

BS EN 62656-3:2015



BSI Standards Publication

Standardized product ontology register and transfer by spreadsheets

Part 3: Interface for Common
Information Model

bsi.

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

National foreword

This British Standard is the UK implementation of EN 62656-3:2015. It is identical to IEC 62656-3:2015.

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

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

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

© The British Standards Institution 2015.

Published by BSI Standards Limited 2015

ISBN 978 0 580 75784 6

ICS 01.040.01; 01.110

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2015.

Amendments/corrigenda issued since publication

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

English Version

**Standardized product ontology register and transfer by
spreadsheets - Part 3: Interface for Common Information Model
(IEC 62656-3:2015)**

Enregistrement d'ontologie de produits normalisés et
transfert par tableurs - Partie 3: Interface pour un modèle
d'information commun
(IEC 62656-3:2015)

Standardisierte Übertragung und Registrierung von
Ontologien für Produkte mittels Tabellen - Teil 3:
Schnittstellen für das allgemeine Informationsmodell
(IEC 62656-3:2015)

This European Standard was approved by CENELEC on 2015-03-31. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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



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

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Foreword

The text of document 3D/234/FDIS, future edition 1 of IEC 62656-3, prepared by IEC/SC 3D "Product properties and classes and their identification", of IEC/TC 3 "Information structures, documentation and graphical symbols", was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 62656-3.

The following dates were fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2015-12-31
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2018-03-31

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

Endorsement notice

The text of the International Standard IEC 62656-3:2015 was approved by CENELEC as a European Standard without any modification.

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

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

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 61360-1	2009	Standard data elements types with associated classification scheme for electric items -- Part 1: Definitions - Principles and methods	EN 61360-1	2010
IEC 61360-4	-	Standard data element types with associated classification scheme for electric components -- Part 4: IEC reference collection of standard data element types and component classes	EN 61360-4	-
-	-		+corrigendum Dec.	-
IEC 61968-1	2003	Application integration at electric utilities - System interfaces for distribution management -- Part 1: Interface architecture and general requirements	EN 61968-1	2004
IEC 61968-11	2010	Application integration at electric utilities - System interfaces for distribution management -- Part 11: Common Information Model (CIM) extensions for distribution	EN 61968-11	2010
IEC 61970-1	2005	Energy management system application program interface (EMS-API) -- Part 1: Guidelines and general requirements	EN 61970-1	2006
IEC 61970-301	2013	Energy management system application program interface (EMS-API) -- Part 301: Common information model (CIM) base	EN 61970-301	2014
IEC 61970-501	2006	Energy management system application program interface (EMS-API) -- Part 501: Common Information Model Resource Description Framework (CIM RDF) schema	EN 61970-501	2006
IEC 62325-301	2014	Framework for energy market communications -- Part 301: Common Information Model (CIM) extensions for markets	EN 62325-301	2014
IEC 62656-1	2014	Standardized product ontology register and transfer by spreadsheets -- Part 1: Logical structure for data parcels	EN 62656-1	2015
ISO 639-1	2002	Codes for the representation of names of languages_ - Part_1: Alpha-2 code	-	-

ISO 8601	2004	Data elements and interchange formats - Information interchange - Representation of dates and times	-	-
IEC/TS 61970-2	2004	Energy management system application program interface (EMS-API) -- Part 2: Glossary	CLC/TS 61970-2	2005
ISO/IEC directives Supplement	2013	Procedures specific to IEC	-	-

CONTENTS

FOREWORD	7
INTRODUCTION	9
1 Scope	10
2 Normative references	10
3 Terms, definitions and abbreviations	11
3.1 Terms and definitions	11
3.2 Abbreviations	13
4 Generic rules and principles for the design of CIM interface	13
4.1 Comparative analysis of the structures and functionalities of POM and CIM	13
4.2 CIM standard series	14
5 Rules of transformation from CIM to POM	15
5.1 General rules	15
5.2 CIM Class	16
5.2.1 General	16
5.2.2 Name	17
5.2.3 Description	17
5.2.4 Generalization-specialization	17
5.3 CIM class attributes	18
5.3.1 General	18
5.3.2 Name	18
5.3.3 Native attribute	18
5.3.4 Type	18
5.3.5 Description	18
5.4 CIM association	19
5.4.1 General	19
5.4.2 CIM association end structure	19
5.4.3 General transformation rule for association	20
5.4.4 General transformation rule for association end	20
5.5 CIM Enumeration class	24
5.5.1 General	24
5.5.2 Name	25
5.5.3 Description	25
5.6 CIM Enumeration Attribute	25
5.6.1 General	25
5.6.2 Name	25
5.6.3 Description	25
5.7 Data type	26
5.7.1 CIM basic datatype	26
5.7.2 CIM named data types	26
5.8 Package	28
5.8.1 General	28
5.8.2 POM relation object	30
5.8.3 CIM packages described in POM relation	31
6 CIM description in IEC 62656 format	33
6.1 General	33
6.2 Class	34

6.2.1	General	34
6.2.2	Code	34
6.2.3	Version number	34
6.2.4	Revision number.....	35
6.2.5	Date of original definition	35
6.2.6	Date of current version	35
6.2.7	Preferred name.....	35
6.2.8	Short name.....	36
6.2.9	Definition	36
6.2.10	Note	36
6.2.11	Superclass	36
6.2.12	Class type	37
6.2.13	Alternate ID	37
6.2.14	CIM package	37
6.2.15	Applicable properties	38
6.3	Property.....	38
6.3.1	General	38
6.3.2	Property data element type.....	38
6.3.3	Definition class	38
6.3.4	Data type.....	39
6.3.5	Applicable relations	39
6.3.6	Condition	39
6.3.7	Unit in text.....	39
6.3.8	Alternative units.....	40
6.4	Data type	40
6.4.1	General	40
6.4.2	Definition class	40
6.4.3	Unit in text.....	40
6.4.4	Alternative units.....	41
6.4.5	Data type.....	41
6.5	Enumeration	41
6.5.1	General	41
6.5.2	Definition class	41
6.5.3	Enumerated list of terms	41
6.5.4	Enumerate code list.....	42
6.6	Term meta-class	42
6.6.1	General	42
6.6.2	Preferred letter symbol in text.....	42
6.7	Relation	42
6.7.1	General	42
6.7.2	Relation type	43
6.7.3	Domain of the relation	43
6.7.4	Domain of the function.....	43
6.7.5	Codomain of the function	43
6.7.6	Domain element type	44
6.7.7	Role of the relation	44
7	XML schema for updated CIM instances in IEC CDD	44
7.1	General.....	44
7.2	Principles.....	

