

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Standardized product ontology register and transfer by spreadsheets –
Part 1: Logical structure for data parcels**

**Enregistrement d'ontologie de produits normalisés et transfert par tableurs –
Partie 1: Structure logique pour les paquets de données**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 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 more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 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 plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 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.



IEC 62656-1

Edition 1.0 2014-08

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Standardized product ontology register and transfer by spreadsheets –
Part 1: Logical structure for data parcels**

**Enregistrement d'ontologie de produits normalisés et transfert par tableurs –
Partie 1: Structure logique pour les paquets de données**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XH

ICS 01.040.01; 01.110

ISBN 978-2-8322-1745-0

**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.....	8
INTRODUCTION.....	10
1 Scope.....	11
2 Normative references	12
3 Terms and definitions	13
4 Parcel use cases and scenarios.....	18
4.1 Typical use cases	18
4.2 Spreadsheet representation of dictionary or library	18
4.3 Use scenario of dictionary parcel format	19
4.4 Use scenario of library parcel format	20
4.5 Use scenario of parcel format of higher layers	21
5 The Parcellized Ontology Model (POM).....	21
5.1 Overview of the parcel structure	21
5.2 Meta dictionary approach	24
5.3 Identification structure.....	25
5.4 Typical modelling constructs of POM	27
5.4.1 Specialization tree versus composition tree	27
5.4.2 Property specialization.....	27
5.4.3 Divide between specialization and generalization.....	28
5.4.4 Property specialization and cardinality.....	29
5.4.5 Property specialization and alternate ID.....	30
5.4.6 Mapping classes and properties by alternate ID	30
5.4.7 Unit with variable prefix.....	31
5.4.8 Dependent condition	31
5.4.9 Use of dependent condition for time dependent property	32
5.4.10 Class valued property	32
5.4.11 Class selector with class valued property and class reference.....	33
5.4.12 Metamorphic or polymorphic classes	33
5.5 Type system extension for data parcels	34
5.5.1 Extended data types and updates from IEC 61360-2:2002	34
5.5.2 ICID_STRING	34
5.5.3 IRDI_STRING	34
5.5.4 STRING_TYPE and its extensions.....	34
5.5.5 STRING_TYPE and its enumerated simple subtypes.....	35
5.5.6 STRING_TYPE and its enumerated reference subtypes	35
5.6 Structure of a parcelling sheet.....	36
5.7 File name extension	37
5.8 CSV representation of parcel format	37
5.9 Basic use of parcels	38
5.10 Header section.....	38
5.10.1 Categories of instructions.....	38
5.10.2 Mandatory	38
5.10.3 Optional - functional.....	39
5.10.4 Optional - informative.....	39
5.10.5 Comment.....	39
5.10.6 Reserved words	39

5.11	Instruction Column	39
5.11.1	General rule.....	39
5.11.2	Class ID.....	39
5.11.3	Preferred name of the class	40
5.11.4	Definition of the class.....	40
5.11.5	Note for the class.....	41
5.11.6	Alternate class ID.....	41
5.11.7	Super alternate class ID	42
5.11.8	Sub-alternate class ID.....	42
5.11.9	Source language	42
5.11.10	Parcel mode	43
5.11.11	Parcel identifier.....	43
5.11.12	Parcel conformance class identifier	44
5.11.13	Default supplier.....	44
5.11.14	Default version.....	45
5.11.15	Default data supplier	45
5.11.16	Default data version	46
5.11.17	Data object identifier name.....	47
5.11.18	Property ID	47
5.11.19	Preferred name of the property.....	48
5.11.20	Definition	49
5.11.21	Note	50
5.11.22	Data type	50
5.11.23	Unit of measurement.....	51
5.11.24	Requirement	52
5.11.25	Alternative units of measurement	52
5.11.26	Variable prefix for the unit	53
5.11.27	Super property	54
5.11.28	Alternate property ID	54
5.11.29	Super alternate ID.....	55
5.11.30	Sub-alternate ID of property	56
5.11.31	Equivalent property ID.....	57
5.11.32	ID for the unit of measurement	57
5.11.33	Property value format.....	58
5.11.34	Identifier encoding	58
5.11.35	Cell delimiter.....	59
5.11.36	Decimal mark.....	59
5.11.37	Pattern constraint.....	60
5.11.38	Relational constraint	60
5.12	Data section for instances	61
5.12.1	General	61
5.12.2	Enumeration types or non quantitative types.....	62
5.12.3	Level type	62
5.12.4	String type	63
5.12.5	Translatable string type.....	63
5.12.6	Boolean type.....	63
5.12.7	Class reference type (Class instance type)	63
5.12.8	Aggregate type	64
5.12.9	Named type	64

5.12.10	Placement types	67
5.12.11	Entity instance type.....	67
6	Use of parcel for Domain Ontology description	67
6.1	Dictionary as an instance of meta-dictionary	67
6.2	Identification of conjunctive parcels	70
6.3	Roles and definition of dictionary parcels.....	70
6.4	Properties of meta-dictionary (meta-ontology).....	71
6.4.1	Overview of meta-classes	71
6.4.2	Meta-properties for dictionary meta-class	72
6.4.3	Meta-properties for class meta-class	73
6.4.4	Meta-properties for property meta-class.....	74
6.4.5	Meta-properties for supplier meta-class	75
6.4.6	Meta-properties for enumeration meta-class	76
6.4.7	Meta-properties for data-type meta-class.....	77
6.4.8	Meta-properties for document meta-class	77
6.4.9	Meta-properties for object meta-class	78
6.4.10	Meta-properties for UoM meta-class	78
6.4.11	Meta-properties for term meta-class	79
6.4.12	Meta-properties for relation meta-class.....	80
7	Use of parcel for meta-ontology (MO) description	84
7.1	Overview of meta-meta-classes	84
7.2	Meta-properties for class meta-meta-class.....	85
7.3	Meta-properties for property meta-meta-class	85
7.4	Meta-properties for term meta-meta-class.....	86
7.5	Meta-properties for relation meta-meta-class	87
8	Mechanism for structural extension	87
8.1	General.....	87
8.2	Example	88
9	Conformance classes for parcelling spreadsheet	88
Annex A (normative)	Information object registration.....	90
Annex B (normative)	Meta-dictionary file and updates.....	91
Annex C (normative)	Reserved words.....	92
Annex D (normative)	Description examples of data types.....	95
Annex E (normative)	Meta-properties used by normative meta-classes	98
Annex F (normative)	Properties for optional meta-classes.....	119
Annex G (normative)	Predefined classes and properties in Meta-Ontology	130
G.1	General.....	130
G.2	Predefined meta-classes in Meta-Ontology	130
G.3	Predefined meta-properties in meta-ontology	133
Annex H (normative)	Predefined meta-relations in meta-ontology.....	153
Annex I (normative)	Axiomatic properties used by each Meta-meta-class	156
Annex J (normative)	Predefined classes and properties in Axiomatic Ontology	164
J.1	General.....	164
J.2	Predefined meta-classes in Axiomatic Ontology	164
J.3	Predefined meta-properties in Axiomatic ontology	166
Annex K (informative)	Mapping of meta-properties to EXPRESS	173
K.1	EXPRESS mapping for mandatory meta-classes	173

K.2	EXPRESS mapping for optional meta-classes	182
Annex L (informative)	Meta-class properties mapped with DIN 4002.....	186
Annex M (informative)	Use case of relation for units and quantities	199
Annex N (informative)	Guide for the use of placement data types.....	202
N.1	Primitive coordinates.....	202
N.2	EXPRESS language codes	203
Annex O (informative)	Foundation in mathematical-logic.....	205
O.1	Class and property as sets	205
O.2	Property specialization explained by set theory	207
O.3	Mathematical basis of POM	209
Bibliography		212
Figure 1	– Parcel use scenario.....	19
Figure 2	– Parcel architecture as four levels of spreadsheets.....	23
Figure 3	– Components of POM architecture depicted as packages	24
Figure 4	– Schematic diagram of Parcellized Ontology Model (POM)	25
Figure 5	– A generalized enumeration	29
Figure 6	– A specialized enumeration.....	30
Figure 7	– Dependent property, condition, and dependent condition.....	32
Figure 8	– STRING_TYPE and its extensions	35
Figure 9	– ENUM_TYPE and its simple subtypes	35
Figure 10	– ENUM_TYPE and its complex subtypes	36
Figure 11	– Structure of a parcelling sheet	37
Figure 12	– Display example of Default data supplier used for IEC 61968-11	46
Figure 13	– Display example of property ID	48
Figure 14	– Display example of preferred name	49
Figure 15	– Display example of definition	50
Figure 16	– Display example of data type	51
Figure 17	– Display example of unit of measurement	51
Figure 18	– Display example of key.....	52
Figure 19	– Display example of alternative units.....	53
Figure 20	– Display example of variable prefix unit	54
Figure 21	– Display example of Super-property for properties	54
Figure 22	– Display example of alternate property ID.....	55
Figure 23	– Display example of super alternate property ID	56
Figure 24	– Display example of sub-alternate property ID	56
Figure 25	– Display example of equivalent property ID.....	57
Figure 26	– Display example of ID for the unit of measurement.....	58
Figure 27	– Display example of value format	58
Figure 28	– Display example of pattern constraint.....	60
Figure 29	– Display example of relational constraint	61
Figure 30	– Display example of ENUM_INT_TYPE or ENUM_CODE_TYPE.....	62
Figure 31	– Display example of LEVEL_TYPE	63
Figure 32	– Display example of TRANSLATABLE_STRING_TYPE	

Figure 33 – Display example of BOOLEAN_TYPE	63
Figure 34 – Display example of CLASS_INSTANCE_TYPE	64
Figure 35 – Display example of SET OF STRING_TYPE	65
Figure 36 – Display example of LIST OF STRING_TYPE	65
Figure 37 – Display example of LIST OF TRANSLATABLE_STRING_TYPE	65
Figure 38 – Display example of SET OF LEVEL OF INT_MEASURE_TYPE	66
Figure 39 – Display example of SET OF SET OF STRING_TYPE	66
Figure 40 – Display example of NAMED TYPE	67
Figure 41 – Configuration of a dictionary parcel	68
Figure 42 – Parcels for Domain Library and Domain Ontology (Dictionary)	70
Figure 43 – Relation, function, and predication	84
Figure 44 – Definition example of the Relation meta-class	84
Figure M.1 – Example of UoM meta-class for defining units for length	199
Figure M.2 – Sample specification of the relation meta-class for quantity and system of units of measurement	200
Figure M.3 – Quantity and system of units of measurement expressed as relations	201
Figure N.1 – Local coordinate system and the primitive coordinates	202
Figure N.2 – Extracts of EXPRESS codes for placement types	203
Figure N.3 – Extracts of EXPRESS codes for CSG primitives	204
Figure O.1 – Class, property and property-value function	206
Figure O.2 – Class and Property and its characteristic function	206
Figure O.3 – Property specialization by restriction of the domain	207
Figure O.4 – Property specialization by restriction of the codomain	208
Figure O.5 – Property specialization by limiting the selectable function set	208
Figure O.6 – Architecture of POM	209
Figure O.7 – Examples of instances at DL layer	210
Table 1 – Description of the property ID code	48
Table 2 – Example of correspondence within multiple languages	66
Table 3 – Meta-classes for building a domain-dictionary	71
Table 4 – Formula specification for property constraint	82
Table 5 – Conformance classes	89
Table C.1 – Key words for instruction in class header	92
Table D.1 – Description examples for simple data types	95
Table D.2 – Description examples for complex data types	96
Table E.1 – Meta-properties used by dictionary meta-class	99
Table E.2 – Meta-properties used by class meta-class	101
Table E.3 – Meta-properties used by property meta-class	105
Table E.4 – Meta-properties used by supplier meta-class	109
Table E.5 – Meta-properties used by enumeration meta-class	111
Table E.6 – Meta-properties used by datatype meta-class	113
Table E.7 – Meta-properties used by document meta-class	115
Table F.1 – Meta-properties used by object meta-class	120

Table F.2 – Meta-properties used by UoM meta-class.....	121
Table F.3 – Meta-properties used by term meta-class	124
Table F.4 – Meta-properties used by relation meta-class	127
Table G.1 – List of meta-classes in Meta-Ontology	131
Table G.2 – List of meta-properties defined at meta-ontology (MO) layer	134
Table H.1 – List of meta-relations predefined at MO layer.....	154
Table I.1 – Axiomatic properties used by class meta-meta-class	157
Table I.2 – Axiomatic properties used by property meta-meta-class	159
Table I.3 – Axiomatic properties used by term meta-meta-class	161
Table I.4 – Axiomatic properties used by relation meta-meta-class.....	162
Table J.1 – Predefined meta-classes in Axiomatic Ontology.....	165
Table J.2 – List of axiomatic meta-properties defined at Axiomatic Ontology (AO) layer.....	167
Table K.1 – Mapping to EXPRESS modelling languages for meta-properties of dictionary meta-class	173
Table K.2 – Mapping to EXPRESS modelling languages for meta-properties of property meta-class.....	176
Table K.3 – Mapping to EXPRESS modelling languages for meta-properties of supplier meta-class.....	178
Table K.4 – Mapping to EXPRESS modelling languages for meta-properties of enumeration meta-class.....	179
Table K.5 – Mapping to EXPRESS modelling languages for meta-properties of datatype meta-class.....	180
Table K.6 – Mapping to EXPRESS modelling languages for meta-properties of document meta-class	181
Table K.7 – Mapping to EXPRESS modelling languages for meta-properties of object meta-class.....	183
Table K.8 – Mapping to EXPRESS modelling languages for meta-properties of terminology meta-class	184
Table L.1 – Meta-properties for the definition of a class or a property, mapped with DIN 4002.....	187
Table L.2 – Meta-properties for the definition of an enumeration, mapped with DIN 4002.....	193
Table L.3 – Meta-properties for the definition of a data type, mapped with DIN 4002	195
Table L.4 – Meta-properties for the definition of a UoM, mapped with DIN 4002	197

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**STANDARDIZED PRODUCT ONTOLOGY REGISTER
AND TRANSFER BY SPREADSHEETS –****Part 1: Logical structure for data parcels****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 62656-1 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/226/FDIS	3D/229/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 web site 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

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

- Part 1: Logical structure for data parcels;
- Part 2: Application guide for use with IEC CDD;
- Part 3¹: Interface for common information model.

