternative languages as supported by translated data, the relevant characters or ideographs of the related language-specific character set are available as defined by the Unicode Standard.

EXAMPLE CJK (Chinese-Japanese-Korean) characters.

Syntax rule:

```
MValue = 'M', (space | variableLengthIndicator), numberOfCharacters;
```

The meaning of M-value format components for value representation is as follows:

- 'M': the value shall be a string, or several substrings.
- variableLengthIndicator: if variableLengthIndicator is present, the string may contain fewer characters than indicated by numberOfCharacters. The string shall contain at least one character.

G.6.4 Number value format

A “Number value format (N)” defines the value format of a string that contains numeric digits only. Thus, the content shall be taken from the characters of row 00, cell 2B, cell 2D, cells 30 to 39, or cell 45 of the Basic Multilingual Plane (BMP) (Plane 00 of Group 00) of ISO/IEC 10646.

NOTE For alternative languages as supported by translated data, the relevant characters or ideographs of the related language-specific character set are available as defined by the Unicode Standard.

EXAMPLE Table G.2 shows the transposition of the European digits “0” to “9” into Arabic digits.

Table G.2 – Transposing European style digits into Arabic digits

European digits	9	8	7	6	5	4	3	2	1	0
Arabic digits	٩	٨	٧	٦	٥	٤	٣	٢	١	٠

Syntax rule:

```
NValue = 'N', (space | variableLengthIndicator), numberOfCharacters;
```

The meaning of N-value format components for value representation is as follows:

- 'N': the value shall be a string, or several substrings, of numeric digits.
- `variableLengthIndicator`: if `variableLengthIndicator` is present, the string may contain fewer characters than indicated by `numberOfCharacters`. The string shall contain at least one character.

G.6.5 Mixed alphabetic or numeric characters value format

A “Mixed alphabetic or numeric characters value format (X)” defines the value format of a string that contains alphanumeric characters, i.e. any combination of characters from A-value format or N-value format.

NOTE For alternative languages as supported by translated data, the relevant characters or ideographs of the related language-specific character set are available as defined by the Unicode Standard.

Syntax rule:

```
XValue = 'X', (space | variableLengthIndicator), numberOfCharacters;
```

The meaning of X-value format components for value representation is as follows:

- 'X': the value shall be a string, or several substrings, of alphanumeric, i.e. any combination of alphabetic and numeric, characters.
- `variableLengthIndicator`: if `variableLengthIndicator` is present, the string may contain fewer characters than indicated by `numberOfCharacters`. The string shall contain at least one character.

G.6.6 Binary value format

A “Binary value format (B)” defines the value format of a string that contains binary characters, i.e. “0” or “1”. Thus the content shall be taken from the characters of row 00, cell 30 or 31, of the Basic Multilingual Plane (BMP) (Plane 00 of Group 00) of ISO/IEC 10646.

NOTE For alternative languages as supported by translated data, the relevant characters or ideographs of the related language-specific character set are available as defined by the Unicode Standard.

Syntax rule:

```
BValue = 'B', (space | variableLengthIndicator), numberOfCharacters;
```

The meaning of B-value format components for value representation is as follows:

- 'B': the value shall be a string, or several substrings, consisting of binary values, i.e. the characters “0” or “1” or sequences thereof.
- `variableLengthIndicator`: if `variableLengthIndicator` is present, the string may contain fewer characters than indicated by `numberOfCharacters`. The string shall contain at least one character.

G.7 HTML5 format

In cases where values have to be presented in a non-numerical form, e.g. as a mathematical formula or in pictorial form, HTML5 (see W3C Recommendation 28:2014) may be used.

EXAMPLE In case of three phase electrical systems, numeric values are often presented as multiples of $\frac{1}{\sqrt{3}}$ or of $\sqrt{3}$.

Syntax rule:

```
Html5Value = 'HTML5';
```

G.8 Value examples

Table G.3 contains some examples for values that may be contained in numbers and strings characterized by the above value format scheme.

Table G.3 – Number value examples

Value format	Examples of possible values
NR1 3	123; 001; 000;
NR1..3	123; 87; 5
NR1S 3	+123; +000;
NR1S..3	-123; +1; 0; -12;
NR2 3.3	123.300; 000.400; 000.420;
NR2..3.3	321.233; 1.234; 23.56; 9.783; .72; 324.
NR2S 3.3	-123.123; +123.300;
NR2S..3.3	-123.123; +12.3; 0.1; +.4; -3.; 0.; .0;
NR3 3.3E4	123.123E0004; 003.000E1000;
NR3 3.3ES4	123.123E+0004; 123.123E0004; 123.000E-0001
NR3..3.3E4	123.123E0004; .123E0001; 5.E1234
NR3S 3.3ES4	+123.123E+0004; 123.000E-0001;
NR3S..3.3ES4	-123.123E+0004; +1.00E-01; .0E0; +3.E-1; -.2E-1000;
NR4 3N2D2	001; 02/03; 012; 00/01; 123; 03/04; 000; 01/04;
NR4..3N2D2	1; ½; 12; 123; ¼; ¼;
NR4S 3N2D2	+001; -02/03; -012; 00/01; 123; 03/04; -000; +01/04;
NR4S..3N2D2	-1; -¼; 12; +123; -¾; +¼;
A 19	My name is Reinhard, abcdefghijklmnopqrs
A..3	Abc; de; G
X..5	A23RN1; B1; ca
M..10	A23RN1; B1; ca. 256 µm;
N (nx5)	12345; 1234512345; 222223333344444;
N..(nx5)	1234; 12345; 34512345; 1234512345; 23333344444; 222223333344444; -3; 5E2;
B 1	0; 1;
B 3	011; 101;

Annex H (informative)

Modelling notation

H.1 General

For specifying the data model of this document, class diagrams of Unified Modelling Language (UML) are used. Clauses H.2 to H.5 give a short introduction of the utilized UML constructs. For a detailed description see ISO/IEC 19505-1.

H.2 UML Class

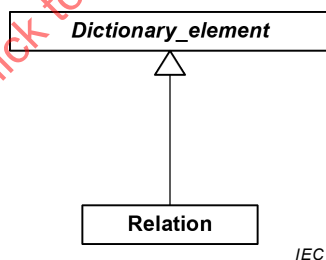
The UML class diagrams in Figure H.1 to Figure H.3 show the concepts used to model the information objects, their attributes and relationships.

NOTE An IEC 61360-1 concept can comprise more than one UML class information object.

UML classes have attributes specifying their characteristics. Relationships between the classes reveal how they are structured in terms of each other. Classes are related in a variety of ways, as described in Clauses H.3 to H.5.

H.3 Generalization

A generalization is a relationship between a general class (superclass) and a specific class (subclass). A generalization provides for the subclass to inherit all information including relationships from the superclass. Since a subtype inherits all the characteristics from its superclass, the subclass shall specify only the additional information as characteristics.



IEC

Figure H.1 – Example for generalization

Figure H.1 is an example of generalization. In this example **Relation** is a more specific type of **Dictionary_element**. **Relation** inherits all attributes and associations from **Dictionary_element**.

EXAMPLE 1 The UML class "**Relation**" in Figure H.1 is a subtype of the UML_class "**Dictionary_element**".

Abstract supertypes have a name which is written in italics.

EXAMPLE 2 The UML class "***Dictionary_element***" in Figure H.1 is an abstract supertype.

H.4 Simple association

An association is a conceptual connection between classes. Each association has two "association ends". Each association end describes the role the target class (i.e. the class the association end goes to) has in relation to the source class (i.e. the class the association end

comes from). Each association end has cardinality (multiplicity), which specifies the number of information objects that may participate in the given relationship.

EXAMPLE Figure H.2 shows a case where the UML class "**Property**" has an optional reference to a UML class "**Drawing**".

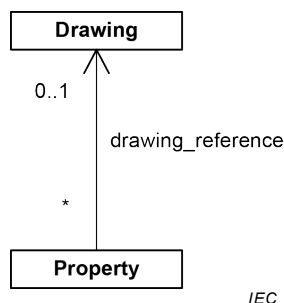


Figure H.2 – Example of an association

In this example, a **Property** information object may refer to zero or one **Drawing** information objects whereas any **Drawing** may be referenced by zero, one or many **Property** information objects.

NOTE An asterisk "*" stands for "any number". If no cardinality is specified "any number (*)" is assumed.

H.5 Modularization with UML package

For a better overview the UML model was split into modules. Each of these can easily be presented on a single page. For implementation of the modularization the UML construct "package" was used.

A package is a collection of UML elements that can be used for dividing a UML model into smaller, better manageable parts. Almost all UML elements can be grouped into UML packages. Each package establishes an own namespace for its members. A namespace may import elements from other namespaces. If a member of a namespace with the name N is an element with the name x, then the member can be referred to by a qualified name of the form "N::x".

EXAMPLE Figure H.3 shows a case where the UML class "Item_class" references the UML class "Value_list", which is owned by the UML package "Value_list_concept".

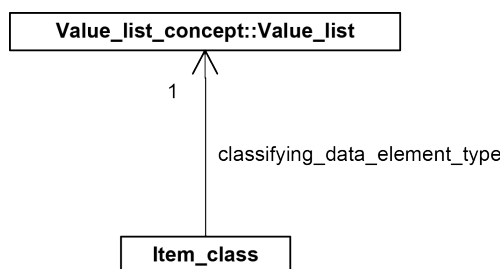


Figure H.3 – Example of an association to a UML class owned by another UML package

In this document, UML packages are used for grouping UML elements. Thus the model is organized in modules and a more clear presentation of its elements becomes possible. Figure H.4 shows the modules used in this document.

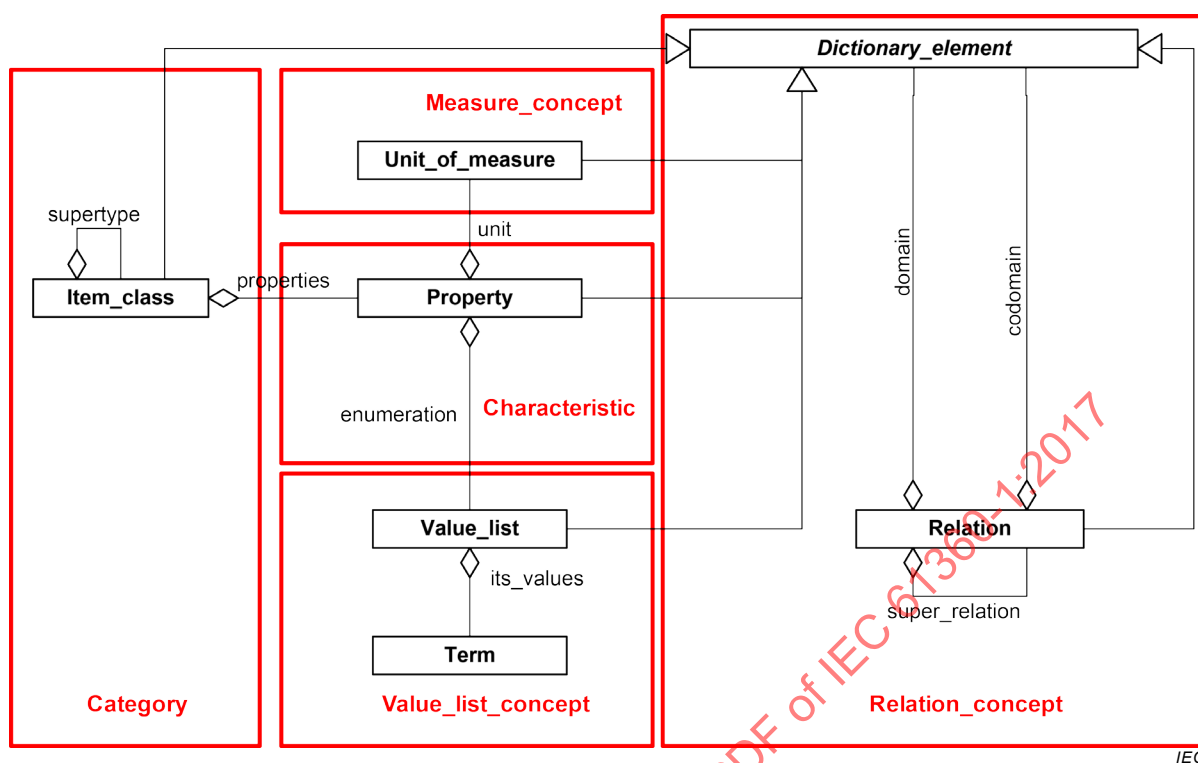


Figure H.4 – Division of the IEC 61360-1 model into modules

Bibliography

IEC 60027 (all parts), *Letter symbols to be used in electrical technology*

IEC 60747 (all parts), *Semiconductor devices*

IEC 60191-4:2013, *Mechanical standardization of semiconductor devices – Part 4: Coding system and classification into forms of package outlines for semiconductor device packages*

IEC 60748 (all parts), *Semiconductor devices – Integrated circuits*

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

IEC 61987 (all parts), *Industrial-process measurement and control – Data structures and elements in process equipment catalogues*

IEC 62683, *Low-voltage switchgear and controlgear – Product data and properties for information exchange*

ISO/IEC 6523-1, *Information technology – Structure for the identification of organizations and organization parts – Part 1: Identification of organization identification schemes*

ISO/IEC 11179-4, *Information technology – Metadata registries (MDR) – Part 4: Formulation of data definitions*

ISO/IEC 11179-6, *Information technology – Metadata registries (MDR) – Part 6: Registration*

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

ISO/IEC Guide 77-2, *Guide for specification of product properties and classes – Part 2: Technical principles and guidance*

ISO 704, *Terminology work – Principles and methods*

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

ISO 1087-1:2000, *Terminology work – Vocabulary – Part 1: Theory and application*

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

ISO 9735-1, *Electronic data interchange for administration, commerce and transport (EDIFACT) – Application level syntax rules (Syntax version number: 4, Syntax release number: 1) – Part 1: Syntax rules common to all parts*

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

ISO 10303-42:2014, *Industrial automation systems and integration – Product data representation and exchange – Part 42: Integrated generic resource: Geometric and topological representation*

ISO 13584-24:2003, *Industrial automation systems and integration – Parts library – Part 24: Logical resource: Logical model of supplier library*

ISO 13584-25:2004, *Industrial automation systems and integration – Parts library – Part 25: Logical resource: Logical model of supplier library with aggregate values and explicit content*

ISO 22274:2013, *Systems to manage terminology, knowledge and content – Concept-related aspects for developing and internationalizing classification systems*

W3C Recommendation 10:2014, *Mathematical Markup Language (MathML)*

W3C Recommendation 28:2014, *HTML5 – A vocabulary and associated APIs for HTML and XHTML*

BANEY, Douglas M., et al., *Theory and Measurement Techniques for the Noise Figure of Optical Amplifiers*. *Optical Fiber Technology*. 2000, **6**, 122-154. ISSN 1068-5200

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

SOMMAIRE

AVANT-PROPOS	172
INTRODUCTION.....	174
1 Domaine d'application	176
2 Références normatives	176
3 Termes, définitions et abréviations	177
3.1 Termes et définitions	177
3.2 Abréviations	183
4 Bases des concepts des dictionnaires normalisés de l'IEC 61360.....	184
5 Identification du dictionnaire	186
5.1 Généralités	186
5.2 Fournisseur du dictionnaire.....	186
5.3 Code.....	187
5.4 Numéro de version.....	187
5.5 Date de la version actuelle.....	188
5.6 Numéro de révision	188
6 Propriété	189
6.1 Vue d'ensemble	189
6.2 Spécification de l'objet d'information	193
6.3 Classe de type d'élément de données.....	195
6.4 Dépend de	195
6.5 Formule	196
6.6 Symbole littéral préféré	196
6.7 Symbole littéral synonyme	197
6.8 Attributs internes d'une Propriété	197
6.9 Code d'autre unité	197
6.10 Code de liste d'unités	198
6.11 Code_de_l_unité.....	198
6.12 Classe de définition	199
6.13 Référence de dessin	199
7 Attributs_d'identification	199
7.1 Vue d'ensemble	199
7.2 Spécification de l'objet d'information	200
7.3 Code.....	201
7.4 Nom préféré.....	202
7.5 Numéro de révision.....	203
7.6 Nom abrégé	203
7.7 Nom synonyme	204
7.8 Numéro de version.....	204
8 Attributs_sémantiques	205
8.1 Vue d'ensemble	205
8.2 Spécification de l'objet d'information	205
8.3 Définition	205
8.4 Note.....	207
8.5 Remarque	207
8.6 Document source de définition	207
9 Attributs_administratifs	208

9.1	Vue d'ensemble	208
9.2	Spécification de l'objet d'information	208
9.3	Date obsolète	208
9.4	Proposé le	209
9.5	Publié dans	209
9.6	Publié par	209
9.7	Comité responsable	210
9.8	Révision publiée le	210
9.9	Niveau de statut	210
9.10	Version commencée le	211
9.11	Version publiée le	211
10	Attributs_de_valeur	212
10.1	Vue d'ensemble	212
10.2	Spécification de l'objet d'information	212
10.3	Autres unités de mesure	213
10.4	Type de données	214
10.4.1	Généralités	214
10.4.2	Type simple	215
10.4.3	Type énumération	221
10.4.4	Type d'instance de classe	222
10.4.5	Type référence de classe	222
10.4.6	Type agrégat	223
10.4.7	Type niveau	225
10.4.8	Type grand objet	226
10.4.9	Type placement	226
10.4.10	Dépendances en fonction du type de données	227
10.5	Nombre de chiffres significatifs	229
10.6	Identificateur de classe référencé	230
10.7	Unité de mesure	231
10.8	Format de valeur	231
11	Propriété_de_condition	233
11.1	Spécification de l'objet d'information	233
11.2	Type d'élément de données de propriété	234
12	Propriété_de_condition_dépendante	234
12.1	Spécification de l'objet d'information	234
12.2	Dépend de	235
12.3	Type d'élément de données de propriété	235
13	Propriété_dépendante	236
13.1	Spécification de l'objet d'information	236
13.2	Dépend de	236
13.3	Type d'élément de données de propriété	237
14	Propriété_non_dépendante	237
14.1	Spécification de l'objet d'information	237
14.2	Type d'élément de données de propriété	238
15	Information_de_traduction	238
15.1	Spécification de l'objet d'information	238
15.2	Date de la révision de traduction actuelle	238
15.3	Code de langue	239

15.4	Traducteur responsable	239
15.5	Traducteur responsable codé	239
15.6	Révision de traduction	240
16	Liste de valeurs	240
16.1	Spécification de l'objet d'information	240
16.2	Attributs internes de Value_list	242
16.3	Classe de définition	243
16.4	Liste d'articles énumérés	243
17	Terme	243
17.1	Spécification de l'objet d'information	243
17.2	Symbole littéral préféré dans texte	244
17.3	Attributs internes de Term	245
17.4	Classe de définition	245
18	Dessin	245
18.1	Modèle d'information	245
18.2	Spécification de l'objet d'information	246
18.3	Code	247
18.4	Indicateur descriptif	248
18.5	Titre du dessin	248
18.6	Format de fichier	249
18.7	Nom de fichier	249
18.8	Numéro de révision	249
18.9	Numéro de version	250
18.10	Attributs internes de Drawing	251
19	Unité de mesure	251
19.1	Vue d'ensemble	251
19.2	Spécification de l'objet d'information	252
19.3	Unité principale	253
19.4	Unité en texte	253
19.5	Unité XML	254
19.6	Attributs internes de Unit_of_measure	254
19.7	Référence de dessin	254
20	Création de variantes de langues	255
20.1	Vue d'ensemble	255
20.2	Attributs d'une Propriété dépendant de la langue	255
20.3	Attributs d'une Item_class dépendant de la langue	255
20.4	Attributs d'un Drawing dépendant de la langue	255
20.5	Attributs d'une Unit_of_measure dépendant de la langue	255
20.6	Attributs d'une Relation dépendant de la langue	256
21	Classe_d_article	256
21.1	Arbre de classification	256
21.2	Arbre de composition	259
21.3	Utilisation de schémas auxiliaires de codage et de classification des valeurs	260
21.4	Modèle d'information	261
21.5	Spécification de l'objet d'information	263
21.6	Type de classe	265
21.7	Nom codé	265
21.8	Attributs internes d'une Classe d'article	266

21.9	Type d'élément de données applicable	266
21.10	Association applicable	266
21.11	Type d'élément de données de classification	267
21.12	Référence de dessin	267
21.13	Est un cas de	268
21.14	Superclasse	268
22	Association	268
22.1	Vue d'ensemble	268
22.2	Spécification de l'objet d'information	270
22.3	Résolveur externe pour la formule	272
22.4	Formule	272
22.5	Langage d'interprétation de formule	273
22.6	Type d'association	273
22.7	Rôle de l'association	273
22.8	Attributs internes de Relation	275
22.9	Classe de définition	276
22.10	Co-domaine de fonction	276
22.11	Domaine de fonction	276
22.12	Domaine d'association	277
22.13	Référence de dessin	277
22.14	Super association	277
23	Elément_du_dictionnaire	278
24	Concepts avancés	279
24.1	Vue d'ensemble	279
24.2	Condition	279
24.3	Réutilisation des propriétés	284
24.4	Classes et propriétés pour utilisation commune	286
24.5	Bloc	289
24.6	Cardinalité	292
24.7	Polymorphisme	295
24.7.1	Vue d'ensemble	295
24.7.2	Choix polymorphes directement assignés à la Item_class spécifiée	296
24.7.3	Choix polymorphes assignés à la Item_class spécifiée par le biais d'une Value_list	298
24.8	Association	300
24.8.1	Vue d'ensemble	300
24.8.2	Enumération restreinte	300
24.8.3	Regroupement	302
25	Qualificatifs	305
26	Caractères et jeux de caractères	306
26.1	Vue d'ensemble	306
26.2	Jeux de caractères recommandés	306
26.3	Changement de ligne	306
26.4	Indice	306
26.5	Exposant	306
26.6	Caractères grecs	307
Annexe A (informative)	Modèle de données	309
Annexe B (normative)	Classification type de codes de propriétés	318

B.1	Classification des propriétés	318
B.1.1	Vue d'ensemble	318
B.1.2	Principes	318
B.1.3	Objets d'information Property de propriétés quantitatives	318
B.1.4	Objets d'information Property de propriétés non quantitatives.....	320
B.2	Relevé des codes de classification types pour les propriétés non quantitatives	320
B.3	Relevé des codes de classification types pour les propriétés quantitatives.....	321
Annexe C (informative)	Préparation des nouvelles classes et propriétés	331
C.1	Responsabilités	331
C.2	Éléments de jeux de données recommandés	331
Annexe D (informative)	Règles pour la définition des nouvelles versions et révisions des éléments du dictionnaire	332
D.1	Modifications des attributs des objets d'information Propriété	332
D.2	Modifications des attributs des objets d'information de classe.....	334
Annexe E (informative)	Attribut classifying_data_element_type	335
Annexe F (informative)	Conventions pour les noms et les définitions	336
F.1	Conventions pour l'écriture des définitions	336
F.1.1	Généralités	336
F.1.2	ISO/IEC 11179-4	336
F.1.3	ISO 704	336
F.1.4	Conventions supplémentaires	337
F.2	Conventions pour l'écriture des noms	337
F.2.1	Exigences.....	337
F.2.2	Recommandations	337
F.2.3	Noms des propriétés mécaniques quantitatives	337
F.2.4	Noms des propriétés électriques quantitatives	337
F.2.5	Noms des propriétés électriques non quantitatives	338
Annexe G (normative)	Spécification de format de valeur	339
G.1	Généralités	339
G.2	Notation	339
G.3	Types de formats de valeur de données.....	341
G.4	Méta-identificateur utilisé pour définir les formats	341
G.5	Formats de valeur quantitatifs	341
G.5.1	Généralités	341
G.5.2	Format de valeur NR1.....	341
G.5.3	Format de valeur NR2.....	342
G.5.4	Format de valeur NR3.....	343
G.5.5	Format de valeur NR4.....	343
G.6	Formats de valeurs non quantitatives	344
G.6.1	Généralités	344
G.6.2	Format de valeur alphabétique	345
G.6.3	Format de valeur de caractères mixtes	345
G.6.4	Format de valeur de nombre	346
G.6.5	Format de valeur de caractères mixtes alphabétiques ou numériques.....	346
G.6.6	Format de valeur binaire	346
G.7	Format HTML5.....	347
G.8	Exemples de valeurs.....	347
Annexe H (informative)	Notation de modélisation	349

H.1	Généralités	349
H.2	Classe UML	349
H.3	Généralisation	349
H.4	Association simple	350
H.5	Modularisation avec paquetage UML	350
	Bibliographie.....	353
	Figure 1 – Modèle simplifié de l'IEC 61360-1 (diagramme de classe UML).....	185
	Figure 2 – Modèle d'ensemble de la Caractéristique (diagramme de classe UML).....	193
	Figure 3 – Property (diagramme de classe UML)	195
	Figure 4 – Attributs d'identification (diagramme de classe UML)	201
	Figure 5 – Attributs sémantiques (diagramme de classe UML)	205
	Figure 6 – Attributs administratifs (diagramme de classe UML)	208
	Figure 7 – Attributs de valeur (diagramme de classe UML)	213
	Figure 8 – Exemples de données techniques associées à des lignes de connexion	216
	Figure 9 – Propriété condition (diagramme de classe UML)	234
	Figure 10 – Propriété condition dépendante (diagramme de classe UML)	235
	Figure 11 – Propriété dépendante (diagramme de classe UML)	236
	Figure 12 – Propriété non dépendante (diagramme de classe UML).....	237
	Figure 13 – Information de traduction (diagramme de classe UML)	238
	Figure 14 – Liste de valeurs (diagramme de classe UML)	242
	Figure 15 – Terme (diagramme de classe UML).....	244
	Figure 16 – Modèle d'ensemble du concept de dessin (diagramme de classe UML).....	246
	Figure 17 – Dessin (diagramme de classe UML)	247
	Figure 18 – Modèle d'ensemble du concept de mesure (diagramme de classe UML)	252
	Figure 19 – Unité de mesure (diagramme de classe UML)	253
	Figure 20 – Arbre de classification	257
	Figure 21 – Arbre de composition	260
	Figure 22 – Modèle d'ensemble de données de concept (diagramme de classe UML).....	262
	Figure 23 – Classe d'article (diagramme de classe UML).....	264
	Figure 24 – Modèle d'ensemble du concept d'association (diagramme de classe UML).....	269
	Figure 25 – Association (diagramme de classe UML).....	271
	Figure 26 – Élément de dictionnaire (diagramme de classe UML)	278
	Figure 27 – Point de fonctionnement d'un fusible	280
	Figure 28 – Mise en œuvre sous forme de tableau selon l'IEC 62656-1 de l'exemple de la Figure 27	281
	Figure 29 – (a) Installation de mesure du gain dynamique et du chiffre de bruit et (b) mesure avec une longueur d'onde de saturation de 1550 nm.....	282
	Figure 30 – Mise en œuvre sous forme de tableau selon l'IEC 62656-1 de l'exemple de la Figure 29	283
	Figure 31 – Réutilisation des propriétés (forme tableau selon l'IEC 62656-1).....	286
	Figure 32 – Utilisation d'objets d'information pour associer des informations de capacité avec tolérances à un condensateur fixe (tableau selon l'IEC 62656-1).....	289
	Figure 33 – Objets d'information d'interprétation de Classe et de Propriété formant un bloc	290

Figure 34 – Exemple de bloc (forme tableur selon l'IEC 62656-1)	292
Figure 35 – Objets d'information de Classe et de Propriété formant une cardinalité	293
Figure 36 – Exemple de bloc dont le nombre d'occurrences est limité par PAA506 (forme tableur selon l'IEC 62656-1)	295
Figure 37 – Objets d'information Interprétation de classe et Propriété formant un polymorphisme	296
Figure 38 – Choix polymorphes directement assignés à la Classe d'article spécifiée (forme tableur selon l'IEC 62656-1)	298
Figure 39 – Choix polymorphes assignés à la classe spécifiée par le biais d'une liste de valeurs (forme tableur selon l'IEC 62656-1)	300
Figure 40 – Exemple – Enumération restreinte (forme tableur selon l'IEC 62656-1)	302
Figure 41 – Exemple – Liste d'unités de mesure de la largeur d'un joint à labyrinthe	303
Figure 42 – Exemple – Objets d'information de Figure 41 (forme tableur selon l'IEC 62656-1).....	305
Figure A.1 – Caractéristique (diagramme de classe UML).....	312
Figure A.2 – Concept de dessin (diagramme de classe UML).....	312
Figure A.3 – Catégorie (diagramme de classe UML)	313
Figure A.4 – Concept de mesure (diagramme de classe UML)	314
Figure A.5 – Concept d'association (diagramme de classe UML)	316
Figure A.6 – Concept de liste de valeurs (diagramme de classe UML)	317
Figure H.1 – Exemple de généralisation.....	349
Figure H.2 – Exemple d'association	350
Figure H.3 – Exemple d'une association à une classe UML propriété d'un autre paquetage UML	351
Figure H.4 – Division du modèle de l'IEC 61360-1 en modules	352
Tableau 1 – Exemples de concepts génériques pour les grandeurs individuelles	182
Tableau 2 – Liste des attributs des objets d'information Property définis dans l'IEC 61360-1 et de leurs équivalents dans l'IEC 61360-2	190
Tableau 3 – Identification unique globale	200
Tableau 4 – Dépendances en fonction du type de données.....	229
Tableau 5 – Liste des attributs de Drawing	246
Tableau 6 – Liste des attributs Item_class définis dans l'IEC 61360-1 et de leurs équivalents dans l'IEC 61360-2.....	259
Tableau 7 – Translittération des caractères grecs en caractères latins.....	308
Tableau B.1 – Relevé des classes principales des objets d'information Property	319
Tableau B.2 – Codes de classification de types d'élément de données non quantitatifs	320
Tableau B.3 – Codes de classification pour les grandeurs de chimie physique et de physique moléculaire	321
Tableau B.4 – Codes de classification pour les grandeurs d'électricité et de magnétisme	323
Tableau B.5 – Codes de classification pour les grandeurs de phénomènes périodiques et relatifs	324
Tableau B.6 – Codes de classification pour les grandeurs acoustiques	324
Tableau B.7 – Codes de classification pour les grandeurs thermiques	325
Tableau B.8 – Codes de classification pour les grandeurs d'information	325

Tableau B.9 – Codes de classification pour les grandeurs mécaniques	326
Tableau B.10 – Codes de classification pour les grandeurs de lumière et les rayonnements électromagnétiques relatifs	327
Tableau B.11 – Codes de classification pour les quantités	327
Tableau B.12 – Codes de classification pour les prix et tarifs	328
Tableau B.13 – Codes de classification pour les grandeurs et comptes commerciaux sans dimensions	328
Tableau B.14 – Codes de classification pour les rapports et pourcentages commerciaux	328
Tableau B.15 – Codes de classification pour les grandeurs d'espace et de temps	329
Tableau B.16 – Codes de classification pour les grandeurs de réactions nucléaires et de rayonnements ionisants	330
Tableau D.1 – Vue d'ensemble de la gestion de la configuration des opérations de mises à jour de Property	333
Tableau D.2 – Vue d'ensemble de la gestion de la configuration des opérations de mises à jour de classe	334
Tableau F.1 – Exemple de structure de nom pour les propriétés électriques quantitatives	338
Tableau G.1 – Méta-langage syntaxique EBNF de l'ISO/IEC 14977	340
Tableau G.2 – Transposition des chiffres européens en chiffres arabes	346
Tableau G.3 – Exemples de valeurs de nombres	348

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

TYPES NORMALISÉS D'ÉLÉMENTS DE DONNÉES AVEC PLAN DE CLASSIFICATION –

Partie 1: Définitions – Principes et méthodes

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 61360-1 a été établie par le sous-comité 3D, Propriétés et classes de produits et leur identification, du comité d'études 3 de l'IEC: Structures d'informations, documentation et symboles graphiques.

Elle a le statut d'une norme horizontale conformément au Guide 108 de la CEI.

Cette quatrième édition annule et remplace la troisième édition parue en 2009. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- un support pour les constructions avancées telles que
 - conditions et contraintes,

- blocs,
 - cardinalité,
 - polymorphisme,
 - énumérations générales et restrictions, et
 - mapping;
- une liste de types de données étendue;
 - une harmonisation avec l'IEC 62656-1;
 - un support pour l'IEC TS 62720 et les unités codées;
 - une harmonisation des données sémantiques et administratives entre les divers objets d'information;
 - l'utilisation du langage UML pour la modélisation des données;
 - des descriptions et définitions étoffées;
 - l'introduction d'exemples de constructions de plus haut niveau comme le bloc, la cardinalité ou le polymorphisme, comme préconisations pour assister l'utilisateur de la série IEC 61360.

Le texte de cette Norme internationale est issu des documents suivants:

FDIS	Rapport de vote
3D/295/FDIS	3D/298/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 61360, publiées sous le titre général *Types normalisés d'éléments de données avec plan de classification*, peut être consultée sur le site web de l'IEC.

Les futures normes de cette série porteront dorénavant le nouveau titre général cité ci-dessus. Le titre des normes existant déjà dans cette série sera mis à jour lors de la prochaine édition.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives au document recherché. A cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La série IEC 61360 dans son entier spécifie un dictionnaire à usage général de termes techniques couvrant le domaine de l'électrotechnologie, de l'électronique, ainsi que d'autres domaines associés. Le dictionnaire est spécifié sous forme informatique en tant que dictionnaire de référence. Grâce au dictionnaire, les applications peuvent interagir et partager des données de manière non ambiguë et exempte d'incertitudes sémantiques.

Le présent document s'adresse aux ingénieurs et fournit une présentation détaillée de la structure du dictionnaire et de ses usages du point de vue du fournisseur du dictionnaire ainsi que du point de vue de l'utilisateur du dictionnaire. L'IEC 61360-2 spécifie le modèle de données détaillé du dictionnaire et l'IEC 61360-6 stipule les critères de qualité pour le contenu du dictionnaire.

La référence à un dictionnaire commun est avantageuse dans tous les cas où les informations sur les produits doivent être transférées de manière non ambiguë. Les cas d'usage comprennent des catalogues, des processus de classement, des informations de produits, contenus dans des spécifications ou des contrats.