8	Version control mechanism in POM for the parcellized CIM	46
Annex A (normative)	Mapping rules from CIM to POM – Mapping rules described in POM relation meta-class	47
Annex B (normative)	CIM in IEC 62656 format – Normative meta-properties of the CIM in IEC 62656 model	56
Annex C (informative)	XML Schema for parcellized CIM	63
C.1	XML schema for POM models	63
C.2	XML schema for a class	63
C.2.1	General	63
C.2.2	class.xsd	63
C.2.3	class1.xsd	64
C.2.4	class2.xsd	64
C.2.5	class3.xsd	64
C.2.6	class4.xsd	65
C.3	XML schema for a property	65
C.3.1	General	65
C.3.2	property.xsd	65
C.3.3	property1.xsd	66
C.3.4	property2.xsd	66
C.3.5	property3.xsd	67
C.3.6	property4.xsd	67
C.4	XML schema for a property derived from an association end	68
C.4.1	General	68
C.4.2	association.xsd	68
C.4.3	association1.xsd	68
C.4.4	association2.xsd	69
C.4.5	association3.xsd	70
C.4.6	association4.xsd	70
C.5	XML schema for a data type	70
C.5.1	General	70
C.5.2	datatype.xsd	71
C.5.3	datatype1.xsd	71
C.5.4	datatype2.xsd	71
C.5.5	datatype3.xsd	72
C.5.6	datatype4.xsd	72
C.6	XML schema for an enumeration and its enumerators	73
C.6.1	General	73
C.6.2	enum_term.xsd	73
C.6.3	enum_term1.xsd	73
C.6.4	enum_term2.xsd	74
C.6.5	enum_term3.xsd	74
C.6.6	enum_term4.xsd	75
C.7	XML schema for a CIM package	75
C.7.1	General	75
C.7.2	package.xsd example	75
C.7.3	package1.xsd example	76
C.7.4	package2.xsd example	76
C.7.5	package3.xsd example	76
C.7.6	package4.xsd example	77

Annex D (informative) XML examples for parcellized CIM	78
D.1 XML example for a CIM class with extended attributes	78
D.2 XML example for a property with extended attributes	78
D.3 XML example for a CIM association and aggregation with extended attributes	79
D.4 XML example for a CIM data type with extended attributes	81
D.5 XML example for a CIM enumeration and its enumerators with extended attributes	81
D.6 XML example for a CIM package with extended attributes.....	82
Annex E (informative) Property reference between CIM and IEC CDD – “Power transformer” definitions in both IEC CDD and CIM	84
Annex F (informative) Four Layer architecture of UML and POM – Four layer architecture of MOF (meta object facility).....	87
Annex G (informative) POM Relation usage in IEC 62656-3.....	88
G.1 General.....	88
G.2 Meta-model mapping rules between UML and POM(M2)	88
G.3 UML Association(M1).....	88
G.4 Model level relation definition(M1)	88
Annex H (informative) Parcellized CIM files access – URL for the parcellized CIM files	89
Bibliography.....	90
Figure 1 – CIM generalization example (from Figure 2 of IEC 61970-301:2013)	17
Figure 2 – Simple association example (from Figure 3 of IEC 61970-301:2013)	21
Figure 3 – Aggregation example (from Figure 4 of IEC 61970-301:2013)	21
Figure 4 – Property data element type definition	23
Figure 5 – CIM package example (from Figure 1 of IEC 61970-301:2013).....	30
Figure 6 – POM relation model explanation in UML.....	31
Figure 7 – Class instances of parcellized CIM.....	45
Figure 8 – Conceptual reference mechanism with ID in POM	46
Table 1 – Mapping of CIM objects and POM objects	15
Table 2 – Correspondence between a CIM class in UML and a POM class	17
Table 3 – Mapping of CIM class attribute and POM property	18
Table 4 – CIM association end structure	19
Table 5 –CIM associations defined as POM relations.....	20
Table 6 – Mapping of CIM association end and POM property	21
Table 7 – CIM association ends described as POM properties	22
Table 8 – Mapping of CIM enumeration and POM enumeration	25
Table 9 – Mapping of CIM enumeration attribute and POM term	25
Table 10 – Mapping of CIM basic data type and POM simple data type.....	26
Table 11 – Mapping of CIM data type and a named data type in POM.....	27
Table 12 – Mapping of CIM package and POM relation	29
Table 13 – CIM package described as POM relation	32
Table A.1 – Mapping rule from CIM class to POM class	48
Table A.2 – Mapping rule from CIM class attribute to POM property.....	49
Table A.3 – Mapping rule from CIM association to POM relation	50

Table A.4 – Mapping rule from CIM association end to POM property	51
Table A.5 – Mapping rule between CIM enumeration and POM enumeration.....	52
Table A.6 – Mapping rule between CIM enumeration attribute and POM term	52
Table A.7 – Mapping rule between CIM data type and POM data type	53
Table A.8 – Mapping rule between CIM Package and POM relation	54
Table A.9 – Mapping rule from POM property to CIM aggregation/association/class attribute	55
Table B.1 – Meta-properties for class meta-class	57
Table B.2 – Meta-properties for property meta-class	58
Table B.3 – Meta-properties for datatype meta-class	59
Table B.4 – Meta-properties for enumeration meta-class	60
Table B.5 – Meta-properties for term meta-class	61
Table B.6 – Meta-properties for relation meta-class	62
Table C.1 – POM models and XML schemas.....	63
Table E.1 – Native properties of Power transformer class in the IEC CDD	85
Table E.2 – Native properties of Power transformer class in IEC 61970-301:2013	86

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**STANDARDIZED PRODUCT ONTOLOGY REGISTER
AND TRANSFER BY SPREADSHEETS –****Part 3: Interface for Common Information Model**

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 62656-3 has been prepared by subcommittee 3D, Product properties and classes and their identification, of IEC technical committee 3: Information structures, documentation and graphical symbols.

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

FDIS	Report on voting
3D/234/FDIS	3D/245/RVD

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

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

A list of all parts in the IEC 62656 series, published under the general title *Standardized product ontology register and transfer by spreadsheets*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication 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 62656 series, *Standardized product ontology register and transfer by spreadsheets* is a series of International standards that collectively define the methods for transferring and registering the ontologies of various products and services to and from the ontology registries and applications based on IEC 61360 / ISO13584 common data dictionary model. The IEC common data dictionary, or IEC CDD for short, is one of such registries maintained online as an IEC 61360-4 International Standard based on IEC database procedure stipulated in ISO/IEC Directives Supplement – Procedures specific to IEC. The IEC CDD is a cross-domain data dictionary covering all electro-technical products and services, maintained and updated through a database administered by IEC Central Office.