¹ To be published.

STANDARDIZED PRODUCT ONTOLOGY REGISTER AND TRANSFER BY SPREADSHEETS –

Part 1: Logical structure for data parcels

1 Scope

This part of IEC 62656 specifies the logical structure for a set of spreadsheets, used as “data parcels”, to define, transfer and register product ontologies. Such ontology descriptions in other literatures or disciplines are sometimes called “reference dictionaries”. Thus the logical data structure described in this standard is named “Parcellized Ontology Model” or “POM” for short, and each vehicle of transport of the model is called a “parcel”, and may be used for definition, transfer, and registering of a reference dictionary as a collection of metadata, or for similar purposes for instances belonging to a certain class of the reference dictionary. Moreover, this ontology model allows for modelling or modifying an ontology model per se as a set of instance data, thus it enables an ontology model to evolve over time.

This part of IEC 62656 also includes a standard mapping between the meta-data of dictionary parcels in the spreadsheet format conforming to this standard and the meta-data represented in IEC 61360-2 compliant EXPRESS model for dictionary exchange.

It is assumed that a tool supporting this part of IEC 62656 may read and write a set of spreadsheet data whose semantics and syntax are defined in this part of standard, where the physical file structure of the spreadsheets may be based on the CSV (Comma Separated Values) format, typically used in a commercial spreadsheet application, or any other tabular formats including XML schema compatible or convertible to the CSV format.

The spreadsheet interface structure defined in this part of IEC 62656 contains the following:

- Definition and specification of the logical structure and layout of the spreadsheet interface for definition, transfer, and registering of a reference dictionary;
- Definition and specification of library instance data belonging to a class of a reference dictionary described by a set of spreadsheets conformant to this part of IEC 62656;
- Definition and specification of the meta dictionary that enables the definition and transfer of a reference dictionary as a set of instance data conforming to the meta dictionary;
- Definition and specification of the meta-model as data that enables the definition and transfer of a reference dictionary as a set of instance data conforming to the meta-meta-dictionary
- Specification of the mapping between the dictionary data expressed in the spreadsheet format and the EXPRESS model specified by IEC 61360-2/ISO 13584-42 (with some elements of ISO 13584-25);
- Description of the basic semantic mapping between the dictionary data expressed in the spreadsheet formats defined in this part of IEC 62656 and that of DIN 4002.

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

- Explanation of the CSV format per se, used in spreadsheet applications;
- Presentation of the data parcels conformant to this part of IEC 62656, such as colouring and sizing of the spreadsheets;
- Specification of the dictionary EXPRESS model conformant to IEC 61360 or ISO 13584 series of standards;
- Normative definition of the mappings between an IEC 61360-ISO 13584 compliant dictionary and another that is based upon a standard other than IEC 61360/ISO 13584;

- Specification of the maintenance procedure of this part of IEC 62656.

This standard is closely related with ISO 13584-35, and developed as a superset or generalisation of the latter. A major difference with the ISO 13584-35 is that this IEC standard enables updates and evolutions in a meta dictionary consisting of meta classes, by which the changes and evolution of an ontology model is realised as an update and modification of the meta dictionary, just by updates and modifications of the instances of the meta-meta dictionary. With this capability, mapping and interfacing with other ontology standards are also facilitated.

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

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

IEC/TS 62720:2013, *Identification of units of measurement for computer-based processing*

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

ISO 3166-1:2013, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*

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

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

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

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

ISO/TS 13584-35, 2010, *Industrial automation systems and integration – Parts library – Part 35: Implementation resources: Spreadsheet interface for parts library*

ISO 29002-5:2009, *Industrial automation systems and integration – Exchange of characteristic data – Part 5: Identification scheme*

ISO/IEC 6523-1:1998, *Information technology – Structure for the identification of organizations and organization parts – Part 1: Identification of organization identification schemes*

ISO/IEC 6523-2:1998, *Information technology – Structure for the identification of organizations and organization parts – Part 2: Registration of organization identification schemes*

ISO/IEC 8824-1:2008, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 11179-3:2013, *Information technology – Metadata registries (MDR) – Part 3: Registry metamodel and basic attributes*

ISO/IEC 11179-5:2005, *Information technology – Metadata registries (MDR) – Part 5: Naming and identification principles*

3 Terms and definitions

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

3.1

applicable data element type

data element type defined for an item class and which applies to all items belonging to that class

[SOURCE: IEC 61360-1:2009, 2.23]

3.2

application

use of software or a standard in an industrial domain

3.3

Common Data Dictionary

CDD

IEC 61360 CDD

data dictionary (available as IEC 61360-4 database) to be shared among all electrotechnical domains, based on the data model specified by IEC 61360-2 and conforming to the semantic requirements specified by IEC 61360-1

Note 1 to entry: This note applies to the French language only.

3.4

common dictionary schema

information model for reference dictionary based on the EXPRESS data model defined in IEC 61360-2 and ISO 13584-42

3.5

composite property

collection of properties that can be referenced from a class or classes as a single entity

[SOURCE: ISO 13584-501:2007, 3.8]

3.6

conjunctive parcels

parcel sheets that are used together to define a library, reference dictionary, or meta-dictionary

3.7

data

quantities, characters, images, or symbols on which operations are performed by computers and other automatic equipment, and which may be stored or transmitted in the form of electric signals, records on magnetic or magneto-optic medium, or other types of recording medium including pieces of paper

3.8

data parcel parcel

information structure in a form of a level-pair, comprising a set of properties and a set of tuples of values for the set of properties, with an aim to describe a domain data dictionary, a domain data library or an ontological modelling concept

Note 1 to entry: A data parcel is typically implemented and exchanged as a set of spreadsheets, but the medium of implementation or exchange is not limited to spreadsheets; it may be in any other form.

3.9

data type

representation, interpretation, and structure of values used in computer systems and other automatic equipment

3.10

dictionary data dictionary

set of terms with respective identifiers formulated in a canonical syntax and with commonly accepted definitions designed to yield a lexical or taxonomical framework for knowledge representation in a computer interpretable form, which can be shared by different information systems and communities

3.11

dictionary data

dictionary represented as rows of data conforming to the schema collectively defined by a meta dictionary

3.12

dictionary parcel

set of spreadsheets that are used to define in parts a reference dictionary as the instance of the class that each schema header section of the parcel sheet represents

3.13

dictionary parcel client dictionary parcel client system

parcel client that can read or write dictionary parcel, and may have an optional capability to send them to, or receive them from a server system

3.14

dictionary parcel server dictionary server

parcel server that can provide dictionary parcels over Internet

3.15

entity

class of information defined by common properties

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

**3.16
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]

**3.17
globally unique identifier**

identifier that may or may not be based on ISO/IEC 6523 for the global identification of the source of identifier information, and which provides a globally unique identification of a concept without recourse to the linguistic interpretation of the meaning of the letter sequence of the identifier

**3.18
instance**

unary or n-ary values being identified as a distinct member of a class and characterized by the same set of properties

**3.19
international concept identifier
international concept ID
ICID**

globally unique identifier that is used in the parcelling format for identifying each data concept

Note 1 to entry: This note applies to the French language only.

**3.20
is-a relationship**

inheritance relationship defined in the object oriented paradigm

Note 1 to entry: In ISO 13584 the is-a relationship holds between a family of parts and a generic family of parts to which the former family belongs.

[SOURCE: ISO 13584-24:2003, 3.61]

**3.21
is-case-of relationship**

relationship providing a formal expression of the fact that an object conforms to the partial specification defined by another object

Note 1 to entry: In ISO 13584, all the properties and data types visible or applicable for some family of parts may be imported by all the families of parts that declare to be case-of the former family. These properties and data types may then be used to describe the latter families.

[SOURCE: ISO 13584-24:2003, 3.62]

**3.22
is-part-of**

aggregation part/whole relationship

Note 1 to entry: In ISO 13584 the is-part-of relationship holds between a family of constituent parts and a family of assembled parts to which the constituent parts belongs.

[SOURCE: ISO 13584-24:2003, 3.63]

**3.23
item**

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

3.24

library

set of value instances conforming to a class or a set of classes of a dictionary, and the whole or part of the schema definition that describes the set of instances

3.25

library data supplier supplier

organization that delivers a library in the standard format defined in ISO 13584 and is responsible for its content

[SOURCE: ISO 13584-1:2001, 3.1.10]

3.26

library external file

file, referenced from a library delivery file, that contributes to the definition of a supplier library

Note 1 to entry: The structure and the format of a library external file is specified in the library delivery file that references it.

[SOURCE: ISO 13584-24:2003, 3.71]

3.27

library integrated information model LIIM

EXPRESS schema that integrates resource constructs from different EXPRESS schemas for representing supplier libraries for the purpose of exchange and that is associated with conformance requirements

Note 1 to entry: Three library integrated information models are defined in ISO 13584-24 for representing different kinds of supplier libraries.

Note 2 to entry: This note applies to the French language only.

[SOURCE: ISO 13584-24:2003, 3.72]

3.28

library parcel

parcel sheets that are used to define instances of a class whose properties are collated in their header part

3.29

library parcel client

parcel client that can read or write library parcel, and may have an optional capability to send them to, or receive them from a server system

3.30

library parcel server library parcel server system

parcel server that can provide library parcels over Internet

3.31

meta-class

class representing a category of concepts that is used to instantiate modelling constructs for a domain dictionary, such as (domain) classes, properties, enumerations, or predefined terms

3.32

meta-dictionary

set of meta-classes each of which is characterized by a different set of properties called “meta-properties”, and as an instance of which a reference dictionary may be defined and specified

3.33**meta-property**

property that is used to characterize a meta-class and define the syntactic structure of the meta-class

3.34**MOF****Meta Object Facility**

metadata management framework, and a set of metadata services for development and interoperability of model and metadata driven systems

Note 1 to entry: This note applies to the French language only.

3.35**Office Open XML**

set of XML vocabularies standardized as ISO/IEC 29500, being based on W3C XML Schema, for representing word-processing documents, spreadsheets and presentations

Note 1 to entry: The abbreviation "XML" is derived from the full term "eXtended Markup Language".

3.36**ontological entity**

artefact that is used to represent a category of being of things or relationship among them

3.37**parcel client**

client system or application that can read or write parcelling sheets in general, and may have an optional capability to send them to or receive them from a server system

3.38**parcel editor**

software that edits data parcels, which may have a capability to send or receive the parcels over Internet

3.39**parcelling**

act of defining, exchanging, or transmitting information using parcelling sheets defined in this standard

3.40**parcelling tool**

tool that can process parcel spreadsheets in general, including PCL-clients, PCL-editors, and PCL-servers

3.41**parcel server**

server system or application that can provide parcel spreadsheets in general over Internet

3.42**parcel sheet****parcelling sheet**

normalised use and implementation of a data parcel as a spreadsheet

3.43**part**

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

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

3.44

PLIB

dictionary data model defined by ISO 13584-42 which is common with IEC 61360-2

Note 1 to entry: PLIB is an acronym of the ISO 13584 series of standards.

3.45

property

set of characteristic information that conceptually characterizes a class and the value of which may be actually supplied by a library supplier and used to characterize instances (parts) that belong to the class or its subclasses

3.46

property of parts library

PLIB-property

kind of property that is used strictly in the sense of the property defined in IEC 61360-2 and ISO 13584-42 standards

Note 1 to entry: This note applies to the French language only.