La Commission Electrotechnique Internationale a défini un dictionnaire technique destiné à être utilisé dans les domaines électrotechnique et électronique, tenu à jour par le sous-comité 3D. Ce dictionnaire est appelé le Dictionnaire de données commun de l'IEC (IEC CDD, Common Data Dictionary) et peut être consulté sur la page web de l'IEC (<http://std.iec.ch/iec61360>).

Il convient de ne pas confondre les dictionnaires et les catalogues ou les collections de données maîtres; ceux-ci utilisent les objets du dictionnaire. Dans les catalogues ou les collections de données maîtres, les valeurs sont assignées à des instances d'objets du dictionnaire. Ils sont donc construits d'après les dictionnaires. Par conséquent, les dictionnaires sont normalement échangés avant tout catalogue ou collection de données maîtres.

L'IEC 61360-2 est étroitement associée au présent document. L'IEC 61360-2 contient le modèle d'information utilisant le langage de modélisation EXPRESS. Dans ce modèle, la définition et la structure de l'IEC 61360-1 sont formalisées et présentées sous une forme interprétable par ordinateur.

Le présent document est en grande partie conforme à l'IEC 61360-2:2012 et à l'ISO 13584-42:2010. Cependant, l'usage pratique a montré la nécessité d'étendre le modèle de données de façon sélective. La présente quatrième édition de l'IEC 61360-1 complète donc l'IEC 61360-2 en sémantique comme en syntaxe et introduit des constructions supplémentaires disponibles dans l'IEC 62656-1 pour des avantages pratiques. Par exemple, les constructions telles que les énumérations de nombres réels ou les objets d'association n'existent pas dans l'IEC 61360-2 mais sont nécessaires dans de nombreux domaines techniques.

L'IEC 62656-1 offre des interfaces avec l'IEC CDD. Les demandes de modifications de l'IEC CDD doivent donc être formées conformément à la spécification de l'interface disponible dans cette dernière norme.

Dans certains cas, il peut être difficile de savoir si les mots représentent les noms d'objets d'information ou s'ils traitent simplement d'articles ordinaires, comme pour le terme «propriété».

Pour cette raison, les règles typographiques suivantes sont utilisées dans le présent document:

- gras et première lettre majuscule: nom d'une classe UML, p. ex. «**Propriété**»;

- gras et première lettre minuscule: nom d'un attribut UML ou d'une association UML, p. ex. «**numéro_de_révision**»;
- police normale (non gras), première lettre minuscule: texte banal, p.ex. «propriété».

Il convient que les noms d'objets d'information UML soient traités comme des constantes et qu'ils ne soient pas traduits en d'autres langues.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

TYPES NORMALISÉS D'ÉLÉMENTS DE DONNÉES AVEC PLAN DE CLASSIFICATION –

Partie 1: Définitions – Principes et méthodes

1 Domaine d'application

La présente partie de l'IEC 61360 spécifie les principes de la définition des propriétés et des attributs associés et explique les méthodes de représentation de concepts définis oralement par des constructions de données appropriées disponibles dans l'IEC 61360-2. Il spécifie également les principes permettant d'établir une hiérarchie de classification à partir d'une collection de classes, dont chacune représente un concept technique d'un domaine électrotechnique ou un domaine relatif à l'électrotechnologie.

L'utilisation du présent document facilite l'échange des données techniques au moyen d'une structure définie pour l'information à échanger sous une forme interprétable par ordinateur. Chaque propriété à échanger a une signification définie non ambiguë et une dénomination cohérente, le cas échéant un domaine de valeurs défini, un format prescrit et des unités de mesure définies pour toutes les valeurs quantitatives. Des dispositions sont également données pour:

- a) le contrôle des modifications apportées aux définitions des propriétés par les numéros de version et de révision;
- b) l'introduction de notes et de remarques pour clarifier et aider à l'application des définitions;
- c) l'indication des sources des définitions et des listes de valeurs;
- d) les figures et les formules associées.

NOTE Les comités d'études et sous-comités de l'IEC ou d'autres organismes peuvent utiliser le présent document comme une base pour le développement de leurs propres dictionnaires.

Les sujets concernant les infrastructures des technologies de l'information, tels que les exemples qui suivent, ne font pas partie du domaine d'application du présent document:

- sécurité;
- mécanismes de blocage de bases de données;
- gestion des droits d'accès.

2 Références normatives

Les documents suivants cités dans le texte constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 62656-1:2014, *Enregistrement d'ontologie de produits normalisés et transfert par tableurs – Partie 1: Structure logique pour les paquets de données*

IEC TS 62720, *Identification des unités de mesure pour le traitement assisté par ordinateur*

IEC 80000 (toutes les parties), *Grandeurs et unités*

ISO/IEC 646, *Technologies de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'informations*

ISO/IEC 10646, *Technologies de l'information – Jeu universel de caractères codés (JUC)*

ISO/IEC 11179-3, *Technologies de l'information – Registres de métadonnées (RM) – Partie 3: Métamodèle de registre et attributs de base*

ISO/IEC 19505-1, *Technologies de l'information – Langage de modélisation unifié OMG (OMG UML) – Partie 1: Infrastructure*

ISO 639-1, *Codes pour la représentation des noms de langue – Partie 1: Code alpha-2*

ISO 2382 (toutes les parties), *Technologies de l'information – Vocabulaire*

ISO 3166 (toutes les parties), *Codes pour la représentation des noms de pays et de leurs subdivisions*

ISO 8601:2004, *Éléments de données et formats d'échange – Échange d'information – Représentation de la date et de l'heure*

ISO 10303-11, *Systèmes d'automatisation industrielle et intégration – Représentation et échange de données de produits – Partie 11: Méthodes de description: Manuel de référence du langage EXPRESS*

ISO 13584-26:2000, *Systèmes d'automatisation industrielle et intégration – Bibliothèque de composants – Partie 26: Ressource logique: Identification des fournisseurs d'information*

ISO 13584-42:2010, *Systèmes d'automatisation industrielle et intégration – Bibliothèque de composants – Partie 42: Méthodologie descriptive: Méthodologie appliquée à la structuration des familles de pièces*

ISO/TS 29002-5, *Systèmes d'automatisation industrielle et intégration – Échange de données caractéristiques – Partie 5: Schéma d'identification*

ISO 80000 (toutes les parties), *Grandeurs et unités*

IETF RFC 2045: N. Freed et N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* [en ligne], Novembre 1996 [vu le 01/07/2016]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc2045.txt>

IETF RFC 2046: N. Freed et N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types* [en ligne], Novembre 1996 [vu le 01/07/2016]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc2046.txt>

IETF RFC 2047: K. Moore, *MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text* [en ligne], Novembre 1996 [vu le 01/07/2016]. Disponible à l'adresse <http://www.ietf.org/rfc/rfc2047.txt>

3 Termes, définitions et abréviations

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <http://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <http://www.iso.org/obp>

3.1.1

propriété applicable

propriété nécessairement détenue par chaque partie membre d'une classe de caractérisation

Note 1 à l'article: Chaque partie membre d'une classe de caractérisation possède un aspect correspondant à chaque propriété applicable de cette classe de caractérisation.

Note 2 à l'article: La définition ci-dessus est conceptuelle, il n'existe aucune exigence consistant à recommander l'utilisation de toutes les propriétés applicables d'une classe pour décrire chaque partie de cette classe au niveau du modèle de données.

Note 3 à l'article: Toutes les propriétés applicables d'une superclasse sont également applicables aux sous-classes de ladite superclasse.

Note 4 à l'article: Seules les propriétés définies ou héritées comme visibles et les propriétés importées d'une classe peuvent être des propriétés applicables.

Note 5 à l'article: Pour faciliter l'intégration des bibliothèques de composants et des catalogues électroniques basés sur l'ISO 13584-24:2003 et l'ISO 13584-25, ces parties de l'ISO 13584 exigent que seules les propriétés qui sont applicables à une classe soient utilisées pour caractériser leurs instances dans les bibliothèques de composants et les catalogues électroniques.

[SOURCE: IEC 61360-2:2012, 3.2, modifiée]

3.1.2

attribut

élément de données permettant une description interprétable par ordinateur d'une propriété, d'une relation ou d'une classe

EXEMPLE Le nom d'une propriété, le code d'une classe, l'unité de mesure dans laquelle sont données les valeurs d'une propriété.

Note 1 à l'article: Un attribut ne décrit qu'un seul détail d'une propriété, d'une classe ou d'une relation.

[SOURCE: In English, ISO/IEC Guide 77-2:2008, 2.2]

3.1.3

cardinalité

modèle définissant le nombre de réapparitions d'un concept dans une description

[SOURCE: IEC 61987-10:2009, 3.1.5, modifiée – Dans la définition, "le nombre de fois où un concept réapparaît" est remplacé par "le nombre de réapparitions d'un concept". Notes 1 à 5 omises.]

3.1.4

caractéristique

fonctionnalité distinctive

Note 1 à l'article: Une caractéristique peut être inhérente ou assignée.

Note 2 à l'article: Une caractéristique peut être qualitative ou quantitative.

[SOURCE: In English, ISO 22274:2013, 3.3, modifiée]

3.1.5

classification

division systématique d'un ensemble d'articles en sous-ensembles en fonction de leur différence au niveau de certaines caractéristiques prédéterminées

3.1.6

classe

abstraction d'un ensemble de produits similaires

Note 1 à l'article: Un produit qui satisfait à l'abstraction définie par une classe est appelé membre de la classe.

Note 2 à l'article: Une classe est un concept volontaire pouvant prendre des significations d'extension différentes selon les contextes.

EXEMPLE L'ensemble de produits utilisés par une entreprise particulière et l'ensemble de tous les produits ISO normalisés sont deux exemples de concepts. Dans ces deux contextes (l'entreprise et l'ISO), l'ensemble des produits qui sont considérés comme membres de la classe *roulement à bille unique* peuvent être différents, en particulier du fait que les employés de chaque entreprise ignorent le nombre de produits roulements à bille unique existants.

Note 3 à l'article: Les classes sont structurées par relations d'inclusion de classe.

Note 4 à l'article: Une classe de produits est un concept général tel que défini dans l'ISO 1087-1. Il est donc conseillé d'utiliser les règles définies dans l'ISO 704 pour définir les attributs de désignation et de définition des classes de produits.

Note 5 à l'article: Dans le contexte de la série ISO 13584, une classe est soit une classe de caractérisation associée à des propriétés et utilisable pour caractériser les produits, soit une classe de catégorisation non associée à des propriétés et ne servant pas à caractériser des produits.

[SOURCE: IEC 61360-2:2012, 3.6, modifiée]

3.1.7

propriété de classification

propriété applicable à une classe particulière, comprenant une liste de valeurs dont les valeurs définissent les sous-classes de la classe

3.1.8

concept

unité de connaissance créée par une combinaison unique de caractéristiques

[SOURCE: In English, ISO 22274:2013, 3.7]

3.1.9

composant

produit industriel qui sert une ou des fonctions spécifiques, qui est destiné à être utilisé dans un produit assemblé de niveau supérieur

Note 1 à l'article: Dans son contexte, un composant est considéré comme ne pouvant pas être décomposé ni divisible physiquement. Dans un autre contexte, tel qu'un atelier d'entretien, un composant peut être décomposé.

3.1.10

forme interprétable par ordinateur

représentation spécifique d'informations permettant le traitement du contenu d'information au moyen d'un ordinateur électronique

3.1.11

dessin

illustration utilisée à des fins d'explication

3.1.12

entité

classe d'information définie par des propriétés communes

[SOURCE: In English, ISO 10303-11:2004, 3.3.6]

3.1.13

énumération

liste de constantes nommées appelées énumérateurs, chaque nom d'énumérateur de l'énumération étant non ambigu

3.1.14

caractéristique

aspect d'un article qui peut être capturé par une structure de classe et un ensemble de propriétés et qui ne peut pas exister indépendamment de l'article concerné

[SOURCE: In English, ISO 13584-24:2003, 3.41, modifiée]

3.1.15

objet d'information

partie d'information, définition ou spécification bien définie qui exige un nom afin d'identifier son utilisation dans la communication

3.1.16

article

objet pouvant être capturé par une structure de classe ou par une structure de propriété

Note 1 à l'article: Les articles sont considérés comme concernant les composants, les assemblages et jusqu'aux systèmes.

[SOURCE: IEC 62656-1:2014, 3.23, modifiée – Le terme "élément" est remplacé par "article". Dans la définition, "élément" est remplacé par "objet"]

3.1.17

valeur limite

dans une spécification d'un composant, dispositif, matériel ou système, la plus grande ou la plus petite valeur admissible d'une grandeur au-delà de laquelle il risque d'être endommagé, engendrant des modifications non désirées permanentes des caractéristiques fonctionnelles ou physiques et influençant son fonctionnement

3.1.18

valeur minimale

min

limite inférieure d'une plage de valeurs dans laquelle ladite valeur est significative

EXEMPLE Valeur la plus basse spécifiée d'une grandeur, établie pour un ensemble spécifié de conditions de fonctionnement auxquelles un composant, un dispositif, un matériel ou un système peut fonctionner et s'exécuter conformément aux exigences spécifiées.

Note 1 à l'article: Des informations supplémentaires sur la nature de la valeur peuvent être obtenues d'après la définition de l'objet d'information **Propriété** auquel appartient la valeur.

3.1.19

valeur maximale

max

limite supérieure d'une plage de valeurs dans laquelle ladite valeur est significative

EXEMPLE Valeur la plus élevée spécifiée d'une grandeur, établie pour un ensemble spécifié de conditions de fonctionnement auxquelles un composant, un dispositif, un matériel ou un système peut fonctionner et s'exécuter conformément aux exigences spécifiées.

Note 1 à l'article: Des informations supplémentaires sur la nature de la valeur peuvent être obtenues d'après la définition de l'objet d'information **Propriété** auquel appartient la valeur.

3.1.20**valeur nominale****nom**

valeur d'une grandeur utilisée pour dénommer et identifier un article par sa valeur et ne correspondant pas nécessairement à la valeur réelle de la propriété

Note 1 à l'article: Des informations supplémentaires sur la nature de la valeur peuvent être obtenues d'après la définition de l'objet d'information **Propriété** auquel appartient la valeur.

3.1.21**propriété non quantitative**

propriété qui identifie ou décrit un objet au moyen de codes, d'abréviations, de noms, de références ou de descriptions

EXEMPLE Un contenu d'information typique de propriétés non quantitatives est représenté par des articles tels que les codes, les abréviations, les noms, les références ou les descriptions.

[SOURCE: IEC 61360-2:2012, 3.28, modifiée – "type d'élément de données" est remplacé par "propriété".]

3.1.22**polymorphisme**

modèle qui permet de remplacer un concept simple dans le même contexte par un concept différent plus spécifique (spécialisé)

Note 1 à l'article: Un bloc polymorphe spécialisé peut remplacer un bloc plus général dans le même contexte.

Note 2 à l'article: Un opérateur polymorphe (propriété de contrôle) peut servir à effectuer une sélection entre plusieurs spécialisations.

[SOURCE: IEC 61987-10:2009, 3.1.21, modifiée]

3.1.23**produit**

résultat d'un travail ou d'un processus naturel ou industriel

3.1.24**propriété****type d'élément de données**

paramètre défini adapté à la description et la différenciation d'objets

Note 1 à l'article: Une propriété décrit une caractéristique d'un objet donné.

Note 2 à l'article: Une propriété peut disposer d'attributs tels qu'un code, une version et une révision.

Note 3 à l'article: La spécification d'une propriété peut comprendre des choix de valeurs prédéfinis.

[SOURCE: In English, ISO/IEC Guide 77-2:2008, 2.18, modifiée]

3.1.25**propriété quantitative**

propriété dotée d'une valeur numérique représentant une grandeur physique, une somme d'informations ou un total d'objets

[SOURCE: IEC 61360-2:2012, 3.40, modifiée – "type d'élément de données" est remplacé par "propriété".]

3.1.26**grandeur**

propriété d'un phénomène, d'un corps ou d'une substance que l'on peut exprimer quantitativement au moyen d'un nombre et d'une référence

Note 1 à l'article: Le concept générique de «grandeur» peut être subdivisé en plusieurs niveaux de concepts spécifiques, comme indiqué dans le Tableau 1. La moitié gauche du tableau présente des concepts spécifiques du concept de «grandeur». Ce sont des concepts génériques pour les grandeurs individuelles de la moitié droite.

Tableau 1 – Exemples de concepts génériques pour les grandeurs individuelles

Concepts génériques pour grandeurs individuelles		Grandeurs individuelles
longueur, l	rayon, r	rayon du cercle A, r_A ou $r(A)$
	longueur d'onde, λ	longueur d'onde de la radiation D du sodium, λ_D ou $\lambda(D; Na)$
énergie, E	énergie cinétique, T	énergie cinétique de la particule i dans un système donné, T_i
	chaleur, Q	chaleur de la vaporisation de l'échantillon i d'eau, Q_i
charge électrique, Q		charge électrique du proton, e
résistance électrique, R		résistance électrique de la résistance i dans un circuit donné, R_i
concentration -en- quantité de matière du constituant B, c_B		concentration -en- quantité de matière d'éthanol dans le spécimen de vin i , $c_i(C_2H_5OH)$
nombre volumique du constituant B, C_B		nombre volumique d'érythrocytes dans le spécimen de sang i , $C(Erys; B_i)$
Dureté C Rockwell (charge de 150 kg), HRC(150 kg)		Dureté C Rockwell du spécimen d'acier i , HRC $_i$ (150 kg)

Note 2 à l'article: La référence peut être une unité de mesure, une procédure de mesure, un matériau de référence ou leurs combinaisons.

Note 3 à l'article: Les séries de normes IEC 80000 et ISO 80000 donnent les symboles de grandeurs. Les symboles de grandeurs sont écrits en italique. Un symbole donné peut indiquer des grandeurs différentes.

Note 4 à l'article: Une grandeur telle que définie ici est une grandeur scalaire. Cependant, un vecteur ou un tenseur, dont les composantes sont des grandeurs, est aussi considéré comme une grandeur.

Note 5 à l'article: Le concept de «grandeur» peut être subdivisé génériquement, par exemple, «grandeur physique», «grandeur chimique» et «grandeur biologique», ou «grandeur de base» et «grandeur dérivée».

Note 6 à l'article: Adapté du Guide ISO/IEC 99:2007, définition 1.1, où se trouve une note supplémentaire.

[SOURCE: ISO 80000-1:2009, 3.1]

3.1.27

association

aspect ou qualité qui relie deux choses ou parties ou plus comme étant liées ou appartenant l'une à l'autre

3.1.28

sous-classe

classe inférieure d'un niveau à une autre classe dans une hiérarchie d'inclusion de classe

Note 1 à l'article: Dans le modèle de dictionnaire commun de l'ISO 13584/IEC 61360, les hiérarchies d'inclusion de classe sont définies par la relation est-un. Elles peuvent également être établies par les relations est un cas-de.

[SOURCE: IEC 61360-2:2012, 3.43]

3.1.29

superclasse

classe supérieure d'un niveau à une autre classe dans une hiérarchie d'inclusion de classe

Note 1 à l'article: Dans le modèle de dictionnaire commun de l'ISO 13584/IEC 61360, les hiérarchies d'inclusion de classe sont définies par la relation est-un. Elles peuvent également être établies par les relations est un cas-de.

Note 2 à l'article: Dans le modèle de dictionnaire commun de l'ISO 13584/IEC 61360, une classe a au plus une superclasse spécifiée par une relation est-un.

[SOURCE: IEC 61360-2:2012, 3.44]

3.1.30

valeur type

typ

valeur d'une grandeur utilisée comme valeur représentative ou valeur communément observable

EXEMPLE Valeur type spécifiée d'une grandeur, établie pour un ensemble spécifié de conditions de fonctionnement auxquelles un article peut fonctionner et s'exécuter conformément aux exigences spécifiées.

Note 1 à l'article: Des informations supplémentaires sur la nature de la valeur peuvent être obtenues d'après la définition de l'objet d'information **Propriété** auquel appartient la valeur.

3.1.31

propriété visible

propriété qui possède une définition significative au sein d'un domaine d'application d'une classe de caractéristique donnée, mais qui ne s'applique pas nécessairement aux différents produits appartenant à cette classe

Note 1 à l'article: Les termes «significative au sein d'un domaine d'application d'une classe de caractéristique donnée» signifient qu'un observateur humain peut déterminer, pour tout produit de la classe de caractérisation, si la propriété s'applique et si oui, à quel aspect du produit elle correspond.

Note 2 à l'article: Le concept d'une propriété visible permet de partager la définition d'une propriété parmi des classes de caractérisation où cette propriété ne s'applique pas nécessairement.

EXEMPLE La propriété de longueur non filetée est significative pour toute classe de vis mais elle s'applique uniquement aux vis qui possèdent une partie non filetée. Elle peut être définie comme visible au niveau de la vis tout en étant applicable uniquement dans certaines sous-classes.

Note 3 à l'article: Toutes les propriétés visibles d'une superclasse qui est une classe de caractérisation de produit sont également des propriétés visibles de ses sous-classes.

Note 4 à l'article: Pour faciliter l'intégration des bibliothèques de composants et des catalogues électroniques basés sur l'ISO 13584-24:2003 et l'ISO 13584-25, ces parties de l'ISO 13584 exigent que seules les propriétés qui sont applicables à une classe soient utilisées pour caractériser leurs instances dans les bibliothèques de composants et les catalogues électroniques.

Note 5 à l'article: Cette définition d'une propriété visible remplace la définition précédente de l'ISO 13584-24:2003 qui était la suivante: «une propriété qui est définie pour une certaine classe de produits et qui ne s'applique pas nécessairement aux différents produits de ladite classe de produits».

[SOURCE: IEC 61360-2:2012, 3.46, modifiée]

3.2 Abréviations

Abréviation	Français	Anglais
DI	Identificateur de données	Data Identifier
HTML	Langage de balisage hypertexte	Hypertext Mark-up Language
ICID	Identificateur de concept international	International Concept Identifier
IEC CDD	Dictionnaire de données commun de l'IEC	IEC Common Data Dictionary
IRDI	Identificateur international de données d'enregistrement	International Registration Data Identifier
RAI	Identificateur d'autorité d'enregistrement	Registration Authority Identifier
UML	Langage de modélisation unifié	Unified Modelling Language
URI	Identificateur universel	Uniform Resource Identifier
URL	Adresse universelle	Uniform Resource Locator

4 Bases des concepts des dictionnaires normalisés de l'IEC 61360

Dans la série IEC 61360, les concepts de base sous-jacents au modèle complet de données sont ceux de «dictionnaire», «classe» et «propriété».

Le concept d'un produit est capturé par des entités appelées «classes» et ses caractéristiques sont capturées par des entités appelées «propriétés». Une hiérarchie de classes du présent document est un ordre de classes conforme aux aspects communs des propriétés des classes.

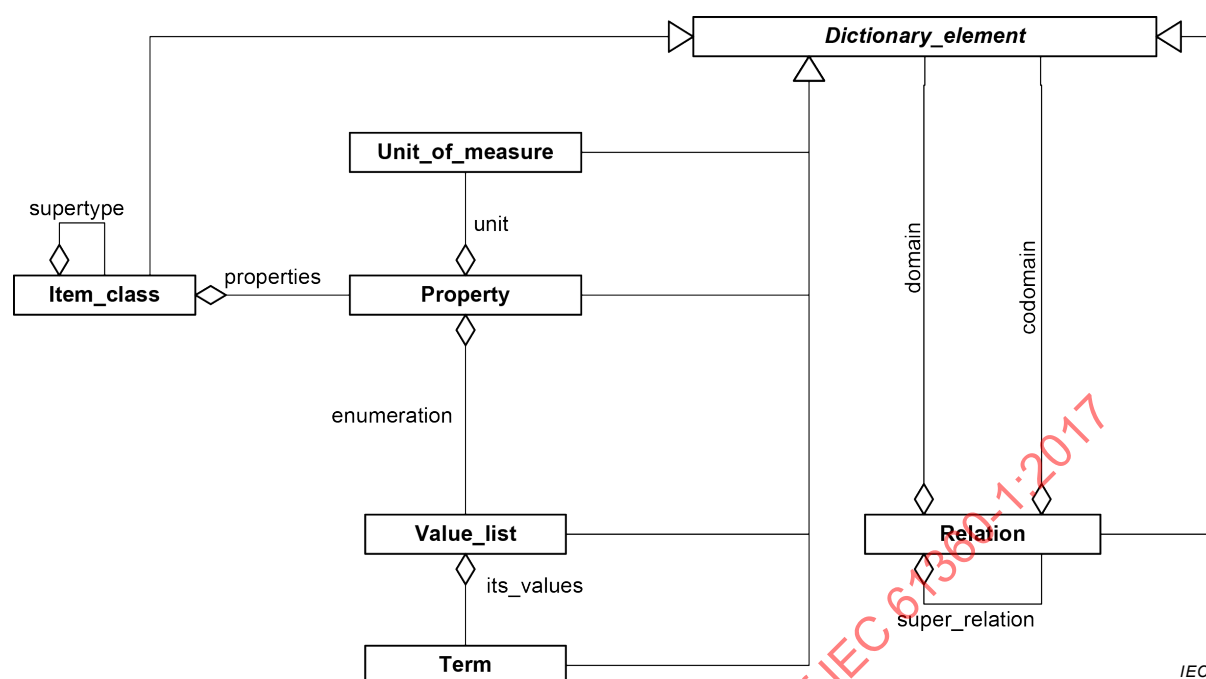
Les articles qui suivent donnent des détails sur ces concepts et leurs relations.

Afin de représenter les concepts, des diagrammes de classe en Langage de modélisation unifié (UML, Unified Modelling Language, voir l'ISO/IEC 19505-1), sont utilisés aux Articles 6 à 23. L'Article 24 utilise une présentation en format tableur introduite par l'IEC 62656-1:2014 pour la représentation des structures contenant des instances.

NOTE 1 Les modèles UML existent à des fins d'explication, un modèle de données détaillé pour la mise en œuvre est donné dans l'IEC 61360-2.

La Figure 1 représente un modèle général simplifié des objets d'information de l'IEC 61360-1 et de leurs relations.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017



IEC

Anglais	Français
<i>Dictionary_element</i>	<i>Elément du dictionnaire</i>
Unit_of_measure	Unité de mesure
Item_class	Classe d'article
Property	Propriété
Value_list	Liste de valeurs
Term	Terme
Relation	Association
supertype	supertype
properties	propriétés
unit	unité
enumeration	énumération
domain	domaine
codomain	co-domaine
super_relation	super_association
its_values	ses valeurs

Figure 1 – Modèle simplifié de l'IEC 61360-1 (diagramme de classe UML)

NOTE 2 L'énumération des associations de la Figure 1 est mise en œuvre par le biais des types de données d'énumération.

Le modèle de dictionnaire de l'IEC 61360 est un modèle à héritage simple. Il arrive toutefois que certaines propriétés définies dans une autre classe qui réside dans d'autres branches de la hiérarchie de spécialisation deviennent nécessaires et pertinentes.

Dans ce cas, le modèle propose le mécanisme est un cas-de pour importer les propriétés d'une version fixe (voir 24.3). Une fois importées dans une classe, ces propriétés deviennent applicables dans la classe et héritées dans ses sous-classes en tant que propriétés applicables. L'importation de propriétés est réalisée en énumérant les propriétés à importer dans l'attribut «propriétés_importées» de la classe cas-de.

A noter que lorsqu'une relation cas-de est installée, la version de la propriété importée est fixe et même si la définition de la propriété est modifiée dans la classe source, la propriété importée n'est pas modifiée.

Voir les diagrammes de classe UML détaillés de l'IEC 61360-1 à l'Annexe A. L'Annexe H présente brièvement les constructions UML utilisées.

L'Annexe C donne des conseils sur la préparation de nouvelles classes et propriétés.

5 Identification du dictionnaire

5.1 Généralités

Le Dictionnaire de données commun de l'IEC (IEC CDD, IEC Common Data Dictionary) contient une collection de classes et de propriétés. Tous les objets d'information d'un dictionnaire sont identifiés de manière unique au moyen d'un identificateur de version en tant que dossier des modifications significatives d'un objet du dictionnaire. Cela permet un contrôle des versions pour chaque objet du dictionnaire. C'est la raison pour laquelle l'identification d'un objet du dictionnaire est indépendante des publications successives du dictionnaire dans son entier.

Afin de pouvoir communiquer le contenu du dictionnaire, il peut être utile d'identifier par une référence le jeu complet d'articles du dictionnaire. C'est pour cette raison que l'IEC CDD dans son entier est identifié par les attributs suivants donnés en 5.2 à 5.6.

5.2 Fournisseur du dictionnaire

Nom de l'attribut: **dictionary_supplier**

Définition de l'attribut: identification de tout organisme autorisé à enregistrer les articles du dictionnaire

Description: L'identificateur est spécifié dans l'ISO 13584-26:2000. L'identification de la source pour le dictionnaire de référence de l'IEC 61360-4 est la suivante: «0112/2///61360_4».

D'autres sources peuvent exiger des identifications différentes.

NOTE 1 L'identification de la source de la norme n'inclut pas de numéro d'édition, contrairement à ce qui est défini par l'ISO 13584-26:2000.

NOTE 2 Le trait d'union «-» de la norme 61360-4 est remplacé par un tiret inférieur «_».

EXEMPLE «0112/2///61987» est l'identification source pour les données de la série IEC 61987.

Obligation: obligatoire

Type de caractère des valeurs: Constante chaîne comme spécifié

NOTE 3 Le séparateur barre oblique «/» est utilisé dans un fichier physique conformément à l'ISO 10303-21. Dans d'autres environnements comme le XML, d'autres séparateurs peuvent être utilisés.

5.3 Code

Nom de l'attribut:	code
Définition de l'attribut:	identificateur unique de l'article du dictionnaire
Description:	Le contenu du code pour le Dictionnaire de données commun de l'IEC doit être le suivant: «IEC CDD»
Obligation:	obligatoire
Type de caractère des valeurs:	Constante chaîne comme spécifié

5.4 Numéro de version

Nom de l'attribut:	version_number
Définition de l'attribut:	numéro utilisé pour contrôler les versions du Dictionnaire de données commun de l'IEC
Description:	Il convient que le version_number du dictionnaire soit composé de 6 chiffres. Bien que le modèle d'information commun EXPRESS ISO/IEC autorise un nombre plus élevé de chiffres, pour que le dictionnaire soit enregistré dans l'IEC CDD ou soit compatible avec la série IEC 61360, il convient de n'utiliser que 6 chiffres. Il est admis d'utiliser un plus grand nombre de chiffres à des fins spécifiques. Les numéros des versions consécutives commençant par «000001» doivent être donnés en suivant l'ordre croissant. Une nouvelle version du dictionnaire peut être générée si au moins une classe ou une propriété est ajoutée ou lorsque des modifications affectant le numéro de version sont apportées. Lorsqu'une nouvelle version du dictionnaire est publiée, cet ensemble de données contient toutes les modifications apparues depuis la dernière publication. NOTE Des modifications intermédiaires du dictionnaire peuvent être déterminées en vérifiant l'attribut date_of_current_version.
Obligation:	obligatoire
Type de caractère des valeurs:	Chiffres de «0» à «9».

5.5 Date de la version actuelle

Nom de l'attribut:	date_of_current_version
Définition de l'attribut:	date associée au contenu actuel du Dictionnaire de données commun de l'IEC
Description:	L'information de date du dictionnaire doit être constituée de 10 chiffres et doit respecter le format «AAAA-MM-JJ» comme spécifié dans l'ISO 8601:2004. A chaque fois que le contenu du dictionnaire est modifié, la valeur de l'attribut doit être mise à jour. EXEMPLE «1994-03-21».
Obligation:	obligatoire
Type de caractère des valeurs:	Chiffres de «0» à «9» et «-».

5.6 Numéro de révision

Nom de l'attribut:	revision_number
Définition de l'attribut:	numéro utilisé pour le contrôle administratif du dictionnaire de données commun (IEC CDD)
Description:	Il convient que le revision_number du dictionnaire soit composé de 3 chiffres. Bien que le modèle d'information commun EXPRESS ISO/IEC autorise un nombre plus élevé de chiffres, pour que le dictionnaire soit enregistré dans l'IEC CDD qu'il soit compatible avec la série IEC 61360, il convient de n'utiliser que 3 chiffres. Il est admis d'utiliser un plus grand nombre de chiffres à des fins spécifiques. Les numéros des révisions consécutives commençant par «000» doivent être donnés dans l'ordre croissant. Un seul revision_number du dictionnaire, unique par son identificateur, est valable à un moment donné. Un nouveau revision_number du dictionnaire doit être généré si un attribut d'un objet du dictionnaire est modifié sans affecter l'utilisation ni la signification des objets du dictionnaire. EXEMPLE Modifications éditoriales telles que la correction d'erreurs de frappe. Le contenu de revision_number doit être repris à la valeur de départ «1» lorsque le version_number est incrémenté.
Obligation:	obligatoire
Type de caractère des valeurs:	Chiffres de «0» à «9».

6 Propriété

6.1 Vue d'ensemble

Une propriété est une caractéristique d'un objet tel qu'un composant (voir 3.1.24, Note 1). Une propriété, lorsqu'elle est utilisée, a toujours une valeur associée, comme une valeur numérique et son unité, ou une valeur qualitative tirée d'une liste prédéfinie.

Dans l'Article 6, les différents attributs des propriétés rencontrées dans les spécifications sont explicités. Voir le Tableau 2 pour la liste de ces attributs. Ces attributs sont relatifs à l'identification, la description, la valeur des caractéristiques et à ses relations.

