eCl@ss®

CLASSIFICATION AND PRODUCT DESCRIPTION

Technical Specification 11

Conceptual Data Model

eCl@ss e.V.

07.08.2020

Authors and contributors

eCl@ss Center of Research and Development (CRD):

Ms. Petra Eder

Mr. Oliver Hadasch

Mr. Philippe Juhel

Mr. Thorsten Kroke

Mr. Gerald Lobermeier

Mr. Stefan Mülhens

Mr. Nikolaus Ondracek

Mr. Frank Scherenschlich

Mr. Josef Schmelter

and

Dr. Christian Block

Christian Hoffmann

Please send remarks to crd@eclass.de

Revision History

Date	Version	Change	Who
2020-04-30	0.3	Reformatted chapters and tables, reworked into more wiki like structure, added Text to the Diagram descriptions, incorporated comments of Christian Block	Ondracek
2020-06-25	0.4	Reworked and commented	Juhel
2020-07-16	0.5	Agreed version finalized	Ondracek
2020-08-07	1.0	Formatted and published	Hoffmann

Introduction

eCl@ss is a cross-industry master-data business standard for products and services information to be exchanged in a computer-sensible form across all borders – across sectors, countries, languages and organizations. The eCl@ss data dictionary is based on ISO 13584-42 and IEC 61360-2. It is used for the exchange of product data for procurement, eCommerce, engineering tools, etc.

The information model used by eCl@ss is the foundation of its classification and the structuration of the product descriptions. It also allows the reuse of generic descriptions in multiple business domains, the mapping to other dictionaries and ensure the upgradability of the eCl@ss dictionary across its successive versions called releases.

This document is describing an abstract form of the eCl@ss information model. It is helpful for understanding its logical structure and the relations between its elements.

The document is intended to become the backbone in the eCl@ss wiki for common terminology and model structure.

It is intended to expert groups members for developing product data models and to IT experts for implementing the use of eCl@ss dictionary in their information system.

Table of contents

1 Scope

This document defines the data model concepts of the eCl@ss dictionary, its Structure Elements and their Relations.

The concepts defined in this current version are used in eCl@ss 11.x and further Releases.

Out of scope:

- Business transaction of business objects
- Mappings (between Releases)
- Mappings (between standards)
- Change Management Rules
- File format serializations (CSV, XML, …)
- Future developments (e.g. Qualifier, Relations)
- ISO / IEC related mappings and comparisons
- Security, confidentiality and access right management

2 Normative references

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

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

ISO 639-2:1998, *Codes for the representation of names of languages — Part 2: Alpha-3 code*

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

ISO 8601-1:2019, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO 10303-42:2019, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation*

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

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

ISO/IEC 11179-6:2015, *Information technology — Metadata registries (MDR) — Part 6: Registration*

3 Terms and definitions

3.1 General terms

The purpose of this section is not to provide extensive terms and definition but to ensure that there is consistent understanding on the approaches and terms used in this document.

3.1.1 Conceptual Data Model - Conceptual Model

Data Model that represents an abstract view of the real world

[SOURCE: ISO/IEC 11179-1:2015, 3.2.5]

Note:

A **Conceptual Data Model** is the most abstract form of information model. It is helpful for communicating ideas to a wide range of stakeholders because of its simplicity. Therefore platform-specific information, such as data types, indexes and keys, is omitted from a Conceptual Information Model. Other implementation details, such as procedures and interface definitions, are also excluded.

3.1.2 Data Model

Graphical and/or lexical representation of data, specifying their Properties, Structure, and Interrelationships

[SOURCE: ISO/IEC 11179-1:2015, 3.2.7]

3.1.3 Physical Model

A **Physical Data Model** visually represents the structure of data as implemented by for example a relational database schema.

Note:

Providing a visual abstraction of the database structure, an important benefit of defining a Physical Data Model is that you can automatically derive the database schema from the model. This is possible due to the richness of meta-data captured by a Physical Data Model and its close mapping to aspects of the database schema, such as database Tables, Columns, Primary keys and Foreign keys.