3.47

reference dictionary

dictionary that is based on ISO 13584-IEC 61360 data model

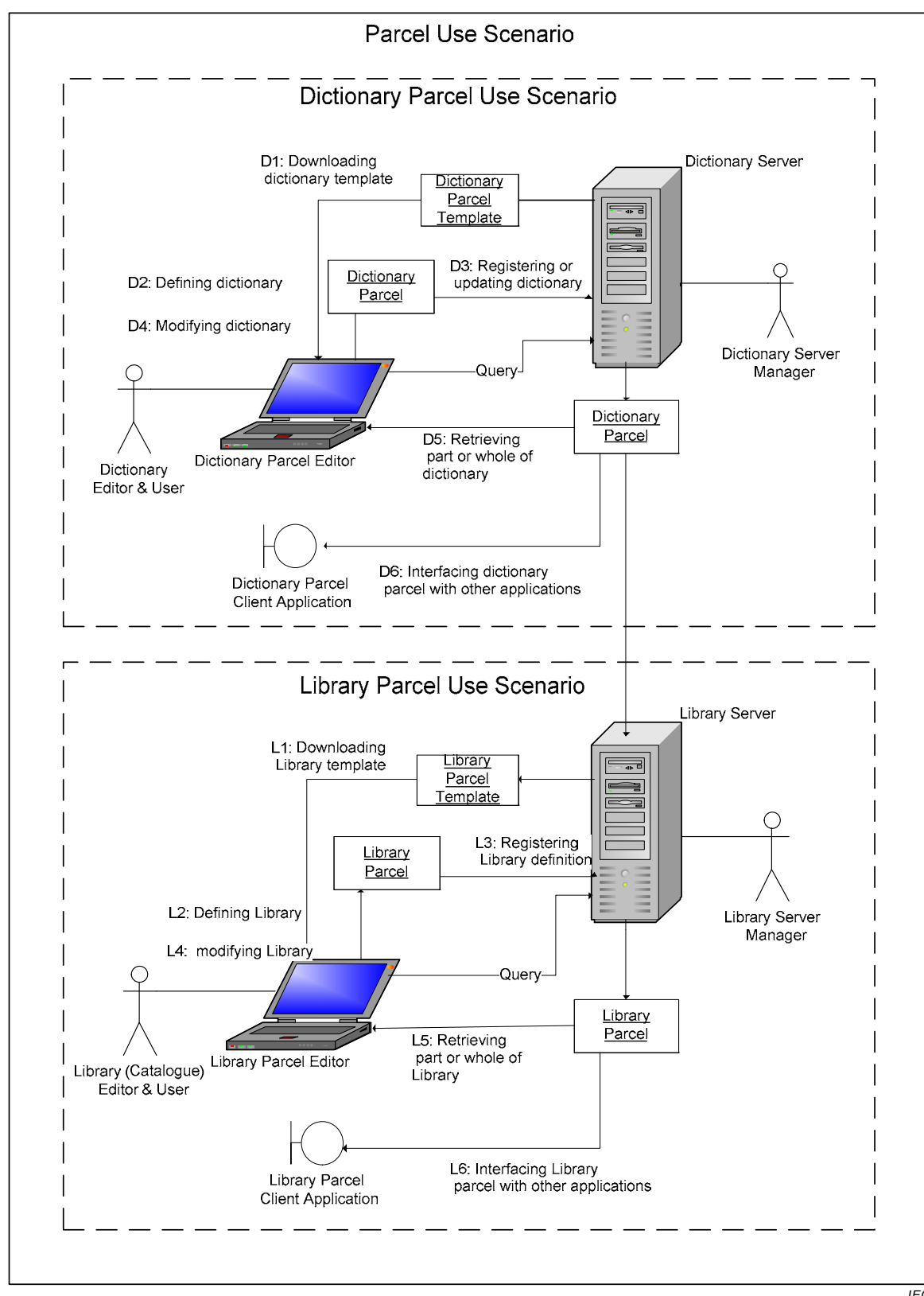
4 Parcel use cases and scenarios

4.1 Typical use cases

The spreadsheet interface structure defined in this standard may be used either for exchanging a dictionary (an aggregation of metadata) or a library (of instances) conforming to a dictionary (metadata) of objects of a given representation layer, including the metadata of the objects that can be mapped one-to-one to the entirety or a subset of IEC 61360-2/ISO 13584-42 dictionary model. Moreover this standard may be applied to transport or store part or whole of a meta-model, such as (but not limited to) IEC 61360-ISO 13584 dictionary model as a payload (content) of the format, by employing a higher abstraction layer (meta-meta data) to which the payload must conform. However, the most typical use case of the format defined in this standard is for the transport and register of the dictionary and library, which embodies the classification and specification of products by their properties and values. Thus, for ease of understanding, the following 4.2 to 4.5 are dedicated to the illustration of how the parcel format can be used in industry for the above use cases.

4.2 Spreadsheet representation of dictionary or library

The spreadsheet interface structure defined in this standard may be used either for expressing a dictionary or a library. Two interface formats for dictionary and library are in fact two different and specialized uses of the same format, whose underlying physical file structure, i.e., “spreadsheet” is widely recognized and processible by many software applications. For the ease of identification of the two uses, and for the distinction of the specialized spreadsheets from general purpose ones, one for dictionary representation of parts library shall be called “dictionary parcel format”, and the other for library representation of parts library shall be called “library parcel format”, in the following part of the standard. Among the parcel formats used either for dictionary or for library, several spreadsheets need to be processed together, in order to represent consistently a dictionary or a library. Hence, the word “parcelling” comes from the mere fact that both formats use a certain number of spreadsheets packed together, each of which is called “parcel” in this standard, and each one of the spreadsheets represents a semantically different group of EXPRESS entities from one another, although syntactically very similar in structure. The typical use scenario of the spreadsheet interface for dictionary and library is illustrated in Figure 1.

**Figure 1 – Parcel use scenario****4.3 Use scenario of dictionary parcel format**

The dictionary parcel format may be used in the following use-cases as typical ones:

- D1: Retrieval of a set of blank dictionary parcels from a dictionary server as a dictionary template for defining a new reference dictionary;
- D2: Definition of the content of a reference dictionary;
- D3: Registering of the content of a reference dictionary in a dictionary server;
- D4: Modification of the content of a reference dictionary;
- D5: Retrieving part or whole of a reference dictionary from a dictionary server;
- D6: Interfacing dictionary parcel with other software applications or engineering tools.

Among the above cases, the use cases D1 through D3 are concerned with defining a new reference dictionary with use of spreadsheet software, while the use cases D4 through D6 are for the reuse of information of a reference dictionary which has been registered beforehand. The benefit of putting a reference dictionary into a set of spreadsheets, i.e., a tabular or matrix form is to make it easy for human readers to understand the content, while from an information processing view point, a spreadsheet format is quite close to the data forms suitable for interfacing with relational databases. In addition, the advantage of assigning a global identifier to each of the attributes of dictionary elements is that item-referencing structure of the data model becomes independent from the change of names of the attributes, necessitated by an evolution of standard or merging of standards, and it helps to map between the attributes of two or more different dictionary standards. Furthermore, the parcel format defined in this standard provides a unique functionality called “alternate ID” that enables identification of an item by either identifier, used temporally or locally. Annex L to this part of IEC 62656 may well be used to establish a mapping between this part of IEC 62656 and the DIN 4002, with the help of alternate ID.

4.4 Use scenario of library parcel format

The library parcel format may be used in the following typical use-cases:

- L1: Retrieval of a set of blank library parcels from a library server as a library template for defining a new library;
- L2: Definition of the content of a library;
- L3: Registering of the content of a library in a library server;
- L4: Modification of the content of a library;
- L5: Retrieving part or whole of a library from a library server;
- L6: Interfacing library parcel with other software applications or engineering tools.

Among the above use-cases, as explained in L6 has the purposes of interfacing with other software applications such as DTP (Desk-Top Publishing) applications, PDM (Product Data Management) applications or ERP (Enterprise Resource Planning) systems or other engineering applications that support spreadsheet formats for data input and output, which are especially important. In fact, many commercial DTP, PDM and analytic engineering tools have a built-in spreadsheet input/output interface, and once the data are read into a commercial spreadsheet application available on the market, conversion of data between one format (or use) and another on the spreadsheet is quite easy and can be almost intuitively done by a domain engineer, who has no special knowledge or prior training about information processing. This demarcates the parcel format from other conventional exchange formats of parts library. In other words, the parcel format may be used as an interface between a tool or application that reads and writes data in a spreadsheet format and another tool or application that processes conventional ISO 13584 dictionary or library data in ISO 10303-21 step physical file format. In this concern, it must be noted that any column which is not headed by a property ID of applicable property which is supposed to be in the class designated by the class ID of the parcel sheet shall be ignored by the system compliant to this standard. This feature is useful for a wide range of applications based on this part of IEC 62656, for information may be embedded in the parcel sheet for other processing, and the same spreadsheet may be used in multiple ways by different applications for their own specific purposes. Thus the parcel formats serve not just for data exchange between ISO 13584 compliant systems, but also as an interface between an ISO 13584 compliant system and

others that are compliant to other standards or protocols but may read and write spreadsheets for data input or output.

4.5 Use scenario of parcel format of higher layers

The parcel format may be used for purposes other than dictionary or library exchange, in particular for defining and exchanging an ontology model as an instance of the model of the immediately upper abstraction layer. The level to be instantiated is called “meta-ontology” abbreviated as “MO”, while the level that gives schema to the MO is called “axiomatic ontology” abbreviated as “AO” in POM. Thanks to this capability, one can extend or modify an ontology model and ultimately it enables schema evolution. For example, adding an extra attribute to a class or to a property is relatively easy in POM. In addition, the method of defining an MO is quite similar to defining a domain dictionary (ontology) with dictionary parcels. Use scenario for defining a meta-ontology may be similar to the dictionary use scenario, explained in a previous section. Meanwhile, the overall structure of a parcel is explained in Clause 5.

5 The Parcellized Ontology Model (POM)

5.1 Overview of the parcel structure

The generic parcel model architecture, or “POM (Parcellized Ontology Model)” for short, has four layers of abstraction, as shown in Figure 2, each of which comprises a level pair. Such a layer is embodied as a set of spreadsheets each sheet of which is a parcel comprising a pair of header and data sections as illustrated in Figure 4. If we explain this in a light of a hierarchical meta-modelling architecture, commonly used in UML (Unified Modelling Language) and MOF (Meta-Object Facility) related literature, a layer consisting of M1 and M0 as header and data section is used for modelling a Domain Library (DL), while a layer consisting of M2 and M1, is used for modelling a dictionary or Domain Ontology (DO). In addition, in order to articulate the modelling structure of the M2, we need a layer corresponding to meta-dictionary or meta-ontology (MO) consisting of the pair M3- M2, as the header section and data section, in a form of data matrix, often embodied as a spreadsheet.

Adding such a superstructure (M3-M2) to the parcel modelling architecture above the dictionary representation layer (M2-M1) makes an implementation of this standard immune to a premature obsolescence, often caused by a partial or minor modification of the data model. The key to achieving this is that updating or modifying a predefined set of meta-meta data as “data” in another layer sufficiently accommodates the gradual evolution of a data model over time. Conversely, constructs of a data modelling standard at M2 being reduced to a set of meta-metadata, can be kept as a set of database records, again in a form of rows and columns of a spreadsheet. As a consequence, another superstructure (M4-M3) at a higher layer may be added just as a logical extension to specify the meta-metadata, in order to explain the constructs of a meta-modelling standard (at M3 layer) in a homogeneous way. In short this layer forms an “axiomatic ontology”. However the Axiomatic Ontology layer (M4-M3) may be kept invariant or constant over time, for its functions are axiomatic and definitions are self-explanatory or axiomatic in nature – not any higher modelling layer is meaningful or significant for our purpose. Thus, the choice of top layer being implemented in a database or being kept in a paper document would not and should not largely influence the effectiveness of this standard. In other words the axiom shall be designed so that it would not unnecessarily restrict the capability of the layers below. Figure 3 illustrates a simplified view of the POM architecture, for ease of understanding.

Meanwhile, regardless of the deepness of the layers of meta-modelling, the underlying structure for accommodating the payload, i.e., the spreadsheet-like structure, and the basic functions to retrieve and manipulate the relevant pieces of data from the payload should remain the same. In all cases, it must be able to represent an ordering among the classes (is_a tree) on a plane of a given layer of abstraction, where the properties of an upper class may be inherited into its subclasses, and a class may be composed of other classes as its parts or components or “blocks” (composition or whole-part hierarchy). In addition, there will be some relational mechanism to relate a member or members of one class to another.

For the above purposes, if a set of dictionary metadata and its instance library data are represented and always stored in a self-similar structure, it is sufficient. The parcel structure being standardized here as a specialised use of spreadsheets is one of possible representation frameworks. Since it is in essence a multi-layer representation of a dictionary and library, it is sufficient to illustrate how a dictionary and its library are used at a particular layer, for example, M1 and M0. Hence, although 5.2 to 5.12.11 are dedicated to the representation of M1 and M0 objects as examples, the basic use scenarios are exactly the same for all other layers.