Lors de la spécification des propriétés et de leur type de données associé, il convient de s'assurer que la valeur anticipée est significative. En particulier, l'utilisation de valeurs `BOOLEAN_TYPE` peut mener à des spécifications défectueuses si la signification des valeurs «TRUE (vrai)» et «FALSE (faux)» n'est pas spécifiée correctement. Une liste de valeurs donne souvent des informations plus précises que des valeurs booléennes et il convient d'utiliser celle-ci de préférence.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

Tableau 2 – Liste des attributs des objets d'information Property définis dans l'IEC 61360-1 et de leurs équivalents dans l'IEC 61360-2

Noms d'attributs dans l'IEC 61360-1	Objets d'information dans l'IEC 61360-2	Numéro de paragraphe
code	code	7.3
code (de valeur)	value_code	7.3
code (pour liste de valeurs)	value_code	7.3
code_of_alternative_unit	a	6.9
code_of_unit	a	6.11
code_of_list_of_units	a	6.10
condition_property	condition_DET	11.2
data_element_type_class	det_classification	6.3
data_type	data_type	10.4
date_of_current_translation_revision	a	15.2
definition	definition	8.3
definition (de valeur)	preferred_name of meaning	8.3
definition (pour liste de valeurs)	preferred_name of meaning	8.3
dependent_condition_property	a	12
dependent_property	dependent_P_DET	13
drawing_reference	figure	6.13
formula	formula	6.5
language_code	a	15.3
non_dependent_property	non_dependent_P_DET	14
note	note	8.4
obsolete_date	is_deprecated/ date_of_current_version	9.3
preferred_letter_symbol	preferred_symbol	6.6
preferred_name	preferred_name	7.4
proposal_date	a	9.4
published_by	a	9.6
published_in	a	9.5
referenced_class_identifier	domain of class_instance_type	10.6
remark	remark	8.5
responsible_committee	a	9.7
responsible_translator	a	15.4
responsible_translator_coded	a	15.5
revision_number	revision	7.5
revision_released_on	date_of_current_revision	9.8
short_name	short_name	7.6
source_document_of_definition	source_doc_of_definition	8.6
source_document_of_definition (pour liste de valeurs)	source_doc_of_value_domain	8.6
status_level	a	9.9
synonymous_letter_symbol	synonymous_symbol	6.7
synonymous_name	synonymous_name	7.7
translation_revision	a	15.6
unit_of_measure	unit	10.7
value_format	value_format	10.8

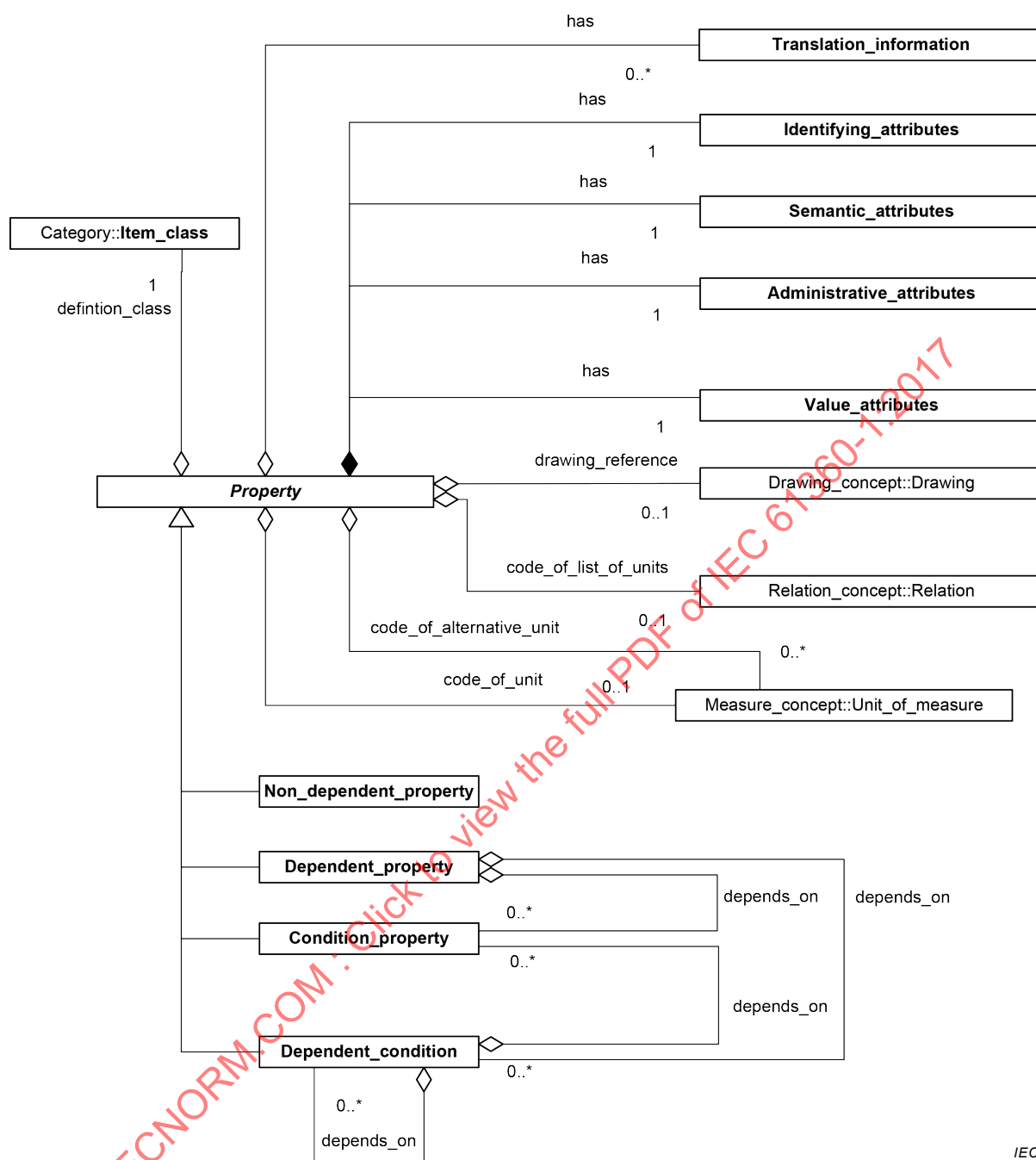
Noms d'attributs dans l'IEC 61360-1	Objets d'information dans l'IEC 61360-2	Numéro de paragraphe
value_list	value_domain	16
version_initiated_on	date_of_original_definition	9.10
version_number	version	7.8
version_released_on	date_of_current_version	9.11
^a Non défini dans l'IEC 61360-2:2012.		

A partir des principes décrits dans l'ISO/IEC 11179-3, les attributs sont divisés en quatre groupes principaux:

- attributs d'identification,
- attributs sémantiques,
- attributs de valeur,
- attributs de relation.

La Figure 2 représente le modèle d'ensemble du module «Caractéristique» utilisant le langage UML comme technique de présentation.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017



IEC

Anglais	Français
Category::Item_class	Catégorie::Classe_d_article
definition_class	classe_de_définition
Property	Propriété
Non_dependent_property	Propriété_non_dépendante
Dependent_property	Propriété_dépendante
Condition_property	Propriété_de_condition
Dependent_condition	Condition_dépendante
Translation_information	Information_de_traduction
Identifying_attributes	Attributs_d'identification
Semantic_attributes	Attributs_sémantiques

Anglais	Français
Administrative_attributes	Attributs_administratifs
Value_attributes	Attributs_de_valeur
Drawing_concept::Drawing	Concept_de_dessin::Dessin
Relation_concept::Relation	Concept_d_association::Association
Measure_concept::Unit_of_measure	Concept_de_mesure::Unité_de_mesure
has	possède
drawing_reference	référence_du_dessin
code_of_list_of_units	code_de_liste_d_unités
code_of_alternative_units	code_d_autres_unités
depends_on	dépend_de
code_of_unit	code_de_l_unité

**Figure 2 – Modèle d'ensemble de la Caractéristique
(diagramme de classe UML)**

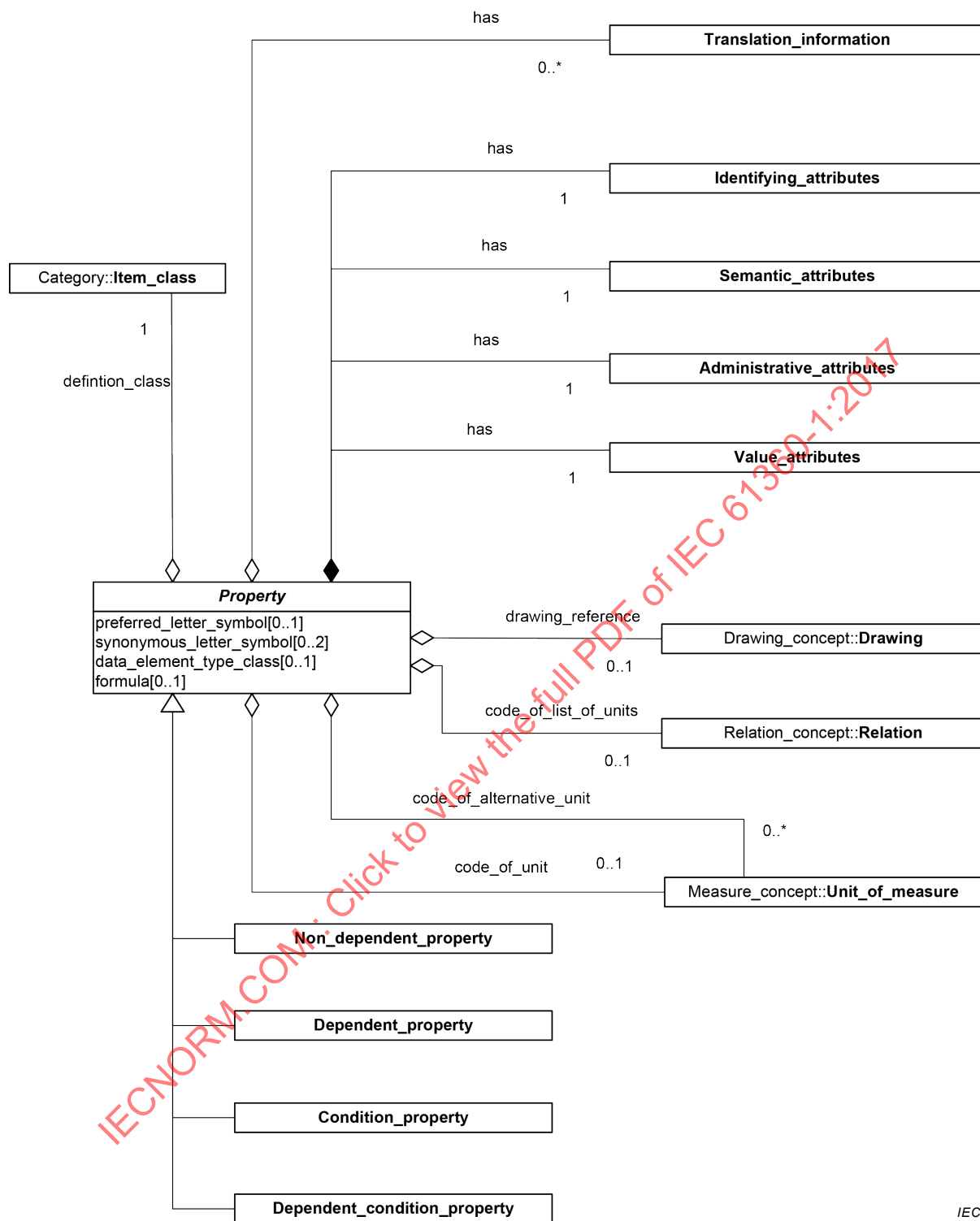
6.2 Spécification de l'objet d'information

Nom: **Property**

Définition: objet d'information qui caractérise la ou les **Item_class** à laquelle ou auxquelles il est associé

Description:

La Figure 3 représente le modèle détaillé de l'objet d'information «**Property**» utilisant le langage UML comme technique de présentation.



IEC

Anglais	Français
Category::Item_class	Catégorie::Classe_d_article
definition_class	classe_de_définition
Property	Propriété
preferred_letter_symbol(0...1)	symbole_littéral_préféré(0...1)
synonymous_letter_symbol(0...2)	symbole_littéral_synonyme(0...2)
data_element_type_class(0...1)	classe_de_type_d_élément_de_données(0...1)
formula(0...1)	formule(0...1)

Anglais	Français
Non_dependent_property	Propriété_non_dépendante
Dependent_property	Propriété_dépendante
Condition_property	Propriété_de_condition
Dependent_condition_property	Propriété_de_condition_dépendante
Translation_information	Information_de_traduction
Identifying_attributes	Attributs_d'identification
Semantic_attributes	Attributs_sémantiques
Administrative_attributes	Attributs_administratifs
Value_attributes	Attributs_de_valeur
Drawing_concept::Drawing	Concept_de_dessin::Dessin
Relation_concept::Relation	Concept_d_association::Association
Measure_concept::Unit_of_measure	Concept_de_mesure::Unité_de_mesure
has	possède
drawing_reference	référence_du_dessin
code_of_list_of_units	code_de_liste_d_unités
code_of_alternative_units	code_d'autres_unités
code_of_unit	code_de_l'unité

Figure 3 – Property (diagramme de classe UML)

6.3 Classe de type d'élément de données

Nom de l'attribut: **data_element_type_class**

Définition de l'attribut: classification de types de propriété similaires

Description: La classification de la propriété est décrite en détail en Annexe B.

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

6.4 Dépend de

Nom de l'attribut: **depends_on**

Définition de l'attribut: ensemble d'un ou plusieurs codes d'objets d'information de **Property** (voir 7.3) jouant le rôle des conditions desquelles la **Property** en question dépend

Description: Voir également 24.2.

Obligation: conditionnel

Condition: doit avoir une valeur si **data_element_type_class** a l'une des valeurs suivantes

DEPENDENT_P_DET

DEPENDENT_C_DET

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

6.5 Formule

Nom de l'attribut: **formula**

Définition de l'attribut: règle ou énoncé sous forme mathématique exprimant la sémantique d'une **Property**

Description: Le contenu de **formula** ne doit modifier aucune information essentielle de la signification de la définition. La représentation graphique d'une formule doit être mémorisée dans un fichier sous un format d'usage courant.

NOTE 1 MathML (voir la Recommandation du W3C 10:2014) est le langage recommandé pour la présentation des formules mathématiques.

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

6.6 Symbole littéral préféré

Nom de l'attribut: **preferred_letter_symbol**

Définition de l'attribut: marque ou caractères utilisés comme signe pour exprimer un objet d'information

EXEMPLE 1 Les caractères représentant les éléments chimiques dans la classification périodique (Ag = argent) ou ceux représentant des concepts électriques (f = fréquence, T_{amb} = température ambiante).

Description: Le contenu de **preferred_letter_symbol** d'une **Property** doit être identique au symbole littéral des normes associées.

EXAMPLE 2 Série IEC 60027, série IEC 60747, série IEC 60748 ou séries IEC 80000 et ISO 80000.

Il convient de consulter les normes relatives à l'article concerné pour trouver des symboles littéraux supplémentaires. Il convient que le premier caractère du symbole littéral préféré d'une **Condition_property** soit un «@»

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

6.7 Symbole littéral synonyme

Nom de l'attribut:	synonymous_letter_symbol
Définition de l'attribut:	marque alternée ou chaîne de caractères utilisée comme signe pour décrire un objet d'information qui diffère du contenu du preferred_letter_symbol mais qui représente le même concept
Description:	Le recours à des symboles littéraux synonymes est quelquefois inévitable pour des raisons historiques. Le nombre de symboles littéraux synonymes doit être limité à deux. Il convient que le premier caractère du symbole littéral synonyme d'une Condition soit un «@».
Obligation:	facultatif
Type de caractère des valeurs:	Voir l'Article 26.

6.8 Attributs internes d'une Propriété

Une **Property** possède des attributs qui sont regroupés pour une meilleure lisibilité dans les objets d'information

- **Identifying_attributes** (Attributs d'identification);
- **Semantic_attributes** (Attributs sémantiques);
- **Administrative_attributes** (Attributs administratifs);
- **Translation_information** (Information de traduction);
- **Value_attributes** (Attributs de valeur).

Ces objets d'information sont relatifs à la **Property** par la relation «possède». Les objets d'information doivent être conservés avec la classe de référence, c.-à-d. qu'ils sont internes aux instances de la **Property**.

6.9 Code d'autre unité

Nom de l'attribut:	code_of_alternative_unit
Définition de l'attribut:	un ou plusieurs identificateurs de classe des objets d'information Unit_of_measure pouvant être utilisés à la place de l'unité spécifiée par code_of_unit
Description:	L'unité de mesure doit être spécifiée au moyen des codes spécifiés dans l'IEC TS 62720 ou dans les dictionnaires d'unités conformes.

NOTE 1 Les dictionnaires peuvent être fournis sous forme interprétable par ordinateur ou en format papier.

NOTE 2 L'identificateur d'unité utilisé dans l'attribut **code_of_alternative_unit** peut se restreindre à une unité fournie par un serveur pour des unités de mesure.

Obligation: facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

NOTE 3 En principe, le contenu de **code_of_alternative_unit** présente la forme RAI#DI##VI, où RAI est l'identificateur de l'autorité d'enregistrement, DI est l'identificateur de données et VI est l'identificateur de la version, comme décrit dans l'ISO/IEC 11179-5:2009.

6.10 Code de liste d'unités

Nom de l'attribut: **code_of_list_of_units**

Définition de l'attribut: identificateur de classe spécifiant une **List_of_units** offrant une collection d'unités de mesure possibles

Description: L'unité de mesure peut être spécifiée au moyen des codes spécifiés dans l'IEC TS 62720 ou dans les dictionnaires d'unités conformes.

NOTE 1 Les dictionnaires peuvent être fournis sous forme interprétable par ordinateur ou en format papier.

NOTE 2 L'identificateur d'unité utilisé dans l'attribut **code_of_list_of_units** peut se restreindre à une unité fournie par un serveur pour des unités de mesure.

Obligation: facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

NOTE 3 En principe, le contenu de **code_of_list_of_units** présente la forme RAI#DI##VI, où RAI est l'identificateur de l'autorité d'enregistrement, DI est l'identificateur de données et VI est l'identificateur de la version, comme décrit dans l'ISO/IEC 11179-5:2009.

6.11 Code de l'unité

Nom de l'attribut: **code_of_unit**

Définition de l'attribut: identificateur de classe de l'unité de mesure préférée

Description: L'unité de mesure peut être spécifiée au moyen des codes spécifiés dans l'IEC TS 62720 ou dans les dictionnaires d'unités conformes.

NOTE 1 Les dictionnaires peuvent être fournis sous forme interprétable par ordinateur ou en format papier.

NOTE 2 L'identificateur d'unité utilisé dans l'attribut **code_of_unit** peut se restreindre à une unité fournie par un serveur pour des unités de mesure.

Obligation: facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

NOTE 3 En principe, le contenu de **code_of_unit** présente la forme RAI#DI##VI, où RAI est l'identificateur de l'autorité d'enregistrement, DI est l'identificateur de données et VI est l'identificateur de la version, comme décrit dans l'ISO/IEC 11179-5:2009.

6.12 Classe de définition

Nom de l'attribut: **definition_class**

Définition de l'attribut: classe dans laquelle l'objet d'information est défini

Description: La **class** référencée par la **definition_class** spécifie le domaine d'application. En dehors de ce domaine d'application, les instances ne sont pas valides.

Obligation: obligatoire

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

6.13 Référence de dessin

Nom de l'attribut: **drawing_reference**

Définition de l'attribut: référence à un **Drawing** illustrant la signification d'une **Property**

Obligation: facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

7 Attributs d'identification

7.1 Vue d'ensemble

L'objet d'information **Identifying_attributes** collecte les attributs qui spécifient l'identité d'un objet d'information.

Pour identifier un objet d'information de manière unique, une combinaison de caractères ne dépendant pas de la langue doit être utilisée. L'identificateur doit se composer du code à six caractères suivi d'un trait d'union et du numéro de version à trois chiffres de l'objet d'information.

Afin de garantir le caractère unique mondial des identificateurs dans le dictionnaire de référence, il convient que l'identification de l'autorité d'enregistrement soit ajoutée à l'identificateur. Le concept d'identification de l'autorité d'enregistrement est défini dans l'ISO/IEC 6523-1. Tout d'abord, un plan d'identification d'organisme est choisi parmi les plans enregistrés en utilisant le code international de désignation (ICD, International Code

Designator). Ensuite, dans ce plan, l'organisme est identifié par l'identificateur d'organisme (OI, Organization Identifier) (voir Tableau 3).

Pour le Dictionnaire de données commun de l'IEC, le code ICD «0112» doit être utilisé pour désigner un «organisme d'enregistrement de normalisation». Pour l'identificateur d'organisme (OI), le chiffre «2» doit être utilisé pour désigner l'«IEC».

Une identification de source pour les données du dictionnaire de données commun de l'IEC est: «0112/2///61360_4». Le «61360_4» ajouté à la fin est l'identificateur de norme (SI, Standards Identifier) qui est décrit dans l'ISO 13584-26 comme une extension de l'ISO/IEC 6523-1.

Le Dictionnaire de données commun de l'IEC peut comprendre de nombreuses collections d'articles de dictionnaire. Ces collections ne créent pas de nouveaux espaces d'identification, c.-à-d. que les identificateurs doivent être uniques dans tout l'espace d'adresse.

EXEMPLE 1 «0112/2///61987» pour la série IEC 61987 ou «0112/2///62683» pour l'IEC 62683.

EXEMPLE 2 L'identification unique globale d'un objet d'information particulier peut être «0112/2///61340_4#AAF307#005».

Tableau 3 – Identification unique globale

Nom	Description	Obligation	Type de données	Longueur maximale
Code international de désignation (ICD)	Identification du plan d'identification d'un organisme	Obligatoire	Nombre entier	4
Identificateur d'organisme (OI)	Identification d'un organisme dans un plan d'identification	Obligatoire	Chaîne	35

7.2 Spécification de l'objet d'information

Nom: **Identifying_attributes**

Définition: collection d'attributs identifiant un objet d'information

Description:

La Figure 4 représente le modèle détaillé de l'objet d'information «**Identifying_attributes**» utilisant le langage UML comme technique de présentation.

Identifying_attributes	
code[1] version_number[1] revision_number[1] preferred_name[1] synonymous_name[0..2] short_name[0..1]	
IEC	
Anglais	Français
Identifying_attributes	Attributs d'identification
code(1)	code(1)
version_number(1)	numéro_de_version(1)
revision_number(1)	numéro_de_révision(1)
preferred_name(1)	nom_préféré(1)
synonymous_name(0...2)	nom_synonyme(0...2)
short_name(0...1)	nom_abrégé(0...1)

**Figure 4 – Attributs d'identification
(diagramme de classe UML)**

7.3 Code

Nom de l'attribut: **code**

Définition de l'attribut: identificateur unique à six caractères d'un objet d'information

NOTE L'IEC 61360-2:2012 spécifie une longueur maximale de l'attribut «code» de 14 caractères. La différence est due à l'information ajoutée au code, telle que l'identificateur international de données d'enregistrement (voir l'ISO/IEC 11179-6:2005).

Description: Les trois premiers caractères doivent être alphabétiques, les trois derniers numériques (format AAANN). Le caractère «X»¹ ne doit pas être utilisé comme premier caractère. Les codes sont émis séquentiellement et ne doivent pas avoir de relation avec la signification de l'objet d'information.

Un nouvel (autre) objet d'information avec un nouvel identificateur doit être défini en cas de changement d'au moins un attribut de l'objet d'information affectant la signification et/ou la communication de l'objet d'information.

La modification éventuelle de l'identificateur de manière à générer un nouvel objet d'information ou la modification éventuelle de numéro de version de l'objet d'information doivent être déterminées séparément pour chaque cas.

De tels attributs sont:

definition²;

unit_of_measure;

depends_on;

data_type.

La suppression de l'association à une **Condition_Property**, c'est-à-dire la modification du contenu de **depends_on**, doit donner lieu à une modification de **code**.

Obligation: obligatoire

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

7.4 Nom préféré

Nom de l'attribut: **preferred_name**

Définition de l'attribut: désignation d'un ou plusieurs mots assignés à un article

Description: Le contenu du **preferred_name** doit être identique à celui utilisé dans les normes internationales, s'il existe. Il est conseillé de limiter la longueur du nom à 35 caractères même si elle peut être étendue jusqu'à 255 caractères.

Voir l'Annexe F pour les conventions pour les noms.

¹ Pour une utilisation locale dans des environnements privés, il est admis d'utiliser des codes commençant par XAA et allant jusqu'à XZZ inclus et ceux-ci sont par conséquent exclus du Dictionnaire de données commun de l'IEC.

² Le fait d'étendre le contexte sémantique d'un objet d'information qui ne dénature pas la signification définie à l'origine en incluant une autre **Item_class** dans la définition donne lieu à une modification de version.

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

7.5 Numéro de révision

Nom de l'attribut: **revision_number**

Définition de l'attribut: numéro utilisé pour le contrôle administratif d'un objet d'information

Description: Il convient que le **revision_number** d'un objet d'information soit composé de 3 chiffres. Bien que le modèle d'information commun EXPRESS ISO/IEC autorise un nombre plus élevé de chiffres, pour que le dictionnaire soit enregistré dans l'IEC CDD ou soit compatible avec la série IEC 61360, il convient de n'utiliser que 3 chiffres. Il est admis d'utiliser un plus grand nombre de chiffres à des fins spécifiques. Les numéros des révisions consécutives commençant par «000» doivent être donnés dans l'ordre croissant. Un seul **revision_number** par objet d'information, unique par son identificateur, est valable à un moment donné.

Un nouveau numéro de révision d'un objet d'information doit être généré si un attribut de l'objet d'information est modifié sans que cela n'affecte ni son utilisation ni sa signification ou lorsque des corrections de fautes de frappe et d'orthographe sont apportées.

Le **revision_number** doit être ramené à sa valeur de départ «000» lorsque le **version_number** est modifié.

Voir l'Annexe D pour les règles de définition des nouvelles versions ou révisions des éléments du dictionnaire.

Obligation: obligatoire

Type de caractère des valeurs: Chiffres de «0» à «9».

7.6 Nom abrégé

Nom de l'attribut: **short_name**

Définition de l'attribut: représentation abrégée du contenu du **preferred_name**

Description: Le premier caractère du contenu d'un **short_name** doit être une lettre:

- Il convient que le nom abrégé soit précédé d'un caractère «@» pour une **Condition_property**;
- le nom abrégé doit être une représentation abrégée du contenu du **preferred_name**. Il convient que la longueur du nom abrégé soit limitée à 18 caractères.

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

7.7 Nom synonyme

Nom de l'attribut: **synonymous_name**

Définition de l'attribut: désignation par un ou plusieurs mots qui diffère du nom préféré donné mais représente le même concept

Description: Le nombre de noms synonymes doit être limité à deux. Pour des raisons d'ordre pratique, il convient que la longueur du **synonymous_name** soit limitée à 70 caractères.

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

7.8 Numéro de version

Nom de l'attribut: **version_number**

Définition de l'attribut: numéro utilisé pour contrôler les versions d'un objet d'information

Description: Il convient que le **version_number** d'un objet d'information soit composé de 3 chiffres. Bien que le modèle d'information commun EXPRESS ISO/IEC autorise un nombre plus élevé de chiffres, pour que l'objet d'information soit enregistré dans l'IEC CDD ou soit compatible avec la série IEC 61360, il convient de n'utiliser que 3 chiffres. Il est admis d'utiliser un plus grand nombre de chiffres à des fins spécifiques. Les numéros des versions consécutives commençant par «001» doivent être donnés en suivant l'ordre croissant. Un seul numéro de version par objet d'information, unique de par son identificateur, est valable à un moment donné.

Une nouvelle version d'un objet d'information doit être générée si au moins un attribut de l'objet d'information a été modifié et peut affecter son utilisation, mais sans affecter sa signification.

Voir l'Annexe D pour les règles de définition des nouvelles versions ou révisions des éléments du dictionnaire.

Obligation: obligatoire

Type de caractère des valeurs: Chiffres de «0» à «9».

8 Attributs_sémantiques

8.1 Vue d'ensemble

L'objet d'information **Semantic_attributes** collecte les attributs qui spécifient la signification d'un objet d'information.

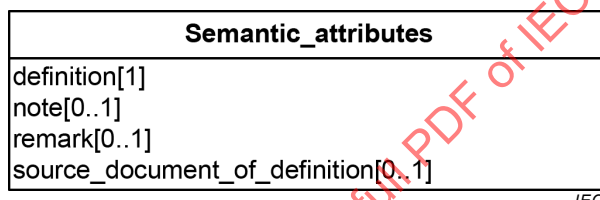
8.2 Spécification de l'objet d'information

Nom: **Semantic_attributes**

Définition: collection d'attributs spécifiant le contenu sémantique d'un objet d'information

Description:

La Figure 5 représente le modèle détaillé de l'objet d'information «**Semantic_attributes**» utilisant le langage UML comme technique de présentation.



IEC

Anglais	Français
Semantic_attributes	Semantic_attributes
definition(1)	définition(1)
note(0...1)	note(0...1)
remark(0...1)	remarque(0...1)
source_document_of_definition(0...1)	document_source_de_définition(0...1)

**Figure 5 – Attributs sémantiques
(diagramme de classe UML)**

8.3 Définition

Nom de l'attribut: **definition**

Définition de l'attribut: énoncé décrivant la signification d'un objet d'information de manière unique et non ambiguë pour permettre de le différencier des autres objets d'information

EXEMPLE 1 Définition de «distance d'arc»:

distance d'arc
valeur de la distance la plus courte dans l'air en dehors de l'isolant entre les parties métalliques supportant normalement la tension de fonctionnement à leurs bornes

Description:

Conventions et exigences:

NOTE Un objet d'information normalisé dans des éditions antérieures de l'IEC 61360-1 peut être mis à jour seulement sur demande de modification.

- a) Le contenu de la définition d'un objet d'information doit être dérivé de la définition d'origine indiquée dans les dernières normes IEC ou ISO correspondantes, le cas échéant.
- b) Si possible, les définitions des objets d'information de propriété doivent être indépendantes des classes. Ainsi la réutilisation des propriétés comme parties d'objets d'information d'autres classes est prise en charge.
- c) Il convient d'utiliser l'ISO 704 comme base pour la rédaction de l'énoncé de la définition.
- d) L'unité de mesure ne doit pas être incluse dans l'énoncé de la définition.
- e) Il convient que l'information de niveau n'apparaisse pas dans l'énoncé de la définition.

NOTE L'information de niveau peut être spécifiée au moyen du type de données **level_type**.

- f) Il convient que le contexte sémantique soit inclus dans l'énoncé de la définition si cela est essentiel pour la compréhension de sa signification.
- g) Si le concept exige une limitation de son applicabilité, cela doit être exprimé explicitement dans l'énoncé de la définition.

EXEMPLE 2 Il existe différentes sémantiques pour le terme «tension assignée»; pour les produits $\geq 1,5$ kV, le terme tension assignée exprime la tension maximale pour laquelle un produit a été conçu et peut fonctionner. Actuellement, cela ne s'applique pas aux produits inférieurs à 1,5 kV.

- h) Si des relations de dépendance sont parties inhérentes du concept, celles-ci doivent être incluses dans l'énoncé de la définition.

EXEMPLE 3 Enoncé de la définition de la Property «temps de recouvrement inverse»:

«valeur du temps exigée pour que le courant inverse d'une diode revienne à une valeur spécifiée, lorsqu'elle est commutée à partir d'un courant direct spécifié vers une tension inverse spécifiée, dans des conditions spécifiées»

- i) Dans le cas où les conditions sont spécifiées, il convient que l'énoncé de la définition se termine par les mots «à la ou aux condition(s) spécifiée(s)».
- j) Si le concept représente une valeur moyenne, la méthode de calcul de la moyenne doit être identifiée par un qualificatif qui désigne la méthode tel que «moyenne arithmétique», «moyenne géométrique», «médian» ou «mode», ou bien dans le **preferred_name** ou dans la **definition**.

Voir l'Annexe F pour les conventions pour les définitions.

Obligation:

obligatoire

Type de caractère des valeurs: Voir l'Article 26.

8.4 Note

Nom de l'attribut: **note**

Définition de l'attribut: énoncé donnant des informations supplémentaires sur la **définition**, considérées comme essentielles à sa compréhension

Description: EXEMPLE La **Property** «temps de recouvrement inverse» est plus amplement explicitée par une **note**: «Le temps de recouvrement inverse est mesuré en tant qu'intervalle de temps entre t_0 , le point d'intersection entre le courant direct et l'axe du courant nul et le moment quand pour des valeurs décroissantes de i_R une ligne passant par les points $0,9 I_{RM}$ et $0,25 I_{RM}$ croise l'axe du courant nul».

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

8.5 Remarque

Nom de l'attribut: **remark**

Définition de l'attribut: information supplémentaire sous forme de texte

Description: Le contenu d'une **remark** ne doit pas modifier la signification de la **définition**.

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

8.6 Document source de définition

Nom de l'attribut: **source_document_of_definition**

Définition de l'attribut: référence au document source d'où provient le contenu de la **définition**

Description: Le document doit être reconnu par le comité concerné de l'ISO ou de l'IEC comme étant largement accepté et faisant autorité, et comme étant également publiquement accessible.

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

9 Attributs_administratifs

9.1 Vue d'ensemble

L'objet d'information **Administrative_attributes** collecte les attributs qui servent à administrer un objet d'information.

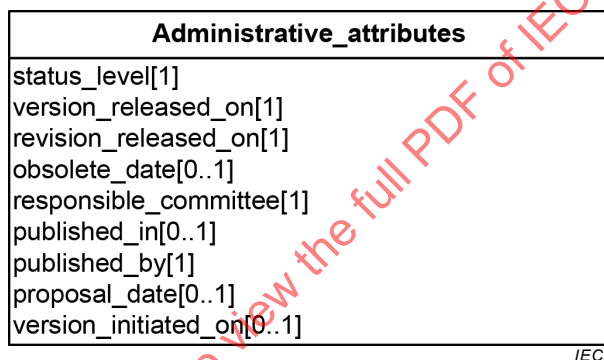
9.2 Spécification de l'objet d'information

Nom: **Administrative_attributes**

Définition: collection d'attributs spécifiant les informations utilisées pour gérer un objet d'information

Description:

La Figure 6 représente le modèle détaillé de l'objet d'information «Administrative_attributes» utilisant le langage UML comme technique de présentation.