3.1.4 Serialization

A **Serialization** is a special case of a **Physical Data** which represents a series of data exchange structures.

Note:

A serialization may be represented e.g. by an XML Schema (XML-Exchange File), CSV File or JSON.

3.2 Abbreviations

IRDI	International Registration Data Identifier

4 eCl@ss-Conceptual-Data-Model

4.1 Release and Representations

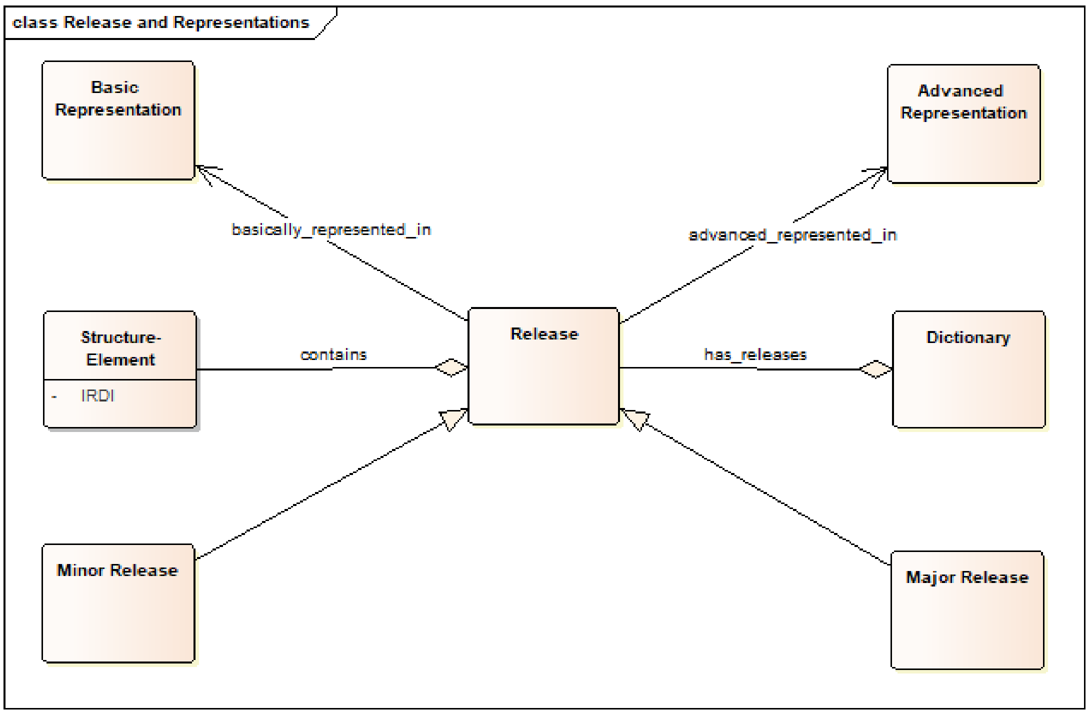

Figure 1: eCl@ss Conceptual Data Model - Release and Representation

eCl@ss is a formal semantic DICTIONARY which is used for product description and product classification.

The eCl@ss dictionary is maintained in RELEASE (S). Each Release is containing an exhaustively defined and reproduceable set of STRUCTURE-ELEMENT (S).

eCl@ss separates Releases into MINOR-RELEASE and MAJOR-RELEASE, depending on business rules on the Releases's content.

Releases are delivered in different Representations, currently in BASIC-REPRESENTATION and ADVANCED-REPRESENTATION.