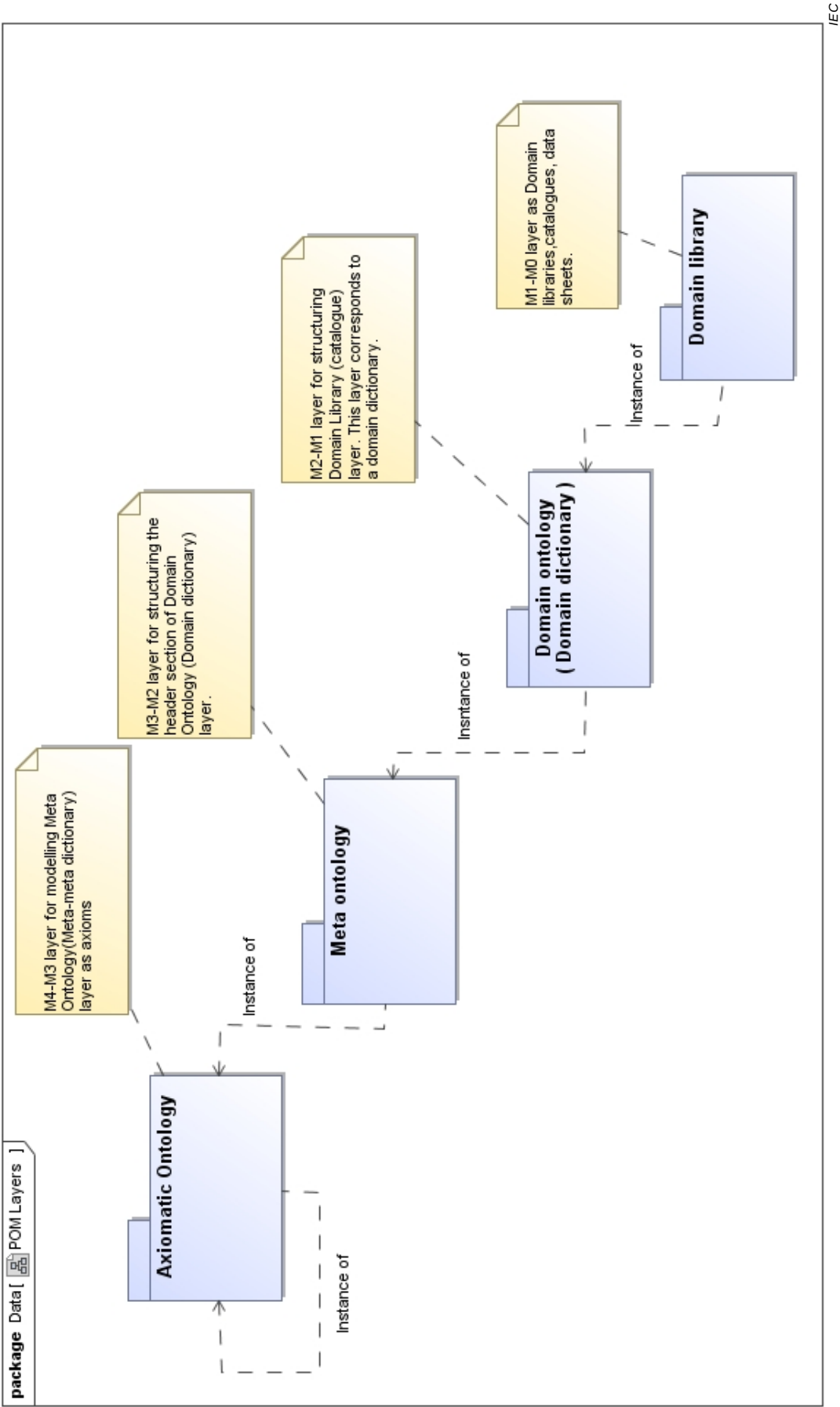
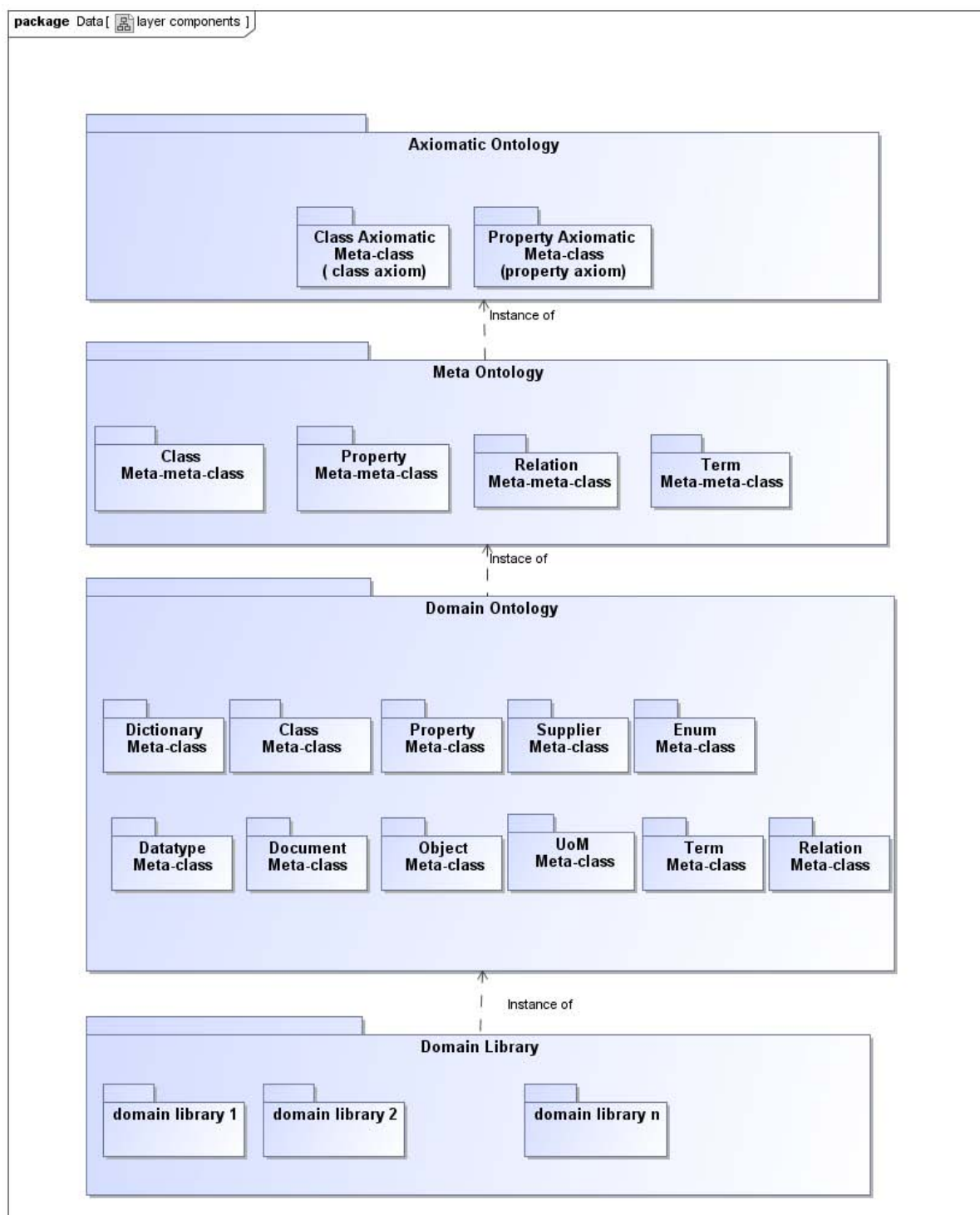


Figure 2 – Parcel architecture as four levels of spreadsheets



IEC

Figure 3 – Components of POM architecture depicted as packages

5.2 Meta dictionary approach

For flexibly modelling either a library or a dictionary on the same base structure as a spreadsheet file, this part of IEC 62656 employs a “meta dictionary approach” with a special emphasis on separating the modelling layer of meta-ontology (meta-dictionary) from that of Domain Ontology (dictionary). In other words, the axiomatic meta-class of M4-M3 as an observer language gives syntax to the meta-ontologies at M3-M2 as a target language. Whereas the meta-ontology as an observer language gives syntax to the Domain Ontology at

M2-M1 as the target dictionary, which in turn describes the semantics of Domain Library, i.e., the real world objects at M1-M0. This layered logical architecture is rooted in Reduction Theorem (See Annex O) in mathematical logic or formal metaphysics, but its application in data modelling is relatively rare. The advantage of taking this approach is that the data model for the reference dictionary represented in this part of IEC 62656 may be flexibly updated or modified without changing the underlying base-structure, modelled and represented as parcels. Since all modelling constructs, such as EXPRESS entities in IEC 61360-2/ISO 13584-42 with a specific name, can be reduced to a set of instances of the meta-ontology layer (depicted in the second column from the left, as MO) defining the meta-classes and meta-properties being identified by a globally unique identifier based on ISO/IEC 6523-1 and ISO/IEC 6523-2, most of the updates and changes in an ontology (dictionary) model may be realized through an addition or a deletion of the instances, or a modification of the values of those instances defined in the meta-meta-classes, i.e. a set of parcels at MO layer.

The meta-dictionary approach explained above is schematically depicted in Figure 4 just for the ease of understanding of the method.

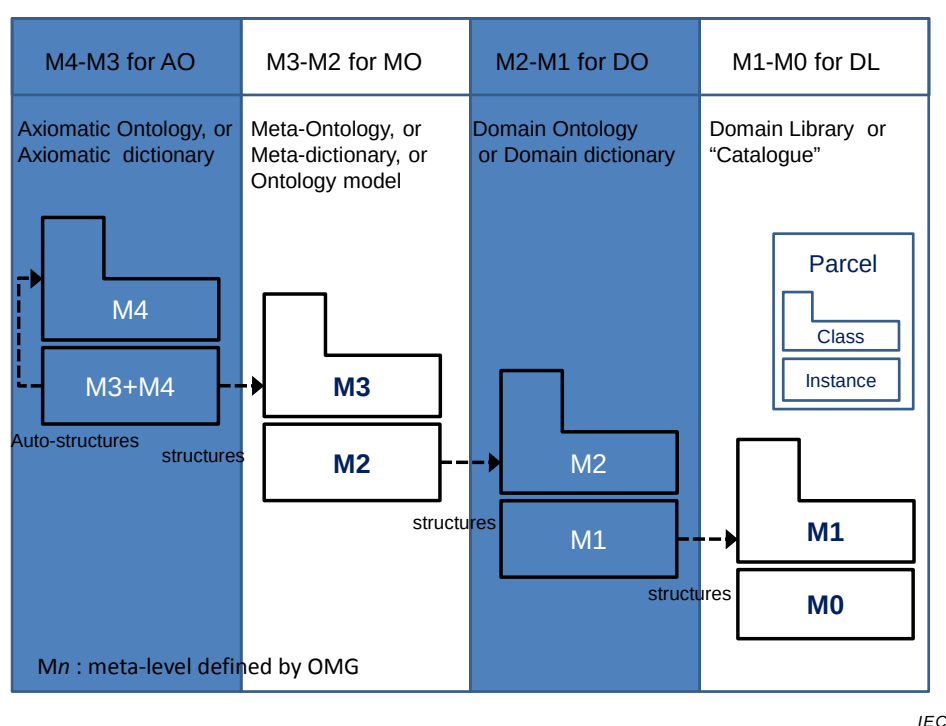


Figure 4 – Schematic diagram of Parcelized Ontology Model (POM)

5.3 Identification structure

This part of IEC 62656 references ISO/IEC 11179 series of standards, in particular ISO/IEC 11179-3 and ISO/IEC 11179-5, as the basic framework for the identification of the types of concepts used to describe elements of data in a parcel sheet. Each data element plays a role similar to “administered item” defined in ISO/IEC 11179-3 as the identifier of a registered item.

Hence, the parcel sheets defined in IEC 62656 may simply be used as a medium for gathering or collecting metadata, or data elements of a reference dictionary, to be registered into a registration authority based on ISO/IEC 11179 series of standards. However, in this standard, such an identifier is also applied and assigned to each construct that structures metadata, and thus its functions need to be further differentiated from that of IRDI (International registration Data Identifier) mechanism for the administered item described in the ISO/IEC 11179-5 standard.

In this standard, we call an element to be identified within a set of metadata or a meta-model as “concept element”. Examples of the concept elements include not only the concepts of class and property, but their constituent key attributes, such as their name, ID, data type, unit of measurement, definition, symbol, note, etc., regardless of their appellations. Conversely, such an identifier of the concept element is referred to as “international concept identifier” or “concept identifier” for short where the context is clear and “ICID” by acronym. However, except for the cases of classifying and identifying the constructs of a metamodel, the function of ICID is identical to IRDI. So it is regarded as an extension of the IRDI.

In parcel format, each ICID is unique across the boundary of organizations, and it has the following sequence:

ICID::= 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, and RAI, DI, and VI themselves may be composed of several data elements. For the coding of RAI of an ICID, ISO/IEC 6523-1 and ISO/IEC 6523-2 shall be referenced. Note that in this standard, two consecutive pound signs, i.e., “##” are used to separate DI and VI, while RAI and DI are separated by one single “#” character. This allows us a short hand notation of the concept identifier eliminating either RAI or VI or both where the context for DI is clear.

In addition to the significant sequence as an IRDI, an ICID may entail a notational appendix, starting with “###”, as specified below;

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

where the segment starting with “###” is just an appended note as mnemonics for human reader guidance on the referenced ICID and it shall not affect the meaning of the ICID. In other words, this may be removed and added anytime. This is a trivial addition, however, quite useful as a mnemonics to indicate what a property or a class an ICID refers to. Needless to say an ICID with or without a note shall have exactly the same effect on machine interpretation.

DI may be entailed by a language identifier, such as “.en”, “.fr” as if it were an extension of DI. Sometimes, there is a necessity to clearly mark a national language variant in the same language. For this purpose, a 2-letter language code based on ISO 639-1 and a 2-letter country code based on ISO 3166-1 can be combined with a hyphen to form a joint-identifier for the language variant. This may be further extended to form a regional, a communal, or an organizational language variant, as long as the third extension contains distinctly different numbers of letters. However, for practical reasons, the ICID restricts the range of the third extension letters from 4 to 10 alphanumeric characters. As a consequence, for example, “en-US-Hawaii” is an example of a correct extension for ICID. To keep the language variant specification interoperable with the one specified by the Internet Engineering Task Force (IETF), it is recommendable to see the reference [6]².

Thus, “P501_P000170##001” means an identification of property whose property identification, so called “property_BSU” in ISO 13584 standard, is “P501_P000170” and its version is 1(one) whose information supplier shall be determined by the context, while “0120/1///13584_501_1#P501_P000170” means a property whose information supplier code is “0120/1///13584_501_1” and its property identification is “P501_P000170”, while the version is to be determined by the context. This shorthand convention is extremely useful in a query when a user does not know exactly the current version of a property or class in a reference dictionary maintained in a dictionary server for retrieving missing pieces of information.

² Numbers in square brackets refer to the Bibliography.

For the meta-classes and meta-properties defined in this standard, the RAIs and versions also apply. They are assumed to have “0112/2///62656_1” as for RAI, and they are assumed to start from version 1 (one), in the initial release of this standard. If the descriptions of RAIs and versions are missing from the meta-classes and meta-properties, it should be understood that they are supplied as default values as above.