IEC

Anglais	Français
Administrative_attributes	Attributs administratifs
status_level(1)	niveau_de_statut(1)
version_released_on(1)	version_publiée_le(1)
revision_released_on(1)	révision_publiée_le(1)
obsolete_date(0...1)	date_obsolète(0...1)
responsible_committee(1)	comité_responsable(1)
published_in(0...1)	publié_dans(0...1)
published_by(1)	publié_par(1)
proposal_date(0...1)	proposé_le(0...1)
version_initiated_on(0...1)	version_commencée_le(0...1)

**Figure 6 – Attributs administratifs
(diagramme de classe UML)**

9.3 Date obsolète

Nom de l'attribut: **obsolete_date**

Définition de l'attribut: date à laquelle le niveau de statut (voir 9.9) a reçu la valeur «remplacé»

EXEMPLE «2010-06-28»

Obligation: conditionnel

Type de caractère des valeurs: format de date sous la forme AAAA-MM-JJ comme spécifié par l'ISO 8601

9.4 Proposé le

Nom de l'attribut: **proposal_date**

Définition de l'attribut: date à laquelle l'objet d'information a été téléchargé dans la base de données

EXEMPLE «2003-05-14»

Obligation: facultatif

Type de caractère des valeurs: format de date sous la forme AAAA-MM-JJ comme spécifié par l'ISO 8601

9.5 Publié dans

Nom de l'attribut: **published_in**

Définition de l'attribut: numéro d'identité d'un document disponible pour le public dans lequel l'article et sa définition ont été publiés en premier.

Il convient que le numéro d'édition et/ou l'année soient inclus suivant le cas.

EXEMPLE «IEC 61175:2009»

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

9.6 Publié par

Nom de l'attribut: **published_by**

Définition de l'attribut: organisme responsable de la publication comme défini par «publié dans»

EXEMPLE «IEC»

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

9.7 Comité responsable

Nom de l'attribut:	responsible_committee
Définition de l'attribut:	Comité Technique ou sous-comité de l'IEC ou de l'ISO chargé de tenir à jour l'objet d'information
	EXEMPLE «SC3D»
Obligation:	obligatoire
Type de caractère des valeurs:	Voir l'Article 26.

9.8 Révision publiée le

Nom de l'attribut:	revision_released_on
Définition de l'attribut:	date à laquelle l'objet d'information portant le numéro de révision actuel a été autorisé
	EXEMPLE «2004-05-20»
Obligation:	obligatoire
Type de caractère des valeurs:	format de date sous la forme AAAA-MM-JJ comme spécifié par l'ISO 8601

9.9 Niveau de statut

Nom de l'attribut:	status_level				
Définition de l'attribut:	nom de l'étape du processus de normalisation à laquelle se trouve l'objet d'information				
Description:	Les niveaux de statuts suivants peuvent être attribués: <table> <tr> <td>Proposal (Proposition)</td><td>niveau du statut d'un nouvel objet d'information issu de son enregistrement et de son identification dans la base de données, jusqu'à son acceptation dans le cadre de travaux et jusqu'à ce qu'une résolution soit obtenue sur la manière de procéder après les préliminaires</td></tr> <tr> <td>Draft (Projet)</td><td>niveau du statut d'un nouvel objet d'information qui a été accepté pour donner lieu à des travaux à la suite des préliminaires avec la procédure de base de données normale ou étendue, jusqu'au moment où il est décidé s'il fera ou non partie de la norme</td></tr> </table>	Proposal (Proposition)	niveau du statut d'un nouvel objet d'information issu de son enregistrement et de son identification dans la base de données, jusqu'à son acceptation dans le cadre de travaux et jusqu'à ce qu'une résolution soit obtenue sur la manière de procéder après les préliminaires	Draft (Projet)	niveau du statut d'un nouvel objet d'information qui a été accepté pour donner lieu à des travaux à la suite des préliminaires avec la procédure de base de données normale ou étendue, jusqu'au moment où il est décidé s'il fera ou non partie de la norme
Proposal (Proposition)	niveau du statut d'un nouvel objet d'information issu de son enregistrement et de son identification dans la base de données, jusqu'à son acceptation dans le cadre de travaux et jusqu'à ce qu'une résolution soit obtenue sur la manière de procéder après les préliminaires				
Draft (Projet)	niveau du statut d'un nouvel objet d'information qui a été accepté pour donner lieu à des travaux à la suite des préliminaires avec la procédure de base de données normale ou étendue, jusqu'au moment où il est décidé s'il fera ou non partie de la norme				

NOTE 1 La transition vers le **status_level** "projet" a lieu à partir du **status_level** "proposé".

Standard (Norme)	niveau du statut d'un nouvel objet d'information qui a été mis à disposition pour être utilisé comme partie de la norme
	NOTE 2 La transition vers le status_level "norme" a lieu à partir du status_level "projet".
Superseded (Remplacé)	niveau du statut d'un objet d'information qui ne fait plus partie de la norme, quelle qu'en soit la raison
	NOTE 3 La transition vers le status_level "remplacé" a lieu à partir du status_level "norme". Sur la page de l'objet d'information, une note ou une référence à un objet d'information de remplacement indique la raison de cette obsolescence.
Rejected (Rejeté)	statut d'un objet d'information qui a été entré dans la base de données comme partie d'une demande de modification, mais qui n'a pas été approuvé comme partie de la norme
	NOTE 4 La transition vers "rejeté" a lieu à partir du status_level "proposé" ou "projet".
Obligation:	obligatoire
Type de caractère des valeurs:	Voir l'Article 26.

9.10 Version commencée le

Nom de l'attribut:	version_initiated_on
Définition de l'attribut:	date à laquelle l'objet d'information portant le numéro de version actuel a été publié initialement
	EXEMPLE «2004-05-15»
Obligation:	facultatif
Type de caractère des valeurs:	format de date sous la forme AAAA-MM-JJ comme spécifié par l'ISO 8601

9.11 Version publiée le

Nom de l'attribut:	version_released_on
Définition de l'attribut:	date à laquelle l'objet d'information portant le numéro de version actuel a été autorisé
	EXEMPLE «2004-05-15»
Obligation:	obligatoire
Type de caractère des valeurs:	format de date sous la forme AAAA-MM-JJ comme spécifié par l'ISO 8601

10 Attributs_de_valeur

10.1 Vue d'ensemble

L'objet d'information **Value_attributes** collecte les attributs qui spécifient la valeur numérique ou non numérique d'une **Property**.

Chaque objet d'information **Property** peut avoir un domaine de valeur qui couvre la plage de valeurs admissibles, qu'elle soit spécifiée de manière implicite par le **value_format** ou de manière explicite par la liste des valeurs autorisées.

Pour les objets d'information **Property** quantitatifs, le domaine de valeur peut être exprimé en une plage de valeurs donnant les valeurs minimale et maximale ou en une énumération explicite des valeurs discrètes possibles.

Pour les objets d'information **Property** non quantitatifs, le domaine de valeur peut être exprimé en une énumération des valeurs admissibles sous forme de chaînes de texte.

NOTE L'attribut **value_format** (voir 10.8) peut être utilisé comme indicateur des valeurs prévues pouvant être attribuées à une **Property** pendant son utilisation. De plus, il peut servir à effectuer un contrôle de qualité de base sur les données fournies.

Pour les objets d'information **Property** avec ensembles de valeurs prédéfinies, un mécanisme d'utilisation des valeurs, représenté par les codes de valeur et les significations associées des valeurs, est décrit à l'Article 16.

Lorsqu'une **Property** sert de propriété de classification, une liste de valeurs doit être spécifiée.

10.2 Spécification de l'objet d'information

Nom: **Value_attributes**

Définition: collection d'attributs spécifiant la ou les valeurs et unités d'un objet d'information

Description:

La Figure 7 représente le modèle détaillé de l'objet d'information «Value_attributes» utilisant le langage UML comme technique de présentation.

Value_attributes	
data_type[1] value_format[0..1] unit_of_measure[0..1] alternative_units_of_measure[0..*] referenced_class_identifier[0..1] number_of_significant_digits[0..1]	
IEC	
Anglais	Français
Value_attributes	Attributs_de_valeur
data_type(1)	type_de_données(1)
value_format(0...1)	format_de_valeur(0...1)
unit_of_measure(0...1)	unité_de_mesure(0...1)
alternative_units_of_measure(0...*)	autres_unités_de_mesure(0...*)
referenced_class_identifier(0...1)	identificateur_de_classe_référence(0...1)
number_of_significant_digits(0...1)	nombre_de_chiffres_significatifs(0...1)

Figure 7 – Attributs de valeur (diagramme de classe UML)

10.3 Autres unités de mesure

Nom de l'attribut: **alternative_units_of_measure**

Définition de l'attribut: unités pouvant être utilisées en variante pour spécifier la valeur d'une propriété

NOTE 1 Les unités autres autorisent la recommandation d'unités supplémentaires en plus de l'unité préférée (voir 10.7). Les raisons qui peuvent conduire à choisir d'autres unités sont par exemple des principes de mesure spécifiques, des domaines d'application particuliers ou des appareils législatifs spéciaux.

NOTE 2 D'autres unités peuvent être mises à disposition dans des groupes d'unités dédiés (voir 24.8.3).

Description:

Conventions et exigences:

- a) D'autres unités peuvent être fournies en plus de l'unité préférée. Ces unités peuvent ou non être des unités SI.

EXEMPLE bar, psi, ou Torr à la place de Pa.

- b) Les autres unités peuvent être spécifiées au moyen du code de l'unité spécifié dans l'IEC TS 62720 ou dans les dictionnaires d'unités conformes.

NOTE 3 Les dictionnaires peuvent être fournis sous forme interprétable par ordinateur ou en format papier.

- c) La conversion en une unité différente ne fait pas partie du domaine d'application du présent document.
- d) La prescription de l'utilisation d'une unité spécifique dans un but de transaction incluant de possibles préfixes ne fait pas partie du domaine d'application du présent document.

NOTE 4 Les objets d'information **Property** avec un **data_type** «CLASS_REFERENCE_TYPE» ne comportent pas d'unité.

- e) Il est fortement recommandé de ne pas spécifier le même concept dans différentes propriétés si seuls les préfixes des unités associées sont différents. Des exceptions peuvent se produire si certaines unités sont très répandues, en particulier dans le domaine des affaires, tel que «nm» (nanomètre) dans la fabrication des semiconducteurs, ou «kV» (kilovolt) dans le domaine de la distribution de puissance.
- f) Il est exigé que chaque unité contenue dans l'attribut `alternative_units_of_measure` soit associée à une représentation en chaîne, dont la représentation textuelle peut être utilisée pour caractériser l'unité au niveau de l'instance.

Obligation:

facultatif

Type de caractère des valeurs: Voir l'Article 26.

10.4 Type de données

10.4.1 Généralités

Nom de l'attribut:

data_type

Définition de l'attribut:

identifie une qualité spécifique de la valeur d'une **Property**

Description: Le contenu de **data_type** est divisé en catégories:

- type simple;
- type énumération;
- type agrégat;
- type instance de classe;
- type référence de classe;
- données de grand objet;
- type placement;
- type niveau.

Obligation: obligatoire

Type de caractère de la valeur: Voir l'Article 26.

10.4.2 Type simple

Un type de données indiquant que la valeur d'une **Property** est une valeur simple numérique ou non numérique. Le format de la valeur doit être défini conformément à 10.8.

Dans chaque désignation de type où l'on trouve «_TYPE», «_TYPE» peut être omis.

Pour un type simple, les types de données valides sont les suivants:

- **BOOLEAN_TYPE** (type booléen)

Le type de données **BOOLEAN_TYPE** peut être utilisé pour des valeurs représentant des valeurs vraies en algèbre logique ou booléenne ou un modèle de valeurs vraies.

L'utilisation de «TRUE» comme valeur booléenne indique un résultat vrai ou précis. «FALSE» comme valeur booléenne exprime un résultat faux ou imprécis. Les autres valeurs sont déconseillées.

EXEMPLE 1 Les valeurs booléennes déconseillées sont des valeurs telles que «OUI» ou «NON» et «O» ou «N».

Si une valeur contient une seule valeur vraie, le format de données doit contenir «B1».

NOTE 1 L'utilisation de valeurs **BOOLEAN_TYPE** peut mener à des spécifications défectueuses si la signification des valeurs n'est pas spécifiée correctement. Une liste de valeurs donne souvent des informations plus précises que des valeurs **BOOLEAN_TYPE**.

De plus certains types de présentations peuvent perdre leur signification si des valeurs **BOOLEAN_TYPE** sont utilisées.

EXEMPLE 2 Dans les schémas de circuit, les textes principaux sont souvent omis en raison des restrictions d'espace ou de la volonté d'éviter une traduction, et seules les valeurs sont présentées. La Figure 8 représente des exemples de données techniques associées à des lignes de connexion présentées sous forme abrégée (voir l'IEC 61082:2014, Figure 38). Dans ce cas, l'utilisation de valeurs **BOOLEAN_TYPE** peut rendre les présentations difficiles à interpréter, car ces valeurs n'ont pas de signification sans texte.

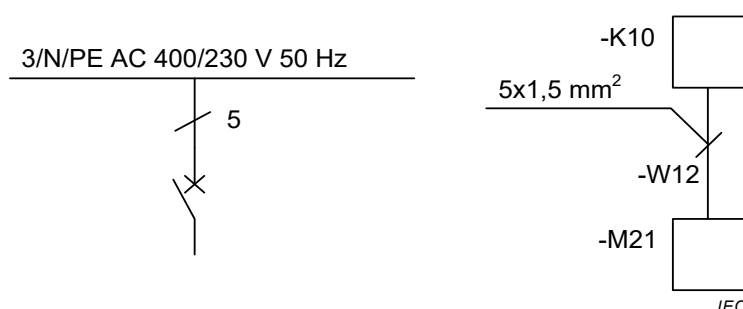


Figure 8 – Exemples de données techniques associées à des lignes de connexion

- **BINARY_TYPE** (type binaire)

Le type de données **BINARY_TYPE** peut être utilisé pour des valeurs contenant une séquence de bits, chaque bit étant représenté par «0» ou «1». Le format de la valeur doit être «B n » avec $n > 1$. Il convient que la partie de texte de la propriété, c.-à-d. le contenu des attributs «définition», «note» ou «remarque», donne des lignes directrices sur l'interprétation du schéma binaire.

NOTE 2 Aucune limite supérieure n'est spécifiée pour la valeur de « n ». Les mises en œuvre actuelles (en 2014) prennent en charge les séquences de bits («chaînes de bits») d'une taille allant jusqu'à 2 gigaoctets.

- **STRING_TYPE** (type chaîne)

Le type de données **STRING_TYPE** peut être utilisé pour des valeurs contenant une quelconque séquence d'éléments, généralement des caractères, utilisant un codage de caractères.

Un ou plusieurs types de données plus spécifiques suivants peuvent être utilisés à la place de **STRING_TYPE**:

- **TRANSLATABLE_STRING_TYPE** (type chaîne traduisible);
- **NON_TRANSLATABLE_STRING_TYPE** (type chaîne non traduisible);
- **NON_QUANTITATIVE_CODE_TYPE** (type code non quantitatif (déconseillé));
- **DATE_TIME_TYPE** (type date et heure);
- **DATE_TYPE** (type date);
- **TIME_TYPE** (type heure);
- **IRDI_STRING** (chaîne IRDI);
- **URI_TYPE** (type URI).
- **HTML5_TYPE** (type HTML5).

- **TRANSLATABLE_STRING_TYPE**

Le type de données **TRANSLATABLE_STRING_TYPE** peut être utilisé pour des valeurs contenant des chaînes mais qui sont supposées être représentées en chaînes différentes dans des langues différentes.

- **NON_TRANSLATABLE_STRING_TYPE**

Le type de données **NON_TRANSLATABLE_STRING_TYPE** peut être utilisé pour des valeurs contenant des chaînes mais qui doivent être représentées de la même manière dans toutes les langues, c.-à-d. que les chaînes ne doivent pas être traduites.

EXEMPLE 3 L'adresse MAC d'un ordinateur, p. ex. «00-80-41-ae-fd-7e».

- **DATE_TYPE**

Le type de données **DATE_TYPE** peut être utilisé pour des valeurs contenant une date du calendrier spécifiée conformément à une représentation conforme à l'ISO 8601.

Une valeur dont le type de données est **DATE_TYPE** doit satisfaire à la représentation lexicale qui suit, qui est un sous-ensemble des représentations lexicales autorisées par

l'ISO 8601. La représentation lexicale de DATE_TYPE est la représentation lexicale réduite (tronquée) de DATE_TIME_TYPE: «AAAA-MM-JJ». Aucune troncature à gauche n'est permise. Un qualificatif facultatif de fuseau horaire est autorisé pour DATE_TIME_TYPE. Afin d'intégrer les valeurs des années extérieures à la plage comprise entre «0001» et «9999», des chiffres supplémentaires peuvent être ajoutés à gauche de cette représentation et un signe «-» précédent est autorisé.

EXEMPLE 4 «1999-05-31» est la représentation DATE_TYPE de: «31 mai 1999».

Il convient que l'attribut **format** contienne la valeur «M..10».

- TIME_TYPE

Le type de données TIME_TYPE peut être utilisé pour des valeurs définissant une heure spécifiée conformément à une représentation conforme à l'ISO 8601. Une valeur de TIME_TYPE représente une heure qui revient tous les jours.

NOTE 3 Seul un sous-ensemble des représentations lexicales autorisées par l'ISO 8601 est autorisé pour les valeurs de TIME_TYPE.

NOTE 4 La restriction des représentations de l'ISO 8601 mentionnée ci-dessus est identique à celle définie par le schéma XML.

NOTE 5 La représentation lexicale autorisant un indicateur facultatif de fuseau horaire, les valeurs TIME_TYPE sont partiellement ordonnées, car il n'est pas possible de déterminer l'ordre de deux valeurs dont l'une possède un fuseau horaire et l'autre non.

Une valeur dont le type de données est TIME_TYPE doit satisfaire à la représentation lexicale qui suit, qui est un sous-ensemble des représentations lexicales autorisées par l'ISO 8601. La représentation lexicale pour TIME_TYPE est la représentation lexicale tronquée à gauche de DATE_TIME_TYPE «hh:mm:ss.sss» avec un indicateur facultatif suivant de fuseau horaire.

EXEMPLE 5 «13:20:00-05:00» est la représentation en TIME_TYPE de: 13:20, en heure normale de l'Est qui a 5 heures de retard sur le temps universel coordonné (UTC).

Il convient que l'attribut **format** contienne la valeur «M..14».

- DATE_TIME_TYPE

Le type de données DATE_TIME_TYPE peut être utilisé pour des valeurs définissant un moment spécifié conformément à une représentation conforme à l'ISO 8601.

Une valeur dont le type de données est DATE_TIME_TYPE doit satisfaire à la représentation lexicale qui suit, qui est un sous-ensemble des représentations lexicales autorisées par l'ISO 8601. Cette représentation lexicale est le format étendu de l'ISO 8601 «AAAA-MM-JJThh:mm:ss», où «AAAA» représente l'année, «MM» le mois et «JJ» le jour, précédé d'un signe «-» facultatif. La lettre «T» est le séparateur date/heure et «hh», «mm» et «ss» représentent respectivement les heures, les minutes et les secondes. Des chiffres supplémentaires peuvent être utilisés pour augmenter la précision des fractions de secondes si souhaité, c.-à-d. que le format «ss.ss...» comptant tout nombre de chiffres après la décimale est pris en charge. La partie des fractions de secondes est facultative; les autres parties de la forme lexicale ne sont pas facultatives. Afin d'intégrer les valeurs supérieures à «9999» pour les années, des chiffres supplémentaires peuvent être ajoutés à gauche de cette représentation. Les zéros de tête sont exigés si la valeur de l'année sans eux possède moins de quatre chiffres; sinon, ils sont interdits. L'année «0000» est interdite. Le champ «AAAA» doit compter au moins quatre chiffres, les champs «MM», «JJ», «SS», «hh», «mm» et «ss» exactement deux chiffres chacun (sans compter les fractions de secondes); des zéros de tête doivent être utilisés si le champ sans eux ne compte pas assez de chiffres. Cette représentation peut être immédiatement suivie d'un «Z» pour indiquer le «temps universel coordonné (UTC)» ou, pour indiquer le fuseau horaire, c.-à-d. la différence entre l'heure locale et le temps universel coordonné, immédiatement suivie d'un signe «+» ou «-», suivi de la différence par rapport à l'UTC représentée sous la forme «hh:mm». Si le fuseau horaire est inclus, les heures comme les minutes doivent être présentes.

L'ISO 8601 donne des détails sur les valeurs légales des divers champs. Si le fuseau horaire est inclus, les heures comme les minutes doivent être présentes.

EXEMPLE 6 Pour indiquer 13:20, le 31 mai 1999 pour l'heure normale de l'Est qui a 5 heures de retard sur le temps universel coordonné (UTC), l'on peut écrire: «1999-05-31T13:20:00-05:00».

NOTE 6 La restriction des représentations de l'ISO 8601 mentionnée ci-dessus est identique à celle définie par le schéma XML.

Il convient que l'attribut **format** contienne la valeur «M..25».

- **IRDI_STRING_TYPE**

Le type de données **IRDI_STRING_TYPE** peut être utilisé pour des valeurs conformes aux séquences d'identificateurs globales de la série ISO/IEC 11179.

L'IRDI, le DI et le **code** ont la relation suivante:

IRDI::= RAI'#DI'#VI

NOTE 7 RAI désigne l'identificateur d'autorité d'enregistrement, DI l'identificateur de données et VI l'identificateur de version. L'attribut **code** est utilisé comme DI dans le présent document.

NOTE 8 Dans la mise en œuvre, une validation de données peut être appliquée pour vérifier s'il existe un objet d'information pour le code IRDI référencé.

IRDI_STRING_TYPE peut être utilisé à la place d'un ou plusieurs des types de données spécifiés:

- **ICID_STRING_TYPE**;
- **ISO_29002_IRDI**.

- **ICID_STRING_TYPE**

Le type de données **ICID_STRING_TYPE** peut être utilisé pour des valeurs conformes à **IRDI_STRING_TYPE** lorsque le délimiteur entre RAI et DI est «#» tandis que le délimiteur entre DI et VI est restreint à «##».

L'ICID, le DI et le **code** ont la relation suivante:

ICID::= RAI'#DI'##VI

NOTE 9 RAI désigne l'identificateur d'autorité d'enregistrement, DI l'identificateur de données et VI l'identificateur de version. L'attribut **code** est utilisé comme DI dans le présent document.

NOTE 10 Dans la mise en œuvre, une validation de données peut être appliquée pour vérifier s'il existe un objet d'information pour le code ICID référencé.

- **ISO_29002_IRDI_TYPE**

Le type de données **ISO_29002_IRDI_TYPE** peut être utilisé pour des valeurs contenant un identificateur global qui identifie un article administré dans un registre. La structure de cet identificateur est conforme à la syntaxe d'identificateur définie dans l'ISO/TS 29002-5. L'identificateur doit satisfaire aux exigences spécifiées dans l'ISO/TS 29002-5 pour un «Identificateur international de données d'enregistrement» (IRDI, International Registration Data Identifier).

L'IRDI, le DI et le **code** ont la relation suivante:

IRDI::= RAI'#DI'#VI

NOTE 11 RAI désigne l'identificateur d'autorité d'enregistrement, DI l'identificateur de données et VI l'identificateur de version. L'attribut **code** est utilisé comme DI dans le présent document.

Comme spécifié dans l'ISO/TS 29002-5, la longueur de la chaîne d'identificateur ne doit pas dépasser 290 caractères.

NOTE 12 Conformément à l'ISO/TS 29002-5, un IRDI consiste soit en une chaîne qui ne contient pas le caractère «#», pour identifier un organisme, soit en trois sous-chaînes qui ne contiennent pas le caractère «#» et qui sont séparées par le caractère «#», pour identifier tout autre article administré.

NOTE 13 Si IRDI n'est pas utilisé pour identifier un organisme:

- la première sous-chaîne, appelée «registration authority identifier (RAI)», identifie l'organisme responsable de l'administration de l'article administré;
- la deuxième sous-chaîne, appelée «data identifier (DI)», contient une catégorisation de l'article d'administration représentée par deux caractères suivis du signe moins («-») tel que défini dans l'ISO/TS 29002-5 (p. ex. «classe», «propriété», «unité»), et l'identificateur attribué à l'article administré par le RAI;

- la troisième sous-chaîne correspond au «version identifier (VI)» de l'IRDI.

- **URI_TYPE**

Le type de données URI_TYPE peut être utilisé pour des valeurs contenant une quelconque séquence de caractères qui forme un URI.

NOTE 14 Un URI_type permet en particulier d'exprimer un URL.

- **HTML5_TYPE**

Le type de données HTML5_TYPE peut être utilisé pour des valeurs contenant une quelconque séquence de caractères utilisant la syntaxe de HTML5 (voir la Recommandation du W3C 28:2014). Ce contenu permet une représentation picturale qui peut être impossible si l'on utilise uniquement des caractères.

NOTE 15 Puisque HTML5 prend en charge MathML (voir la Recommandation du W3C 10:2014), des formules mathématiques peuvent être présentées.

EXEMPLE 7 La valeur $\frac{1}{\sqrt{3}}$ 420 kV (242,48711305964282109384248781082 kV) peut être exprimée avec

HTML5_TYPE comme

```
!doctype html
html
  head
    meta charset="UTF-8"
  /head
  body
    math xmlns="http://www.w3.org/1998/Math/MathML"
      mrow
        mrow
          mfrac
            mn1/mn
            msqrt
              mn3/mn
            /msqrt
          /mfrac
        /mrow
      mo · /mo
      mn420/mn
      mi kV /mi
    /mrow
  /math
/body
/html
```

Il existe des cas dans lesquels, outre la présentation de la valeur, sa valeur numérique exacte présente également un intérêt. Dans ces cas, la référence à une **Item_class** est recommandée, pour donner une **Property** à la valeur numérique (type de données INTEGER, REAL, etc.) et une **Property** à la présentation HTML (type de données HTML5).

NOTE 16 La classe est référencée au moyen de CLASS_INSTANCE_REFERENCE.

- **NUMBER_TYPE**

Le type de données NUMBER_TYPE peut être utilisé pour des valeurs contenant des nombres qui peuvent être écrits en valeurs numériques. Ce type de données peut être utilisé lorsqu'une représentation numérique plus spécifique n'est pas nécessaire ou n'est pas souhaitable.

Un ou plusieurs types de données plus spécifiques suivants peuvent être utilisés à la place de NUMBER_TYPE:

- INTEGER (entier),
- RATIONAL (rationnel),
- REAL (réel).

- **INTEGER, INT_TYPE**

Le type de données INT_TYPE peut être utilisé pour des valeurs contenant des nombres qui peuvent être écrits en nombres positifs ou négatifs ou en zéro.

Un ou plusieurs types de données plus spécifiques suivants peuvent être utilisés à la place de INT_TYPE:

- INT_MEASURE_TYPE,
- INT_CURRENCY_TYPE,

- INT_MEASURE_TYPE

Le type de données INT_MEASURE_TYPE peut être utilisé pour des valeurs contenant des valeurs qui sont des mesures du type INTEGER.

Le type de données spécifie en outre une unité ou un identificateur d'unité dans lequel/laquelle les valeurs sont exprimées. Il peut également spécifier d'autres unités ou d'autres identificateurs d'unité qui sont valides si la valeur est explicitement associée à l'une de ces unités.

Une unité ou un identificateur d'unité est obligatoire. Lorsque les deux sont fournis, l'unité est prioritaire.

NOTE 17 L'identificateur d'unité utilisé dans les attributs **code_of_unit**, **code_of_alternative_unit** ou **code_of_list_of_units** peut se restreindre à une unité fournie par un serveur pour des unités de mesure.

- INT_CURRENCY_TYPE

Le type de données INT_CURRENCY_TYPE peut être utilisé pour des valeurs contenant des valeurs qui sont des monnaies entières.

- RATIONAL_TYPE

Le type de données RATIONAL_TYPE peut être utilisé pour des valeurs contenant des valeurs qui sont de type rationnel.

EXEMPLE 8 $\frac{1}{2}$, $\frac{3}{4}$ ou $\frac{7}{2}$ sont des valeurs qui peuvent être représentées par RATIONAL_TYPE.

A la place de RATIONAL_TYPE, le type de données plus spécifique suivant:

- RATIONAL_MEASURE_TYPE

peut être utilisé.

- RATIONAL_MEASURE_TYPE

Le type de données RATIONAL_MEASURE_TYPE peut être utilisé pour des valeurs contenant des valeurs qui sont des mesures du type RATIONAL.

EXEMPLE 9 Les tuyaux d'eau ont très souvent un diamètre de « $\frac{3}{4}$ pouce».

Le type de données spécifie en outre une unité ou un identificateur d'unité dans lequel/laquelle les valeurs sont exprimées. Il peut également spécifier d'autres unités ou d'autres identificateurs d'unité qui sont valides si la valeur est explicitement associée à l'une de ces unités.

Une unité ou un identificateur d'unité est obligatoire. Lorsque les deux sont fournis, l'unité est prioritaire.

NOTE 18 L'identificateur d'unité utilisé dans les attributs **code_of_unit**, **code_of_alternative_unit** ou **code_of_list_of_units** peut se restreindre à une unité fournie par un serveur pour des unités de mesure.

- REAL_TYPE

Le type de données REAL_TYPE peut être utilisé pour des valeurs contenant des nombres qui peuvent être écrits en décimale terminée ou non terminée; un nombre rationnel ou irrationnel.

Un ou plusieurs types de données plus spécifiques suivants peuvent être utilisés à la place de REAL_TYPE:

- REAL_MEASURE_TYPE;
- REAL_CURRENCY_TYPE.

- REAL_MEASURE_TYPE

Le type de données REAL_MEASURE_TYPE peut être utilisé pour des valeurs contenant des valeurs qui sont des mesures du type REAL.

Le type de données spécifie en outre une unité ou un identificateur d'unité dans lequel/laquelle les valeurs sont exprimées. Il peut également spécifier d'autres unités ou d'autres identificateurs d'unité qui sont valides si la valeur est explicitement associée à l'une de ces unités.

Une unité ou un identificateur d'unité est obligatoire. Lorsque les deux sont fournis, l'unité est prioritaire.

NOTE 19 L'identificateur d'unité utilisé dans les attributs **code_of_unit**, **code_of_alternative_unit** ou **code_of_list_of_units** peut se restreindre à une unité fournie par un serveur pour des unités de mesure.

- **REAL_CURRENCY_TYPE**

Le type de données contenant les valeurs réelles des monnaies.

10.4.3 Type énumération

Les types énumération peuvent être utilisés pour spécifier le fait que le contenu d'une valeur doit être pris dans un ensemble prédéfini de valeurs.

Dans chaque désignation de type où l'on trouve «_TYPE», «_TYPE» peut être omis.

Pour un type énumération, les types de données valables sont les suivants:

- **ENUM_CODE_TYPE(enum_id), ENUM_CODE_TYPE(enum_id(code1, code2, ...))**

Le type de données **ENUM_CODE_TYPE** peut être utilisé pour des valeurs référant un jeu de chaînes ou de chiffres, comme valeurs, chacun représentant des significations assignées.

NOTE 1 **enum_id** est un identificateur global pour la liste d'énumération.

NOTE 2 **code1, code2, ...** sont les codes de valeur des valeurs énumérées.

- **ENUM_INT_TYPE(enum_id), ENUM_INT_TYPE(enum_id(code1, code2, ...))**

Le type de données **ENUM_INT_TYPE** peut être utilisé pour des valeurs référant un jeu de valeurs entières comme valeurs.

NOTE 3 **enum_id** est un identificateur global pour la liste d'énumération.

NOTE 4 **code1, code2, ...** sont les codes de valeur des valeurs énumérées.

- **ENUM_REAL_TYPE(enum_id), ENUM_REAL_TYPE(enum_id(code1, code2, ...))**

Le type de données **ENUM_REAL_TYPE** peut être utilisé pour des valeurs référant un jeu de valeurs réelles comme valeurs.

NOTE 5 **enum_id** est un identificateur global pour la liste d'énumération.

NOTE 6 **code1, code2, ...** sont les codes de valeur des valeurs énumérées.

NOTE 7 "**ENUM_REAL_TYPE**" n'est pas disponible dans l'ISO 13584-42/ IEC 61360-2, un mécanisme de contrainte nommé «contrainte d'énumération» peut être utilisé à sa place. Toutefois, la spécification de la contrainte d'énumération est bien plus complexe que **ENUM_REAL**, car la première doit définir tout d'abord le type de données, puis contraindre le sous-ensemble sélectionné avec la compréhension implicite que seul l'un des deux est choisi.

- **ENUM_STRING_TYPE(enum_id), ENUM_STRING_TYPE(enum_id(code1, code2, ...))**

Le type de données **ENUM_STRING_TYPE** peut être utilisé pour des valeurs référant un jeu de valeurs de chaîne, comme valeurs, chacun représentant des significations assignées.

NOTE 8 **enum_id** est l'identificateur global pour la liste d'énumération.

NOTE 9 **code1, code2, ...** sont les codes des valeurs énumérées.

- **ENUM_RATIONAL_TYPE(enum_id),
ENUM_RATIONAL_TYPE(enum_id(code1, code2, ...))**

Le type de données `ENUM_RATIONAL_TYPE` peut être utilisé pour des valeurs référençant un jeu de valeurs de chaîne, comme valeurs, chacun représentant un nombre rationnel.

NOTE 10 `enum_id` est l'identificateur global pour la liste d'énumération.

NOTE 11 `code1`, `code2`, ... sont les codes des valeurs énumérées.

- `ENUM_REFERENCE_TYPE(enum_id)`, `ENUM_INSTANCE_TYPE(enum_id)`,
`ENUM_REFERENCE_TYPE(enum_id(code1, code2, ...))`,
`ENUM_INSTANCE_TYPE(enum_id(code1, code2, ...))`

Le type de données `ENUM_REFERENCE_TYPE`/ `ENUM_INSTANCE_TYPE` peut être utilisé pour des valeurs référençant un jeu de valeurs de chaîne, comme valeurs, chacun représentant l'identificateur global d'une classe.

NOTE 12 `enum_id` est l'identificateur global pour la liste d'énumération.