The common information model originally defined in IEC 61968 and IEC 61970 series of standards, often called by its short name “CIM” provides a standard way to represent all the major objects in an electric utility enterprise typically needed to model the operational aspects of a utility. This model includes public classes and attributes for these objects, as well as the relationships between them. It is known as an information model for energy management system (EMS) of power grids and currently is recognized as a standard ontology model for smart grids. An ontology specification conformant to the CIM data model is available in UML format according to IEC 61970-301, and in RDF format according to IEC 61970-501.

The IEC 62656 series consists of the following parts, under the general title *Standardized product ontology transfer and register by spreadsheets*:

- Part 1: Logical structure for data parcels;
- Part 2: Application guide for use with the IEC common data dictionary (CDD);
- Part 3: Interface for common information model.

STANDARDIZED PRODUCT ONTOLOGY REGISTER AND TRANSFER BY SPREADSHEETS –

Part 3: Interface for Common Information Model

1 Scope

This part of IEC 62656 specifies an interface between IEC 62656 series and meta-model for CIM originally defined in IEC 61968 and IEC 61970 series of standards. The current CIM includes IEC 62325 series and the interface specified in this part of IEC 62656 also applies to the model defined in IEC 62325-301:2014. More specifically, this standard defines a formal mapping between the IEC 62656 and meta-model for CIM in order to import the CIM ontology into the IEC CDD, and to ensure the interoperability of ontologies of two standards, or even among a wider spectrum of standards. For the basis of the mapping from CIM to the data model defined in IEC 62656-1, the UML representation of CIM is referenced.

As a result of the interface specification available from this part of IEC 62656, the smart grid ontology defined in CIM becomes accessible and interoperable in the midst of ontology pieces originating in other ontology standards, encompassing material, environmental, and mechanical, and logistic domains of information. In addition, the specification may also cover the lifecycle of products.

This part of IEC 62656 also defines methods for transforming the IEC CDD content into the format defined in IEC 61968 and IEC 61970 series. However, this standard does not intend any standardization of the elements defined in IEC CDD as part of the CIM ontology, for all the definitions of IEC CDD are already part of an acknowledged horizontal International Standard known as IEC 61360-4-DB. Rather it makes the pieces of information stored in IEC CDD available for further standardization or customization in power electric domains or more specific user communities as a basic resource, in particular for the purpose of extending the CIM series of standards.

The data model of this part is based on IEC 62656-1 which specifies the logical structure for a data parcel, or just simply “parcel,” that is a short name for the medium of registering or transferring of product ontology. Such a parcel may be typically implemented in a leaf of a spreadsheet. The data model built on a set of parcels is called “parcellized ontology model”, which is often abbreviated as “POM” in the remainder of this document.

This part of IEC 62656 specifies:

- the mapping rules and principles for importing CIM UML objects into POM;
- the syntactic and semantic requirements on the parcellized CIM objects;
- the piecewise version control mechanism in POM for the parcellized CIM objects.

The following items are outside the scope of this part of IEC 62656:

- definition of the interface between CIM UML and CIM RDF;
- specification of syntactic and semantic requirements on CIM UML;
- specification of syntactic and semantic requirements on CIM RDF.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For

undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62656-1:2014, *Standardized product ontology register and transfer by spreadsheets – Part 1: Logical structure for data parcels*

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

IEC 61360-4, *Standard data element types with associated classification scheme for electric components – Part 4: IEC reference collection of standard data element types, component classes and terms*, available from <<http://std.iec.ch/iec61360>>

IEC 61968-1:2003, *System interfaces for distribution management – Part 1: Interface architecture and general requirements*

IEC 61968-11:2010, *System interfaces for distribution management – Part 11: Common information model (CIM) extensions for distribution*

IEC 61970-1:2005, *Energy management system application program interface (EMS-API) – Part 1: Guidelines and general requirements*

IEC 61970-2:2004, *Energy management system application program interface (EMS-API) – Part 2: Glossary*

IEC 61970-301:2013, *Energy management system application program interface (EMS-API) – Part 301: Common Information Model(CIM) base*

IEC 61970-501:2006, *Energy management system application program interface(EMS-API) – Part 501: Common Information Model Resource Description Framework(CIM RDF) schema*

IEC 62325-301:2014, *Framework for energy market communications – Part 301: Common information model (CIM) extensions for markets*

ISO 639-1:2002, *Codes for the representation of names of languages – Part 1: Alpha-2 code*

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

ISO/IEC Directives Supplement:2013, *Procedures specific to IEC*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purpose of this document, the terms and definitions given in IEC 61970-2 and IEC 62656-1, as well as the following apply.

3.1.1 attribute

- a) attribute defined in IEC 62656-1, equivalent to a meta-property
- b) attribute of class used in IEC 61970-301 corresponding to a property in POM
- c) attribute of enumeration used in IEC 61970-301 corresponding to a term in POM

Note 1 to entry: In case of b), it is more specifically called as “CIM attribute” or “CIM class attribute” in this standard

3.1.2**aggregationKind**

kind of the UML association comprising normal association, aggregation and composition.

Note 1 to entry: Possible value of aggregationKind is “none” for normal association, “shared” for aggregation and “composition” for a composition, the same as ISO/IEC 19505-2:2012 definitions.

3.1.3**CIM multiplicity at the source**

multiplicity of the instances of the class located at the source side of the two classes connected by an association

3.1.4**CIM basic data type**

basic data type defined in the CIM comprising String, Boolean, Float and Integer

Note 1 to entry: In CIM edition, there are other basic data types available for use

3.1.5**CIM object**

class, attribute, package, or relationship defined in series of IEC 61970, IEC 61968 and IEC 62325 series to represent power system resources

3.1.6**CIM package**

general-purpose means of grouping related model elements of CIM

Note 1 to entry: CIM Package is a synonymous name assigned to the Segment(MDC_P211) attribute of IEC 62656-1:2014

3.1.7**CIM UML ID**

ID of each CIM object, assigned by a UML tool by which the object is designed

3.1.8**INVERSIBLE_P_DET****INVERSIBLE_PROPERTY**

kind of property data element type that can depend on the same type of property

Note 1 to entry: A UML association with unspecified navigability is defined with INVERSIBLE_P_DET because the navigability may be assigned in the future

3.1.9**navigability**

state of accessibility of the instances of a class at one end of an association from the instances of the class at the other end of the association.

Note 1 to entry: In this standard, the navigability of bidirectional, unidirectional and unspecified is distinguished.

3.1.10**parcellized CIM**

collection of CIM objects translated into and represented by a set of spreadsheets conformant to the POM objects defined in IEC 62656-1, with an extended list of attributes added in this part of the standard

3.1.11**POM object**

set of objects defined by a meta-class of IEC 62656-1:2014 and used in this part of IEC 62656.

EXAMPLE Class, property, datatype, enumeration, unit, term and relation.