5.4 Typical modelling constructs of POM

5.4.1 Specialization tree versus composition tree

Like other ontology modelling languages, the POM may represent both an “is-a” tree and a “has-a” tree for describing a relation among classes. The is-a tree may be referenced in another literature as “a hierarchy of specialization” or “a classification tree”, while the has-a tree may also be referenced as a “composition tree” or “part-whole relationship”. The so called “block structure” as described in IEC 61987-10 is a special use of the latter structure. In this regard, some philosophers consider the composition as a subject belonging to another branch of science called “mereology” than “ontology”. In particular, a proposition such that an ontology description system should be equipped with a capacity to represent a has-a tree has been often debated when the number of classes being held by another class, i.e., the “cardinality” or “multiplicity” is under focus. Nevertheless the number of elements in a product sometimes plays a pivotal role in describing the nature of the product: Having two wheels is an essential characteristic of a transport vehicle called “bicycle”. Meanwhile, a four-wheel-drive car must have 4 motorized wheels for its *raison d’être*. In these cases, the structural capability to describe the number of components or parts is indispensable for a practical ontology description. Thus the POM defined in this standard has a capability to describe the part-whole relationship with an additional capacity to denote the cardinality. The formal terms used in IEC 62656 series of standards for an “is-a” type of tree and a “has-a” type of tree are “specialization tree” and “composition tree” respectively. A specialization relationship is achieved by setting appropriately the “super class” attribute of a class to the global identifier of its super-class. On the other hand a composition relationship between an owner class and its subject classes is achieved only by putting as many class reference type properties (maybe in a form of aggregation) in the owner class as the number of the subject classes, where each property is holding the global identifier of exactly one subject class.

5.4.2 Property specialization

There are few ontology modelling frameworks that allow property specialization. For example, the common dictionary model described in IEC 61360-ISO 13584, from which IEC 62656 is evolved, does not have this capability. However this capability is extremely useful in merging, integrating and mapping two or more different domain ontologies. The property specialization is a capacity to derive many specialized properties from one common property prototype. Conversely, many different properties can be related together by finding or adding their common prototype, i.e., a super-type. The POM simply allows this through setting the super-property attribute of each derived property to the ICID of its super-type, i.e., generalized property.

Note that any value of a specialized property may be assignable to its super-property, provided some of the specialized conditions or some details of the value may be lost in the assignment. For a complete understanding of the theoretical basis of the property specialization, some math-logical knowledge is required as a foundation. See Annex O for this purpose.

Given this special functionality, for preventing abuse, the POM stipulates several conditions for applying the property specialization.

A property *P2* is said to be a specialization of another property *P1* if any value that satisfies *P2* also satisfies *P1* although some conditions, attributes (meta properties) or relations of *P2* may be lost in the assignment. If we consider a property as a mapping of element(s) in a domain to an element in the codomain of the mapping (i.e., the “image” of the mapping), where the domain is a subset of a certain collection, or a “class”, in which a property is

applicable, and the codomain is a set of values of the property in which *P1* or *P2* find meaningful values, then it follows in comparison between the 2 mappings representing *P1* and *P2*, at least either one of the domain or codomain of *P2* must be a subset of that of *P1*. Otherwise, in case the mappings themselves are individually identifiable, then some of them are restricted and eliminated for *P2*. In terms of POM, those requirements may be translatable into the following cases:

- a) Value range (codomain) of *P2* is a subset of that of *P1*;
- b) Domain of *P2* is a subset of that of *P1*;
- c) Name scope of *P2* is a subclass of *P1*;
- d) *P2* has all the conditions of *P1* and additionally some more;
- e) *P2* has all the attributes of *P1* and additionally some more;
- f) Referenced classes of *P2* are subclasses of those referenced by *P1*, where *P1* and *P2* are of the class reference type;
- g) Referenced instances of *P2* are a subset of the instances that are referenced by *P1* where *P1* and *P2* are of the class reference type;
- h) Data type of *P2* is a specialized type of or subject to more constraints, including relational ones, than *P1*;
- i) Enumerated list of values of *P2* are all contained in that of *P1*;
- j) Cardinality range of *P2* is narrower than that of *P1*;
- k) *P2* has a list of fewer alternative units or prefixes than *P1*.

In addition to the above, the following case is also considered as a specialization:

- l) the textual definition, including the NOTE, of *P2* semantically means more specialized and narrower extent than that of *P1*.

The relation from a specialized property to its generic type is formed by an attribute named “super property”. This may be represented as a meta-property named “super property” in a property meta-class. Once defined, the ICID(s) of the super property of each property shall be collated in the row following the instruction keyword “#SUPER_PROPERTY” at DO layer.

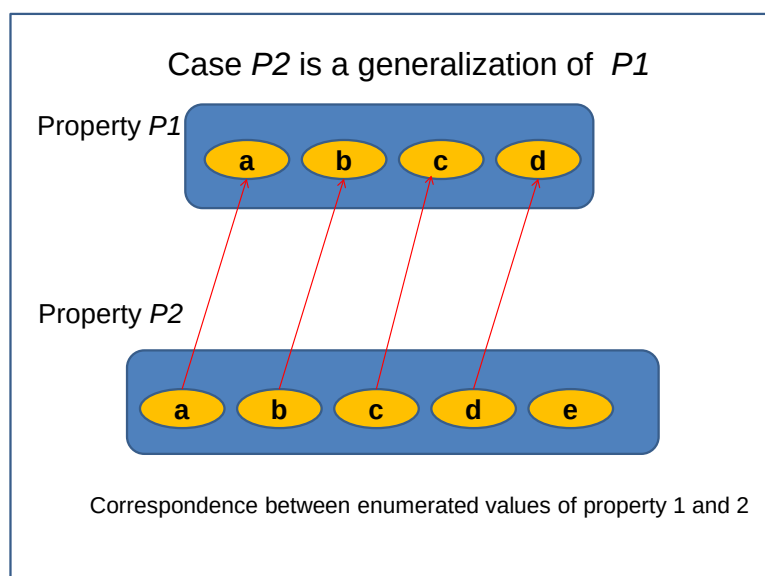
5.4.3 Divide between specialization and generalization

Each value of a specialized property can be regarded as subdivision of a value of its generalized property. If such a relation can be found between two properties, then one is taken as a specialization of another. As a consequence, in the case of the enumeration type properties, if each of the enumerated values of a property can be mapped as a subdivision of some value of another enumeration property, then the former can be considered as a specialized property of the other. This means if the property *P1* has an enumeration (a, b, c, d), and the property *P2* has an enumeration (a, b, c, d, e), in general, *P2* is a generalization of *P1*. However, if “d” and “e” of *P2* are a subdivision of the value “d” of *P1*, then *P2* can be regarded as a specialization of the property *P2*. Such is a case if “d”s in both *P1* and *P2* are typically labelled as “unknown” or “others” and “e” is a concrete value distinct from (a, b, c). Since some of the previously unknown states for *P1* are now classified into a concrete value “e” and separated from the “d” for *P2*, therefore “d” and “e” of *P2* are subdivisions of the “d” of *P1*. The two cases are illustrated in Figure 5 and Figure 6. Instead of “d” as the value representing unknown, one may also provide a value representing the state, called “not-specified”, “no-option” or “null”, if it is appropriate as the value of the property.

In POM by default, there is no provision for relating each element in one enumeration property to another element in another enumeration property. The identification or differentiation of each element shall be done, properly using the ICID of the element and describing or explaining in text the relationship among the elements in the term meta-class. (However, note that an extension can be easily made by creating a “super-term” field in the term meta-class, if necessary).

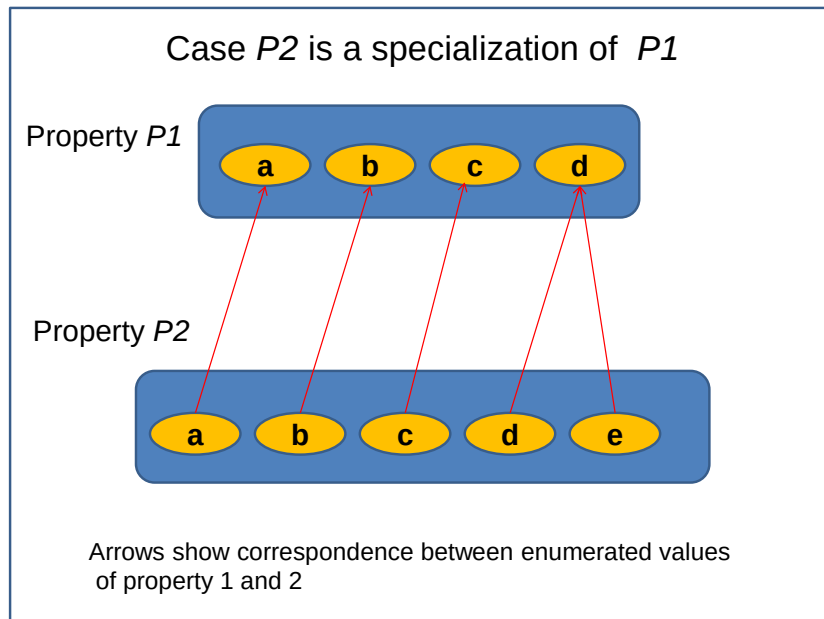
5.4.4 Property specialization and cardinality

In general, a subclass has more or equal number of properties than its superclass. In the same manner, a class that has a property with a cardinality (multiplexity) of four must be more specialized than a class with the same property with a cardinality of one. Here, readers should not confuse the number of instances(multiplicity) used in UML and the number of foldings of a property (multiplexity), i.e., the compoundness of the property. Consequently, a property that references a compounded class shall be more specialized than a property that references a less compound or not compounded class. Note that such a property may be represented as an enumeration of class references. In this case, an enumerated class reference type that references a class with a higher cardinality shall be considered a specialization of a property that references its super-class with less cardinality. This can be translated as a case that a product family with more optional choices for its special parts shall be considered a specialization of a type that has no such options.



IEC

Figure 5 – A generalized enumeration



IEC

Figure 6 – A specialized enumeration

5.4.5 Property specialization and alternate ID

The property specialization is a structural and a-priori relationship that needs to be introduced when one designs or redesigns a Domain Ontology (dictionary) as part of an intent of the ontology. Subtypes of alternate ID, namely sub-alternate ID and super-alternate ID sometimes play a similar role. However, an alternate ID including a sub-alternate ID or super-alternate ID is an *a posteriori* relationship, and may be introduced or added at any time after the design of domain ontologies is complete and put into use. The application of the latter relationship is especially useful when two or more ontologies are registered into a single repository. Note that alternate IDs are not necessarily set in both two properties which are regarded as mutually alternative. Simply setting an alternate ID in one property that a user has the copyright or permission to change must suffice. In contrast, an exact and permanent structural relationship based on property specialization should be established when one needs to perpetuate the relationship and export the knowledge to other users. In other words, alternate IDs are rather temporary relationships and they do not bind the alternate property related by the alternate ID in a way that would not be updated.

Because the property specialization entails a structural change and it is achieved only when one sets permanently an attribute (meta-property), named “super-property” of a property, to the global identifier of another property as the super-property, this shall entail a version-up of the property to be specialized. Conversely the removal or addition of an alternate ID shall not affect the version of a property unless a permanent structural change of the relationship through the property specialization is introduced simultaneously. In brief, the intent of alternate IDs for property is to address the actual cases that two organizations tend to develop two similar properties in parallel and put them to use without knowing each other beforehand. Thus, such a relationship is not necessarily found by an owner of the ontology, but by a user who need to integrate them.

5.4.6 Mapping classes and properties by alternate ID

When one needs to map properties belonging to different ontologies and/or dictionaries, one can describe the mapping functions and their constraints in a relation meta-class. The details of the structure of the relation meta-class are illustrated in a later clause. However, it is important to note that both the relation among classes and the relation among properties can be represented in a relation meta-class. Thus, a sheet of relation meta-class (i.e., relation

parcel) serves as a mapping specification in many cases. It is also worthwhile to note that the relation meta-class has a capability to invoke an external program for actual control of relations, functional calculation or processing of the defined formulae, or translation from a set of property values into another. The minimum requirement for the specification of each relation is that all properties that are related are explicitly defined and the logical dependency among them is clarified.

5.4.7 Unit with variable prefix

Technology rapidly evolves over time: early designs of a circuit within an LSI (Large Scale Integrated circuit), were in the order of micro metres, however, they are currently in the order of nano-metres in state-of-art products. Likewise, when a USB memory was first put on the market, it was in KBs. However, the capacity of the storage devices has continuously increased and it is currently defined in GBs and it will continue to evolve over time. In this situation, to have a fixed prefix to all the units of measurement is not a realistic solution, in particular for the characterization of the products that evolve rapidly over time in their capacity or of their performance. It is also important to recognize that such a change in magnitude of a physical quantity does not and shall not influence the classification of the products, as long as we base the principle of classification on the characterization by properties. In brief, we know that a USB memory is a USB memory whatever its storage capacity might be.

The unit with variable prefix allows a prefix to be selected from a predefined list of prefixes for the same unit of measurement. The actual prefix combined with the base unit shall be indicated in a cell alongside the value cell.

Note that this functionality does not exist in the original IEC 61360-ISO 13584 series of standards. The only possible recourse is to design all these prefixes combined with the base unit as different alternative units in the PLIB systems, but it is not a common practice, nor the true intent of the alternative unit. Meanwhile, even if such alternatives are not available in a PLIB system, one way translation from POM to PLIB is possible when an appropriate fitting of the magnitude of a value is done before an export, whereas from a PLIB property to a POM property being based on the same unit of measurement, it is only necessary to appropriately select a prefix from the list of prefixes of the unit.