NOTE 13 `code1`, `code2`, ... sont les codes des valeurs énumérées.

- `ENUM_BOOLEAN_TYPE(enum_id)`,
`ENUM_BOOLEAN_TYPE((enum_id(code1, code2, ...)))`

Le type de données `ENUM_BOOLEAN_TYPE` peut être utilisé pour des valeurs référençant un jeu de valeurs booléennes.

NOTE 14 `enum_id` est l'identificateur global pour la liste d'énumération.

NOTE 15 `code1`, `code2`, ... sont les codes des valeurs énumérées.

10.4.4 Type d'instance de classe

- `CLASS_INSTANCE_TYPE(icid)`

Le type de données `CLASS_INSTANCE_TYPE` peut être utilisé pour des valeurs contenant une référence à une **Item_class**. La **Item_class** référencée peut résider dans n'importe quelle partie de la hiérarchie de classification et contribue à une collection de zéro, un ou plusieurs objets d'information **Property**. La valeur de la **Property** référencée doit contenir le **code** de la **Item_class** référencée.

Un `CLASS_INSTANCE_TYPE` référence une **Item_class** de manière similaire à `CLASS_REFERENCE_TYPE` (voir 10.4.5), mais à la différence de `CLASS_REFERENCE_TYPE`, les instances conformes à la **Item_class** référencée doivent être conservées ensemble dans la classe de référence, c.-à-d. qu'elles sont internes aux instances de la classe de référence. Cela signifie que si une **Item_class** A compte des instances $A_1, A_2, \dots, A_n$, où A_1 référence l'instance B_1 de la **Item_class** B, alors les instances $A_2, \dots, A_n$ de A se réfèrent à des instances différentes de B.

En d'autres termes, `CLASS_INSTANCE_TYPE` prend pour hypothèse que chaque instance de référence $A_1, A_2, \dots, A_n$ d'une **Item_class** A est liée à son instance «personnelle» de B. Ainsi, si A_1 référence B_1 et si dans l'instance B_1 de la **Item_class** référencée, une valeur d'instance est modifiée, la modification n'est pas visible par toutes les (autres) instances de la **Item_class** de référence $A_2, \dots, A_n$.

NOTE 1 Le «icid» de `CLASS_REFERENCE_TYPE` peut omettre le «VI» (identificateur de version). Dans ce cas, la classe possédant l'identificateur spécifié avec la dernière version peut être référencée.

NOTE 2 Voir l'IEC 62656-1 pour des informations détaillées sur la structure de l'identificateur global ICID.

10.4.5 Type référence de classe

- `CLASS_REFERENCE_TYPE(icid)`

Le type de données `CLASS_REFERENCE_TYPE` peut être utilisé pour des valeurs contenant une référence à une **Item_class**. La **Item_class** référencée peut résider dans n'importe quelle partie de la hiérarchie de classification et contribue à une collection de zéro, un ou plusieurs objets d'information de **Property**. La valeur de la **Property** référencée doit contenir le **code** de la **Item_class** référencée.

S'il y a des classes $A_1, A_2, \dots, A_n$ qui référence un seul et même objet d'information **Item_class** B, alors les instances de $A_1, A_2, \dots, A_n$ référencent la même instance B_1 de B.

En d'autres termes, CLASS_REFERENCE_TYPE prend pour hypothèse que l'instance stockée au côté de la **Item_class** référencée B, est partagée par toutes les classes de référence $A_1, A_2, \dots, A_n$. Ainsi, si dans la **Item_class** référencée, une valeur d'instance est modifiée, la modification est visible dans tous les objets d'information **Item_class** de référence.

NOTE 1 Le «*icid*» dans CLASS_REFERENCE_TYPE peut omettre le «*VI*» (identificateur de version). Dans ce cas, la référence est faite à la dernière version de la classe référencée.

NOTE 2 Voir l'IEC 62656-1 pour des informations détaillées sur la structure de l'identificateur global ICID.

10.4.6 Type agrégat

Type de données contenant des collections de deux valeurs ou plus. Les valeurs individuelles au sein de l'agrégat peuvent être identifiées par un indice numérique. Les jeux de valeurs à une dimension peuvent être représentés en sac, liste ou ensemble. Les jeux de valeurs à plusieurs dimensions peuvent être exprimés en matrices.

Le format applicable à chaque valeur de l'agrégat doit être défini conformément à 10.4.2.

NOTE 1 Un agrégat se relie à un type simple au moyen du mot clé «DE».

- BAG(*b1*, *b2*)

Un type de données BAG a pour domaine des collections non ordonnées de constituants similaires. Les limites inférieure et supérieure, qui sont des entiers, définissent le nombre minimal et maximal de constituants qui peuvent se trouver dans la collection définie par un type de données BAG.

Règles et restrictions:

- L'expression *b1* doit être une valeur entière égale ou supérieure à zéro. Elle donne la limite inférieure, qui est le nombre minimal de constituants qui peuvent se trouver dans un sac. *b1* ne doit pas être la valeur indéterminée («?»).
- L'expression *b2* doit être une valeur entière égale ou supérieure à *b1* ou une valeur indéterminée («?»). Elle donne la limite supérieure, qui est le nombre maximal de constituants qui peuvent se trouver dans un sac. Si cette valeur est indéterminée («?»), la valeur supérieure du nombre de constituants dans un sac n'est pas limitée.
- Si la spécification de la limite est omise, les limites sont (0, ?).

- LIST(*b1*, *b2*)

Un type de données LIST a pour domaine des séquences de constituants similaires. Les limites inférieure et supérieure, qui sont des entiers, définissent le nombre minimal et maximal de constituants qui peuvent se trouver dans la collection définie par un type de données LIST.

Règles et restrictions:

- L'expression *b1* doit être une valeur entière égale ou supérieure à zéro. Elle donne la limite inférieure, qui est le nombre minimal de constituants qui peuvent se trouver dans une liste. *b1* ne doit pas être la valeur indéterminée («?»).
- L'expression *b2* doit être une valeur entière égale ou supérieure à *b1* ou une valeur indéterminée («?»). Elle donne la limite supérieure, qui est le nombre maximal de constituants qui peuvent se trouver dans une liste. Si cette valeur est indéterminée («?»), la valeur supérieure du nombre de constituants dans une liste n'est pas limitée.
- Si la spécification de la limite est omise, les limites sont (0, ?).

- UNIQUE_LIST(*b1*, *b2*)

La valeur prend la forme d'une LIST dans laquelle des valeurs dupliquées ne doivent pas se produire.

Règles et restrictions:

- Les règles et restrictions du type de données LIST s'appliquent.

b) Chaque constituant d'une occurrence d'un type de données UNIQUE_LIST doit être différent de (c.-à-d. instance non égale à) tous les autres constituants de la même liste.

- SET(*b1*, *b2*)

Un type de données SET a comme domaine des collections non ordonnées de constituants similaires. Les limites inférieure et supérieure, qui sont des entiers, définissent le nombre minimal et maximal de constituants qui peuvent se trouver dans la collection définie par un type de données SET.

Règles et restrictions:

- L'expression *b1* doit être une valeur entière égale ou supérieure à zéro. Elle donne la limite inférieure, qui est le nombre minimal de constituants qui peuvent se trouver dans un ensemble. *b1* ne doit pas être la valeur indéterminée («?»).
- L'expression *b2* doit être une valeur entière égale ou supérieure à *b1* ou une valeur indéterminée («?»). Elle donne la limite supérieure, qui est le nombre maximal de constituants qui peuvent se trouver dans une valeur d'ensemble de ce type de données. Si cette valeur est indéterminée («?»), la valeur supérieure du nombre de constituants dans un ensemble n'est pas limitée.
- Si la spécification de la limite est omise, les limites sont (0, ?).
- Chaque constituant d'une occurrence d'un type de données SET doit être différent de (c.-à-d. instance non égale à) tous les autres constituants du même ensemble.

- CONSTRAINED_SET(*b1*, *b2*, *cmn*, *cmx*)

Le type de données CONSTRAINED_SET a comme domaine des collections non ordonnées de constituants qui peuvent être exprimés comme SET (ensemble) de valeurs dont des sous-ensembles peuvent être extraits. Les tailles des sous-ensembles autorisés sont définies par leurs valeurs minimale et maximale. Si ces tailles n'existent pas, tout sous-ensemble est autorisé.

Règles et restrictions:

- Les règles et restrictions du type de données SET s'appliquent.
- Le *cmn* doit être une valeur entière égale ou supérieure à zéro. Elle donne la limite inférieure, qui est le nombre minimal de constituants qui peuvent être extraits. *cmn* ne doit pas être la valeur indéterminée («?»).
- Le *cmx* doit être une valeur entière égale ou supérieure à *cmn* ou une valeur indéterminée («?»). Elle donne la limite supérieure, qui est le nombre maximal de constituants qui peuvent être extraits.

- ARRAY(*b1*, *b2*)

Un type de données ARRAY a comme domaine des collections indexées de taille fixe de constituants similaires. Les limites inférieure et supérieure, qui sont des nombres entiers, définissent la plage des valeurs d'indices et donc la taille de chaque collection matricielle.

m étant la limite inférieure et *n* est la limite supérieure, il y a exactement $n - m + 1$ constituants dans la matrice. Ces constituants sont indexés au moyen d'indices compris entre *m* et *n* inclus. Les limites peuvent être positives, négatives ou égales à zéro mais ne doivent pas être indéterminées («?»).

Règles et restrictions:

- Les deux valeurs dans la spécification de limite, *b1* et *b2*, doivent être des valeurs entières. Aucune ne doit contenir la valeur indéterminée («?»).
- b1* donne la limite inférieure de la matrice. Ce doit être l'indice inférieur valable pour une valeur matricielle de ce type de données.
- b2* donne la limite supérieure de la matrice. Ce doit être l'indice supérieur valable pour une valeur matricielle de ce type de données.
- b1* doit être inférieur ou égal à *b2*.

- OPTIONAL_ARRAY(*b1*, *b2*)

La valeur prend la forme d'une ARRAY (matrice) dans laquelle des valeurs indéterminées («?») peuvent se produire.

Règles et restrictions:

- a) Les règles et restrictions du type de données ARRAY s'appliquent.
- b) Une valeur matricielle de ce type de données peut posséder la valeur indéterminée («?») en une ou plusieurs positions d'indices.

- **UNIQUE_ARRAY(*b1*, *b2*)**

La valeur prend la forme d'une ARRAY dans laquelle des valeurs dupliées ne doivent pas se produire.

Règles et restrictions:

- a) Les règles et restrictions du type de données ARRAY s'appliquent.
- b) Chaque constituant de la matrice doit être différent de (c.-à-d. instance non égale à) tous les autres constituants de la même matrice.

- **UNIQUE_OPTIONAL_ARRAY(*b1*, *b2*)**

La valeur prend la forme d'une ARRAY dans laquelle des valeurs dupliées ne doivent pas se produire. Des valeurs indéterminées («?») peuvent se produire.

Règles et restrictions:

- a) Les règles et restrictions du type de données ARRAY s'appliquent.
- b) Une valeur matricielle de ce type de données peut posséder la valeur indéterminée («?») en une ou plusieurs positions d'indices.
- c) Chaque constituant d'une matrice doit être différent de (c.-à-d. instance non égale à) tous les autres constituants de la matrice.

10.4.7 Type niveau

- **LEVEL_TYPE(MIN,NOM,TYP,MAX)**

Le type de données LEVEL_TYPE peut être utilisé pour des valeurs représentant jusqu'à quatre variantes d'une valeur numérique qui peut être donnée comme

- INT_TYPE,
- INT_MEASURE_TYPE,
- REAL_TYPE,
- REAL_MEASURE_TYPE.

Chaque variante de cette valeur exprime la valeur numérique dans un rôle spécifique:

- MIN minimum de la valeur,
- NOM valeur telle que désignée,
- TYP valeur telle que typiquement présente,
- MAX maximum de la valeur.

Comme l'interprétation des valeurs de type de niveau repose sur l'ordre fixé des niveaux, tous les niveaux spécifiés pour une propriété doivent être représentés. Il n'y a aucune obligation à remplir tous les niveaux avec des valeurs bien qu'il soit conseillé de le faire pour la cohérence des données.

EXEMPLE 1 Dans le cas d'une propriété qui est du LEVEL_TYPE min/nom/typ/max et

- seule la valeur «MIN» doit être transmise, il convient que chacune des autres valeurs contienne une valeur «NIL»;
- seules les valeurs «MIN» et «MAX» doivent être transmises, il convient que chacune des autres valeurs contienne une valeur «NIL».

EXEMPLE 2 Dans le cas d'une propriété qui est du LEVEL_TYPE min/max, exprimant une plage, seules ces deux valeurs doivent être fournies.

NOTE Le signe séparateur utilisé pour séparer les valeurs des niveaux définis dépend de la mise en œuvre.

Il convient de restreindre l'utilisation du type de niveau aux objets d'information **Property** qui sont applicables à des domaines où la signalisation de multiples valeurs sur une seule caractéristique est reconnue comme une pratique courante.

EXEMPLE 3 Les valeurs utilisées dans les industries telles que l'industrie des composants électroniques.

Des valeurs limites peuvent être exprimées en définissant une **Property** spécifiant les niveaux «MIN» et/ou «MAX».

EXEMPLE 4 Une batterie de voiture de 12 V (nominale) contient 6 éléments ayant une tension typique d'environ 2,2 V chacun, ce qui donne une tension typique pour la batterie d'environ 13,5 V. En charge, la tension peut atteindre une valeur maximale d'environ 14,5 V mais on considère qu'elle est complètement déchargée lorsque la tension chute en dessous d'une valeur minimale de 12,5 V.

Une utilisation spécifique du LEVEL_TYPE est le concept de valeur limite. Lorsque la qualification «limite» est ajoutée à une **Property**, pouvant être exprimée dans sa **definition**, la propriété (ainsi que sa valeur) spécifie un seuil au-delà duquel le dispositif risque d'être endommagé, entraînant des modifications permanentes et non voulues des caractéristiques physiques ou fonctionnelles et influençant la performance du dispositif.

EXEMPLE 5 Dans le cas de l'exemple de la batterie pour voiture (voir EXEMPLE 4), la valeur limite maximale peut être exprimée à une valeur de 16,5 V.

Dans le contexte des valeurs limites, les niveaux «NOM» et «TYP» ne s'appliquent pas.

Les agrégats peuvent contenir des valeurs LEVEL_TYPE.

EXEMPLE 6 LIST(1,?) OF LEVEL(MIN,NOM,TYP,MAX) OF REAL_MEASURE_TYPE

10.4.8 Type grand objet

- LOB(*location*, *media_type*, *media_subtype*, *content_transfer_encoding*) (emplacement, type_de_média, sous_type_de_média, codage_de_transfert_de_contenu)

Un grand objet (LOB, large object) est un objet d'information qui est mémorisé logiquement comme valeur d'une **Property** mais qui peut être mémorisé physiquement de façon indépendante (essentiellement séparée) de l'objet d'information **Property**. Les grands objets mémorisent généralement de grandes quantités de données.

EXEMPLE Données vidéo, données audio, images, données xml, archives.

Une valeur de type LOB est constituée d'une liste de quatre valeurs:

- *location*: Spécification de l'endroit duquel il est permis d'accéder au grand objet. Il convient de spécifier cet emplacement de mémorisation de manière à ce que toutes les parties, à savoir le rédacteur et le ou les lecteurs du LOB, puissent accéder aux données.
- *media_type*: Spécification du type de la valeur du LOB conformément au protocole MIME (Multipurpose Internet Mail Extensions), Partie 1. Voir l'IETF RFC 2045:1996.
- *media_subtype*: Spécification du sous-type de la valeur du LOB conformément au protocole MIME (Multipurpose Internet Mail Extensions), Partie 2. Voir l'IETF RFC 2046:1996.
- *content_transfer_encoding*: Spécification du caractère codant la valeur du LOB conformément au protocole MIME (Multipurpose Internet Mail Extensions), Partie 3. Voir l'IETF RFC 2047:1996.

10.4.9 Type placement

Type de données contenant des informations de direction destinées à situer un article par rapport au système de coordonnées de son contexte géométrique. Les valeurs de type placement donnent des informations permettant de définir le système de coordonnées local exigé pour localiser l'article.

NOTE 1 Une spécification détaillée des méthodes mathématiques utilisées par les types placement se trouve dans l'IEC 62656-1 et l'ISO 10303-42.

NOTE 2 Dans le cas de l'ISO 10303-42, AXIS1_PLACEMENT, AXIS2_PLACEMENT_2D, AXIS2_PLACEMENT_3D héritent de l'attribut «emplacement» qui est du type CARTESIAN_POINT en 2D ou 3D, depuis l'entité géométrique

générale «placement». Dans le cas de l'IEC 61360-1, ceci est obtenu en associant un type de condition **Property** du type `placement`, c.-à-d., `CARTESIAN_POINT`, à la forme géométrique qui utilise l'un des types de données `AXIS1_PLACEMENT_2D`, `AXIS1_PLACEMENT_3D`, `AXIS2_PLACEMENT_2D` ou `AXIS2_PLACEMENT_3D`.

- **AXIS1_PLACEMENT_2D**(*ref_dir_X*, *ref_dir_Y*)

Les valeurs spécifient une direction de référence dans un espace bidimensionnel à utiliser comme ligne de symétrie pour les corps symétriques.

- *ref_dir_X*: Composant du vecteur de direction de la direction de référence dans la direction X du système de coordonnées du contexte géométrique.
- *ref_dir_Y*: Composant du vecteur de direction de la direction de référence dans la direction Y du système de coordonnées du contexte géométrique.

- **AXIS1_PLACEMENT_3D**(*ref_dir_X*, *ref_dir_Y*, *ref_dir_Z*)

Les valeurs spécifient une direction de référence dans un espace tridimensionnel à utiliser comme ligne de symétrie pour les corps symétriques.

- *ref_dir_X*: Composant du vecteur de direction de la direction de référence dans la direction X du système de coordonnées du contexte géométrique.
- *ref_dir_Y*: Composant du vecteur de direction de la direction de référence dans la direction Y du système de coordonnées du contexte géométrique.
- *ref_dir_Z*: Composant du vecteur de direction de la direction de référence dans la direction Z du système de coordonnées du contexte géométrique.

- **AXIS2_PLACEMENT_2D**(*ref_dir_X*, *ref_dir_Y*)

Les valeurs spécifient l'axe X local du système de coordonnées local utilisé pour localiser un article.

- *ref_dir_X*: Composant du vecteur de direction de l'axe X local dans la direction X du système de coordonnées du contexte géométrique.
- *ref_dir_Y*: Composant du vecteur de direction de l'axe X local dans la direction Y du système de coordonnées du contexte géométrique.

- **AXIS2_PLACEMENT_3D**((*ref1_dir_X*, *ref1_dir_Y*, *ref1_dir_Z*), (*ref2_dir_X*, *ref2_dir_Y*, *ref2_dir_Z*))

Les valeurs spécifient l'axe local Z et la direction utilisée pour déterminer la direction de l'axe X du système de coordonnées local utilisé pour localiser un article.

- *ref1_dir_X*: Composant du vecteur de direction de l'axe Z local dans la direction X du système de coordonnées du contexte géométrique.
- *ref1_dir_Y*: Composant du vecteur de direction de l'axe Z local dans la direction Y du système de coordonnées du contexte géométrique.
- *ref1_dir_Z*: Composant du vecteur de direction de l'axe Z local dans la direction Z du système de coordonnées du contexte géométrique.
- *ref2_dir_X*: Composant du vecteur de direction de la direction utilisée pour déterminer la direction de l'axe X local dans la direction X du système de coordonnées du contexte géométrique.
- *ref2_dir_Y*: Composant du vecteur de direction de la direction utilisée pour déterminer la direction de l'axe X local dans la direction Y du système de coordonnées du contexte géométrique.
- *ref2_dir_Z*: Composant du vecteur de direction de la direction utilisée pour déterminer la direction de l'axe X local dans la direction Z du système de coordonnées du contexte géométrique.

10.4.10 Dépendances en fonction du type de données

Comme spécifié en 10.1, les objets d'information **Property** peuvent présenter des valeurs numériques ou non numériques conformément aux spécifications des attributs **data_type** et **value_format**. Les valeurs imbriquées, comme des agrégats contenant d'autres agrégats, sont admises.

Le Tableau 4 propose une vue d'ensemble des combinaisons de valeurs de **data_type** autorisées, (voir 10.4.2 à 10.4.9):

- une combinaison autorisée est indiquée comme: «ok»;
- une combinaison qui n'est pas autorisée est indiquée par un tiret: «–».

Le contenu possible de **value_format** (voir 10.8) est également énuméré.

EXEMPLE 1 Une **Property** peut avoir un **data_type** SET (catégorie: type agrégat) de REAL_MEASURE_TYPE (catégorie: réel), qui est marqué «ok» dans le Tableau 4.

EXEMPLE 2 Une **Property** avec un **data_type** LIST (catégorie type agrégat) de ENUM_INT_TYPE (catégorie type énumération) est déconseillée et marquée «–» dans le Tableau 4.

Lorsque l'on spécifie une nouvelle **Property**, les étapes séquentielles suivantes peuvent être suivies pour définir le type de données complet par élimination dans l'ordre indiqué:

- choisir la propriété;
- choisir le type de données;
- choisir la plage de valeur des données;
- choisir le format de la valeur.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017

Tableau 4 – Dépendances en fonction du type de données

	Catégorie de data_type (voir 10.4.1)								
	Type simple	Type agrégat	Énumération	Type d'instance de classe	Type référence de classe	Données de grand objet	Type placement	Type de niveau	
Catégorie de contenu:									Contenu de l'attribut value_format (voir 10.8):
Type simple	–	ok	–	–	–	–	–	^d	A, M, N, X, B, NR1, NR2, NR3, NR4, HTML5
Type agrégat	–	ok	–	–	–	^a	–	–	
Type énumération	–	–	–	–	–	–	–	–	
Type d'instance de classe	–	^b	–	–	–	–	–	–	
Type référence de classe	–	^b	–	–	–	–	–	–	
Données de grand objet	–	–	–	–	–	–	–	–	
Type placement	–	ok	–	–	–	–	–	–	
Type de niveau	ok	–	–	–	–	–	–	–	
Booléenne/binaire	ok	ok	–	–	–	–	–	–	B
Nombre entier	ok	ok	ok	–	–	–	–	ok	NR1
Nombre réel	ok	ok	–	–	–	–	–	ok	NR2, NR3
Nombre rationnel	ok	ok	–	–	–	–	–	ok	NR4
Chaîne	ok	ok	^c	^b	^b	–	ok	–	A, M, N, X, HTML5
^a Matrice contenant le pointeur vers l'objet d'information LOB. ^b Pointeur vers l'objet d'information Item_class . ^c Pointeur vers l'objet d'information Value_list . ^d Uniquement valeurs du nombre type.									

10.5 Nombre de chiffres significatifs

Nom de l'attribut: **number_of_significant_digits**

Définition de l'attribut: prescription de la précision qui doit être obtenue lors de l'assignation d'une valeur numérique à une propriété

Description:

Conventions et exigences:

- a) Les zéros de queue sans signe décimal indiquent un manque de chiffres significatifs, sauf si un signe décimal est ajouté.

EXEMPLE 1 «1000» a un chiffre significatif et «1000,» a quatre chiffres significatifs.

- b) Tous les chiffres (qui suivent un chiffre autre que zéro) sont significatifs lorsqu'un signe décimal est utilisé, ainsi les zéros de queue situés à droite du signe décimal doivent être considérés comme significatifs. Les zéros avec des chiffres significatifs de chaque côté du signe décimal sont également significatifs.

EXEMPLE 2 «30,4», «34,0», et «3,40» ont chacun trois chiffres significatifs mais «340» a seulement deux chiffres significatifs.

- c) Les zéros de tête ne sont pas traités comme étant significatifs.

EXEMPLE 3 Un ou plusieurs zéros de tête ne sont pas traités comme significatifs, p. ex. 00257,7 a quatre chiffres significatifs et 0,00492 a 6 chiffres significatifs.

- d) Le signe du nombre est indépendant et n'est pas compté dans les chiffres significatifs.

EXEMPLE 4 «+30,4», «-34,0», et «3,40» ont chacun trois chiffres significatifs.

Obligation:

facultatif

Type de caractère des valeurs:

Voir l'Article 26.

10.6 Identificateur de classe référencé

Nom de l'attribut:

referenced_class_identifier

Définition de l'attribut:

lien vers un objet d'information **Item_class**

Description:

La classe référencée contient un ensemble d'objets d'information de **Property**. L'objet d'information **Property** à partir duquel la référence est faite doit avoir le **value_format** **CLASS_INSTANCE_TYPE** ou **CLASS_REFERENCE_TYPE** (voir 10.4.4 et 10.4.5).

Obligation:

facultatif

Type de caractère des valeurs:

Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

10.7 Unité de mesure

Nom de l'attribut: **unit_of_measure**

Définition de l'attribut: prescription de l'unité dans laquelle il convient que la valeur d'une propriété soit exprimée

Description: Conventions et exigences:

- a) La préférence doit être donnée à des unités SI sans préfixes multiplicateurs décimaux comme défini dans les séries IEC 80000 et ISO 80000.
- b) Les autres unités peuvent être spécifiées au moyen du code de l'unité spécifié dans l'IEC TS 62720 ou dans les dictionnaires d'unités conformes.

NOTE 1 Les dictionnaires peuvent être fournis sous forme interprétable par ordinateur ou en format papier.

- c) La conversion en une unité différente ne fait pas partie du domaine d'application du présent document.
- d) La prescription de l'utilisation d'une unité spécifique dans un but de transaction incluant de possibles préfixes ne fait pas partie du domaine d'application du présent document.
- e) Il est fortement recommandé de ne pas spécifier le même concept dans différents objets d'information de **Property** si seuls les préfixes des unités associées sont différents. Des exceptions peuvent se produire si certaines unités sont répandues en particulier dans le domaine des affaires, tel que nm (nanomètre) dans la fabrication des semiconducteurs ou kV (kilovolt) dans le domaine de la distribution de puissance.
- f) Il est exigé que chaque unité contenue dans l'attribut **unit_of_measure** soit associée à une représentation en chaîne, dont la représentation textuelle peut être utilisée pour caractériser l'unité au niveau de l'instance.

NOTE 2 Les objets d'information **Property** avec un **data_type** «CLASS_REFERENCE_TYPE» ne comportent pas d'unité.

EXEMPLE L'IRDI de l'unité «%» est "0112/2///62720#UAA000#01"³.

Obligation: conditionnel

Condition: **unit_of_measure** doit être spécifié pour les objets d'information de **Property** quantitatifs

Type de caractère des valeurs: Voir l'Article 26.

10.8 Format de valeur

Nom de l'attribut: **value_format**

³ L'utilisation des identificateurs des unités n'est pas restreinte à l'IEC CDD mais peut exister en dehors de l'IEC CDD.

Définition de l'attribut: spécification du type et de la longueur de la représentation de la valeur d'une **Property**

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

Commentaires Le **value_format** est spécifié comme faisant partie d'une **Property** et il peut être utilisé au niveau de la mise en œuvre pour assurer la cohérence et la vérification des erreurs dans les échanges électroniques.

Au niveau de la mise en œuvre, par exemple dans la mise en place des bases de données d'informations produit ou d'échanges électroniques, il peut être traité comme un format par défaut qui peut être utilisé tel quel, redéfini (restreint ou étendu) ou ignoré totalement en fonction des exigences et du contexte dans lequel l'information est utilisée.

Pour les objets d'information de **Property** dont le **data_type** a la valeur «class_reference_type», aucun **value_format** n'est demandé.

Voir l'Annexe G pour la spécification de la syntaxe et du contenu de **value_format**.

Une vue d'ensemble du contenu de **value_format** est donnée ci-dessous:

a) Types de **value_format** non-quantitatifs:

- A = alphabétique, lettres uniquement;
- M = mixte, tous caractères autorisés;
- N = numérique, chiffres uniquement;
- X = alphanumérique;
- B = binaire, 0 ou 1.

b) Types de **value_format** quantitatifs:

- NR1 = entiers;
- NR2 = nombres rationnels avec signe-décimal (nombres réels);
- NR3 = nombres rationnels avec signe-décimal et signe-exposant (nombres à virgule flottante);
- S = signe (nombres positifs ou négatifs);
- . = signe-décimal;
- E = signe-exposant, base 10: (A)E(B) représente la valeur $A \times 10^B$.

c) Longueur du champ

La longueur du champ d'une valeur non quantitative doit être indiquée par un nombre (par exemple 17). Une longueur de champ variable doit commencer par deux points. Les formats privilégiés suivants ont été définis:

A..3	N..3	X..3	M..3	B 1
A..8	N..8	X..8	M..8	
A..17	N..17	X..17	M..17	
A..35	N..35	X..35	M..35	
A..($n \times 35$)	N..($n \times 35$)	X..($n \times 35$)	M..($n \times 35$)	

Une longueur de champ fixe commence par un espace. Dans ces formats, aucun caractère spécial ne doit exister.

EXEMPLE 1 «A 3», «N 8», «X 17», «M 35»

La longueur de champ d'une valeur quantitative doit être indiquée par une combinaison de chiffres et de caractères (par exemple «3.3ES2»). Les formats normalisés privilégiés suivants ont été définis:

NR1..4
 NR1 S..4
 NR2..3.3
 NR2 S..3.3
 NR3..3.3ES2
 NR3 S..3.3ES2

Une longueur de champ fixe commence par un espace. Dans ces formats, aucun caractère spécial ne doit exister.

EXEMPLE 2 «NR1 4», «NR1 S 4»

11 Propriété_de_condition

11.1 Spécification de l'objet d'information

Nom: **Condition_property**

Définition: objet d'information de propriété qui affecte la valeur d'une autre propriété.

Description: Toute instance de **Condition_property** doit contenir la valeur «CONDITION_DET» dans l'attribut **product_data_element_type** (hérité).

De nombreuses propriétés ont des valeurs qui dépendent de la ou des valeurs d'un ou plusieurs objets d'information de **Property** supplémentaires.

Lorsque la valeur d'une condition est donnée sous forme d'une plage, elle doit être spécifiée par deux objets d'information **Condition_property** représentant les limites supérieure et inférieure de cette plage.

NOTE La **Condition_property** est un sous-type de la **Property**. Elle hérite donc de toutes les relations avec les classes d'attributs du supertype, c.-à-d. les attributs d'identification, les attributs sémantiques, les attributs administratifs, les attributs de valeur et les données de traduction.

La Figure 9 représente le modèle détaillé de l'objet d'information «**Condition_property**» utilisant le langage UML comme technique de présentation.

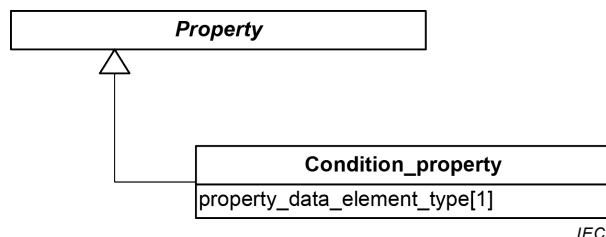


Figure 9 – Propriété condition (diagramme de classe UML)

11.2 Type d'élément de données de propriété

Nom de l'attribut:	property_data_element_type
Définition de l'attribut:	rôle de la Property
Description:	L'un des rôles suivants doit être attribué: CONDITION_DET Property agissant comme condition
Obligation:	obligatoire
Type de caractère des valeurs:	Voir l'Article 26.

12 Propriété de condition dépendante

12.1 Spécification de l'objet d'information

Nom:	Dependent_condition_property
Définition:	condition qui dépend elle-même d'une ou plusieurs autres conditions
Description:	Toute instance de Dependant_condition_property doit contenir la valeur «DEPENDENT_C_DET» dans l'attribut product_data_element_type (hérité). De nombreux objets d'information Condition_property ont des valeurs qui dépendent elles aussi de la ou des valeurs d'une ou plusieurs propriétés supplémentaires. Lorsque la valeur d'un objet d'information Dependent_condition_property est donnée sous forme d'une plage, elle doit être spécifiée par deux objets d'information Dependent_condition_property représentant les limites supérieure et inférieure de cette plage.

NOTE **Dependent_condition_property** est un sous-type de **Condition_property** et de **Property**. Elle hérite donc de toutes les relations avec les classes d'attributs du supertype, c.-à-d. les attributs d'identification, les attributs sémantiques, les attributs administratifs, les attributs de valeur et les données de traduction.

La Figure 10 représente le modèle détaillé de l'objet d'information «**Dependent_condition_property**» utilisant le langage UML comme technique de présentation.

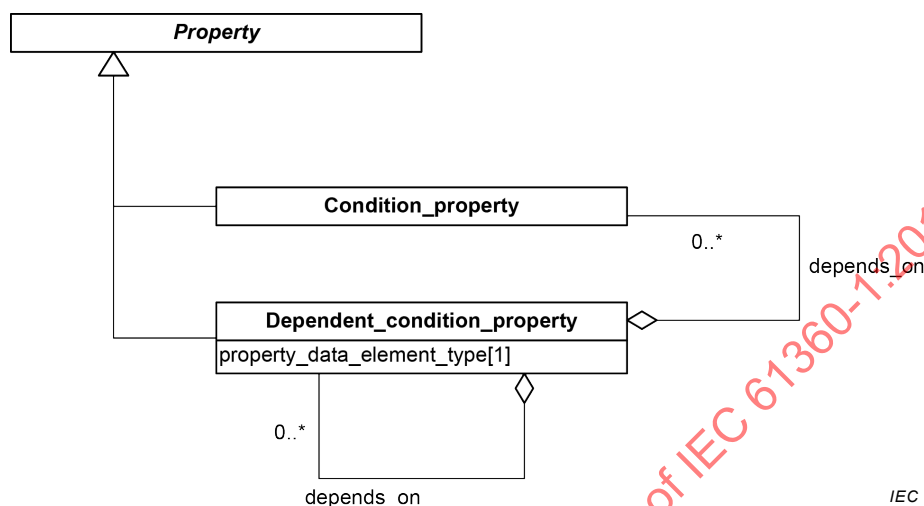


Figure 10 – Propriété condition dépendante
(diagramme de classe UML)

12.2 Dépend de

Nom de l'attribut:

depends_on

Définition de l'attribut:

ensemble d'un ou plusieurs identificateurs d'objets d'information **Condition_property** agissant comme conditions desquelles dépend la caractéristique qui est exprimée par l'objet d'information **Dependent_condition_property**.