4.2 Structure-Element

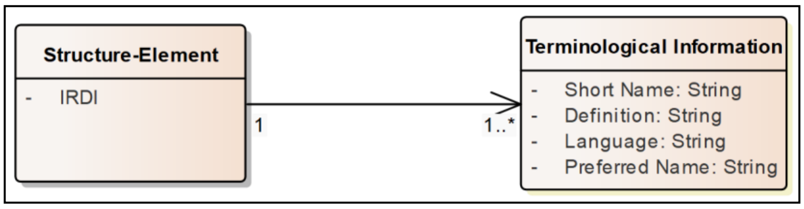

Figure 2: eCl@ss Conceptual Data Model – Structure-Element

A STRUCTURE-ELEMENT represents a semantic representation of a real world concept in the eCl@ss DICTIONARY. The Structure Element is uniquely identified by an IRDI.

For human understanding each Structure Element is described by translatable TERMINOLOGICAL INFORMATION.

4.3 eCl@ss-Conceptual-Data-Model Overview

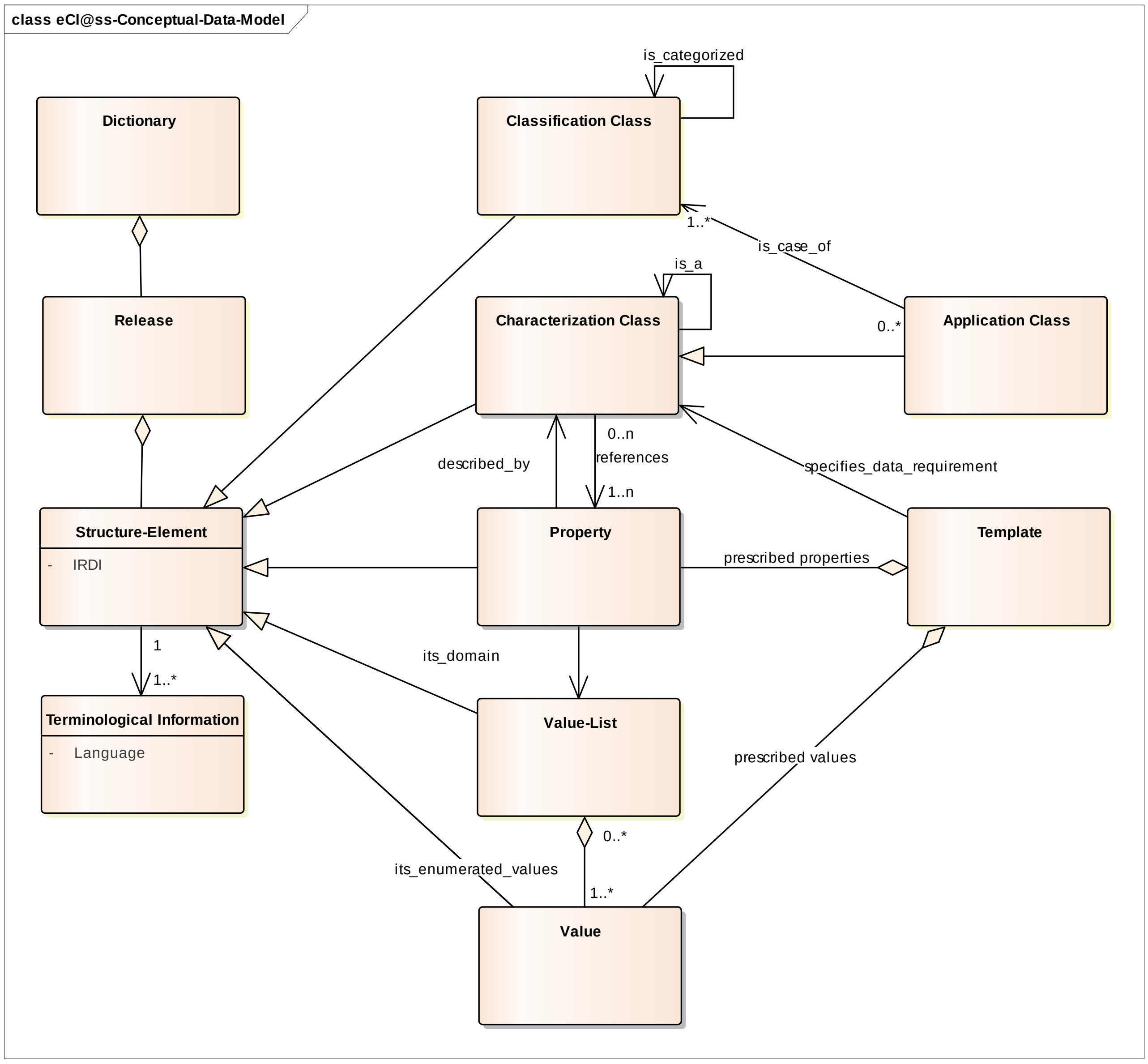

Figure 3: eCl@ss Conceptual Data Model - Overview

eCl@ss is a formal semantic DICTIONARY which is used for product description and product classification.

The eCl@ss dictionary is maintained in RELEASE (S).

Each Release is containing an exhaustively defined and reproduceable set of STRUCTURE-ELEMENT (S) each explained by TERMINOLOGICAL INFORMATION. Between releases apply formal Dictionary Change Management Rules.

If a given concept represents a class of products which may be described by the same set of PROPERTY(S), this class of products is a CHARACTERIZATION CLASS which is identified as an APPLICATION CLASS and categorized in a CLASSIFICATION CLASS.