5.4.8 Dependent condition

Dependent condition is simply a subtype of condition type property (condition_ DET) that relies on other condition properties. The current PLIB model assumes that a dependent property (dependent _P_DET) may sometimes rely on a set of properties called conditions. However each of these properties cannot rely on any other condition properties in the standard. The dependent condition is designed only to denote a recursive dependency among properties. Two different dependent condition properties may rely on the same condition(s), but the dependency shall not be circular. In other words, the property dependency graph shall form an acyclic graph.

Using a dependent condition does not automatically define any mathematical or physical formula that governs the dependency among the properties. If such an exact modelling of relationship needs to be specified, it shall be done in a relation meta-class in parallel.

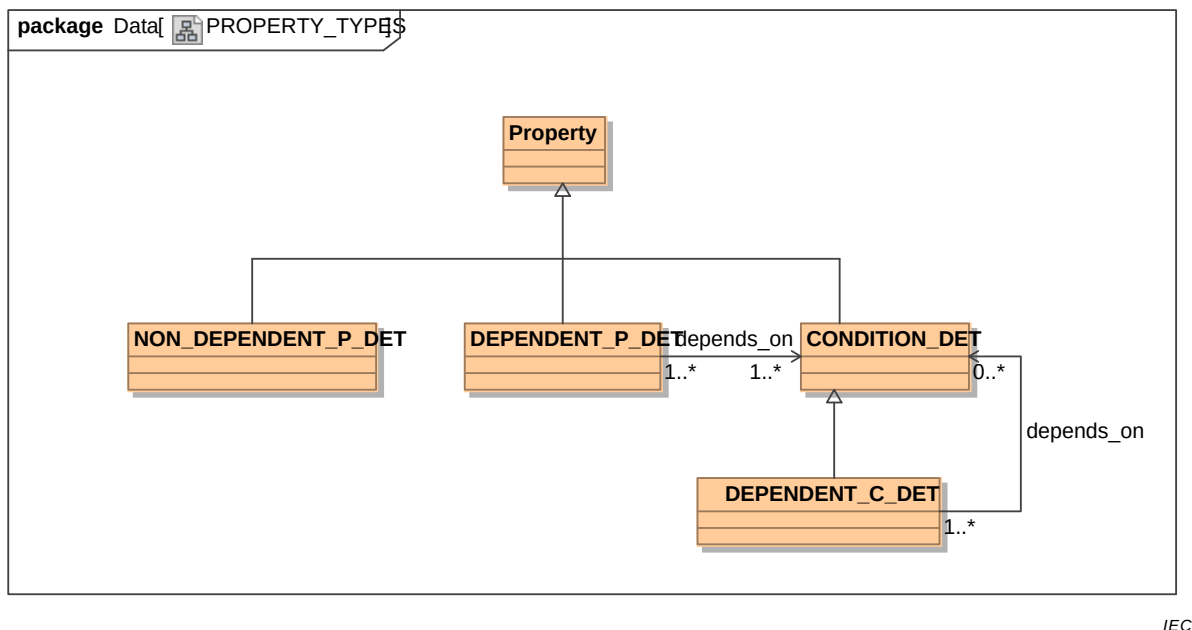


Figure 7 – Dependent property, condition, and dependent condition

Exactly like dependent properties, a dependent condition needs to be declared as “DEPENDENT_C_DET” in a meta property named “Property data element type” (MDC_P020) and its conditions to be depended on shall be set by their respective ICIDs in the meta property named “condition” (MDC_P028).

5.4.9 Use of dependent condition for time dependent property

Most of the products are described by properties which have static values, while some products are especially characterized by properties with time dependent values or values varying over time. Let the first type of the properties be called “static properties” and the latter be called “dynamic properties”. Value of a dynamic property often bears a time stamp recording the measurement time of the value, provided that the property is not designed as an average of several measurements at different moments of time. A property with such a time stamp can be derived from an existing static property by associating a time stamp as a condition to the property. Since some property already has some conditions, it is inevitable that some of the conditions shall be converted into dependent conditions. For example, power consumption of a particular electric component may be highly dependent on the hour of a day. Thus for describing a typical power consumption of the component, it is important to specify the hour of day as condition. Since the power consumption itself may have several other properties as conditions, such as voltage, current, and phase angle, assigning a time stamp to the power consumption means assigning a time stamp to those properties as conditions. In this case, some of the conditions shall be transformed into dependent conditions that depend on another condition containing a time stamp.

5.4.10 Class valued property

Class valued property is a property whose value applies uniformly to all the instances of a class. However the values are assigned in the same manner as other types of properties; being a class valued property does not automatically mean that the value of the property can be assigned simultaneously as the dictionary containing the property is made. In POM as well as PLIB including IEC 61360-2, pre-assignment of a value for a property at the time of dictionary definition is possible only with the enumeration type and CLASS_REFERENCE_TYPE (or CLASS_INSTANCE_TYPE). Thus, to give a single predefined constant value to a class valued property of a class in a given dictionary, the class valued property shall be specified as an enumeration type property with a single choice in its

enumeration list. Note that the POM can make use of any simple type as an enumeration type property, thus such a specification is relatively straightforward.

Class valued properties are assigned to a class through a meta-property of a class meta-class named “class value assignment” (MDC_P017). Note that class valued properties of a class may be inherited from one of its super classes, exactly like other properties. Thus only when those class valued properties are of enumeration type, the values of them are predefined in dictionary for the class. Conversely, in POM, the value of a class valued property other than that of enumeration type may be overloaded by the value given in a lower class into which the class valued property is inherited (as distinct from PLIB). For a class valued type of an enumeration type, the value list shall be the same in all the classes into which the property is inherited into (this is the same as PLIB). Otherwise, the property specialization shall be explicitly used for this purpose.

5.4.11 Class selector with class valued property and class reference

When a class valued property is combined with an enumeration of class references, i.e., ENUM_REFERENCE (or, ENUM_INSTANCE) type, the property of enumerated class references can serve as a class selector for the classes that are referenced. Since the code (value code) of such a type is for human-eye selection, and its paired meaning (value) can be a separate object, the code may be specified as the STRING-encoded value of the class valued property of a class to be referenced (typically the preferred name of the class), and the meaning could be the ICID of the same class by which the latter actual class selection is done. For example, as a class valued property, an enumeration of current types {AC, DC} are given, in a class called “electric motor”, and its subclasses, i.e., AC motor or DC motor, that inherit the property. In this case, either one of the values shall be chosen, representing AC motor or DC motor. In another class, there is a class reference type property that has a list of value pairs, <current type, ICID>. In brief, selecting a current type means selecting an ICID of one of the motor classes to be referenced. This accomplishes a simple class selection using the value code of the class valued property. Note that selecting the value of the class valued property means the selection of the ID of the class to be referenced.

5.4.12 Metamorphic or polymorphic classes

“Polymorphism” is a term used in the IEC 61987 series of standards to describe a kind of selection mechanism similar to that explained in 5.4.11, but rather within an application context. On the other hand, there are some products called “multi-purpose products” each of which is in fact a union of a set of different product classes. Nevertheless, one purpose must be selected when the product is actually put to use. It looks somewhat similar to All-In-One (AIO) products such as a modern facsimile which may serve simultaneously as a fax-machine, a scanner, and a printer, however slightly different in nature. In the case of multi-purpose products, the product provides only one type of functionality in actual use, while in the latter the product is simultaneously multi-functional, or instantly switches between functions. So at any time the functionality of the product may change or at any moment one of the functionalities can be selected. The former type of product may be expressed in this standard as a “metamorphic” product, while the latter as an AIO product. Note that in both cases, the same physical product serves as different types of products, and such a product concept needs to be modelled as a class because it physically exists. Actually some analogue ICs are marketed as multi-purpose ICs and their functionality must be selected before use, for which pin-connections should be selected. A system product or modular product sometimes has a similar characteristic, although in this case, some modules or components of the product are physically replaced to make the product serve for different purposes.

The metamorphic products or concepts can be modelled easily with ENUM_REFERENCE type, or ENUMERATED_INSTANCE type, as explained in 5.4.11.

5.5 Type system extension for data parcels

5.5.1 Extended data types and updates from IEC 61360-2:2002

In POM, there are some extensions to the data type system inherited from IEC 61360-ISO 13584 dictionary model. Extended data types are the following:

- ICID_STRING and IRDI_STRING as extended from STRING_TYPE;
- ENUM_REAL, ENUM_STRING, ENUM_BOOLEAN, ENUM_REFERENCE and ENUM_INSTANCE extended from NON_QUANTITATIVE_TYPE (ENUM_TYPE);
- Measure types of all number types and currency types can be also enumerated.

Note also that for the compatibility with ISO 13584-42:2010 (to-be IEC 61360-2:2012), following subtypes are added for STRING_TYPE:

- DATE_TYPE, TIME_TYPE, and DATE_TIME_TYPE as subtypes of STRING_TYPE:

For the list of data types available for POM, see the tables in Annex D.

5.5.2 ICID_STRING

The ICID_STRING type is a subtype of STRING_TYPE, with a value that has the pattern of an ICID code. In implementation, it is expected that a data validation may be applied to check whether an object surely exists for the referenced ICID code.

5.5.3 IRDI_STRING

The IRDI_STRING type is a subtype of STRING_TYPE, of which the value has the pattern of an IRDI code including ISO 29002-5 compatible code and ICID code. In implementation, it is expected that a data validation may be applied to check whether an object exists for the referenced IRDI code. For ISO_29002_IRDI type, it is virtually treated as an IRDI_STRING, though its type name is different, and no further check on its internal structure is intended in this standard. Note that the IRDI defined in ISO/IEC 11179-6 does not specify what delimiter shall be used for separation among RAI, DI, and VI.

5.5.4 STRING_TYPE and its extensions

The relationships among the newly introduced subtypes of the STRING_TYPE are illustrated in the following diagram. In Figure 8 the subtypes of ENUM_STRING are omitted.

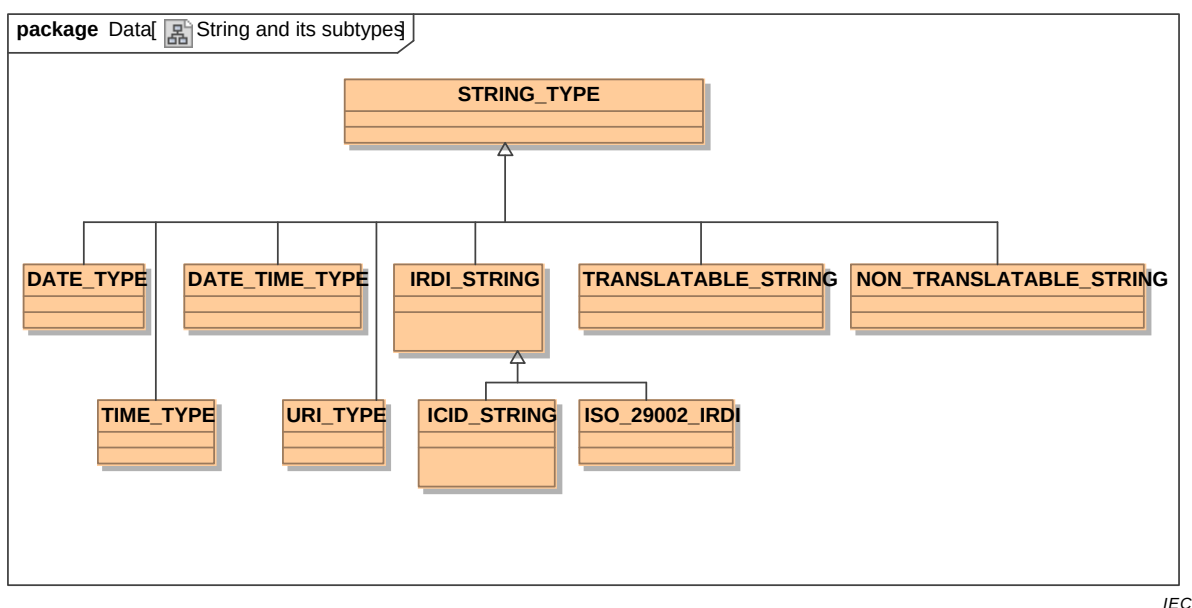


Figure 8 – STRING_TYPE and its extensions

5.5.5 STRING_TYPE and its enumerated simple subtypes

As all the simple types are enumerable in the POM data model, the STRING_TYPE and all its subtypes are also enumerable in the POM data model. In Figure 9, only ENUM_STRING type is shown, however, all the subtypes of the STRING_TYPE may have both the STRING_TYPE and ENUM_TYPE as their parents. Moreover, for number types, all the measure-subtypes and currency-subtypes of what are shown in Figure 9 are also enumerable. For the complete list see Annex D.