Description:

Les conditions sont spécifiées en détail aux Articles 11 et 12

Obligation:

obligatoire

Type de caractère des valeurs:

Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

12.3 Type d'élément de données de propriété

Nom de l'attribut:

property_data_element_type

Définition de l'attribut:

rôle de la **Property**

Description:

L'un des rôles suivants doit être attribué:

DEPENDENT_C_DET

Property agissant comme condition dépendante

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

13 Propriété dépendante

13.1 Spécification de l'objet d'information

Nom: **Dependent_property**

Définition: objet d'information de propriété qui est soumis à des restrictions exprimées par une ou plusieurs propriétés de condition ou propriétés de condition dépendante

Description: Toute instance de **Dependent_property** doit contenir la valeur «DEPENDENT_P_DET» dans l'attribut **product_data_element_type** (hérité).

NOTE **Dependent_property** est un sous-type de **Property**. Elle hérite donc de toutes les relations avec les classes d'attributs du supertype, c.-à-d. les attributs d'identification, les attributs sémantiques, les attributs administratifs, les attributs de valeur et les données de traduction.

La Figure 11 représente le modèle détaillé de l'objet d'information «**Dependent_property**» utilisant le langage UML comme technique de présentation.

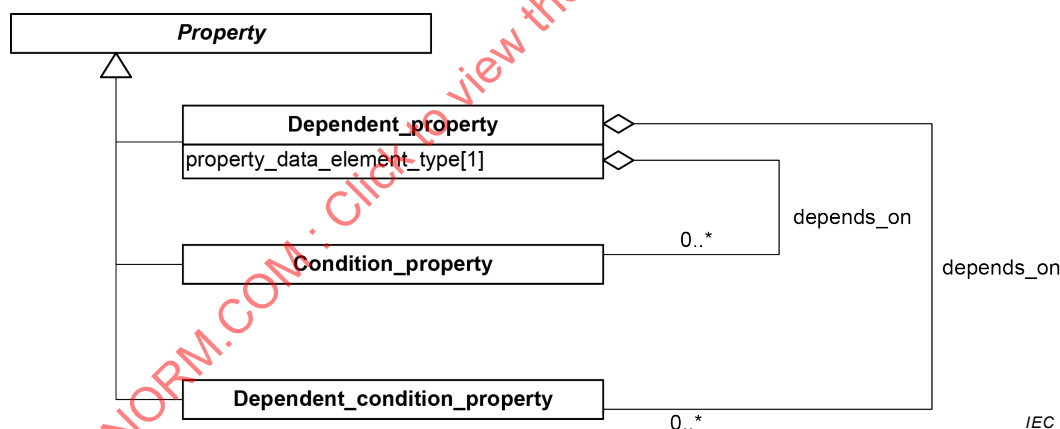


Figure 11 – Propriété dépendante (diagramme de classe UML)

13.2 Dépend de

Nom de l'attribut: **depends_on**

Définition de l'attribut: définition d'un ou plusieurs identificateurs de conditions desquelles dépendent les caractéristiques d'une propriété dépendante ou d'une condition dépendante

Description: Les conditions sont spécifiées en détail aux Articles 11 et 12.

Obligation: obligatoire

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

13.3 Type d'élément de données de propriété

Nom de l'attribut: **property_data_element_type**

Définition de l'attribut: rôle de la **Property**

Description: L'un des rôles suivants doit être attribué:

DEPENDENT_P_DET **Property** agissant comme propriété dépendante

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

14 Propriété_non_dépendante

14.1 Spécification de l'objet d'information

Nom: **Non_dependent_property**

Définition: objet d'information de propriété qui n'est soumis à aucune restriction dans son domaine d'application défini

Description: Toute instance de **Non_dependent_property** doit contenir la valeur «NON_DEPENDENT_P_DET» dans l'attribut **product_data_element_type** (hérité).

NOTE **Non_dependent_property** est un sous-type de **Property**. Elle hérite donc de toutes les relations avec les classes d'attributs du supertype, c.-à-d. les attributs d'identification, les attributs sémantiques, les attributs administratifs, les attributs de valeur et les données de traduction.

La Figure 12 représente le modèle détaillé de l'objet d'information «**Non_dependent_property**» utilisant le langage UML comme technique de présentation.

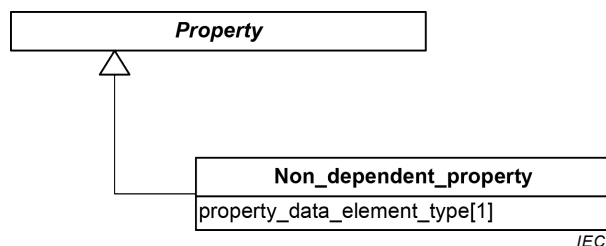


Figure 12 – Propriété non dépendante
(diagramme de classe UML)

14.2 Type d'élément de données de propriété

Nom de l'attribut: **property_data_element_type**

Définition de l'attribut: rôle de la **Property**

Description: L'un des rôles suivants doit être attribué:

NON_DEPENDENT_P_DET **Property** agissant comme propriété non dépendante

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

15 Information_de_traduction

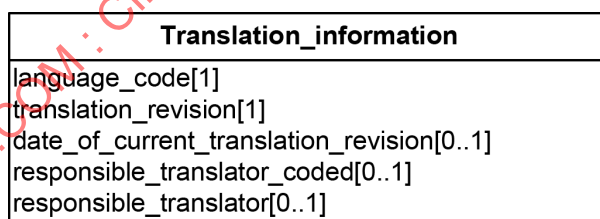
15.1 Spécification de l'objet d'information

Nom: **Translation_information**

Définition: collection d'attributs servant à gérer les objets d'information de contenu localisé tels que **Property** et **Item_class**

Description:

La Figure 13 représente le modèle détaillé de l'objet d'information «**Translation_information**» utilisant le langage UML comme technique de présentation.



IEC

**Figure 13 – Information de traduction
(diagramme de classe UML)**

15.2 Date de la révision de traduction actuelle

Nom de l'attribut: **date_of_current_translation_revision**

Définition de l'attribut: date de la dernière révision de la traduction

EXEMPLE «2005-06-28»

Obligation: facultatif

Type de caractère des valeurs: format de date sous la forme AAAA-MM-JJ comme spécifié par l'ISO 8601

15.3 Code de langue

Nom de l'attribut: **language_code**

Définition de l'attribut: code de la langue dans laquelle le texte traduit est écrit

EXAMPLE «en GB», «en US», «de DE», «de AT»

Description: La valeur du **language_code** doit provenir du code de langue à deux lettres conformément à l'ISO 639-1, suivi par un espace intermédiaire et prolongé par le code pays alpha 2 conformément à l'ISO 3166-1 ou le code de subdivision de pays conformément à l'ISO 3166-2 afin de permettre la distinction entre les variantes particulières d'une même langue.

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

15.4 Traducteur responsable

Nom de l'attribut: **responsible_translator**

Définition de l'attribut: le nom de l'organisme responsable de la traduction identifié par son nom et son code de pays à deux lettres

EXAMPLE «DKE DE»

Description:

Obligation: facultatif

Type de caractère des valeurs: Constante chaîne comme spécifié

15.5 Traducteur responsable codé

Nom de l'attribut: **responsible_translator_coded**

Définition de l'attribut: identification codée de l'organisme responsable de la traduction

EXAMPLE «0112/2»

Description: L'identificateur est spécifié dans l'ISO 13584-26.

Obligation: facultatif

Type de caractère des valeurs: Constante chaîne comme spécifié

15.6 Révision de traduction

Nom de l'attribut:	translation_revision
Définition de l'attribut:	numéro de révision de la traduction correspondante pour administrer les modifications
	EXEMPLE «01»
Obligation:	obligatoire
Type de caractère des valeurs:	Chiffres de «0» à «9».

16 Liste de valeurs

16.1 Spécification de l'objet d'information

Nom:	Value_list
Définition:	ensemble de représentations de valeurs admissibles d'une Property
Obligation:	conditionnel
Condition:	Value_list doit être spécifié pour les objets d'information de propriété de classification. Pour les objets d'information de propriété autre que de classification, Value_list est facultatif.
Description:	Un type de données énumération fait toujours référence à une Value_list pour spécifier l'ensemble des valeurs possibles (voir 10.4.3). Les listes de valeurs qui forment un polymorphisme (voir 24.7) sont considérées comme restrictives car ceux-ci sont exhaustifs et aucune valeur supplémentaire ou modifiée n'est possible. Toutes les autres listes de valeur peuvent être spécifiées comme extensibles par l'utilisateur. Afin d'indiquer explicitement la présence d'une liste de valeurs étendue, l'option «autres» doit être ajoutée à la liste de valeurs en question pour indiquer la liste de valeurs étendue. Les options spécifiant la signification des «autres» doivent être ajoutées en condition de la propriété qui référence la liste de valeurs. Cette condition doit contenir les options supplémentaires sous forme de type de données énumération. Lorsque les options supplémentaires ne sont pas spécifiées, cette énumération reste vide.

NOTE 1 Dans ce cas, la propriété de type énumération (d'origine) est spécifiée par **DEPENDENT_P_DET**. Ces entrées peuvent être intégrées à la liste de valeurs pour une future publication du dictionnaire.

La spécialisation de propriété telle que présentée dans l'IEC 62656-1:2014, 5.4.5, est une autre technique de spécification des propriétés avec des listes de valeurs différentes de celles contenues dans l'ontologie fournie par le fournisseur original du dictionnaire. Cette technique évite toute interférence avec le contenu délivré par le fournisseur original.

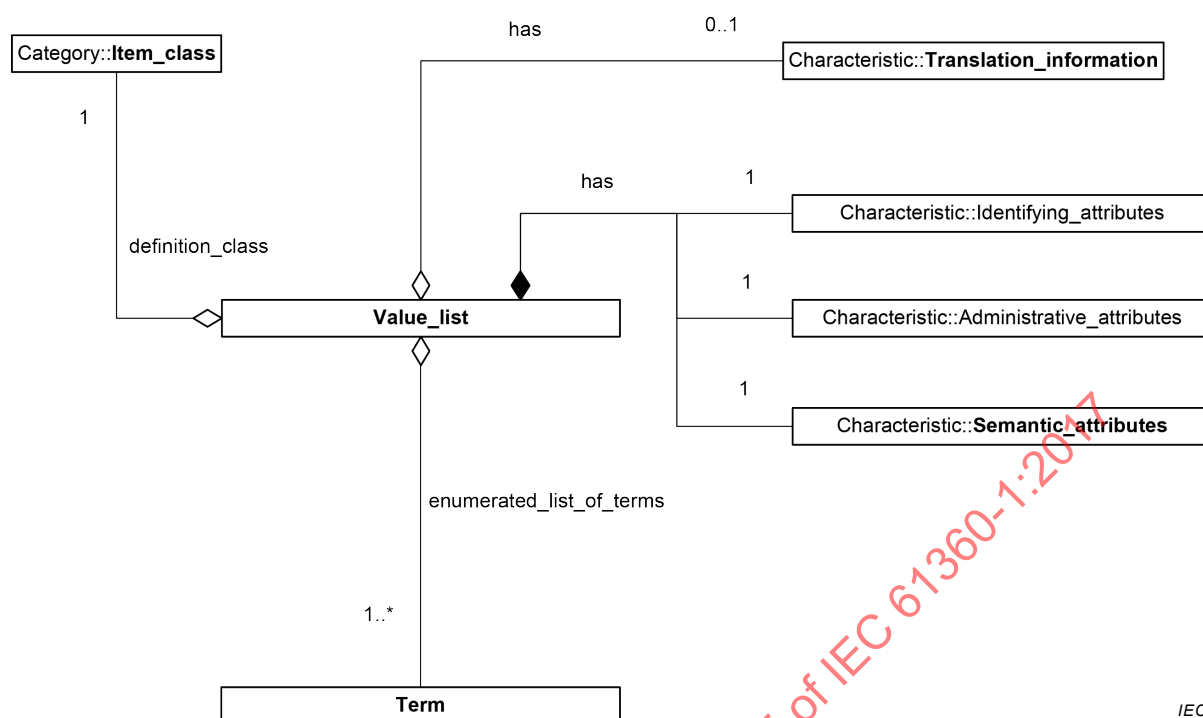
NOTE 2 Lorsque les listes de valeurs doivent être rétrécies, les objets d'information **Relation** peuvent être utilisés (voir 24.8.2).

NOTE 3 Lorsqu'il convient qu'une liste de valeurs soit définitivement étendue, la plage de valeurs peut être étendue par révision de l'objet d'information d'énumération concerné.

NOTE La **Value_list** peut être attribuée à une **Property** par le biais de l'attribut **data_type**. Dans le cas d'une liste de valeurs assignée, cet attribut contient l'un des types de données énumération qui comprennent la référence à la **Value_list** (voir 10.4.3).

La Figure 14 représente le modèle détaillé de l'objet d'information «**Value_list**» utilisant le langage UML comme technique de présentation.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017



IEC

Anglais	Français
Category::Item_class	Catégorie::Classe_d_article
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Characteristic::Identifying_attributes	Caractéristique::Attributs_d_identification
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
Characteristic::Semantic_attributes	Caractéristique::Attributs_sémantiques
definition_class	classe_de_définition
has	possède
Value_list	Liste de valeurs
enumerated_list_of_terms	liste_d_articles_énumérés
Term	Terme

Figure 14 – Liste de valeurs (diagramme de classe UML)

16.2 Attributs internes de Value_list

Value_list possède des attributs qui sont regroupés pour une meilleure lisibilité dans les objets d'information

- **Identifying_attributes;**
- **Semantic_attributes;**
- **Administrative_attributes;**
- **Translation_information.**

Ces conteneurs d'attributs sont relatifs à **Value_list** par la relation «**possède**». Les objets d'information doivent être conservés avec la classe de référence, c.-à-d. qu'ils sont internes aux instances de **Value_list**.

16.3 Classe de définition

Nom de l'attribut:	definition_class
Définition de l'attribut:	classe dans laquelle l'objet d'information est défini
Description:	La classe référencée par definition_class spécifie le domaine d'application des objets d'information Value_list . En dehors de ce domaine d'application, les instances ne sont pas valides.
Obligation:	obligatoire
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

16.4 Liste d'articles énumérés

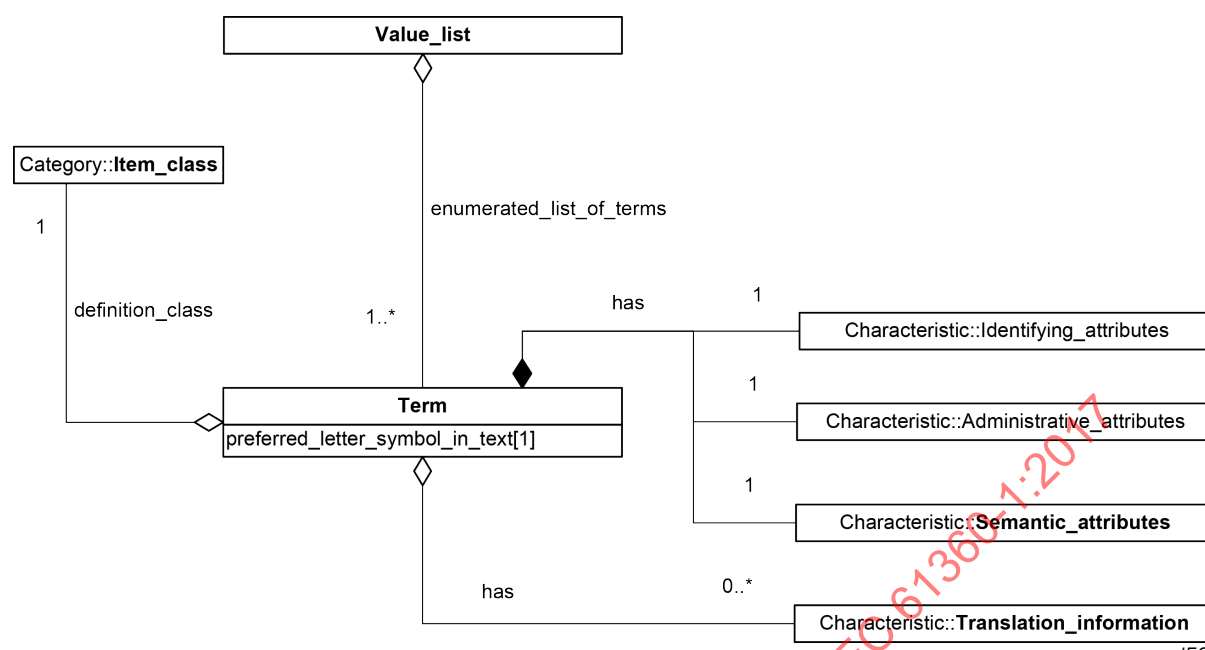
Nom de l'attribut:	enumerated_list_of_terms
Définition de l'attribut:	ensemble de références à des objets d'information qui constituent dans leur totalité le contenu de la liste de valeurs
Description:	
Obligation:	obligatoire
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

17 Terme

17.1 Spécification de l'objet d'information

Nom:	Term
Définition:	représentation d'une valeur admissible d'un élément de value_list
Description:	

La Figure 15 représente le modèle détaillé de l'objet d'information «**Term**» utilisant le langage UML comme technique de présentation.



Anglais	Français
Category::Item_class	Catégorie::Classe_d_article
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Characteristic::Identifying_attributes	Caractéristique::Attributs_d_identification
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
Characteristic::Semantic_attributes	Caractéristique::Attributs_sémantiques
definition_class	classe_de_définition
has	possède
Value_list	Liste de valeurs
enumerated_list_of_terms	liste_d_articles_énumérés
Term	Terme
Preferred_letter_symbol_in_text(1)	Symbole_littéral_préférédans_texte(1)

Figure 15 – Terme (diagramme de classe UML)

17.2 Symbole littéral préféré dans texte

Nom de l'attribut: **preferred_letter_symbol_in_text**

Définition de l'attribut: représentation d'une valeur admissible dans une **Value_list**

Description: S'il est utilisé comme code de valeur d'une propriété de classification, ce code de valeur doit être identique au nom codé de la **Item_class** qui est créée par cette propriété de classification.

Le code de valeur d'une **Classifying_property** doit avoir une longueur maximale de 18 caractères alphanumériques.

Obligation: obligatoire

Type de caractère des Voir l'Article 26.
valeurs:

17.3 Attributs internes de Term

Un **Term** possède des attributs qui sont regroupés pour une meilleure lisibilité dans les objets d'information:

- **Identifying_attributes;**
- **Semantic_attributes;**
- **Administrative_attributes;**
- **Translation_information.**

Ces conteneurs d'attributs sont associés au **Term** par la relation «**possède**». Les objets d'information doivent être conservés avec la classe de référencement, c.-à-d. qu'ils sont internes aux instances du **Term**.

17.4 Classe de définition

Nom de l'attribut: **definition_class**

Définition de l'attribut: classe dans laquelle l'article est défini

Description: La classe référencée par la **definition_class** spécifie le domaine d'application de l'objet d'information **Term**. En dehors de ce domaine d'application, les instances ne sont pas valides.

Obligation: obligatoire

Type de caractère des Lettres majuscules de l'alphabet latin de «A» à «Z» à
valeurs: l'exception des lettres majuscules «O» et «I» de l'alphabet latin
plus les chiffres «0» à «9».

18 Dessin

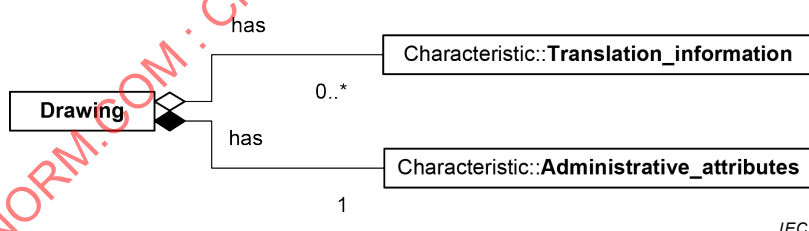
18.1 Modèle d'information

L'Article 18 explicite les différents attributs d'un **Drawing**. Voir le Tableau 5 pour la liste de ces attributs. Ces attributs sont relatifs à l'identification des dessins.

Tableau 5 – Liste des attributs de Drawing

Noms d'attributs dans l'IEC 61360-1	Noms d'attributs dans l'IEC 61360-2	Numéro de paragraphe
code	Code	18.3
date_of_current_translation_revision	a	15.2
descriptive_designator	a	18.4
drawing_title	a	18.5
file_format	a	18.6
file_name	a	18.7
language_code	a	15.3
obsolete_date	a	9.3
proposal_date	date_of_original_definition	9.4
published_by	a	9.6
published_in	a	9.5
responsible_translator	a	15.4
responsible_translator_coded	a	15.5
revision_number	Revision	18.8
status_level	a	9.9
translation_revision	a	15.6
version_initiated_on	a	9.10
version_number	Version	18.9
version_released_on	date_of_current_version	9.11
^a Non défini dans l'IEC 61360-2.		

La Figure 16 représente le modèle d'ensemble du module «Drawing concept» utilisant le langage UML comme technique de présentation.



Anglais	Français
Drawing	Dessin
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
has	possède

Figure 16 – Modèle d'ensemble du concept de dessin (diagramme de classe UML)

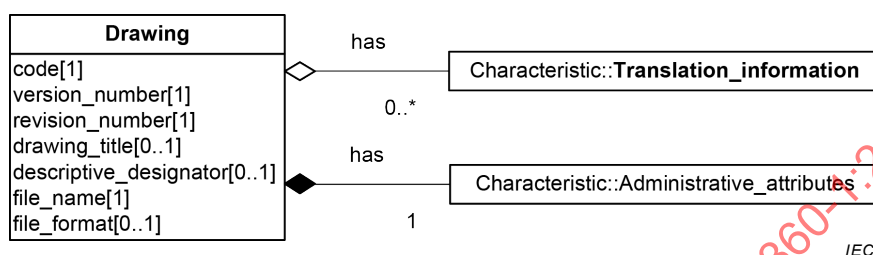
18.2 Spécification de l'objet d'information

Nom: **Drawing**

Définition: collection d'attributs d'identification et administratifs permettant la gestion et l'utilisation de données contenant des informations picturales

Description:

La Figure 17 représente le modèle détaillé de l'objet d'information «**Drawing**» utilisant le langage UML comme technique de présentation.



Anglais	Français
Drawing	Dessin
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
has	possède
code(1)	code(1)
version_number(1)	numéro_de_version(1)
revision_number(1)	numéro_de_révision(1)
drawing_title(0...1)	titre_de_dessin(0...1)
descriptive_designator(0...1)	indicateur_descriptif(0...1)
file_name(1)	nom_de_fichier(1)
file_format(0...1)	format_de_fichier(0...1)

Figure 17 – Dessin (diagramme de classe UML)

18.3 Code

Nom de l'attribut: **code**

Définition de l'attribut: code unique à six caractères pour un **Drawing**

NOTE L'IEC 61360-2:2012 spécifie une longueur maximale de l'attribut **code** de 14 caractères. La différence est due à l'information ajoutée au code, telle que l'identificateur international de données d'enregistrement (voir l'ISO/IEC 11179-6:2005).

Description: Les trois premiers caractères doivent être alphabétiques, les trois derniers numériques (format AAANNN). Le caractère «X»⁴ ne doit pas être utilisé comme premier caractère. Les codes sont émis séquentiellement et il convient qu'ils n'aient pas de relation avec la signification du dessin.

⁴ Pour une utilisation locale, il est admis d'utiliser des codes commençant par XAA et allant jusqu'à XZZ inclus et ceux-ci sont par conséquent exclus de l'ensemble des dessins normalisés.

Obligation: obligatoire

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

18.4 Indicateur descriptif

Nom de l'attribut: **descriptive_designator**

Définition de l'attribut: représentation codée du **drawing_title**

Description: l'indicateur descriptif est utilisé pour identifier le contenu d'un **Drawing** particulier.

Pour assurer la cohérence et la compréhension du **descriptive_designator**, des systèmes de codage peuvent être utilisés. L'exemple décrit l'utilisation d'un système de codage pour spécifier le contenu des dessins de boîtiers de dispositifs à semiconducteurs.

EXEMPLE L'indicateur descriptif présente la forme **XXY-Z-Tnnn-Bmmm** et les caractères ont les significations suivantes:

- X indique la disposition des bornes, définie par la propriété de classification applicable;
- YY indique le style du composant du dispositif à semiconducteurs, défini par la propriété de classification applicable;
- Z indique la forme des bornes, définie par la propriété de classification applicable;
- Tnnn est le code de variante de borne, défini par la propriété de classification applicable;
- Bmmm est le code de variante du corps facultatif, défini par la propriété de classification applicable.

Le **descriptive_designator** d'un **Drawing** doit avoir une longueur maximale de 17 caractères alphanumériques.

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

18.5 Titre du dessin

Nom de l'attribut: **drawing_title**

Définition de l'attribut: légende descriptive d'un ou plusieurs mots appliquée à un dessin

Description: Le **drawing_title** doit comporter un maximum de 70 caractères

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

18.6 Format de fichier

Nom de l'attribut: **file_format**

Définition de l'attribut: le format de données dans lequel le dessin est représenté

Description: Plusieurs formats de données sont autorisés pour un dessin ⁵

Si le format de fichier spécifié par l'attribut **file_format** contredit l'information de format de fichier dérivée de l'attribut **file_name**, le format donné dans l'attribut **file_format** doit être prioritaire.

Obligation: obligatoire

18.7 Nom de fichier

Nom de l'attribut: **file_name**

Définition de l'attribut: le nom du fichier sur un support qui contient les données du **Drawing**

Description: Le **file_name** doit être complet, y compris l'extension du **file_name**.

Si le fichier est stocké sur un serveur à distance, le chemin complet de l'URL (y compris le **file_name**) doit être fourni.

Obligation: obligatoire

18.8 Numéro de révision

Nom de l'attribut: **revision_number**

Définition de l'attribut: numéro utilisé pour le contrôle administratif d'un **Drawing**

⁵ Il est de la responsabilité de l'émetteur d'un dessin de clarifier le format de données utilisé. Les formats possibles incluent les formats bitmap comme BMP, TIFF, GIF et JPEG, les formats vectoriels comme CGM, DXF et HP-GL et les formats document tel que PDF.

Description: Il convient que le **revision_number** d'un dessin soit composé de 3 chiffres. Bien que le modèle d'information commun EXPRESS ISO/IEC autorise un nombre plus élevé de chiffres, pour que le dictionnaire soit enregistré dans l'IEC CDD ou soit compatible avec la série IEC 61360, il convient de n'utiliser que 3 chiffres. Il est admis d'utiliser un plus grand nombre de chiffres à des fins spécifiques. Les numéros des révisions consécutives commençant par «000» doivent être donnés dans l'ordre croissant. Le **Drawing** étant unique de par son identificateur, un seul numéro de révision est valable à un moment donné.

Un nouveau numéro de révision d'un **Drawing** doit être généré si le dessin est modifié sans que cela n'affecte ni son utilisation ni sa signification ou lorsque des corrections de fautes de frappe et d'orthographe sont apportées.

Le **revision_number** doit être ramené à sa valeur de départ «000» lorsque le **version_number** est modifié.

Obligation: obligatoire

Type de caractère des valeurs: Chiffres de «0» à «9».

18.9 Numéro de version

Nom de l'attribut: **version_number**

Définition de l'attribut: numéro utilisé pour indiquer les versions d'un **Drawing** au cours de son cycle de vie

Description: Il convient que le **version_number** d'un dessin soit composé de 3 chiffres. Bien que le modèle d'information commun EXPRESS ISO/IEC autorise un nombre plus élevé de chiffres, pour que le dessin soit enregistré dans l'IEC CDD ou soit compatible avec la série IEC 61360, il convient de n'utiliser que 3 chiffres. Il est admis d'utiliser un plus grand nombre de chiffres à des fins spécifiques. Les numéros des versions consécutives commençant par «000» doivent être donnés en suivant l'ordre croissant. Un seul **version_number** par **Drawing**, unique de par son identificateur, est valable à un moment donné.

Le **version_number** d'un **Drawing** doit être modifié lorsque celui-ci subit une modification importante comme l'addition d'un nouveau paramètre de dimension. Il convient qu'une modification d'une version d'un **Drawing** donne lieu à une modification de la version de l'objet d'information qui fait référence au **Drawing**.

Obligation: obligatoire

Type de caractère des valeurs: Chiffres de «0» à «9».

18.10 Attributs internes de Drawing

Un **Dessin** possède des attributs qui sont regroupés pour une meilleure lisibilité dans les objets d'information:

- **Administrative_attributes**; et
- **Translation_information**.

Ces conteneurs d'attributs sont associés au **Drawing** par la relation «possède». Les objets d'information doivent être conservés avec la classe de référence, c.-à-d. qu'ils sont internes aux instances du **Drawing**.

19 Unité de mesure

19.1 Vue d'ensemble

Les objets d'information de **Property** quantitatifs exigent des informations sur leur grandeur physique et la ou les unités applicables. Sans unité de mesure, les valeurs assignées n'ont aucune signification et de sérieuses erreurs peuvent se produire.

L'attribut **data_element_type_class** (voir 6.3) spécifie la grandeur physique de l'objet d'information **Property**. Ainsi, la présélection d'unités appropriées est prise en charge, et l'identification et l'assignation de l'unité de mesure sont très simplifiées.

NOTE 1 L'Annexe B énumère l'utilisation et les codes de propriétés de classification disponibles.

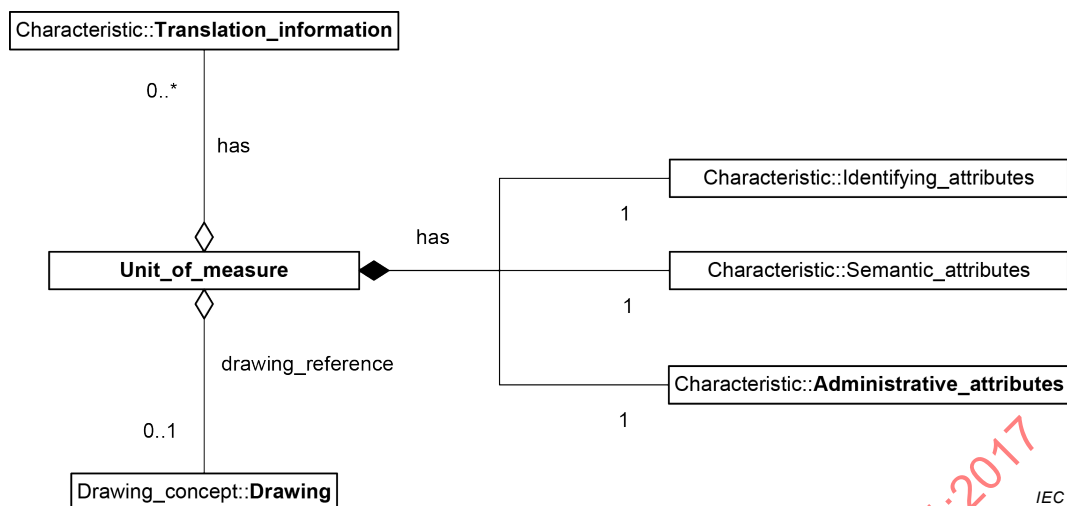
L'unité de mesure peut être spécifiée au moyen des codes spécifiés dans l'IEC TS 62720 ou dans les dictionnaires d'unités conformes.

NOTE 2 L'IEC tient à jour dans l'IEC CDD une collection conforme à l'IEC TS 62720 d'environ 1 500 unités regroupées en grandeurs physiques facilement disponibles, à utiliser avec les objets d'information **Property**. Les attributs de **Property code_of_unit** et **code_of_alternative_unit** permettent de référencer ces objets d'information.

De plus, un système de grandeurs est disponible par le biais de **Relation** et de son attribut **role_of_the_relation** = QUANTITY (voir 22.7). Les systèmes d'unités existants peuvent ainsi être mis en œuvre.

NOTE 3 Le ou les systèmes d'unités sont actuellement mis en œuvre par le biais de l'objet d'information de l'IEC CDD «Liste d'unités» qui est un prédécesseur de l'objet d'information **Relation** référencé ci-dessus.

NOTE 4 Bien qu'ils soient conçus pour faire partie intégrante des environnements conformes à la série IEC 61360, les objets d'information instanciés **Unit_of_measure** peuvent également être référencés par le biais d'environnements IEC autres que conformes à la série IEC 61360 afin de fournir des informations non ambiguës sur l'unité de mesure. Figure 18 représente le modèle d'ensemble du module «Measure concept» utilisant le langage UML comme technique de présentation.



Anglais	Français
Characteristic::Identifying_attributes	Caractéristique::Attributs_d'identification
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Characteristic::Semantic_attributes	Caractéristique::Attributs_sémantiques
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
Unit_of_measure	Unité_de_mesure
has	possède
drawing_reference	drawing_reference
Drawing_concept::Drawing	Concept_de_dessin::Dessin

Figure 18 – Modèle d'ensemble du concept de mesure (diagramme de classe UML)

19.2 Spécification de l'objet d'information

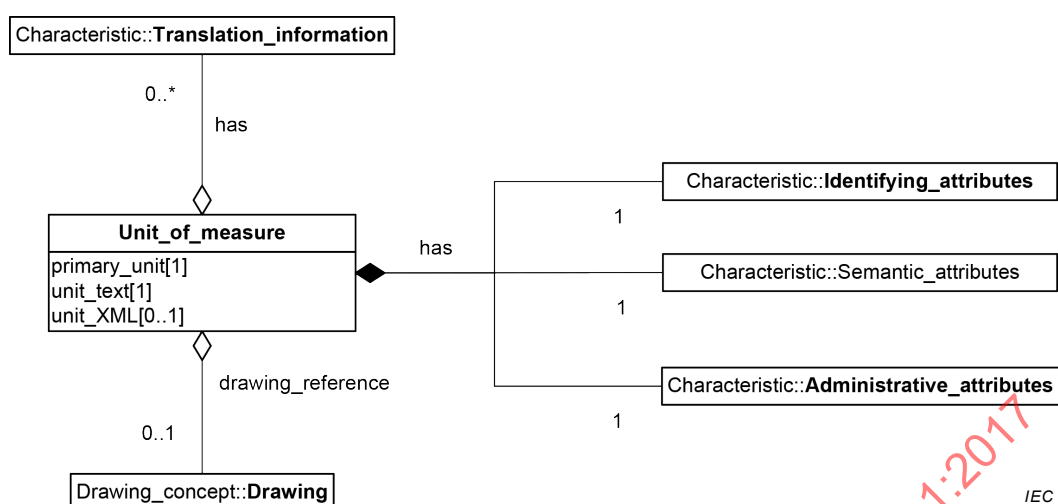
Nom: **Unité_de_mesure**

Définition: grandeur définie d'une grandeur physique, définie et adoptée par convention ou par la loi, utilisée comme norme pour la mesure de la même grandeur physique

EXEMPLE mètre, seconde, degré Celsius, etc.