The enumeration of all values assigned to a property is the Structure-Element VALUE-LIST.

A CHARACTERIZATION CLASS may have zero many TEMPLATE which are defining the formal data requirement specification (according to ISO 22745-30) of this class.

4.4 Classification Class

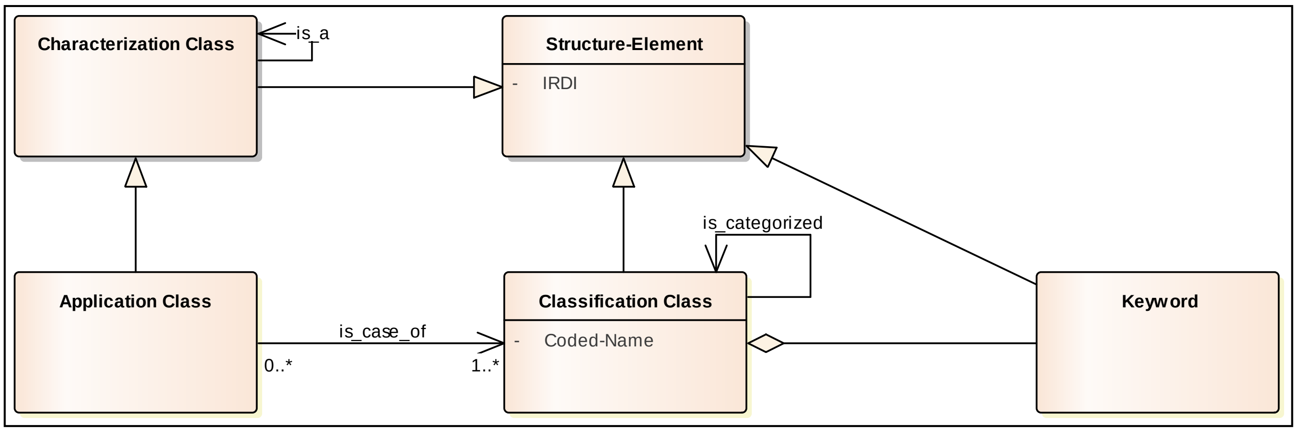

Figure 4: eCl@ss Conceptual Data Model - Classification Class

The CLASSIFICATION CLASS is a CHARACTERIZATION CLASS which helps to categorize the described product(s) are divided into certain categories of similar products. Classification-Classes are hierarchically ordered.

Classification-Classes are hierarchically ordered. The hierarchy is a rooted, ordered tree with fixed height four.

On the fourth level of the eCl@ss classification hierarchy, each Classification-Class has two APPLICATION CLASS (ES) assigned. One of the BASIC-REPRESENTATION and one of the ADVANCED-REPRESENTATION.