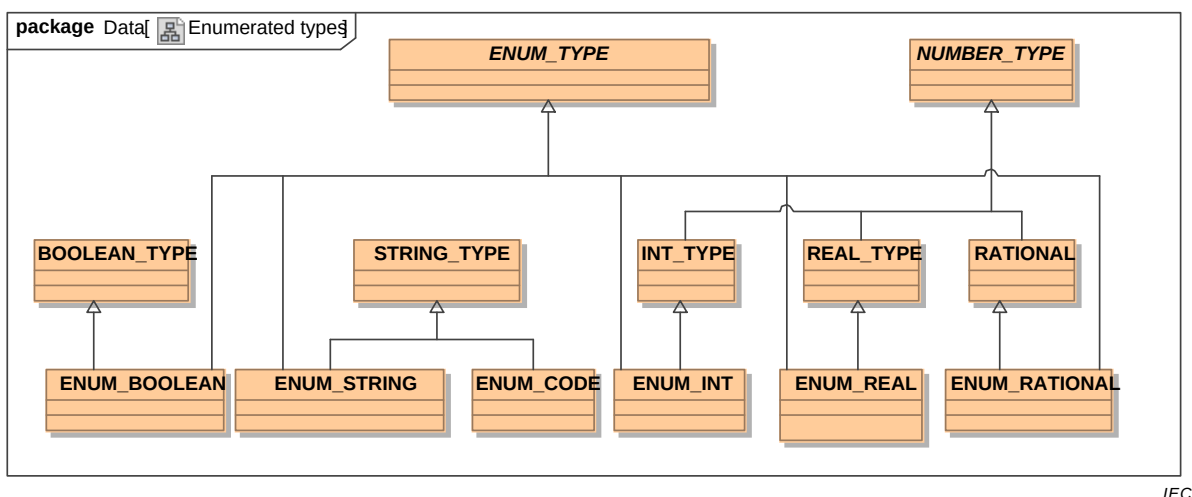


Figure 9 – ENUM_TYPE and its simple subtypes

5.5.6 STRING_TYPE and its enumerated reference subtypes

Among so called “complex types” in PLIB, only CLASS_REFERENCE_TYPE and CLASS_INSTANCE_TYPE are enumerable in POM data model. The reason is simply that the references are of ICID_STRING for both types, although each string signifies a class which may contain several properties. In this sense, ENUM_REFERENCE and ENUM_INSTANCE are no different from ENUM_STRING in their treatment.

Note that in PLIB data model, there is no difference between the two data types, or to be more precise, the `CLASS_REFERENCE_TYPE` is simply a renaming of the `CLASS_INSTANCE_TYPE` that has been used in an older edition of ISO 13584-42 or IEC 61360-2, for ease of understanding to end users. However, in the POM data model, they are similar but have a significant difference in the management of their instances:

In the case of `CLASS_INSTANCE_TYPE`, the instances of the class (i.e., the “part” class) pointed by a property of `CLASS_INSTANCE_TYPE` are embedded in the class where the property resides, i.e, the “whole” class. Thus, if the whole class where the property of `CLASS_INSTANCE_TYPE` is destroyed then the part class where the instances are stored shall be destroyed. While in case of `CLASS_REFERENCE_TYPE`, no such constraint on instances exists. The lives of instances of both the whole and part classes are regarded as independent or not particularly specified. This roughly corresponds to the difference between composition and aggregation relationships in UML model. See Figure 10 for understanding the relationship that complex subtypes of `ENUM_TYPE` have with their super types.

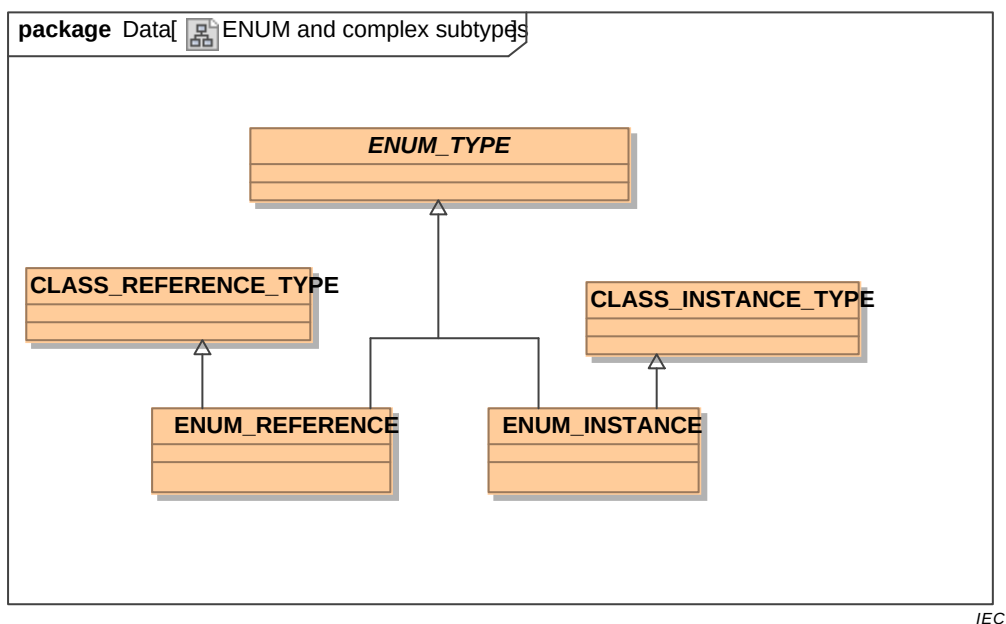


Figure 10 – `ENUM_TYPE` and its complex subtypes

5.6 Structure of a parcelling sheet

Figure 11 illustrates the general structure of a parcelling sheet which is independent of the type of data content, i.e., dictionary or library, carried by the parcelling sheets. For ease of illustration, both information supplier and version are omitted from the property ID.

Instruction column		Cell columns				
Class header section	#SOURCE_LANGUAGE:=EN					
	#CLASS_ID:=C001					
	#CLASS_NAME.EN:= Regional cuisine					
	#PARCEL_CC:=1					
Schema header section	#PROPERTY_ID	P001	P002	P003	P004.EN	P005.EN
	#ALTERNATE_ID	APID001	APID002	APID003	APID004	APID005
	#PROPERTY_NAME.EN	Maker ID	Salt	Sugar	Locality	Speciality
	#DEFINITION.EN	maker ide...	mass of salt in..	mass of sugar ..	place or ...	cuisine that...
	#DATATYPE	STRING_TYPE	REAL_MEASURE_TYPE	REAL_MEASURE_TYPE	STRING_TYPE	STRING_TYPE
	#VALUE_FORMAT	M..100	NR2..7.4	NR2..7.4	M..100	M..100
	#UNIT		kg	kg		
Data section	#REQUIREMENT	KEY			KEY	KEY
		HM	0.01	0.02	Kagoshima	Sakezushi
		NO	0.02	0.01	Vienna	Schnitzel
		NM	0.03	0.02	Osaka	Takoyaki
		LO	0.01	0.03	Sichuan	Mapodoufu
		WW	0.02	0.01	Hagen	Sauerkraut
		GP	0.07	0.04	Poitiers	Pot au feu

IEC

Figure 11 – Structure of a parcelling sheet

As shown in the figure, a parcel sheet may be divided into two horizontal sections; a header section and a data section. The parcel sheet may also be divided vertically into an instruction column, and a set of cell columns. As a further subdivision of the horizontal division, the header section may be divided into a class header section, and a schema header section. The class header section describes the information about this parcelling sheet, thus it may also be called the “class header”, while the schema header section describes the data schema of this spreadsheet as a class, thus it gives the “schema” of the instance data described in the data section of this parcelling sheet.

5.7 File name extension

When it is necessary to differentiate the parcel format conforming to this standard from a spreadsheet for general use, the file name of a file for library parcelling shall be terminated by a file name extension “.pcl”, signifying the parcel format for library. For the specification of dictionary, on the other hand, the extension “.pcd” shall be used.

5.8 CSV representation of parcel format

When representing a parcel sheet in CSV format, data elements in each definition line are separated by a comma “,”, and all the values for a property delimited by a pair of commas shall be arranged in a column. If a data value itself contains comma, then the data value shall be enclosed between a start and an end double-quotation mark, such as “ , “10,000”, ” where 10,000 is a value and the outer-most pair of double quotation marks are just added for marking the quotation into this document.

It is also assumed that each line shall be terminated by a line feed code used in many popular spreadsheet software applications, i.e., “0d0a” in hexadecimal coding which is automatically added by most of the typical spreadsheet applications that are compatible with a leading spreadsheet software commercially available on the market.

Note that a parcel may be exchanged using other formats, such as Office Open XML or some other proprietary formats of software³ vendors. Hence this standard only defines the spreadsheet view of the parcel data, and lists the CSV representation as a referential example. As long as data may be convertible and it generates the same spreadsheet view, any other format may be used for actual transaction.

5.9 Basic use of parcels

The library use of the spreadsheet structure, or library parcel format for short, shall consist of a set of parcelling sheets, each of which contains in its header section, a list of applicable properties that characterize a class of a given dictionary, and some rows of the instance data in data section that belong to this class. When some of those properties are of enumeration type, i.e., either `non_quantitative_string` type or `non_quantitative_int` type, then an additional parcelling sheet called “enumeration parcel” may accompany the parcel sheet as a library of instances, and it may list all the option codes and meanings for the enumeration types used in the library parcel. But this accompaniment of enumeration parcel is not mandatory for the use of library parcel, because although the presence of such an enumeration sheet may help users to enter a precise option value into a property of enumeration type, it is not indispensable as long as the users know the enumerated values. And in more typical cases, the sender and receiver will share the dictionary, prior to the library data exchange.

As noted earlier, in each parcelling sheet, there is a header section and a data section. The header section consists of lines each of which shall start with a pound sign “#” and then entail a reserved word. If the word is not recognized as a reserved word, then the line shall be simply treated as a comment line.

5.10 Header section

5.10.1 Categories of instructions

Each line shall be interpreted using the semantic function described in the cell of the first column. As noted above, instructions to and from an information system conformant to this standard which reads or writes a parcel format file shall be started with a letter “#”. There are four categories of instructions:

- Mandatory;
- Optional – functional;
- Optional – informative;
- Comment.

The words following the symbol “#” and designated in the above categories except the one for comment line shall be treated as reserved words, and shall not be used for other purposes.

5.10.2 Mandatory

Reserved words of this category following the letter “#” are mandatory, therefore, they shall be present in a library exchange file. They are required for any system conformant to this part of IEC 62656 for analysing the property definitions specified by a user. Thus they are also functional in nature.

³ One of such software tools is the MS-EXCEL ® of Microsoft ® Corporation.

The MS-EXCEL ® of Microsoft ® Corporation is an example of a suitable product available commercially. This information is given for the convenience of users of this standard and does not constitute an endorsement by IEC of this product.

5.10.3 Optional - functional

If a reserved keyword of this category following the letter “#” is present in a file, the values of the cells in the line following the second element shall be processed according to the function implied by the keyword. Thus changing the description in the line may change the behaviour of the system which interprets the parcelling sheet. An instruction of this category may or may not exist in an exchange file.

5.10.4 Optional - informative

If a reserved keyword of this category following the letter “#” is present in a file, the values of the cells in the line following the second element are provided to the users of this parcelling sheet just as informative messages by a system conformant to this standard. Since the instruction is informative, changing the information contained in the line would not change the behaviour of the system which interprets the parcelling sheet. An instruction of this category may or may not exist in an exchange file.

5.10.5 Comment

The line started with a letter “#” and not followed by a reserved word shall be interpreted as a comment line. It is used to note a comment within the line. This means an instruction of the categories of mandatory, optional - functional, and optional - informative may be commented out with an additional “#” inserted before the instruction word.

5.10.6 Reserved words

The reserved words of the categories specified in 5.10.1 are summarized in Table C.1 of Annex C.

5.11 Instruction Column

5.11.1 General rule

There are three general description rules that every keyword in the instruction column shall follow:

- Rule 1 – comment out
 - Any line including a line with a keyword may be commented out using “#”.
- Rule 2 – precedence of mandatory instructions
 - Mandatory instructions shall be described before the start of data section used for the description of instances, i.e., library data.
- Rule 3 – precision for the description of keywords
 - Keyword shall be written exactly as specified in this document.

The following 5.11.2 to 5.11.38 define the specific rules and meanings of each keyword.

5.11.2 Class ID

Keyword	#CLASS_ID
Name:	Class identifier
Definition:	globally unique identifier of a class based on ISO/IEC 6523, which is characterized by the properties described in the same parcel, and to which the rows of instance data contained in the data section belong
Description:	Character sets used for the class ID shall be in accordance with the class BSU specification defined in IEC 61360/ISO 13584-42 standards, in case

the parcel is used for the exchange of dictionaries conformant to ISO 13584-IEC 61360 common dictionary schema.

Category:	mandatory
Format:	The keyword “#CLASS_ID” shall be described in the first column and the class code shall be described after the keyword separated by the symbol “:=” (colon-equal). The cells in the second column and after shall be ignored.
Example 1:	#CLASS_ID:=0112/1///13584_501_1#P501_C000001##000000001,
Example 2:	#CLASS_ID:=0112/1///13584_501_1#P501_C000001,

5.11.3 Preferred name of the class

Keyword:	#CLASS_NAME.<lang>
Name:	Preferred name of the class
Definition:	preferred name of the class specified by the class ID, optionally in the language designated by a language code
Description:	The information is optional and informative in a library parcel format because by the specification of a class ID already designates mandatory pieces of information about the class, including the preferred name(s). When an appropriate name is not available for the corresponding language, the cell may be kept open.
Category:	optional - informative
Format:	For IEC 61360/ISO 13584 conformant classes, the class name shall follow the naming convention that is defined in IEC 61360-1/ISO 13584-42.
Example:	#CLASS_NAME.EN:=Environment measuring instrument,

5.11.4 Definition of the class

Keyword:	#CLASS_DEFINITION.<lang>
Name:	Definition of the class
Definition:	statement about the meaning and concept of the class, in the language designated by a language code
Description:	The definition available in this field is informative, and supposed to be provided by a parcelling tool which holds the dictionary to which the class belongs. When an appropriate definition is not available in the corresponding language, the cell may be kept open.
Category:	optional - informative

Format: Definition shall follow the rules and principles described in Part 2 of the ISO/IEC Directives

Example: #CLASS_DEFINITION.EN:= set of voltage amplifiers of which each amplifier can be described with the same group of data element types,

5.11.5 Note for the class

Keyword: #CLASS_NOTE

Name: Note for the class

Definition: statement that provides additional information about the definition of the class that is essential for the understanding of the definition of the class designated by the class ID described in the same parcel

Description: This may be used for describing information useful for the interpretation of the definition of the class. The information is provided by a system conformant to this standard, and changing the value for this keyword shall not change the behaviour of the system.