Description:

La Figure 19 représente le modèle détaillé de l'objet d'information «**Unit_of_measure**» utilisant le langage UML comme technique de présentation.



IEC

Anglais	Français
Characteristic::Identifying_attributes	Caractéristique::Attributs_d_identification
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Characteristic::Semantic_attributes	Caractéristique::Attributs_sémantiques
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
Unit_of_measure	Unité_de_mesure
primary_unit(1)	unité_principale(1)
unit_text(1)	unité_en_texte(1)
unit_XML(0...1)	unité_XML(0...1)
has	possède
drawing_reference	drawing_reference
Drawing_concept::Drawing	Concept_de_dessin::Dessin

Figure 19 – Unité de mesure (diagramme de classe UML)

19.3 Unité principale

Nom de l'attribut: **primary_unit**

Définition de l'attribut: unité exprimée en unités de base SI

Description:

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

19.4 Unité en texte

Nom de l'attribut: **unit_in_text**

Définition de l'attribut: présentation textuelle de l'unité

Description: Les unités peuvent comprendre des formules mathématiques sophistiquées; l'expression présentée ici peut donc dévier de la présentation mathématique correcte de l'unité

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

19.5 Unité XML

Nom de l'attribut: **unit_XML**

Définition de l'attribut: présentation de l'unité exprimée en code xml

Description:

Obligation: facultatif

Type de caractère des valeurs: caractères conformes à la spécification xml

19.6 Attributs internes de Unit_of_measure

Une **Unit_of_measure** possède des attributs qui sont regroupés pour une meilleure lisibilité dans les objets d'information:

- **Administrative_attributes;**
- **Identifying_attributes;**
- **Semantic_attributes;**
- **Translation_information.**

Ces conteneurs d'attributs sont relatifs à l'**Unit_of_measure** par la relation «possède». Les objets d'information doivent être conservés avec la classe de référence, c.-à-d. qu'ils sont internes aux instances de l'**Unit_of_measure**.

19.7 Référence de dessin

Nom de l'attribut: **drawing_reference**

Définition de l'attribut: référence à l'identificateur d'un dessin qui contient une représentation picturale du concept

Commentaires

Obligation: facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

20 Création de variantes de langues

20.1 Vue d'ensemble

La Figure 13 représente les attributs qui contiennent des informations administratives supplémentaires à propos des propriétés ou classes localisées. Ces attributs doivent être tenus à jour pour chaque objet d'information localisé.

Un objet d'information dans une langue autre que l'anglais doit seulement être préparé si la version anglaise de référence précédente est disponible.

NOTE A cause des retards dans l'élaboration des versions nationales, les utilisateurs peuvent se retrouver face à des incohérences entre la version anglaise de référence et les différentes langues supplémentaires.

20.2 Attributs d'une **Property** dépendant de la langue

Les attributs suivants d'une **Property** dépendent de la langue et peuvent être traduits:

- **definition**,
- **note**,
- **preferred_name**,
- **remark**,
- **short_name**,
- **synonymous_name**.

20.3 Attributs d'une **Item_class** dépendant de la langue

Les attributs suivants d'une **Item_class** dépendent de la langue et peuvent être traduits:

- **definition**,
- **note**,
- **preferred_name**,
- **remark**,
- **short_name**,
- **synonymous_name**.

20.4 Attributs d'un **Drawing** dépendant de la langue

Les attributs suivants d'un **Drawing** dépendent de la langue et peuvent être traduits:

- **drawing_title**.

20.5 Attributs d'une **Unit_of_measure** dépendant de la langue

Les attributs suivants d'une **Unit_of_measure** dépendent de la langue et peuvent être traduits:

- **definition**,
- **note**,
- **preferred_name**,
- **remark**,
- **short_name**,
- **synonymous_name**.

20.6 Attributs d'une Relation dépendant de la langue

Les attributs suivants d'une **Relation** dépendent de la langue et peuvent être traduits:

- **definition**,
- **note**,
- **preferred_name**,
- **remark**,
- **short_name**,
- **synonymous_name**.

21 Classe_d_article

21.1 Arbre de classification

Pour la classification des articles, le principe de la division d'un ensemble d'articles en sous-ensembles peut être appliqué de façon répétée, ce qui permet de créer un arbre hiérarchique de classes utilisant ce que l'on appelle les structures «est-un». L'arbre doit commencer par la classe racine et comprend plusieurs superclasses et sous-classes.

Les principes de classification sont les suivants:

- il convient qu'une superclasse comprenne au moins deux sous-classes;
- il convient qu'une superclasse ait une propriété de classification définissant ces sous-classes;
- une sous-classe doit posséder exactement une superclasse;
- une sous-classe devient elle-même une superclasse lorsqu'elle possède des sous-classes.

La classe de base (classe racine) d'un arbre de classification n'a pas de superclasse. Cette classe n'a que des objets d'information de sous-classe. Cette classe de base doit porter la valeur «UNIVERSE» comme **code** et ne doit ni être impliquée dans une autre relation, ni référencer un autre objet d'information.

NOTE 1 L'union de deux nœuds d'univers reste un univers.

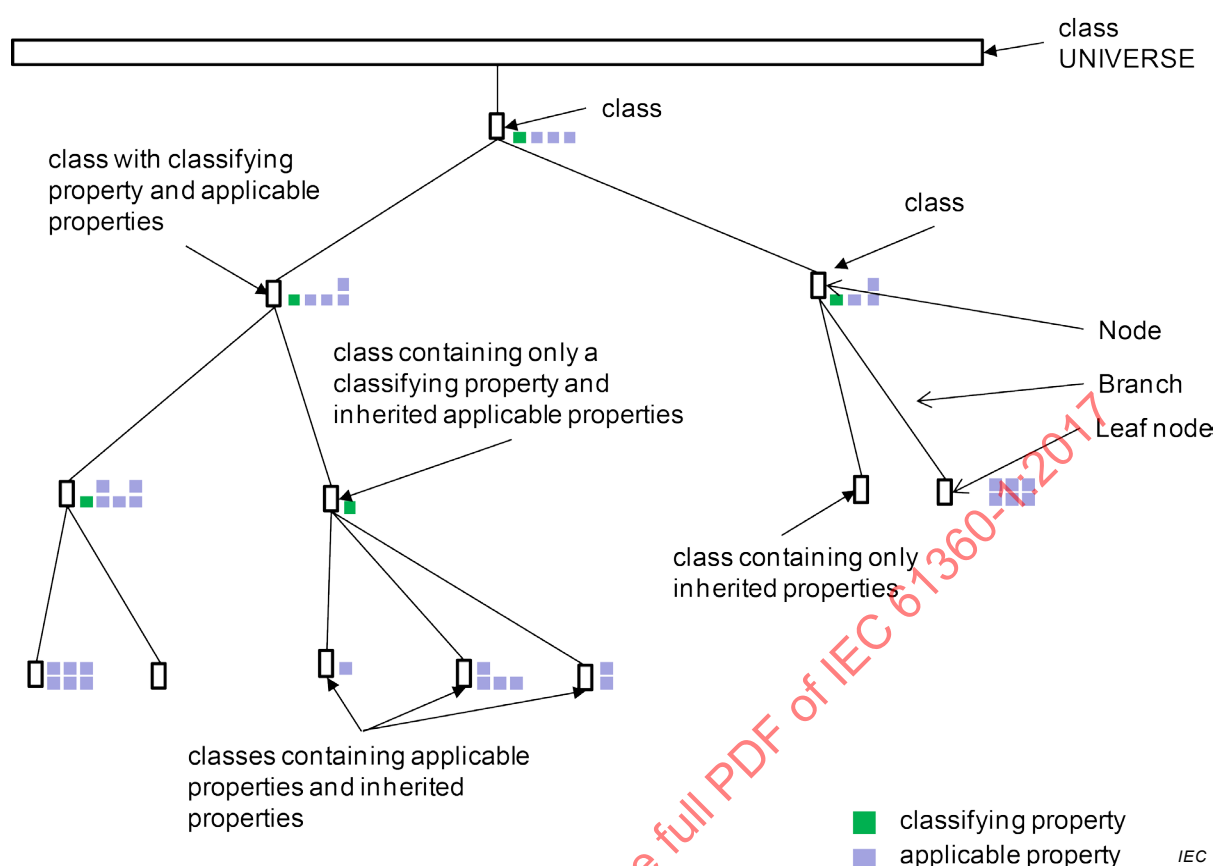
NOTE 2 Universe est une classe virtuelle signifiant la Classe universelle en logique mathématique: la classe de toutes les classes.

L'objectif d'une classification est de diviser l'ensemble d'articles confus dans le domaine examiné en ensembles plus petits et plus faciles à gérer qui peuvent être étudiés séparément.

EXEMPLE L'une des classifications les plus réussies toujours utilisées est la classification du règne animal définie par Carl Linné en 1735.

Une propriété qui est applicable à une classe doit également être applicable à toutes les sous-classes de cette classe. Les propriétés qui ne sont applicables qu'à un nombre limité de sous-classes doivent être spécifiées dans la ou les sous-classes concernées.

Une structure en arbre est ainsi créée avec des superclasses et des sous-classes, comme représenté à la Figure 20. Sur cette figure, chaque classe représente sous forme de points la collection des propriétés applicables à cette classe. L'ensemble des propriétés applicables à une classe inclut toutefois également les propriétés héritées de classes de niveau supérieur (superclasses). Ainsi, lorsqu'une classe est représentée vide, elle inclut néanmoins tous les objets d'information de propriété hérités.



Anglais	Français
class UNIVERSE	Classe UNIVERSE
class	classe
class with classifying property and applicable properties	Classe avec propriété de classification et propriétés applicables
class containing only a classifying property and inherited applicable properties	Classe contenant uniquement une propriété de classification et les propriétés applicables héritées
Node	Nœud
Branch	Branche
Leaf node	Nœud de feuille
class containing only inherited properties	Classe contenant uniquement les propriétés héritées
classes containing applicable properties and inherited properties	Classes contenant les propriétés applicables et les propriétés héritées
classifying property	propriété de classification
applicable property	propriété applicable

Figure 20 – Arbre de classification

La classification des articles décrite ici est établie selon les indications ci-dessous.

- L'ordre hiérarchique des classes est déterminé par spécialisation, ce qui signifie qu'une classe comprend toujours sa ou ses sous-classes de niveau inférieur suivant. La classification s'arrête au niveau auquel aucune autre division des articles ne doit être réalisée.
- Parmi la collection de propriétés applicables à un nœud particulier, une propriété est la propriété de classification.

- c) Les propriétés de classification, applicables à toutes les classes comprenant des sous-classes de l'arbre, doivent présenter les caractéristiques suivantes:
- chacune correspond à un attribut élémentaire unique et non à une combinaison de différents attributs, c'est-à-dire qu'ils possèdent des listes de valeurs homogènes;
 - les sous-classes désignées par les valeurs d'une propriété de classification sont complémentaires; elles couvrent la totalité du domaine spécifié par leur superclasse sans se chevaucher. Les champs dits à valeurs multiples, pour lesquels au moins deux valeurs s'appliquent simultanément, ne sont pas autorisés;
 - les codes de valeurs des propriétés de classification doivent être uniques dans le plan de classification complet;
 - aucune classe poubelle n'est autorisée, telle que «divers», «restant», «varié»;
 - il est admis d'utiliser tout attribut significatif dans la classification; leur choix et leur ordre sont déterminés par le fait qu'ils mènent ou non à des propriétés spécifiques significatives pour les utilisateurs;
 - les critères de classification utilisés sont généralement relatifs à la fonctionnalité, mais d'autres critères peuvent intervenir, tels que la technologie, l'application, le matériau ou la géométrie; le choix peut varier dans un arbre de classification entre articles de types différents;
 - les propriétés de classification ont toujours la valeur de type de données: «ENUM_CODE_TYPE».
- d) Pour toutes les propriétés de classification, la visibilité et l'applicabilité sont garanties:
- une propriété est spécifiée comme applicable à une classe si elle est explicitement associée à cette classe. Elle doit satisfaire à une caractéristique essentielle des articles représentés par cette classe;
 - une condition devient applicable à une classe lorsqu'elle est référencée par l'une des propriétés relatives à cette classe.
- e) Les termes relatifs aux branches de l'arbre, étant une valeur de la propriété de classification appropriée, doivent présenter les caractéristiques suivantes:
- ils sont précis – aucun terme vague ou ambigu comme «usage général», «économique», «grande vitesse», etc. n'est autorisé;
 - leur signification est clairement définie;
 - ils sont conformes à une norme internationale, le cas échéant;
 - ils ne sont pas synonymes – s'il y a des synonymes, un terme doit être choisi comme le nom préféré et les autres se rapportent à lui;
 - la signification particulière des homonymes doit être expliquée par le contexte.

Lorsque l'IEC CDD est étendu, les nouveaux ensembles de classes et de propriétés doivent y être ajoutés. Quatre approches peuvent être appliquées.

- 1) Ajout de classes et de propriétés individuelles en divers points dans l'arbre de classe existant de l'IEC CDD.
- 2) Ajout de branches complètes aux arbres de classe de l'IEC CDD. Dans les branches, les nouvelles classes doivent présenter des structures spécifiques au domaine adaptées aux concepts caractérisés.
- 3) Ajout d'un ou plusieurs arbres complets à la structure de données de l'IEC CDD qui couvrent totalement le domaine concerné. Dans de nombreux cas, les domaines (ré)utilisent les concepts des domaines voisins; par exemple, le domaine de l'automatisation de processus peut comprendre les condensateurs, qui sont contenus dans le domaine des composants électroniques. Pour éviter la duplication de classes et/ou de propriétés, il convient que la classe «condensateurs» soit importée au moyen de la relation est un cas-de.
- 4) Ajout d'un ou plusieurs arbres complets sans association spécifique à des classes déjà existantes. Ce cas n'est pas souhaitable car il peut entraîner des duplications de classes

ou de propriétés dans les données de l'IEC CDD. Il convient que ces arbres soient soumis à révision et migrés vers l'approche 3) à moyen terme.

Lors de l'extension du dictionnaire par le contenu d'un nouveau domaine, la classe UNIVERSE joue un rôle important. Par définition, la classe UNIVERSE n'a aucune restriction. La classe UNIVERSE englobe donc toutes les autres classes par définition et tout nouvel arbre doit avoir cette classe pour classe de base.

Tableau 6 – Liste des attributs Item_class définis dans l'IEC 61360-1 et de leurs équivalents dans l'IEC 61360-2

Noms d'attributs dans l'IEC 61360-1	Noms d'attributs dans l'IEC 61360-2	Numéro de paragraphe
applicable_data_element_type	described_by	21.9
applicable_relation	a	21.10
classifying_data_element_type	a	21.11
class_type	a	21.6
code	code	7.3
coded_name	short_name	21.7
definition	definition	8.3
drawing_reference	simplified_drawing	21.12
note	note	8.4
obsolete_date	is_deprecated/ date_of_current_version	9.3
preferred_name	preferred_name	7.4
proposal_date	date_of_original_definition	9.4
published_in	a	9.5
published_by	a	9.6
remark	remark	8.5
responsible_committee	a	9.7
revision_number	revision	7.5
revision_released_on		9.8
source_document_of_definition	source_doc_of_definition	8.6
status_level	a	9.9
superclasse	its_superclass	21.14
synonymous_name	synonymous_name	7.7
version_initiated_on	a	9.10
version_number	version	7.8
version_released_on	date_of_current_version	9.11
^a Non défini dans l'IEC 61360-2:2012.		

21.2 Arbre de composition

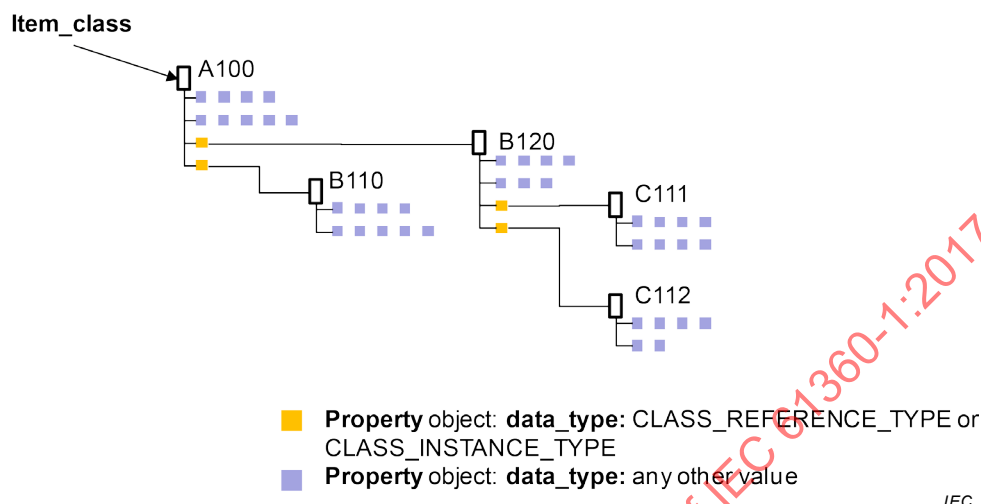
Un arbre de composition exprime la structure partie-tout entre un article et ses membres, par exemple entre un produit et ses pièces ou entre une collection d'informations et ses constituants.

EXEMPLE 1 La nomenclature d'une pièce représente un arbre de composition.

Ces relations entre un article et ses membres sont appelées relations «possède-un». La relation «possède-un» est mise en œuvre par un objet d'information **Property** de type «CLASS_INSTANCE_TYPE» ou «CLASS_REFERENCE_TYPE» (voir 10.4.4 et 10.4.5).

Sur la Figure 21, A100 est composé de deux constituants B110 et B120, tandis que B120 est lui-même composé de C111 et C112. Il n'y a pas d'héritage dans les arbres de composition.

EXEMPLE 2 Sur la Figure 21, A100 peut être un débitmètre constitué de deux modules, une interface d'entrée et une interface de sortie, l'interface de sortie offrant deux configurations: interface 20 mA (C111) et interface 0-24 V (C112).



Anglais	Français
Item_class	Classe d'article
Property object: data_type: CLASS_REFERENCE_TYPE or CLASS_INSTANCE_TYPE	Objet Propriété: type de données: CLASS_REFERENCE_TYPE ou CLASS_INSTANCE_TYPE
Property object: data_type: any other value	Objet Propriété: type de données: toute autre valeur

Figure 21 – Arbre de composition

Le Tableau 6 énumère les différents attributs des classes rencontrés dans les spécifications. Ces attributs sont relatifs à l'identification, la description et les relations.

21.3 Utilisation de schémas auxiliaires de codage et de classification des valeurs

Pour assurer la cohérence et la compréhension des propriétés, l'utilisation de systèmes de codages et de schémas de classification des parties de définition et de leurs valeurs associées est recommandée. L'exemple décrit l'utilisation d'un schéma de classification qui spécifie les formes des contours d'un boîtier de dispositif à semiconducteurs.

NOTE Les schémas auxiliaires soient référencés dans l'attribut «note» lorsqu'ils sont utilisés.

EXEMPLE Le plan de classification des formes des contours d'un boîtier de dispositif à semiconducteurs est basé sur les trois caractéristiques de boîtier suivantes conformément aux codes définis dans l'IEC 60191-4:

- modèle de contour,
- position de borne,
- forme de borne.

Sur la base de ces caractéristiques, cinq niveaux de classification sont définis comme suit:

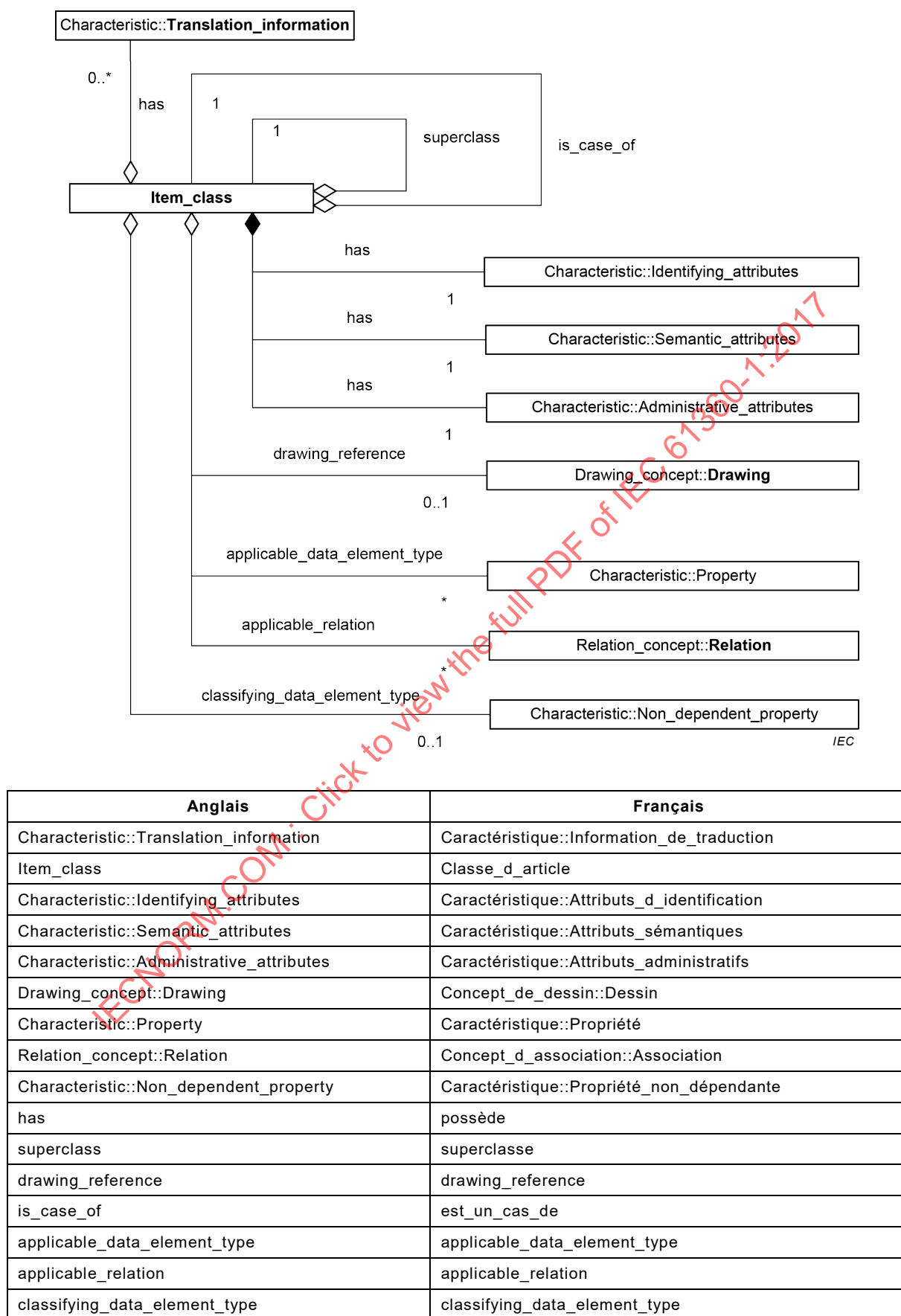
- niveau 1: modèle de contour,
- niveau 2: position de borne,
- niveau 3: forme de borne,
- niveau 4: variante de borne,
- niveau 5: variante de corps (facultatif).

La spécification de chaque classe au niveau le plus bas de l'arbre de classification du boîtier de dispositif à semiconducteurs contient un attribut qui fait référence à un dessin unique spécifiant les paramètres de dimension. Voir l'IEC 60191-4:2013, Article 4 pour la description des paramètres.

21.4 Modèle d'information

La Figure 22 représente le modèle de l'objet d'information «**Données_de_concept**».

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017



**Figure 22 – Modèle d'ensemble de données de concept
(diagramme de classe UML)**

21.5 Spécification de l'objet d'information

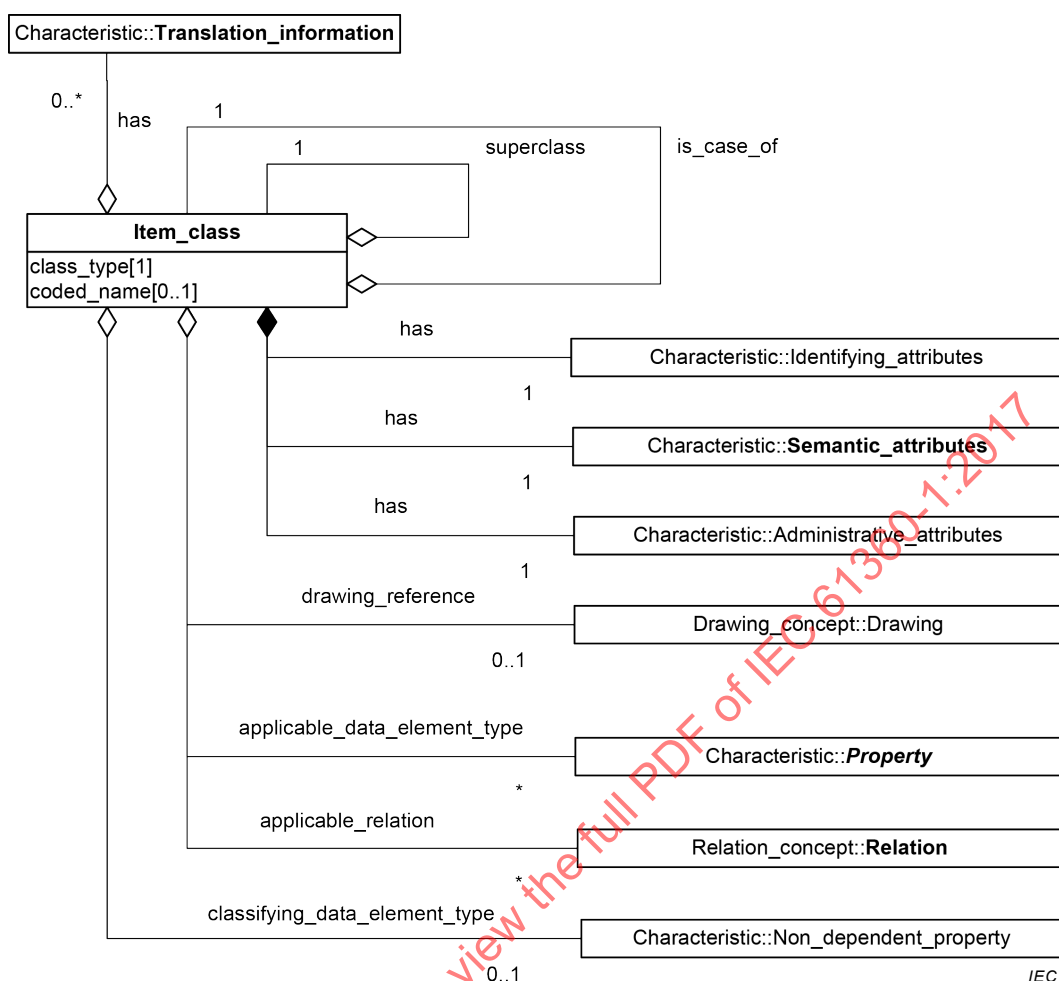
Nom: **Item_class**

Définition: description d'un ensemble d'objets d'information qui partagent les mêmes caractéristiques

Description:

La Figure 23 représente le modèle détaillé de l'objet d'information «**Item_class**» utilisant le langage UML comme technique de présentation.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017



Anglais	Français
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Item_class	Item_class
Characteristic::Identifying_attributes	Caractéristique::Attributs_d_identification
Characteristic::Semantic_attributes	Caractéristique::Attributs_sémantiques
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
Drawing_concept::Drawing	Concept_de_dessin::Dessin
Characteristic::Property	Caractéristique::Propriété
Relation_concept::Relation	Concept_d_association::Association
Characteristic::Non_dependent_property	Caractéristique::Propriété_non_dépendante
class_type(1)	type_de_classe(1)
coded_name(0...1)	nom_codé(0...1)
has	possède
superclass	superclasse
drawing_reference	drawing_reference
is_case_of	est_un_cas_de
applicable_data_element_type	applicable_data_element_type
applicable_relation	applicable_relation
classifying_data_element_type	classifying_data_element_type

Figure 23 – Classe d'article (diagramme de classe UML)

21.6 Type de classe

Nom de l'attribut:	class_type
Définition de l'attribut:	information supplémentaire sur le type de hiérarchie contenant la classe
Description:	Valeurs autorisées: ITEM_CLASS La classe fait partie d'une hiérarchie est-un et est relative à sa superclasse par héritage (voir l'IEC 61360-2:2012, 5.8) CATEGORICAL_CLASSES La classe fait partie d'une hiérarchie de catégorisation et est relative à sa superclasse par est_un_cas_de (voir l'IEC 61360-2:2012, 5.8) FEATURE_CLASS Toutes les classes inférieures à la classe «AAA233 – Caractéristiques» représentent des concepts couramment utilisés qui peuvent être utilisés indépendamment ou être référencés depuis d'autres classes. Il convient d'utiliser CLASS_INSTANCE_REFERENCE si le référencement s'applique.
Obligation:	obligatoire
Type de caractère des valeurs:	Voir l'Article 26.

NOTE **COMPONENT_CLASS** et **MATERIAL_CLASS** étaient définis dans le modèle de dictionnaire de l'IEC 61360-1:1995. Ces types sont déconseillés et supprimés du présent document. (Voir également l'IEC 61360-2:2012, 5.8.1.)

21.7 Nom codé

Nom de l'attribut:	coded_name
Définition de l'attribut:	représentation codée du preferred_name qui doit être égale au code de valeur correspondant (voir 21.7) de la propriété de classification
Description:	Le premier caractère du coded_name doit être une lettre. Le coded_name d'une Item_class doit avoir une longueur maximale de 18 caractères alphanumériques.
Obligation:	obligatoire
Type de caractère des valeurs:	Voir l'Article 26.

21.8 Attributs internes d'une Classe d'article

Une **Item_class** possède des attributs qui sont regroupés pour une meilleure lisibilité dans les objets d'information:

- **Administrative_attributes;**
- **Identifying_attributes;**
- **Semantic_attributes;**
- **Translation_information.**

Ces conteneurs d'attributs sont relatifs à la **Item_class** par la relation «**possède**». Les objets d'information doivent être conservés avec la classe de référence, c.-à-d. qu'ils sont internes aux instances de la **Item_class**.

21.9 Type d'élément de données applicable

Nom de l'attribut: **applicable_data_element_type**

Définition de l'attribut: propriété applicable nécessairement détenue par chaque partie membre d'une classe de caractérisation

NOTE 1 Chaque partie membre d'une classe de caractérisation possède un aspect correspondant à chaque propriété applicable de cette classe de caractérisation.

NOTE 2 La définition ci-dessus est conceptuelle; il n'existe aucune exigence d'utilisation de toutes les propriétés applicables d'une classe pour décrire chaque partie de cette classe au niveau du modèle de données.

NOTE 3 Toutes les propriétés applicables d'une superclasse sont également applicables aux sous-classes de ladite superclasse.

NOTE 4 Seules les propriétés définies ou héritées comme visibles et les propriétés importées d'une classe peuvent être des propriétés applicables.

[La définition et les Notes proviennent de l'IEC 61360-2:2012, 3.2.]

Obligation: conditionnel

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

Etre énumérée en **applicable_data_element_type** ou être héritée depuis un supertype n'implique pas qu'une **Property** spécifique soit facultative ou obligatoire. Il n'existe aucune obligation pour l'utilisateur de donner des valeurs à un quelconque objet d'information **Property**.

NOTE 5 Les contrats entre le propriétaire d'une base de données et les utilisateurs peuvent stipuler l'existence d'un contenu obligatoire dans la base de données en question.

21.10 Association applicable

Nom de l'attribut: **applicable_relation**

Définition de l'attribut: **Relation** définie pour une **Item_class**

Obligation: facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

Etre énumérée en **applicable_relation** n'implique pas qu'une **Relation** spécifique soit facultative ou obligatoire.

NOTE Les contrats entre le propriétaire d'une base de données et les utilisateurs peuvent stipuler l'existence d'un contenu obligatoire dans la base de données en question.

Cet attribut n'est pas concerné par l'héritage.

21.11 Type d'élément de données de classification

Nom de l'attribut: **classifying_data_element_type**

Définition de l'attribut: **Value_list** applicable à une **Item_class** particulière, traitant une propriété élémentaire simple ayant une liste de valeurs, dont les valeurs définissent les sous-classes

Description: Les codes de valeur (c.-à-d. **preferred_letter_symbol_in_text**, voir 17.2) des valeurs doivent être identiques aux **coded_names** (voir 21.7) des sous-classes. Il est aussi souhaitable que le **preferred_name** d'une sous-classe soit le même que le **preferred_name** associé au code valeur.

Voir l'Annexe E pour plus d'informations.

Obligation: facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

21.12 Référence de dessin

Nom de l'attribut: **drawing_reference**

Définition de l'attribut: référence à l'identificateur d'un **Drawing** illustrant la signification d'une **Item_class**

Description: L'identificateur d'un **Drawing** doit se composer de la combinaison du code de dessin à six caractères suivi d'un trait d'union et du numéro de version à trois chiffres des références du dessin.