3.1.12

property

- a) instance of property meta-class, or the meta-class per se as schema; defined in IEC 62656-1
- b) property defined in IEC 61970-2 that represents a specific aspect, characteristic, attribute or relation used to describe a network resource

Note 1 to entry: To clearly distinguish between these two cases, in case of a), it is called “POM property” or “property in POM”. In case of b), it is called “CIM property” in this part of IEC 62656; whenever necessary.

3.2 Abbreviations

For the purposes of this standing document, the following abbreviations apply.

CIM	common information model
IEC CDD	IEC common data dictionary
POM	parcellized ontology model
RDF	resource description framework
RDFS	resource description framework schema
UML	unified modeling language
XML	extensible markup language

4 Generic rules and principles for the design of CIM interface

4.1 Comparative analysis of the structures and functionalities of POM and CIM

The current CIM is modelled with the UML which is widely accepted among IT engineers for designing concepts of products and services. Thus UML is used as a convenient design tool to model the CIM. However, from the users' perspective, the current CIM modelled in UML is totally devoid of mechanism to locally update or extend the entities of the model on an entity-by-entity basis, while the POM is adequately equipped to record the history of changes in a continuous manner. In other words, the current CIM standard is designed to update the model as a whole in a file at once. However, this poses a significant obstacle to the maintenance of the model as an ontology in a database and to the updates of the elements of the database on an individual basis. Since every update shall entail a holistic change of the model, and such a change will disrupt the operation of the objects already installed and used in various applications and systems.

To address this concern, an advanced framework for ontology maintenance and change management for the elements of the ontology in a database form is required, which shall be equipped with;

- a) ontology evolution management
- b) ontology extension mechanism
- c) hierarchy management for both the specialisation (is-a) and composition (has-a).

The above three requirements are fully met by POM, while the CIM in UML or in RDF is designed for other requirements and objectives.

Moreover, by scope, the CIM concentrates on information exchange among electric-power related systems and products, thus the information covered by IEC 61968, IEC 61970 and IEC 62325 series is not complete for describing the whole range and life-cycle of the products and services used in the power grids: Namely, manufacturing, installation, maintenance, transport, and disposal related characteristics of the products and services are not covered. Although some parts of IEC 61968 and IEC 62325 series are intended to relate pieces of information of different domains, they do not and cannot standardize the entire domains of electro-technical ontology, simply because such is not within the scope of one domain TC/SC.

It requires almost all the pieces of information of all the technical committees of IEC: That is why the generic description of product classes, properties and identification covering all technical domains is assigned to a horizontal committee. With the IEC 61360 CDD managed by such a committee, integration and maintenance of all the range of electro-technical ontologies is made possible.

Hence, this part of IEC 62656 provides the means and methods for importing the CIM defined ontology into IEC CDD, and thereby allows the interoperable use of the ontologies of both IEC CDD and CIM in a harmonious manner for the benefit of all electro-technical systems and applications. The case of “Transformer” class is listed in Annex E as a typical example to show how CIM and IEC CDD definitions complement each other.

With this perspective in mind, this part of IEC 62656 defines in the first place the rules and principles of mapping between CIM and POM meta-models with an aim to import the objects of the former into the IEC CDD, in a form of the latter. It also defines additional attributes to the current CIM model that are indispensable for managing the content in the IEC CDD database. In terms of the choice of source data representation, this standard directly maps the objects of CIM UML into POM, since CIM is defined originally in UML. Consequently, CIM profiles, which are subsets of the CIM model, can also be transformed into POM because they are required to be consistent with the CIM UML definitions.

Secondly, this standard intends to specify rules and principle for transforming the objects conforming to IEC 61360 represented in IEC 62656-1, into a format that conforms to the CIM standard, namely, the CIM RDFS format. This virtually means that objects in IEC 61360 CDD are transformable into pieces of ontology that can be regarded as a possible extension of CIM.

The choice of this format, RDFS, is for practical reasons: This part of IEC 62656 does not intend to fully automate the generation of CIM objects from the IEC CDD. A converter based on this standard may create CIM-like objects in RDFS, and the information contained in the objects shall be useful for updating or extending the CIM ontology of the next generation. However, the selection and validation from both the syntactic and semantic perspectives need to be performed by experts in the electric power domain, and the techno-semantic standardization in this perspective is out of the scope of this part of IEC 62656: It is a horizontal sub-committee responsible for development and maintenance of the model and content of the IEC CDD across the domains of IEC, but it does not necessarily have the expertise of each specific technical domain.

Nevertheless, the steps of integration of the CIM objects into IEC CDD are significantly facilitated by the methods described in this standard. For this standard fully describes the rules for syntactic transformation of CIM objects into IEC CDD compatible forms through the projection of the former constructs onto the parcel data structures. This standard additionally provides a set of basic rules and methods useful for semantic validation of the parcellized CIM content in IEC CDD, with an aim to assure the interoperability of the ontologies originating in different standard committees.

While the detailed semantic validation shall lie in the hands of human experts, the IEC CDD change management rules for the content stored in the IEC CDD in combination with the quality guide manual will provide a practical guide and help for this purpose.

In Clause 5, the rules and principles of the mapping between CIM and POM are defined as normative content. The detailed description of the mappings is summarized in the Annex A. Reverse transformation of the objects of the IEC CDD into CIM RDF and its relationship with the original CIM-RDF are listed in Annex C as informative.

4.2 CIM standard series

The common information model (CIM) specifies the semantics of the application programming interface (API) for energy management systems (EMS). Since the CIM is an abstract model

that represents the semantics of all the typical objects in an electric utility enterprise, it comprises multiple standard parts each of which addresses a specific field and topic concerning the power grid ontology. IEC 61970-301 is the basic one of them, and it defines a base set of CIM, as a set of packages, which provides a logical view of the functional aspects of EMS including SCADA. IEC 61970-302 is another part of the series, and it defines financial and energy scheduling aspects. IEC 61968 is another series of CIM that deals with other logical views, such as utility operations including assets, location, activities, documentation, and work management. In addition, IEC 62325-301 is a standard part which defines CIM extensions for energy market communications. There is a single and unified information model that comprises all the parts of CIM, even though they have different individual standard documents.

The methodology and model defined in IEC 61970-301:2013 is selected as the logical basis for this part of standard.

5 Rules of transformation from CIM to POM

5.1 General rules

As shown in Table 1, UML meta-model available in CIM shall be projected onto the POM meta-model. First of all, each primary class in CIM shall be mapped to a class in POM. Each attribute of CIM class shall be transformed into a corresponding property in POM. An enumeration in CIM shall be transformed to an enumeration in POM and the attributes of an enumeration in CIM shall be transformed into the corresponding terms in POM. An association between classes in CIM, including aggregation, shall be also transformed into a property in POM. Each data type in CIM shall be mapped into an instance of datatype meta-class, thus it is treated as a user defined data type that augments built-in data types of POM, and a package of CIM shall be transformed into a relation in POM.