Each Classification-Class may have zero to n untranslated KEYWORD (S) assigned.

4.5 Characterization Classes

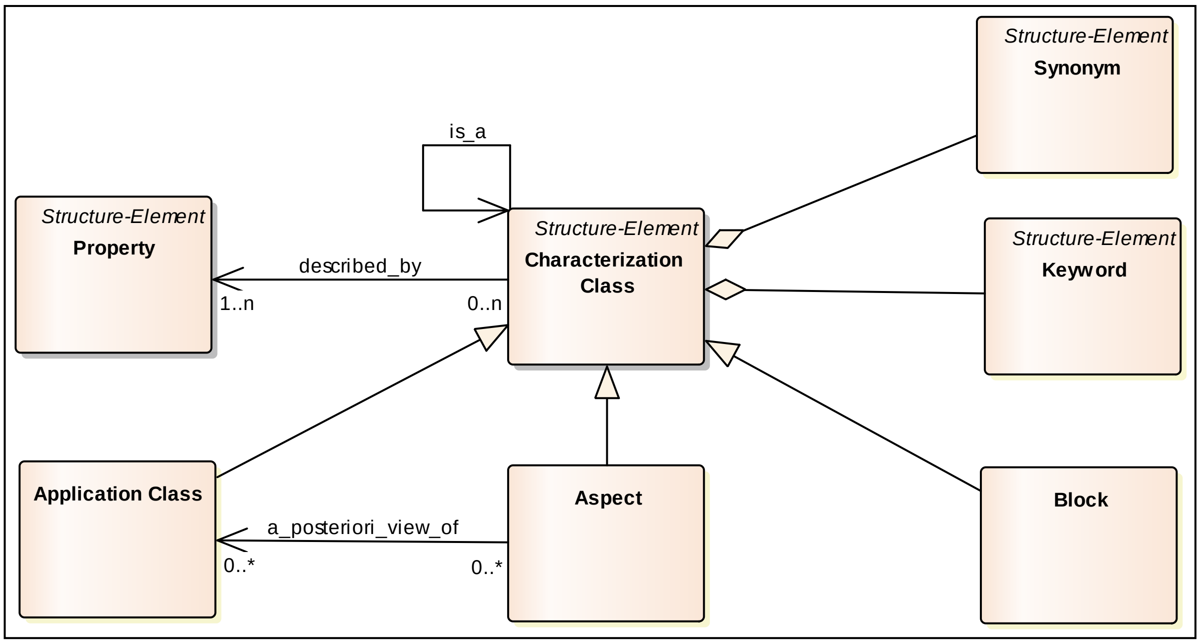

Figure 5: eCl@ss Conceptual Data Model - Characterization Classes

Characterization Class is a STRUCTURE-ELEMENT which is grouped hierarchically by the specialization relation (is_a). The specialization relation implies the substitution principle of the generic by the more special. Characterization Classes must have at least one property assigned.

A special case of Characterization Class is the APPLICATION CLASS, which is used for product description. It is mandatorily connected to the 4th level of the eCl@ss Classification-Class hierarchy.

Another case of Characterization Class is the ASPECT which is used for describing a special external perspective on an Application Class.

A BLOCK is also a Characterization Class which is used for describing sub-concepts of an Application class.

Note: Blocks are referenced by PROPERTY of DATA-TYPE Reference.

Characterization Class may have zero to n KEYWORD and/or SYNONYM assigned.