Obligation: facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

21.13 Est un cas de

Nom de l'attribut:	is_case_of
Définition de l'attribut:	(autre) classe qui est totalement ou partiellement conforme à la spécification de la classe
Description:	is_case_of permet de réutiliser des classes déjà définies.
Obligation:	facultatif
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

21.14 Superclasse

Nom de l'attribut:	superclass
Définition de l'attribut:	classe désignée comme la classe parent canonique de la classe
Description:	S'il n'existe aucun objet d'information Item_class de niveau supérieur, le terme «UNIVERSE» doit être utilisé.
Obligation:	obligatoire
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

22 Association

22.1 Vue d'ensemble

Relation sert à décrire une relation nommée au moyen d'un identificateur général unique. **Relation** englobe deux catégories; l'une est appelée PREDICATION et l'autre FUNCTION. La différence entre elles est que PREDICATION a un domaine mais pas de co-domaine et FUNCTION doit avoir un domaine et un co-domaine.

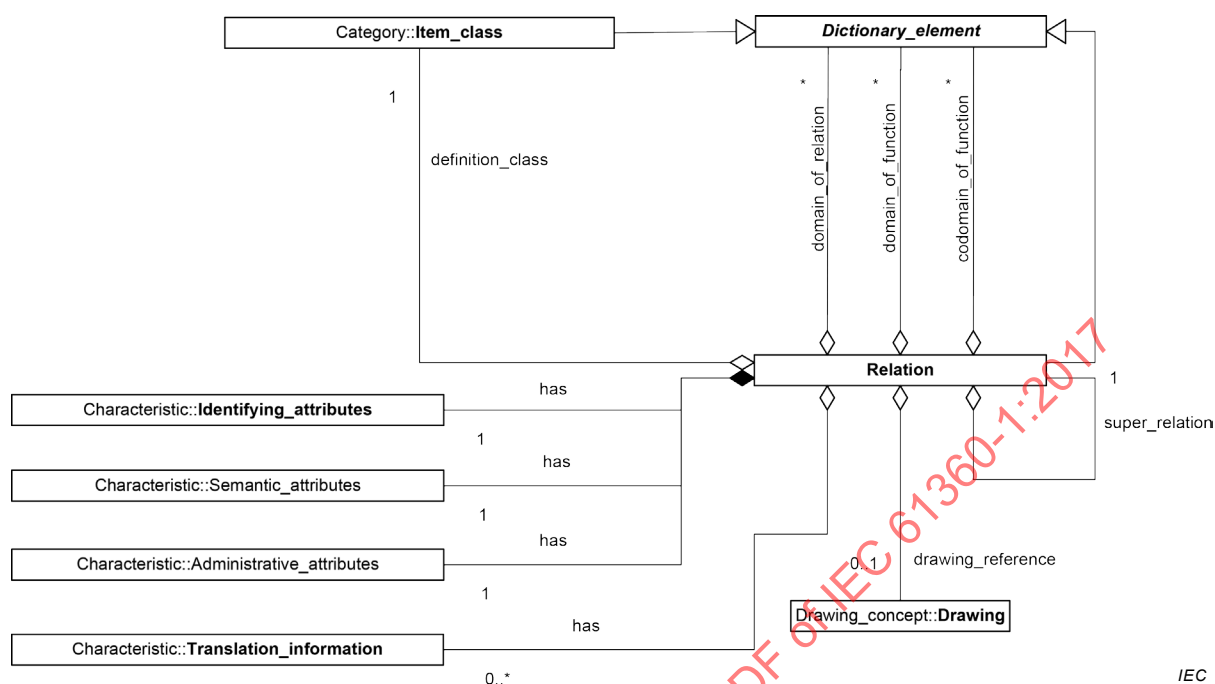
NOTE Lorsqu'il existe une relation R entre $S_1, S_2, \dots, S_n$ alors R peut être défini mathématiquement comme un ensemble R qui est un sous-ensemble du produit cartésien des n ensembles, c.-à-d. $R \subseteq S_1 \times S_2 \times \dots \times S_n$. Bien entendu, d'un point de vue mathématique, une fonction ainsi qu'une prédication sont des types d'association, et donc, les caractéristiques de base de l'association sont conservées dans ces deux catégories.

Différentes relations entre les objets d'information peuvent être modélisées comme instances de l'association:

- contrainte sur une ou plusieurs propriétés, en extension d'une contrainte sur une propriété unique;
- mécanisme de regroupement des objets d'information, en particulier des objets d'information hétérogènes;
- mapping, correspondance, ou transition entre objets d'information.

Chaque rôle est désigné par l'attribut **role_of_the_relation**.

La Figure 24 représente le modèle d'ensemble du module «Relation_concept» utilisant le langage UML comme technique de présentation.



Anglais	Français
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Characteristic::Identifying_attributes	Caractéristique::Attributs_d_identification
Characteristic::Semantic_attributes	Caractéristique::Attributs_sémantiques
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
Drawing_concept::Drawing	Concept_de_dessin::Dessin
drawing_reference	référence_de_dessin
Category:Item_class	Catégorie:Classe_d_article
Dictionary_element	Élément_du_dictionnaire
Relation	Association
definition_class	classe_de_définition
domain of relation	domaine d'association
has	possède
domain_of_function	domaine_de_fonction
codomain_of_function	co-domaine_de_fonction
super_relation	super_association

**Figure 24 – Modèle d'ensemble du concept d'association
(diagramme de classe UML)**

22.2 Spécification de l'objet d'information

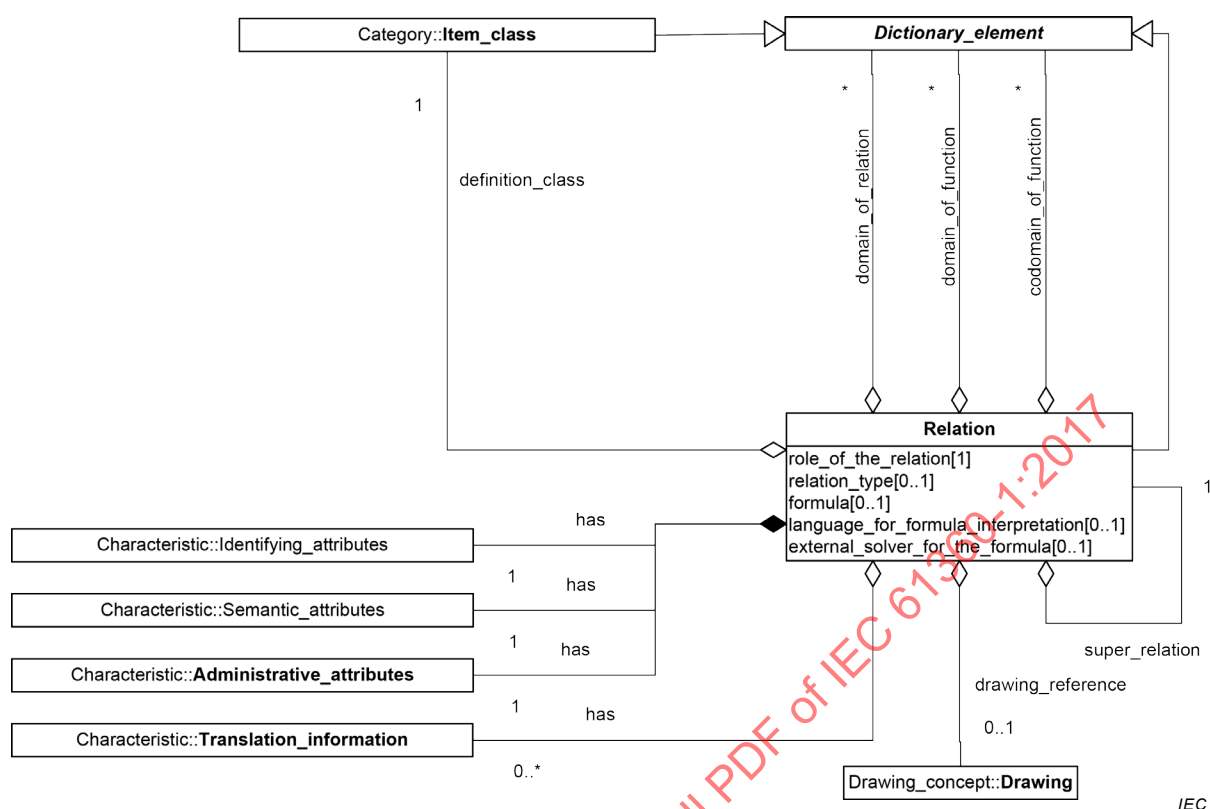
Nom: **Association**

Définition: objet d'information reliant deux instances ou plus d'une classe de type ou d'une propriété dans le but d'appliquer des conditions secondaires spécifiques

Description:

La Figure 25 représente le modèle détaillé de l'objet d'information «**Relation**» utilisant le langage UML comme technique de présentation.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017



Anglais	Français
Characteristic::Translation_information	Caractéristique::Information_de_traduction
Characteristic::Identifying_attributes	Caractéristique::Attributs_d_identification
Characteristic::Semantic_attributes	Caractéristique::Attributs_sémantiques
Characteristic::Administrative_attributes	Caractéristique::Attributs_administratifs
Drawing_concept::Drawing	Concept_de_dessin::Dessin
drawing_reference	drawing_reference
Category::Item_class	Catégorie::Classe_d_article
Dictionary_element	Élément_du_dictionnaire
Definition_class	Classe de définition
Relation	Association
role_of_the_relation(1)	rôle_de_l_association(1)
relation_type(0...1)	type_d_association(0...1)
formula(0...1)	formule(0...1)
language_for_formula_interpretation(0...1)	langue_d_interprétation_de_formules(0...1)
external_solver_for_the_formula(0...1)	résolveur_externe_pour_la_formule(0...1)
domain of relation	domaine d'association
has	possède
domain_of_function	domaine_de_fonction
codomain_of_function	co-domaine_de_fonction
super_relation	super_association

Figure 25 – Association (diagramme de classe UML)

22.3 Résolveur externe pour la formule

Nom de l'attribut: **external_solver_for_the_formula**

Définition de l'attribut: référence à un résolveur externe du contenu de la **formula** qui est transmis à l'environnement du système de paquets et à partir duquel la valeur de la fonction est retournée

Description:

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

22.4 Formule

Nom de l'attribut: **formula**

Définition de l'attribut: de formule logique ou mathématique au moyen de laquelle l'effet de contrainte sur le domaine et/ou la valeur du co-domaine en cas d'association fonctionnelle, peut être calculé

enum_constraint (<i>value1</i> , <i>value2</i> , ...) ^a	ENUMERATION
string_pattern_constraint (<i>pattern</i>) ^b	STRING_TYPE (type chaîne)
string_range_constraint (<i>min</i> , <i>max</i>) ^c	STRING_TYPE (type chaîne)
range_constraint (<i>min</i> , <i>max</i> , OPTIONAL <i>min_inclusive</i> , OPTIONAL <i>max_inclusive</i> ^d , OPTIONAL <i>complement</i> ^e)	INT, REAL, RATIONAL
cardinality_constraint (<i>bound1</i> , <i>bound2</i>) ^f	Types d'agrégats
subclass_constraint (<i>icid1</i> , <i>icid2</i> , ...) ^g	CLASS_REFERENCE ou CLASS_INSTANCE
subtype_constraint (<i>type1</i> , <i>type2</i> , ...) ^h	ENTITY INSTANCE

^a *value1*, *value2*: l'un des éléments d'une énumération

^b *modèle*: une expression générique

^c *min*, *max*: INTEGER

^d *min_inclusive*, *max_inclusive*: BOOLEAN, sa valeur par défaut est TRUE

^e *Complement*: BOOLEAN, sa valeur par défaut est FALSE

^f *bound1*, *bound2*: limite de la cardinalité

^g *icid1*, *icid2*, ...: une liste des ICID de classes auxquelles la propriété de type CLASS_REFERENCE ou CLASS_INSTANCE est autorisée à faire référence

^h *type1*, *type2*, ... types de données tels que définis dans l'ISO 10303-11:2004

Description:

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

22.5 Langage d'interprétation de formule

Nom de l'attribut: **language_for_formula_interpretation**

Définition de l'attribut: langage informatique ou système de description logique au moyen duquel le contenu d'une **formule** peut être interprété

Description:

Obligation: facultatif

Type de caractère des valeurs: Voir l'Article 26.

22.6 Type d'association

Nom de l'attribut: **relation_type**

Définition de l'attribut: type de l'association

valeurs autorisées:

FUNCTION	association f telle que pour toute entité a il existe exactement une entité b à laquelle a est associée par f . [SOURCE: IEC 60050-103:2009, 103-01-01, modifiée – Notes omises]
----------	---

PREDICATION	fonction booléenne $P: X \rightarrow \{\text{vrai, faux}\}$ dont la valeur correspond à la fonction caractéristique ou à la fonction d'indicateur de l'association
-------------	--

Description:

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

22.7 Rôle de l'association

Nom de l'attribut: **role_of_the_relation**

Définition de l'attribut: utilisation spécifique de l'association

Valeurs autorisées:

ARROW
(flèche)

Relation représente une transition d'une collection d'objets d'information à une autre collection d'objets d'information (différents).

La source doit être spécifiée dans **domain_of_the_function** et la destination doit être désignée dans le **codomain_of_the_function**.

Le **relation_type** doit être défini sur «FUNCTION».

NOTE 1 L'attribut **codomain_of_the_function** peut être vide.

PACKAGE
(paquetage)

Les objets d'information énumérés dans **domain_of_the_function** sont des objets d'information considérés comme du même type et qui vont ensemble.

Le **relation_type** doit être défini sur «FUNCTION». L'entrée est **domain_of_function**, le paquetage est toujours référencé dans **codomain_of__function**.

EXEMPLE 1 Un «paquetage» utilisé par le langage informatique orienté objet «Ada».

EXEMPLE 2 «Module» dans «Modula-2» et «C++».

CONSTRAINT
(contrainte)

Contrainte de valeur conforme à l'IEC 61360-2/ ISO 13584-42. L'association est utilisée pour modéliser une contrainte numérique sur une propriété ou sur un ensemble de propriétés.

Le **relation_type** doit être défini sur «PREDICATION».

NOTE 2 Dans le cas CONSTRAINT, l'association est assignée à la **Item_class** par le biais de l'attribut **applicable_relation**.

QUANTITY
(grandeur)

Relation est utilisé pour modéliser le concept d'une grandeur physique comprenant un ensemble d'unités de mesure.

Le **relation_type** doit être défini sur «PREDICATION» ou «FUNCTION».

L'attribut **domain_of_the_relation** collecte les ID des unités de mesure.

EXEMPLE 3 Les unités «millimètre», «micromètre» et «nanomètre», etc., y compris «mètre», qui couvrent la grandeur «mesure de longueur SI».

L'attribut **codomain_of_the_function** contient l'ID de l'unité SI cohérente.

NOTE 3 **codomain_of_the_function** peut être vide.

EXEMPLE 4 «mètre» comme unité SI cohérente de mesure de longueur.

Le **role_of_the_relation** en tant que grandeur peut être utilisé pour modéliser une hiérarchie entre des grandeurs. L'attribut **super_relation** contient alors le lien vers l'association de niveau supérieur suivante.

EXEMPLE 5 Un objet d'information **Relation** peut comprendre toutes les mesures de longueur d'unités SI telles que «mètre», «millimètre», «micromètre» ou «nanomètre». L'autre objet d'information d'association comprend les mesures de longueur impériales avec les unités impériales telles que «mile», «yard», «pied» ou «pouce». Un troisième objet d'information d'association qui est référencé par le contenu des attributs «super relation» de la «relation SI» et de la «relation impériale» représente la grandeur générale «mesure de longueur» comprenant des unités de mesure de longueur SI et des unités de mesure de longueur impériales.

NOTE 4 Il est possible que les futures éditions de l'IEC 61360-1 étendent cette liste de mots clés.

NOTE 5 D'autres normes élaborées à partir du présent document peuvent étendre cette liste de mots clés d'une manière conforme à la sémantique de l'association.

Description:

Obligation: obligatoire

Type de caractère des valeurs: Voir l'Article 26.

22.8 Attributs internes de Relation

Une **Relation** possède des attributs qui sont regroupés pour une meilleure lisibilité dans les objets d'information:

- **Administrative_attributes;**
- **Identifying_attributes;**
- **Semantic_attributes;**
- **Translation_information.**

Ces conteneurs d'attributs sont relatifs à **Relation** par la relation «possède». Les objets d'information doivent être conservés avec la classe de référence, c.-à-d. qu'ils sont internes aux instances de **Relation**.

22.9 Classe de définition

Nom de l'attribut:	definition_class
Définition de l'attribut:	Item_class qui a un domaine d'application dans lequel l'association est significative
Obligation:	obligatoire
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

22.10 Co-domaine de fonction

Nom de l'attribut:	codomain_of_function
Définition de l'attribut:	ID du ou des objets d'information de propriété ou classe qui servent de co-domaine (plage de valeurs) d'une fonction NOTE 1 Ce qui peut dériver d'une fonction est appelé le co-domaine. Ce qui dérive effectivement d'une fonction est appelé la plage. NOTE 2 codomain_of_function peut être vide.
Description:	
Obligation:	conditionnel
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

22.11 Domaine de fonction

Nom de l'attribut:	domain_of_function
Définition de l'attribut:	liste d'ID d'un ou plusieurs objets d'information de propriété ou de classe qui servent de domaine pour l'association fonctionnelle identifiée par Relation NOTE Ce qui peut aller dans une fonction est appelé le domaine.
Description:	

Obligation:	conditionnel
	Cet attribut ou le domain_of_relation doit être utilisé comme fonction
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

22.12 Domaine d'association

Nom de l'attribut:	domain_of_relation
Définition de l'attribut:	liste d'ID d'un ou plusieurs objets d'information de propriété ou de classe qui servent de domaine pour la Relation fonctionnelle identifiée par les ID d'association
	NOTE Ce qui peut aller dans une association est appelé le domaine.
Description:	
Obligation:	conditionnel
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

22.13 Référence de dessin

Nom de l'attribut:	drawing_reference
Définition de l'attribut:	référence à un Drawing illustrant la signification d'une Relation
Commentaires	
Obligation:	facultatif
Type de caractère des valeurs:	Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

22.14 Super association

Nom de l'attribut:	super_relation
Définition de l'attribut:	type général de la Relation
Description:	S'il n'existe aucun objet d'information Relation de niveau supérieur, le terme «UNIVERSE» doit être utilisé.
Obligation:	facultatif

Type de caractère des valeurs: Lettres majuscules de l'alphabet latin de «A» à «Z» à l'exception des lettres majuscules «O» et «I» de l'alphabet latin plus les chiffres «0» à «9».

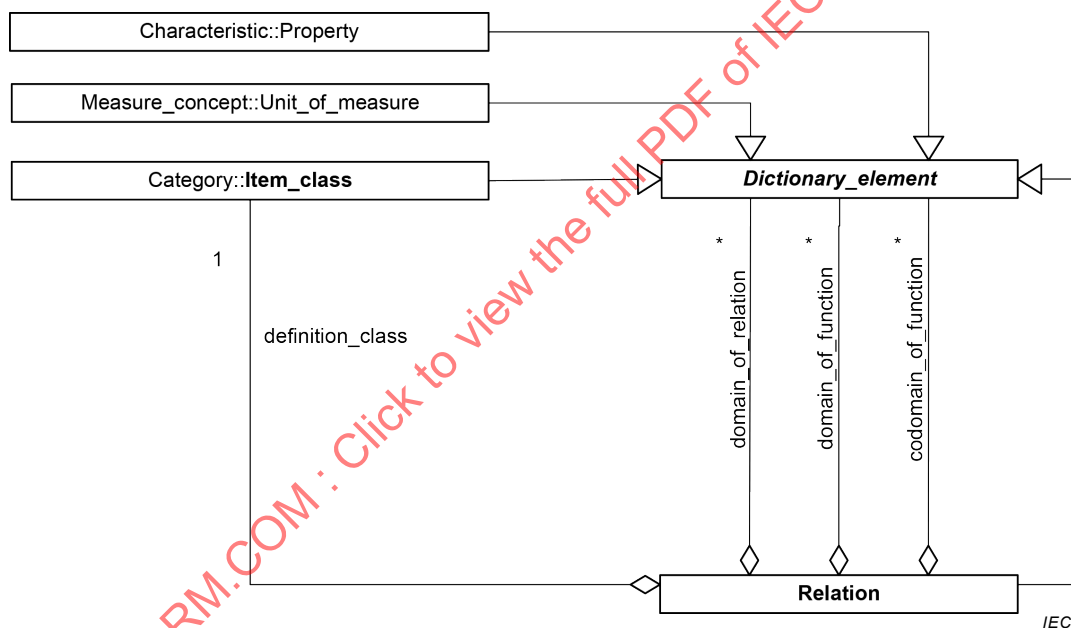
23 Elément_du_dictionnaire

Nom: **Dictionary_element**

Définition: généralisation de **Item_class**, **Property**, **Unit_of_measure** et **Relation** pour pouvoir créer des associations entre eux

Description:

La Figure 26 représente le modèle détaillé de l'objet d'information «**Dictionary_element**» utilisant le langage UML comme technique de présentation.



Anglais	Français
Characteristic::Property	Caractéristique::Propriété
Measure_concept::Unit_of_Measure	Concept_de_mesure::Unité_de_mesure
Category::Item_class	Catégorie::Classe_d'article
definition_class	classe_de_définition
Dictionary_element	Elément_du_dictionnaire
Relation	Association
domain of relation	domaine d'association
domain_of_function	domaine_de_fonction
codomain_of_function	co-domaine_de_fonction

Figure 26 – Elément de dictionnaire (diagramme de classe UML)

24 Concepts avancés

24.1 Vue d'ensemble

Afin de permettre une spécification ou une utilisation des éléments du dictionnaire efficace et exempte d'erreurs, des concepts sont fournis pour prendre en charge la contrainte ou la réutilisation de propriétés, regroupements ou descriptions d'articles configurables déjà spécifiés.

24.2 Condition

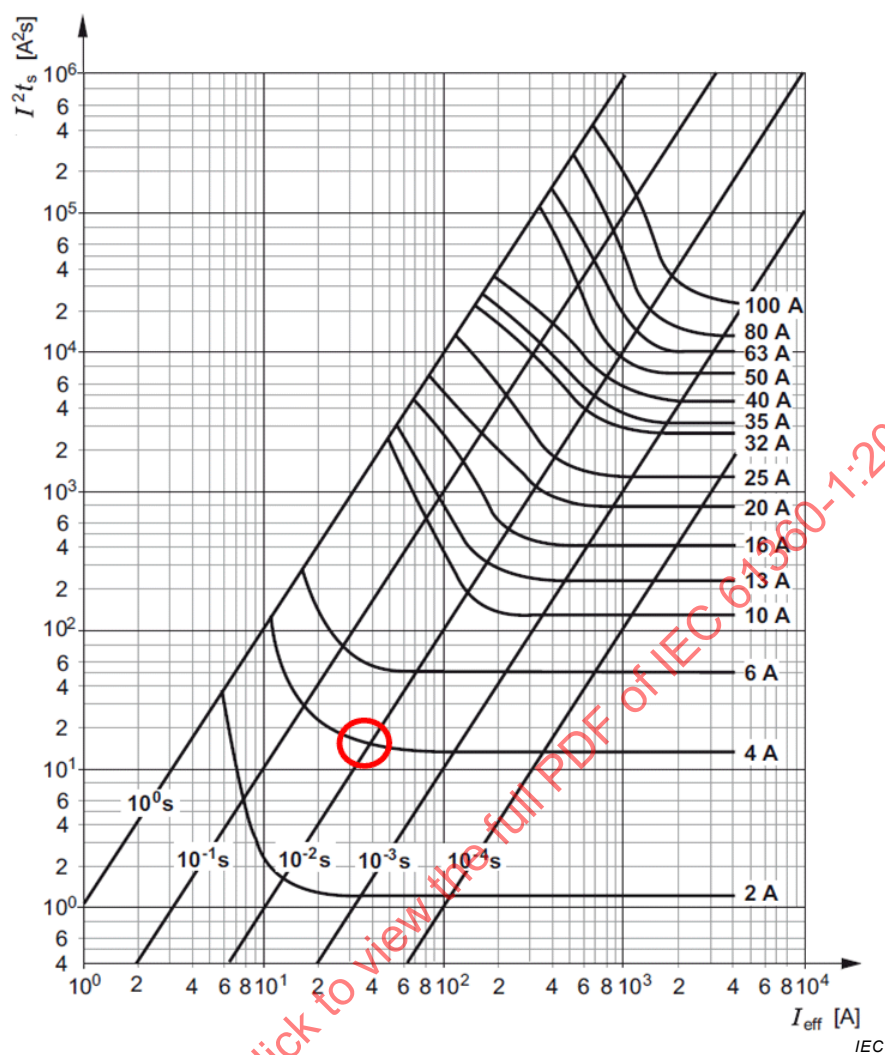
Dans de nombreux cas, les énoncés ou les propositions sont valables uniquement lorsque des informations supplémentaires sont disponibles, comme des conditions secondaires ou des états initiaux. Les conditions permettent de spécifier des informations qui doivent être connues pour l'interprétation non ambiguë d'une valeur donnée.

Une **Property** peut faire référence à un nombre quelconque de conditions. Les conditions représentent toujours des informations supplémentaires. Les conditions peuvent à leur tour être soumises à d'autres conditions.

NOTE Le présent document ne propose pas de concepts tels que les conditions facultatives ou les conditions obligatoires. Des préconditions peuvent toutefois être exprimées. Dans l'exemple ci-dessous, la condition «Temps de réponse» est une précondition pour les conditions «Temps de fusion» et «Temps d'extinction d'arc».

Les conditions vides peuvent exister, c.-à-d. des objets d'informations **Property** sans valeur associée.

EXAMPLE 1 Pour spécifier la valeur I^2t_s d'un fusible, plusieurs conditions secondaires doivent être prises en compte (voir Figure 27 et Figure 28).



Légende

$I^2 t_s$ valeur de fusion $I^2 t$

I_{eff} valeur efficace du courant de court-circuit prévu

Figure 27 – Point de fonctionnement d'un fusible

#CLASS_ID:= MDC_C003						
#CLASS_NAME.en:= Property meta-class						
#PROPERTY_NAME.en	code	preferred_name	formula	property_data_element_type	condition	unit_of_measure
#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en	MDC_P024	MDC_P020	MDC_P028	MDC_P023
	PAA300	melting i^2t value		DEPENDENT_P_DET	PAA301,PAA302,PAA305	A²s
	PAA301	r.m.s. value of the protective short circuit current		CONDITION_DET		A
	PAA302	rated current		CONDITION_DET		A
	PAA303	response time	$T=f(t_s, t_i)$	DEPENDENT_C_DET	PAA303,PAA304	s
	PAA303	melting time		CONDITION_DET		s
	PAA304	arc quenching time		CONDITION_DET		s

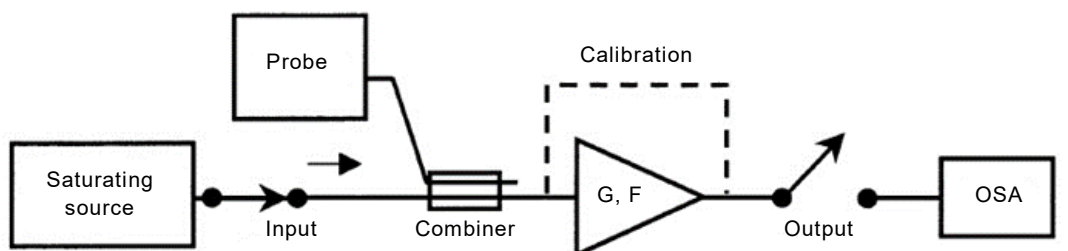
IEC

Anglais	Français
#CLASS_ID:=MDC_C003	#CLASS_ID:=MDC_C003
#CLASS_NAME.en:=Property meta-class	#CLASS_NAME.en:=Property meta-class (Méta-classe Propriété)
#PROPERTY_NAME.en	#PROPERTY_NAME.en
#PROPERTY_ID	#PROPERTY_ID
code	code
preferred_name	nom_préfééré
formula	formule
property_data_element_type	type_d_élément_de_données_de_propriété
condition	condition
unit_of_measure	unité_de_mesure
melting i^2t value	valeur de fusion i^2t
DEPENDENT_P_DET	DEPENDENT_P_DET
CONDITION_DET	CONDITION_DET
DEPENDENT_C_DET	DEPENDENT_C_DET
r.m.s. value of the protective short circuit current	valeur efficace du courant de court-circuit de protection
rated current	courant assigné
response time	temps de réponse
melting time	temps de fusion
arc quenching time	temps d'extinction d'arc

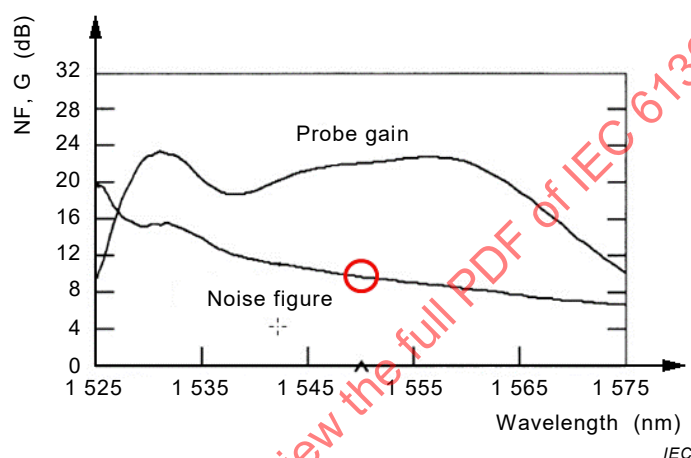
NOTE Tous les attributs non pertinents ont été omis pour que la figure reste simple.

Figure 28 – Mise en œuvre sous forme de tableau selon l'IEC 62656-1 de l'exemple de la Figure 27

EXEMPLE 2 Pour la mesure du bruit d'amplificateurs optiques, il existe plusieurs méthodes. Une condition peut être utilisée pour fournir l'information sur l'environnement de mesure. Voir l'exemple de Figure 29 et Figure 30 [SOURCE: Douglas M. Baney, et al, *Theory and Measurement Techniques for the Noise Figure of Optical Amplifiers*].



a) Installation de mesure du gain dynamique et du chiffre de bruit



b) mesure avec une longueur d'onde de saturation de 1550 nm

Anglais	Français
Saturating source	Source de saturation
probe	sonde
calibration	étalonnage
input	entrée
combiner	multiplexeur
DUT	DUT
output	sortie
OSA	OSA
probe gain	gain de la sonde
noise figure	chiffre de bruit
NF, G, dB	NF, G, dB
wavelength, nm	longueur d'onde, nm

Légende

DUT dispositif en essai
F facteur de bruit
G gain
NF chiffre de bruit
OSA analyseur de spectre optique

Figure 29 – (a) Installation de mesure du gain dynamique et du chiffre de bruit et (b) mesure avec une longueur d'onde de saturation de 1550 nm

#CLASS_ID:= MDC_C003							
#CLASS_NAME.en:= Property meta-class							
#PROPERTY_NAME.en	code	preferred_name	formula	property_data_element_type	condition	unit_of_measure	drawing_reference
#PROPERTY_ID	MDC_P001_6	MDC_P004_1.en	MDC_P024	MDC_P020	MDC_P028	MDC_P023	MDC_P008_2
	PAA320	noise figure	$10\log_{10}(F)$	DEPENDENT_P_DET	PAA321, PAA322	dB	
	PAA321	wavelength		CONDITION_DET		nm	
	PAA322	noise figure measurement setup		CONDITION_DET			PAA330
#CLASS_ID:= MDC_C007							
#CLASS_NAME.en:= Document meta-class							
#PROPERTY_NAME.en	code	drawing_title	file_name	file_format			
	MDC_P001_8						
	PAA330	noise figure measurement setup picture	figure001	pdf			

IEC

Anglais	Français
#CLASS_ID:=MDC_C003	#CLASS_ID:=MDC_C003
#CLASS_NAME.en:=Property meta-class	#CLASS_NAME.en:=Property meta-class (Méta-classe Propriété)
#PROPERTY_NAME.en	#PROPERTY_NAME.en
#PROPERTY_ID	#PROPERTY_ID
code	code
preferred_name	nom_préféré
formula	formule
property_data_element_type	type_d_élément_de_données_de_propriété
condition	condition
unit_of_measure	unité_de_mesure
drawing_reference	référence_de_dessin
noise figure	chiffre de bruit
wavelength	longueur d'onde
noise figure measurement setup	installation de mesure du chiffre de bruit
DEPENDENT_P_DET	DEPENDENT_P_DET
CONDITION_DET	CONDITION_DET
#CLASS_ID:=MDC_C007	#CLASS_ID:=MDC_C007
#CLASS_NAME.en:=Document meta-class	#CLASS_NAME.en:=Document meta-class (Méta-classe Document)
drawing_title	drawing_title
file_name	nom_de_fichier
noise figure measurement setup picture	image de l'installation de mesure du chiffre de bruit
file_format	format_de_fichier
figure001	figure001

Figure 30 – Mise en œuvre sous forme de tableur selon l'IEC 62656-1 de l'exemple de la Figure 29

24.3 Réutilisation des propriétés

Il existe des situations dans lesquelles il convient de référencer les objets d'information **Property** pour éviter la duplication de concepts.

L'attribut **is_case_of** permet d'importer une ou plusieurs propriétés «étrangères» depuis une autre classe. Il importe peu que cette classe fasse partie du même arbre de classes ou réside dans un autre arbre de classes. Les relations importées sont héritées dans les sous-classes de la même manière que les associations qui ne sont pas assignées par **is_case_of**.

EXEMPLE Il peut exister une classe «ordinateur personnel» dans un arbre de classe «ordinateur» et il peut exister une classe «disque dur» dans un arbre «mémoire». L'attribut **is_case_of** est utilisé pour éviter la duplication des objets d'information de **Property** «vitesse du disque» et «capacité de stockage». Figure 31 La représente les objets d'information relatifs.

IECNORM.COM : Click to view the full PDF of IEC 61360-1:2017