Category: optional - informative

Format: A string of alphanumeric characters of any length, except “:=” as a part of the string.

Example: #CLASS_NOTE.EN:= This class corresponds to the class in IEC 61360-4 CDD

5.11.6 Alternate class ID

Keyword: #ALTERNATE_CLASSID

Name: Alternate class ID

Definition: alternate class identifier of the class specified by the class ID noted in the same parcel

Description: This may be used for or for identification within a local system, or for mapping with an external system, another International standard or some other standard that uses a different identification scheme than that is used for the class ID, for historical or domain specific reasons.

Category: optional - informative

Format: A string of alphanumeric characters of any length, excluding “:=” or “,” as a constituent.

Example: #ALTERNATE_CLASS_ID:=147/101001///ECALS_10_1#XJA418,

5.11.7 Super alternate class ID

Keyword:	#SUPER_ALT_CLASSID
Name:	Super-alternate class ID
Definition:	alternate class identifier of the class specified by the class ID in the same parcel, where the extent of instances of the class indicated by this alternate ID includes the extent of instances of the class indicated by the class ID
Description:	This may be used for identification within a local system, or for mapping with an external system, another international standard or some other standard that uses a different identification scheme than that is used for the class ID, for historical or domain specific reasons.
Category:	optional - informative
Format:	A string of alphanumeric characters of any length, excluding “:=” or “,” as a constituent.
Example:	# SUPER_ALT_CLASSID:=147/101001/// ECALS_10_1#XJA418,

5.11.8 Sub-alternate class ID

Keyword:	#SUB_ALT_CLASSID
Name:	Sub-alternate class ID
Definition:	alternate class identifier of the class specified by the class ID in the same parcel, where the extent of instances of the class identified by this alternate ID is included in the extent of instances of the class identified by the class ID
Description:	This may be used for identification within a local system, or for mapping with an external system, another International standard or some other standard that uses a different identification scheme than that is used for the class ID, for historical or domain specific reasons.
Category:	optional – informative
Format:	A string of alphanumeric characters of any length, excluding “:=” or “,” as a constituent.
Example:	#SUB_ALT_CLASSID:=147/101001/// ECALS_10_1#XJA418,

5.11.9 Source language

Keyword:	#SOURCE_LANGUAGE
Name:	Source language

Definition:	designation of the source language in line with ISO 639, in which the text of dictionary definition was originally made
Category:	optional - informative
Format:	The keyword “#SOURCE_LANGUAGE” and its assigned value shall be described in the instruction column. The cells in and after the second column shall be ignored. The language code according to ISO 639 enables the identification of the language used as the original.
Mapping:	It shall be mapped to the following entity of IEC 61360-2 or ISO 13584-42; “dictionary_element.administration\administrative_data.source_language”.
Example1:	#SOURCE_LANGUAGE:=EN-US,
Example2:	#SOURCE_LANGUAGE:=FR,
Example3:	#SOURCE_LANGUAGE:=JA,

5.11.10 Parcel mode

Keyword:	#PARCEL_MODE
Name:	Parcel use mode
Definition:	Designation of the use mode of the parcel indicating if the parcel is intended for defining a full content of an ontology or a library or just part of it for updating the content or for test purpose
Description:	When FULL is specified it is possible to conduct an integrity check within the conjunctive parcels. If UPDATE is specified, combination of the conjunctive parcels and the target content to be updated forms a complete content and an integrity check should be possible. If PARTIAL is specified, no integrity check is intended for the content.
Category:	optional - functional
Format:	Either FULL, UPDATE or PARTIAL may be assigned as value
Example:	#PARCEL_MODE:=PARTIAL,

5.11.11 Parcel identifier

Keyword:	#PARCEL_ID
Name:	Parcel identifier
Definition:	designation of conjunctive parcels, i.e., the parcels that are used together in the same unit of exchange, and in parts they describe a dictionary or library

Description: Conjunctive parcels are required to have the same alphanumeric letter sequence for this identifier. PARCEL_ID may include neither a comma nor a double quotation mark. When this ID is omitted for dictionary parcelling, the other parcels processed together shall be construed as conjunctive parcels. In case of library parcelling, it may be used to couple library instances with part of the relevant dictionary information, especially with an enumeration parcel, in order for an application to properly display the meanings of option codes for non-quantitative (enumeration) type property values.

Category: optional - functional

Format: alphanumeric

Example: #PARCEL_ID:=2006-06-25 08:19:49,

5.11.12 Parcel conformance class identifier

Keyword: #PARCEL_CC

Name: Parcel conformance class identifier

Definition: designation of the conformance class number according to either IEC 62656 or ISO 13584-35 conformance classes, for the data carried in the data section, in the parcel

Description: When there is a local extension, the value for parcel conformance class identifier shall be clearly marked for such an extension.

Category: optional - informative

Format: In case of IEC 62656-1 based conformance classes, identifiers assignable to this meta property shall be selected according to the Table 5. If the conformance class ID = 2 or 2x is selected and the assumed meta-ontology (MO) is a specialized version of IEC 61360-2, a token may be passed in a pair of parentheses placed immediately after the conformance class ID (CCID). In case of ISO 13584-35 based conformance classes an integer value between one (1) and ten (10) shall be used, and the acronym "P35" shall be noted in the parentheses

Example: #PARCEL_CC:=9(P35),

5.11.13 Default supplier

Keyword: #DEFAULT_SUPPLIER

Name: Default supplier

Definition: prefix to be added to the shorthand notations of class ID and property ID in the header section, with an aim to make a full supplier identifier sequence designating the information supplier

Description: This shorthand notation of IDs shall be used only in the header section. This short cut notation is allowed for definition purpose only, and it is assumed that the parcelling file containing the shorthand notation would be pre-processed to yield a full identifier sequence, before it is actually sent to external systems compliant to this part of IEC 62656.

Category: optional - functional

Example: #DEFAULT_SUPPLIER:=0112/1///62656_1_1,

5.11.14 Default version

Keyword: #DEFAULT_VERSION

Name: Default version

Definition: postfix to be appended to the shorthand notations of the class ID and property ID in the header section to denote the version of them, with an aim to make a full identifier sequence

Description: This shorthand notation of IDs shall be used only in the header section. This short cut notation is allowed for definition purpose only, and it is assumed that the parcelling file containing the shorthand notation would be pre-processed to yield a full identifier notation, before it is actually sent to external systems compliant to this part of IEC 62656. The version shall be issued in an ascending order.

Category: optional - functional

Format: Only positive integer number of one to maximum nine digits shall be used. Thus the allowed version is an integer between 1 and 999999999. Leading zeros before the first non-zero digit shall be ignored. Thus both 000001 and 001 are equal to 1.

Example: #DEFAULT_VERSION:= 1,

5.11.15 Default data supplier

Keyword: #DEFAULT_DATA_SUPPLIER

Name: Default data supplier

Definition: prefix to be added to the shorthand notations of ICID in the data section, with an aim to make a full identifier sequence

Description: This prefix shall be specified and applied column by column to the shorthand notations of instances in the data section of a parcel. This shorthand notation is for the convenience of definition and for the ease of visual recognition of the instance data by human users. If a parcel containing this shorthand notation is to be sent to an external system that is not capable of processing the shorthand notation, the data should be pre-processed to yield a full notation, before it is actually sent to

the external system.

The shorthand notation of the data is possible only when the property is of `STRING_TYPE`, and transformable to an `ICID_STRING`

Category: optional - functional

Example: `#DEFAULT_DATA_SUPPLIER, 0112/2///61987_11_1, , ,`

Spreadsheet view: It will be displayed by a spreadsheet application as in Figure 12.

#CLASS_ID:= MDC_C003;				
#CLASS_NAME:= Property meta class;				
#DEFAULT_SUPPLIER:= 0112/2///IEC62656_1_1;				
#PROPERTY_ID	MDC_P001_6	MDC_P002_1	MDC_P004_1	MDC_P022
#PROPERTY_NAME	Code	Version Number	Preferred name	Data type
#DATA_TYPE	ICID_STRING	STRING	STRING	STRING
#DEFAULT_DATA_SUPPLIER	0112/2///61987_11_1			
#DEFAULT_DATA_VERSION	001			
	AAD001	001	die ID	STRING
	AAD002	001	die name	STRING
	AAD003	001	die version	STRING

Figure 12 – Display example of Default data supplier used for IEC 61968-11

5.11.16 Default data version

Keyword: `#DEFAULT_DATA_VERSION`

Name: Default data version

Definition: postfix to be appended to the shorthand notations of class ID and property ID in the data section to denote the version of the items, with an aim to yield a full identifier sequence

Description: This postfix shall be specified and applied column by column to the shorthand notations of instances in the data section of a parcel. This shorthand notation is for the convenience of definition and for the ease of visual recognition of the instance data by human users. If a parcel containing this shorthand notation is to be sent to an external system that is not capable of processing the shorthand notation, the data should be pre-processed to yield a full notation, before it is actually sent to the external system. The shorthand notation of the data is possible only when the property is of `STRING_TYPE`, and transformable to an `ICID_STRING`

Category: optional - functional

Format: Only positive integer numbers of one to maximum nine digits shall be used. Thus the allowed version is an integer between 1 and 999999999.

In case of a number with preceding zeros, such as 001 or 000000010, zeros placed before a positive integer number such as 1 or 10, shall be ignored, when it is converted into a value for the version.

Example: #DEFAULT_DATA_VERSION, ,001

Spreadsheet view: It will be displayed by a spreadsheet application as in Figure 12.

5.11.17 Data object identifier name

Keyword: #OBJECT_ID_NAME

Name: Data object identifier name

Definition: name of the data object identification system, by which each row of instance in the data section shall be uniquely and globally identified

Description: Currently either GUID or UUID may be selected for use.

Category: optional - functional

Example: #OBJECT_ID_NAME:=GUID,

5.11.18 Property ID

Keyword: #PROPERTY_ID

Name: Property identifier

Definition: globally unique identifier of a property based on ISO/IEC 6523 referenced for the definition of instances in the data section of the same parcel

Description: This corresponds to the specification of the property BSU of a known applicable property that is used for library exchange in ISO 13584-25 in accordance with the common dictionary schema defined in ISO 13584-42 and IEC 61360-2 standards.

When a property ID listed in the #PROPERTY_ID line is not recognizable as an applicable property of the class designated by the class ID described in the same parcel, all the information in the cells in the same column designated by the property ID shall be ignored as comments.

Category: mandatory

Format: The keyword “#PROPERTY_ID” is described in the first column. The ICIDs of properties are described in the second and following columns . A

property ID shall be encoded as in the following table.

Table 1 – Description of the property ID code

level	property ID code description
level 1	supplier_bsu.code + sep_id + property_bsu.code + sep_cv + property_bsu.version
level 2	supplier_bsu.code + sep_id + class_bsu.code + sep_cv + class_bsu.verison + sep_id + property_bsu.code + sep_cv + property_bsu.version
level 3	supplier_bsu.code + sep_id + property_bsu.code
level 4	property_bsu.code + sep_cv + property_bsu.version
level 5	property_bsu.code

NOTE 1 If the data type of the property is level_type, a column is allocated for each level defined in the data type. Such property code is described as “property code” + “sep_id” + “levels”. The levels are identified by three-letter alphabets and are described as in the following way: “MIN”, “NOM”, “TYP”, “MAX”.

NOTE 2 If the data type of the property is translatable_string, a column is allocated for each language. Such property code is described as “property code” + “sep_id” + “language code”. The language code identifies a language according to ISO 639-1. Values are e.g. “EN” for English in general, “FR” for French, “RU” for Russian, “DE” for German. If a language variant needs to be specified, a country code based on ISO 3166-1 can be added to the ISO 639-1 code, as in “en-GB” for UK English, and “en-US” for US English, with a hyphen being inserted between the two codes.

For the use of level 4 and level 5 notations, default supplier ID shall be declared prior to the use of level 3 and level 4 types of property code description. See “#DEFAULT_SUPPLIER” for more detail.

Level 2 should be used only for maintaining compatibility with the legacy dictionaries based on ISO 13584-42:1998, and IEC 61360-2:1998 and 2002. For any other purposes, the use of level 2 shall be strictly prohibited.

Example: This example is described using the level 3 notation (i.e. the versions of property are omitted).

```
#PROPERTY_ID,0140/TOPAS#P000001,0140/TOPAS#P001089.MAX,
0140/TOPAS#P001089.MIN,0140/TOPAS#000894.EN,
0114/TOPAS#P000894.FR,
```

Spreadsheet view: It will be displayed by a spreadsheet application as in Figure 13.

#PROPERTY_ID	0140/TOPAS# P000001	0140/TOPAS# P001089.MAX	0140/TOPAS# P001089.MIN	0140/TOPAS# P000894.EN	0140/TOPAS# P000894.FR
--------------	------------------------	----------------------------	----------------------------	---------------------------	---------------------------

Figure 13 – Display example of property ID

5.11.19 Preferred name of the property

Keyword: #PROPERTY_NAME.<lang>

Name: Preferred name of the property

Definition: preferred name of the property specified by the property ID, optionally in the language designated by a language code

Description: The information is provided by a parcel server, and any change of the information of this attribute by user shall not affect the behaviour of the system. Language for the preferred name may be specified using ISO 639. A further extension may be specified by attaching a country