4.6 Property

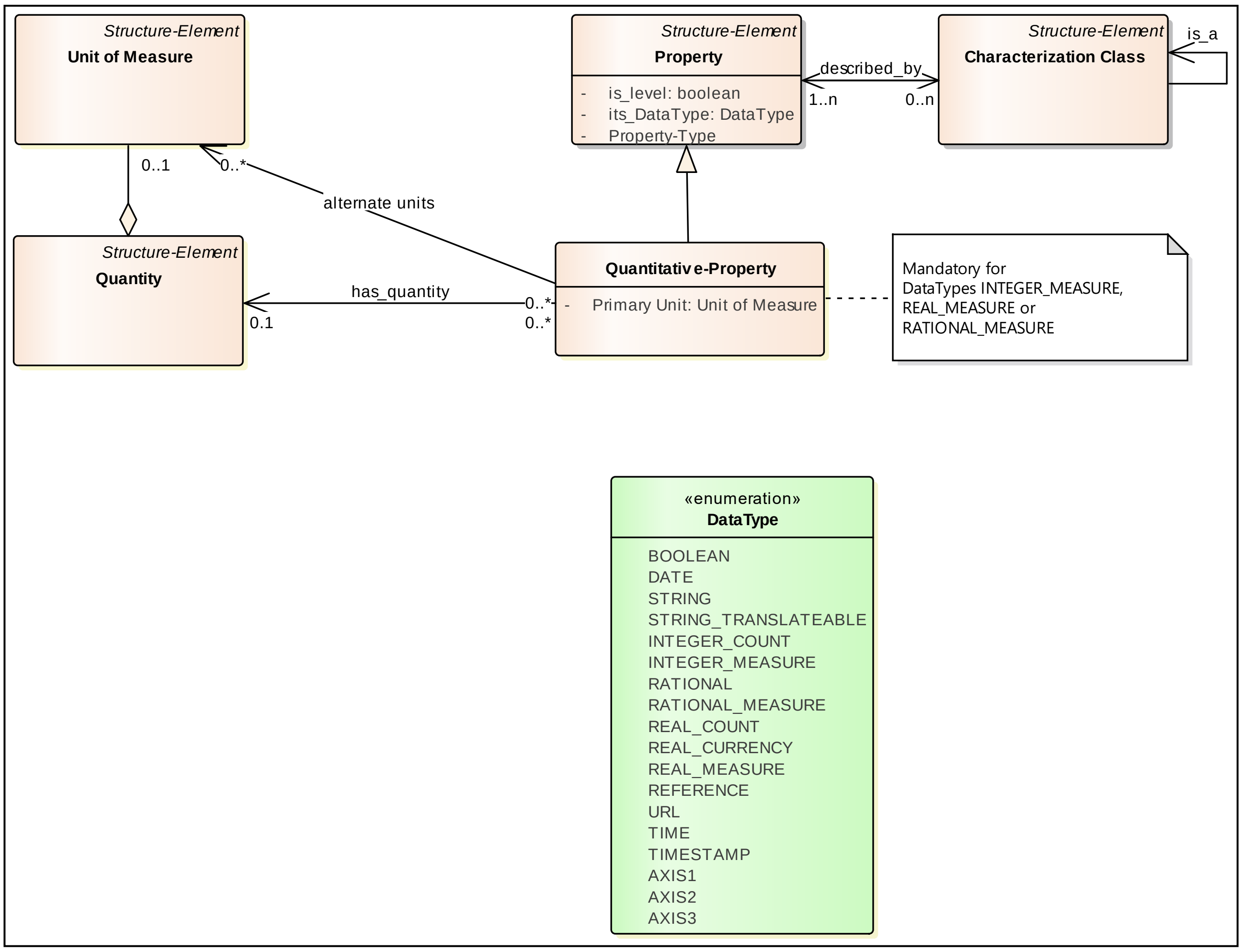

Figure 6: eCl@ss Conceptual Data Model - Property

A PROPERTY is a Structure-Element which represents a set of values, either by the description of its data-type or by an relation to its VALUE (S) which determine the enumeration of the most useful characteristics of the described item (e.g. property: color, value: red).

PROPERTY (S) are used for describing the products which are instances of a CHARACTERIZATION CLASS.

Each Property has mandatory exactly one DATA-TYPE assigned, which describes or enumerates the set of possible VALUE (S) in a VALUE-LIST.

QUANTITATIVE-PROPERTY are properties used to represent numeric Values, which are assigned to a UNIT OF MEASURE.