Annex A lists the detail of the transformation rules exemplified in Table 1 in a form of a set of relations in POM: Note that the POM has a built-in capacity to describe various mapping rules between ontologies as relations. However, it is important to be aware of the distinction between the mapping table and the parcellized CIM model, which is produced as a result of the transformation rules described in Table 1.

Table 1 – Mapping of CIM objects and POM objects

UML meta-model			POM meta-model
Meta object	(1)	Class	Class
	(2)	Enumeration class	Enumeration
	(3)	Datatype class	Datatype
	(4)	Attributes of primary class	Property
	(5)	Attributes of enumeration	Term
	(6)	Package	Relation
NOTE Primary class means a class defined in the UML except classes for enumeration and data type.			

Details of the transformation rules for each object are explained in the following subclauses 5.2 to 5.8.

5.2 CIM Class

5.2.1 General

From a data modelling point of view, a class in UML is an abstraction of an information structure that represents a static component of an information system. The components are drawn as classes with their explicit attributes, and the relationships among the classes are shown as associations. A class may have a hierarchical relationship with a generalized class or a specialized class, referred to as a “parent class” or a “child class” respectively, where inheritance transcends the attributes of the parent class to its offspring. In short, a class in UML is an element of generic software design tool or methodology for object-oriented programming than a structure for representing a concept of products or services in the real world. Thus it fits for generally guiding the overall software design. However it is not necessarily the best for maintaining and updating the product and service specifications as one information entity called ontology that is separate from the body of procedural software description, for its content grows in volume and whose specification of elements evolves individually over time.

In contrast, a class in POM is an abstraction of a product concept with a collection of properties each of which is designed as an abstraction of “product property” that has in general a unit of measure and a corresponding data type, when the type is quantifiable. Moreover, a class is generated as an instance of a class meta-class, i.e., a “factory” of classes or sometimes short-named as “class parcel” and is designed to be defined and updated within a database. Therefore, a class is supposed to have, a preferred name, synonymous names, definitions, multiple language translations, globally unique IDs, versions and revisions, a set of characteristics, etc., as part of a default and pre-defined structure for representing a concept of products or services. Needless to say, inheritance and part-whole relationships are built-in as part of the model. In short, a class in POM is more specifically designed as part of a database structure for representing a concept of products and services in the real world.

With this understanding, CIM classes can be transformed into POM classes, as specified by a set of mapping rules summarized in Table 2.

In the table, the column of “UML meta-model in CIM” is for the original UML meta-model applied in the CIM. Here are two sub-columns: the “UML object” is for the UML object name, and the “CIM definition” is for the corresponding CIM definition of the UML object on the left side.

The column of “POM meta-model” is for the meta-model from POM. Here are three sub-columns: the “POM object” is for the POM object name corresponding to the CIM UML concept, “Attribute name” is for the POM attribute name and “Attribute ID” is for the ID defined in POM for the POM attribute name on the left.

The Description column is for noting a human interpretable instruction for guidance in transforming the UML meta-model in CIM into the corresponding meta-model of POM.

Table 2 – Correspondence between a CIM class in UML and a POM class

UML meta-model in CIM		POM meta-model			Description
UML object	CIM definition	POM object	Attribute name	Attribute ID	
Class	Name	Class	Short name	MDC_P004_3.<lang>	<lang> in POM shall be a two-letter ISO 639 code. As CIM only gives name in English, lang here shall be “en”.
	Description		Note	MDC_P007_1.<lang>	<lang> in POM shall be a two-letter ISO 639 code. As CIM only gives note in English, lang here shall be “en”.
	Generalization (inverse of Specializes)		Superclass	MDC_P010	

5.2.2 Name

The name of a CIM class shall be transformed to the “short name” (MDC_P004_3.<lang>) attribute of the POM class. At the same time, as the POM requires a “Preferred name”(MDC_P004_1.<lang>) for each POM class as a mandatory item, the original CIM name shall be copied to the “Preferred name” of the same class as an initial tentative name when transforming from CIM to POM for the first time. The “Preferred name” of a class in the POM may be updated according to the maintenance procedure for database standards specified in the ISO/IEC Directives Supplement.

5.2.3 Description

The “Description” attribute of a CIM Class is to explain or describe a class. Therefore, the “Description” of a CIM class shall be mapped to the “Note”(MDC_P007_1.<lang>) in a POM class. At the same time, as the POM requires a “Definition”(MDC_P005.<lang>) for each POM class as a mandatory item, the original CIM note shall be copied to the “Definition” of the same POM class as an initial tentative definition when transforming from CIM to POM for the first time. The “Definition” in the POM may be updated according to the maintenance procedure for database standards specified in the ISO/IEC Directives Supplement.

5.2.4 Generalization-specialization

UML has a capacity to describe the generalization-specialization relationship between two classes. In case of the example shown in Figure 1, the class “PowerSystemResource” is a generalization of the class “Equipment”. Specialization is the inverse of the generalization. The generalization is also known as an “is-a” relationship which can be described by a “Superclass” attribute of a class in POM. With this example, class “Equipment” has a superclass named “PowerSystemResource” in POM.

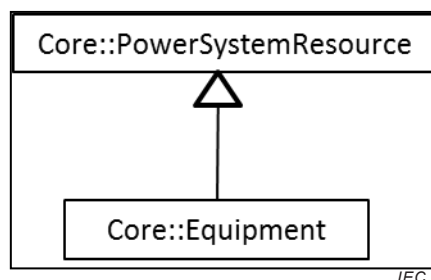


Figure 1 – CIM generalization example
(from Figure 2 of IEC 61970-301:2013)

5.3 CIM class attributes

5.3.1 General

Attributes of a UML class are the descriptive elements of a class, while a property of POM describes one characteristic aspect of a class. Thus, the UML class attributes can be mapped into POM properties with due specification of data type, etc.

Table 3 – Mapping of CIM class attribute and POM property

UML meta-model in CIM		POM meta-model			Description
UML object	CIM definition	POM object	Attribute name	Attribute ID	
Attribute of class	Name	Property	Short name	MDC_P004_3.<lang>	<lang> in POM shall be a two-letter ISO 639 code. As CIM only gives name in English, lang here shall be "en".
	(for native attribute)		Definition class	MDC_P021	for native attribute, its definition class shall be described here
	Type		Data type	MDC_P022	
	Description		Note	MDC_P007_1.<lang>	<lang> in POM shall be a two-letter ISO 639 code. As CIM only gives note in English, lang here shall be "en".