Properties are assigned to a QUANTITY, in order to allow conversion of values referencing other UNIT OF MEASURE belonging to the same physical QUANTITY.

4.7 Depending Properties and Conditions

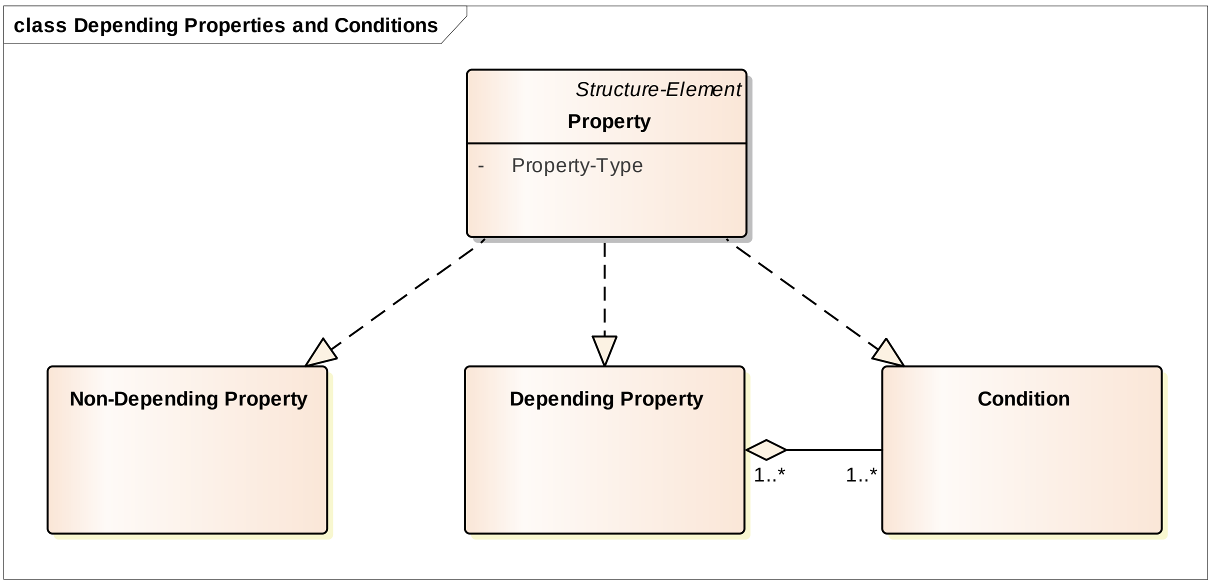

Figure 7: eCl@ss Conceptual Data Model - Depending Properties and Conditions

Properties may act in different ways, either the property's value is not depending on any other value, the PROPERTY is a NON-DEPENDING PROPERTY.

However, there are cases, where a Property's Value makes only sense if another Property's Value is mandatory existing in a product. In this case the Property is a DEPENDING PROPERTY which depends on a non-empty-set of Properties, its CONDITIONS.