5.3.2 Name

The name of a CIM class attribute shall be transformed to the "Short name" (MDC_P004_3.<lang>) attribute of the POM property. At the same time, the original name of the CIM class attribute shall be copied to the "Preferred name" of the same POM property as an initial tentative name when transforming from CIM to POM for the first time. The "Preferred name" of a property in the POM may be updated according to the maintenance procedure for database standards specified in the ISO/IEC Directives Supplement.

5.3.3 Native attribute

In CIM, there are two types of class attributes: one type is a set of native attributes, which are defined in the class itself. The other type is the set of inherited attributes, which are inherited from the super class or ancestor classes.

POM provides the following structure for differentiating the native attributes and the inherited attributes:

- Each native attribute shall be transformed into a POM property which has its "Definition class" in the class in which the property is defined. At the same time, the property shall be added to the attribute of "Applicable properties" (explained at 6.2.15.) of the POM class.
- Inherited attributes of a class shall be collected from the "Applicable properties" of its upper classes.

5.3.4 Type

The type of a CIM attribute shall be transformed into the data type of a POM property which defines the possible data type of the instance values.

5.3.5 Description

The "Description" of the CIM class attribute shall be transformed into the "Note" attribute of a POM property. The note shall be copied to the "Definition" attribute of the same POM

property as a tentative definition, when the conversion from CIM to POM is performed for the first time. The “Note” in POM will not be changed to keep the originality of CIM, and the “Definition” in POM will be updated according to the maintenance procedure for database standards specified in the ISO/IEC Directives Supplement.

5.4 CIM association

5.4.1 General

An association is a conceptual connection among classes. In UML, an association may have a “name” and have more than two “association ends” members. For each association end, attributes of “aggregationKind”, “navigability” and “multiplicity” are defined. An aggregation in UML is a specialized case of an association. Compared to the normal association, it describes a part-whole relationship between classes. Furthermore, there is a special type of aggregation defined as “composition”, which restricts that the composite object shall have responsibility for the existence and storage of the composed objects, i.e., both objects should have the same life cycle according to the composition. As the aggregation and composition are specified types of association, three kinds of them shall be described with the same model in the POM. In this subclause 5.4, how to represent an association including aggregation and composition in POM is explained.

In the CIM, an association name is not specified normally. Each association end contains the information including multiplicity shown in Table 4. Attributes of “aggregationKind” and “navigability” are not explained directly in the association end structure in the CIM, but they can be accessed from the UML class diagram or the corresponding XMI (XML Metadata Interchange) file for the CIM.

In this part of the standard, each association end with its aggregationKind, navigability and multiplicity is mapped to a POM property. Details of the mapping rule are explained in 5.4.3.

Even though composition has not yet been applied in the CIM models, the method for representing composition in POM is introduced in this standard for a possible use of the construct in a future edition of CIM.

5.4.2 CIM association end structure

Each association end contains the information shown in Table 4, which is editorially adapted from the original defined in IEC 61970-301:2013.

As described in the CIM standard, the first column describes the multiplicity of instances of the class (per one instance of the class connected at the named end) from which the association having the end name is extended (i.e., the multiplicity of the class for which the association ends are being described). In the second column, the multiplicity of the association end, namely the range of instances of the class connected to the association end with a specific association end name in plain text is enclosed within a pair of brackets placed at the head of the name. The name of the class connected to the association end is listed in the “type” column. A multiplicity of zero indicates an optional association. A multiplicity of “*” indicates any number is allowed. For example, a multiplicity of [1..*] indicates that a range of instances from 1 to any number is allowed.

Table 4 – CIM association end structure

Multi from	[Mult to] name	Type	description
[0..*]	[1..1]BaseVoltage	BaseVoltage	The base voltage used for all equipment within the VoltageLevel

5.4.3 General transformation rule for association

In the UML, each association should have no less than two association ends. Each association end is defined as a member of the association. As only binary association which includes two association ends is available in the current CIM, the representation of a binary association is expressed in this standard.

In the IEC 62656-3, each association end is transformed to an instance of a POM property and each association is transformed to an instance of the POM relation. The association ends owned by an association are defined in the “Domain of the Relation” of the relation derived from the UML association.

For each association explained in the POM relation, values of following attributes shall be assigned as:

- Relation type(MDC_P200) has a value of “Predication”
- Role of the relation(MDC_P210) has a value of “UML Association”
- Domain of the relation(MDC_P201) has a value of a list of property ID representing association end owned by this association

For an association without a given name, the code assigned to the corresponding relation can be copied to the “Preferred name”(MDC_P004_1) attribute.

Table 5 shows samples of how to describe an association in the POM relation.

In Table 5, “Asso1” represents the association shown in Figure 2 with its association ends definition in Table 7 in the first two records. “Asso2” represents the association shown in Figure 3 with its association ends definition in Table 7 in the third and fourth records.

Table 5 –CIM associations defined as POM relations

MDC_P001_13	MDC_P004_1.<lang>	MDC_P007_1.<lang>	MDC_P200	MDC_P201	MDC_P202	MDC_P203	MDC_P208	MDC_P210
Code	Preferred name	Note	Relation type	Domain of the relation	Domain of the function	CoDomain of the function	Domain element type	Role of the relation
Asso1			Predication	{Property_code1,Property_code2}				UML Association
Asso2			Predication	{Property_code3,Property_code4}				UML Association

5.4.4 General transformation rule for association end

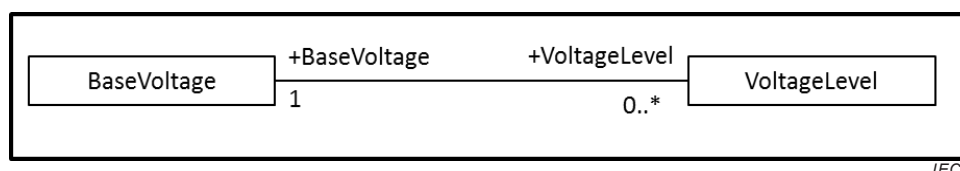
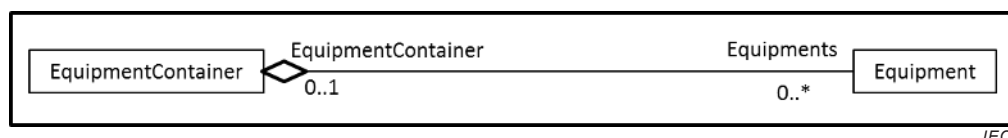
5.4.4.1 General

The CIM association end shall be mapped to a POM property. In this property, attribute of “applicable relations” should list the ID of the relation which represents the association containing this association end. The navigability of an association end is distinguished with the “property data element type” and “condition” attributes of the property. The aggregationKind is distinguished from the “remark” attribute of the property. Table 6 summarises the representation of association end of CIM in POM.

Table 6 – Mapping of CIM association end and POM property

UML meta-model in CIM		POM meta-model			Description
UML object	CIM definition	POM object	Attribute name	Attribute ID	
Association end	Association end name	Property	Short name	MDC_P004_3. <lang>	<lang> in POM shall be a two-letter ISO 639 code. As CIM only gives name in English, lang here shall be "en".
	Description		Note	MDC_P007_1. <lang>	<lang> in POM shall be a two-letter ISO 639 code. As CIM gives note only in English, lang here shall be "en".
			Definition class	MDC_P021	For an association end, its definition class shall be set to the class which is connected to.
			Property data element type	MDC_P020	For a property derived from an association end, the value of this attribute shall be set to "INVERSIBLE_P_DET", for other properties from CIM, the value of this attribute shall be set to "NON_DEPENDENT_P_DET"
			Condition	MDC_P028	For a property derived from an association end with navigability, the ID of the property representing the association end at the other side shall be described in this attribute.
			Applicable relations	MDC_P230	This attribute is to describe the association containing this association end.
			Remark	MDC_P007_2	This attribute is to describe the aggregationKind which has value of "none", "shared" or "composition".
	[Mult to] name		Data type	MDC_P022	For a property derived from an association end, it is required to have the datatype as "class reference type", with a reference to its connected class. For the association end example shown in Table 4, it shall be defined as a property with the following datatype: CLASS_REFERENCE_TYPE(ICID of the class BaseVoltage).
	Type				

Let us use a simple association shown in Figure 2, an aggregation shown in Figure 3 as samples to explain the transformation rules.

**Figure 2 – Simple association example (from Figure 3 of IEC 61970-301:2013)****Figure 3 – Aggregation example (from Figure 4 of IEC 61970-301:2013)**

For these two samples, Table 7 shows their corresponding POM descriptions. Each property in Table 7 represents an association end with their multiplicities.

Table 7 – CIM association ends described as POM properties

MDC_P001_6	MDC_P004_1. en	MDC_P005. en	MDC_P007_2. en	MDC_P021	MDC_P230	MDC_P020	MDC_P028	MDC_P022
Code	Preferred name	Definition	Remark	Definition class	Applicable relations	Property data element type	Condition	Data type
Property_code1	VoltageLevel	The VoltageLevels having this BaseVoltage.	#aggregationKind=none	ICID of BaseVoltage	Asso1	INVERSIBLE_P_DET		List(0,n) of class_reference_type(ICID of class VoltageLevel)
Property_code2	BaseVoltage	The base voltage used for all equipment within the VoltageLevel.	#aggregationKind=none	ICID of class VoltageLevel	Asso1	INVERSIBLE_P_DET		class_reference_type of(ICID of class BaseVoltage)
Property_code3	Equipments	The association is used in the naming hierarchy.	#aggregationKind=shared	ICID of Equipment Container	Asso2	INVERSIBLE_P_DET		List(0,n) of class_reference_type(ICID of class Equipment)
Property_code4	Equipment Container		#aggregationKind=none	ICID of Equipment	Asso2	INVERSIBLE_P_DET		List(0,1) of class_reference_type(ICID of class EquipmentContainer)

5.4.4.2 Name of an association end

As CIM does not name an association but gives a name to each end of an association, the name of the association end, for example, “VoltageLevel” is transformed to the “short name” of a POM property which is held in a class with the preferred name “BaseVoltage”. At the same time, the name of the association end shall be copied to the “preferred name” of the same POM property as an initial tentative name, when transforming from CIM to POM for the first time.

The multiplicity of an association end at the target class, which is defined as [mult to] in the CIM, shall be described in the datatype as List(0,n) of class_reference_type. For example, the association end “VoltageLevel” described in the first record of Table 7, the multiplicity at the target class, i.e., at “BaseVoltage” class is specified as “(0,n)” in the datatype as List(0,n) of class_reference_type. For the second record of Table 7, the multiplicity at the target class, i.e., at “VoltageLevel” class is omitted as the multiplicity is “1”. Details are explained in 5.4.4.8.

5.4.4.3 Description

A “description” which explains an association shall be transformed to a “Note” attribute (MDC_P007_1.<lang>) of the POM property where <lang> is assumed to be “en”. At the same time, since the POM requires a “definition”(MDC_P005.<lang>) for each property, the original CIM note shall be copied to the “definition” attribute of the same POM property when transforming from CIM to the POM for the first time. The “definition” in the POM will be updated according to the maintenance procedure for database standards specified in the ISO/IEC Directives Supplement.

5.4.4.4 Remark

In this part of the standard, “Remark” is used to describe the “aggregationKind” of each association end. The “aggregationKind” which is the same as that of the UML, may have value of “none” for normal association, or “shared” for an aggregation, or “composition” for a composition.

In this part of the standard, in order to distinct the aggregationKind definition from other Remark, the aggregationKind shall be described with the following format:

#AggregationKind=none|shared|composition

Samples are shown in the “Remark” column of Table 7.

5.4.4.5 Property data element type and condition

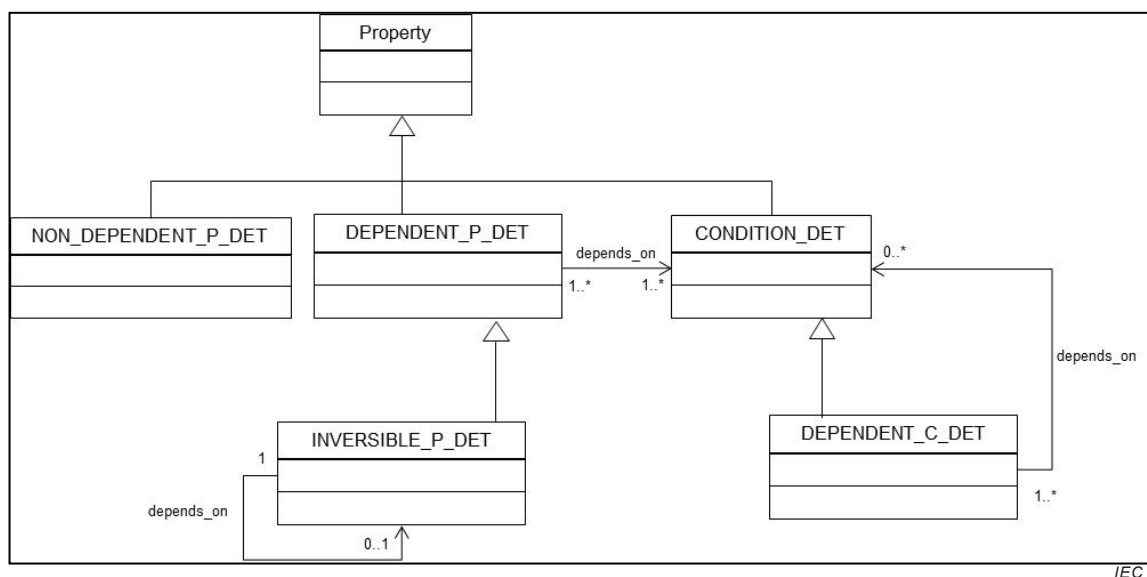


Figure 4 – Property data element type definition

In this standard, “INVERSIBLE_P_DET” is added as a new “property data element type” in order to describe the association end which may have navigability. Figure 4 shows the relationship among property and their subtypes and their dependent conditions, with the addition of “INVERSIBLE_P_DET”. In the IEC 61360 series and IEC 62656-1, a property of “DEPENDENT_P_DET” is assumed to depend on a property of “CONDITION_DET” type. In this standard, a property of “INVERSIBLE_P_DET” can depend on another property of “INVERSIBLE_P_DET”. Thus, a property derived from an association end can depend on a property derived from the association end on the opposite side. For other usages of the “INVERSIBLE_P_DET”, the cardinality of “depends_on” of the “INVERSIBLE_P_DET” does not need to be 0 or 1, but can be set to any number by necessity.

In this standard, the “INVERSIBLE_P_DET” combined with the attribute of “Condition” is applied to record the navigability of an association end.

In the UML, there are three types of navigability for an association end:

- Non-navigable

NOTE 1 In the UML class diagram, an association end with non-navigability is drawn with a X mark.

- Navigable

NOTE 2 In the UML class diagram, an association end with navigability is drawn with an arrow.

– Unspecified

NOTE 3 In the UML class diagram, an association end with unspecified navigability is drawn with a simple line. In the CIM, most association ends are defined with unspecified navigability.

Each association end shall be transformed into a POM property with the “property data element type” as “INVERSBILE_P_DET”. For an association with two navigable association ends, “Condition” of the property (e.g. Property_A) representing the association end shall have the value of the property ID (e.g. Property_B) representing the association end on the opposite side.

5.4.4.6 Applicable relations

For a POM property which is derived from an association end of CIM, this attribute is used to describe the association which contains this association end. As the association itself is described as an instance of POM relation, the corresponding ICID of the relation shall be described in this attribute. Examples are shown in Table 7.

5.4.4.7 Native and inherited association ends

In the CIM, there are two kinds of association ends for a class: native association ends and inherited association ends from its upper classes.

- Each native association end shall be transformed into a POM property which has its “definition class” in the class in which the property is defined. At the same time, the property shall be added to the “applicable properties (explained in 6.2.15)” of the POM class.
- Inherited association ends of a class shall be collected from the “applicable properties” of its upper class and ancestor classes.

5.4.4.8 Data type

POM is equipped with a property of “class_reference_type” which is capable of describing an association between two classes. Therefore, a property transformed from a CIM association end into POM shall be of “class_reference_type”. Furthermore, to describe the multiplicity of instances of a class to which the association end is connected, POM may make use of an ordered set of properties of “class_reference_type”, i.e. the data type of the property shall be described as “List(0,n) of class_reference_type (ICID)” where the ICID signifies the global ID of the class to which the association end is connected. When the multiplicity is 1, the data type does not need to be defined as “List(1) of class_reference_type(ICID)” but only as “class_reference_type(ICID)”.

For an association of “Composition” type, it is recommended to use data type of “class_instance_type” instead of “class_reference_type” because the former one restricts that the composed objects must have the same lifecycles of the composite objects.

5.5 CIM Enumeration class

5.5.1 General

In the CIM, UML stereotypes of “primitive”, “enumeration” and “datatype” are applied. A class with primitive type of the stereotype standards for a common class, a class of an enumeration type of the stereotype stands for an enumeration class and a class with datatype of the stereotype is for a datatype class. An enumeration class defines all possible values for this enumeration and enumeration class has no hierarchical structures.

POM defines an object as enumeration which is provided to describe the possible values for a property. Therefore, the CIM enumeration shall be described as a POM enumeration.

POM defines an object as datatype which shall be provided to describe the datatype of the CIM. The detail of the datatype is explained in 5.7.

Table 8 – Mapping of CIM enumeration and POM enumeration

UML meta-model in CIM		POM meta-model		
UML object	CIM definition	POM object	Attribute name	Attribute ID
Enumeration	Name	Enumeration	Short name	MDC_P004_3.<lang>
	Description		Note	MDC_P007.<lang>

5.5.2 Name

CIM enumeration class name shall be transformed to the “short name” of the POM enumeration. At the same time, the name of the enumeration class shall be copied to the “Preferred name” of the same POM enumeration as an initial tentative name, when transforming from CIM to POM for the first time. And the “Preferred name” in POM will be updated according to the maintenance procedure for database standards specified in the ISO/IEC Directives Supplement.

5.5.3 Description

The “Description” of the CIM enumeration class attribute shall be transformed to “Note” attribute of a POM enumeration. And when it is the first time to transform from CIM to POM, the “Note” shall be copied to the POM “Definition” attribute too. The “Note” in POM will not be changed to keep the originality of CIM, and the “Definition” in POM will be updated according to the maintenance procedure for database standards specified in the ISO/IEC Directives Supplement.

5.6 CIM Enumeration Attribute

5.6.1 General

In CIM, enumeration is defined as a class and its possible values are defined as its native attributes. POM defines an object as term to provide a means to define repeatedly used terms, constants or individual values used in a data model. While transforming from CIM to the POM, each possible enumeration value of the CIM shall be defined as one instance of the POM term. As shown in Table 9, attributes of an enumeration shall be transformed to instances of the POM term.

Table 9 – Mapping of CIM enumeration attribute and POM term

UML meta-model in CIM		POM meta-model		
UML object	CIM definition	POM object	Attribute name	Attribute ID
Attribute of enumeration	Name	Term	Short name	MDC_P004_3.<lang>
	Description		Note	MDC_P007.<lang>

5.6.2 Name

Name of an enumeration attribute shall be transformed to the “Short name” of the POM term. At the same time, the name of the CIM enumeration attribute shall be copied to the “Preferred name” of the same POM term as an initial tentative name, when transforming from CIM to POM for the first time. The “Preferred name” in POM will be updated according to the maintenance procedure for database standards specified in the ISO/IEC Directives Supplement.

5.6.3 Description

“Description” of an enumeration attribute shall be transformed to the “Note” of the POM term. And when it is the first time to transform from CIM to POM, the “Note” shall be copied to the