4.8 Value

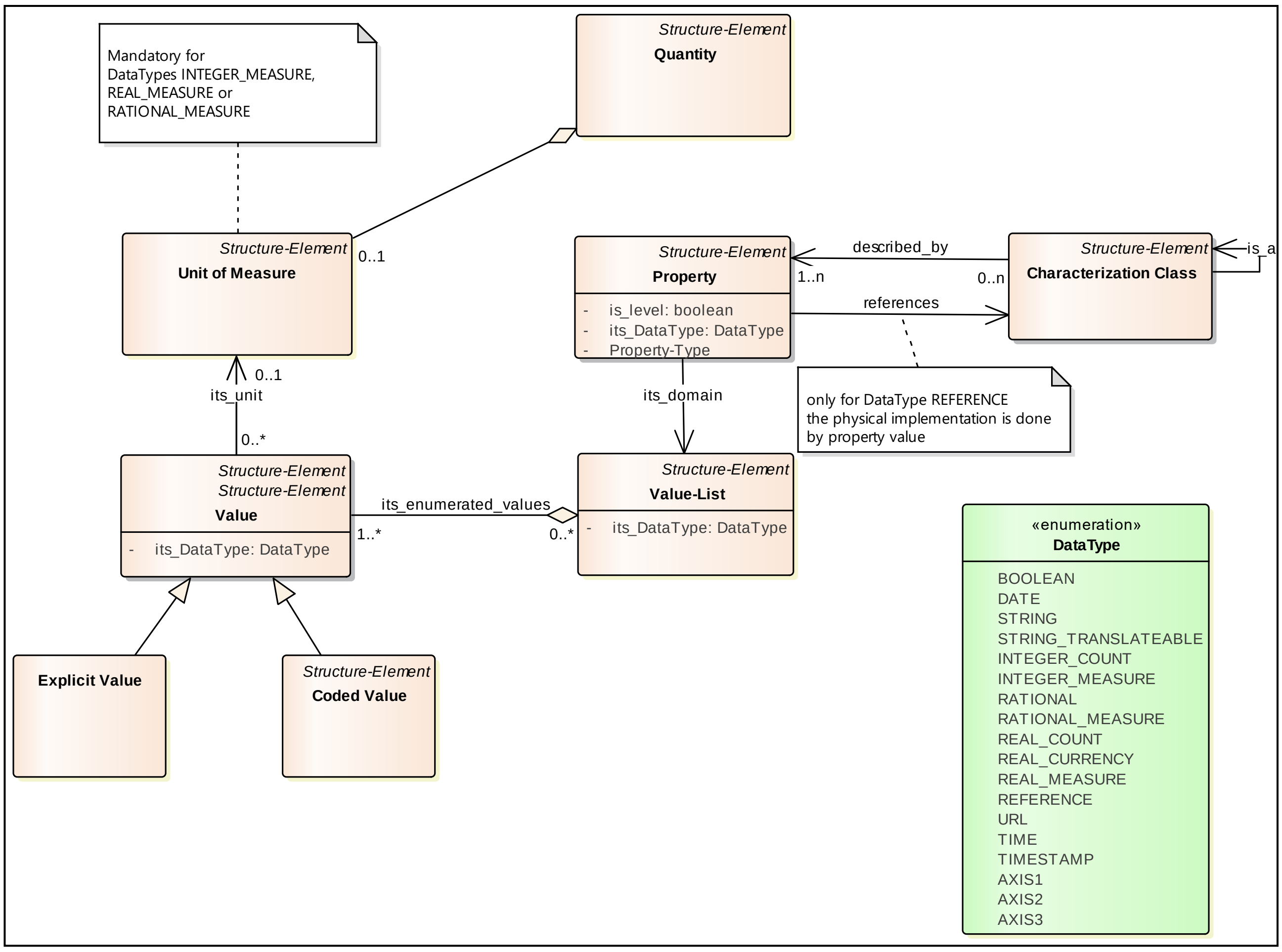

Figure 8: eCl@ss Conceptual Data Model - Value

VALUE (S) are enumerations within the DATA-TYPE of a PROPERTY which restrict the set of possible Values for the Property.

Note: This enumeration of Values is referred to as VALUE-LIST.

Note: A Property does not necessarily have a VALUE-LIST assigned. However, a Property always has a mandatory DATA-TYPE.

Values have a Data-Type which must match the Date-Type of the Property.

Values may be subdivided into:

- CODED VALUE which are Structure-Elements and can be looked up in the dictionary and can be translated, and
- EXPLICIT VALUE which are commonly used concepts, like numbers (e.g. 10,25) which do not need look up in dictionaries.

Values may have optionally a UNIT OF MEASURE assigned, provided the value is intended for QUANTITATIVE-PROPERTY (IES).

For a Property of Data-Type REFERENCE the Values are references (i.e. IRDI) to CHARACTERIZATION-CLASS, what represents the set of instances which are abstracted by this Class.

4.9 Unit of Measure

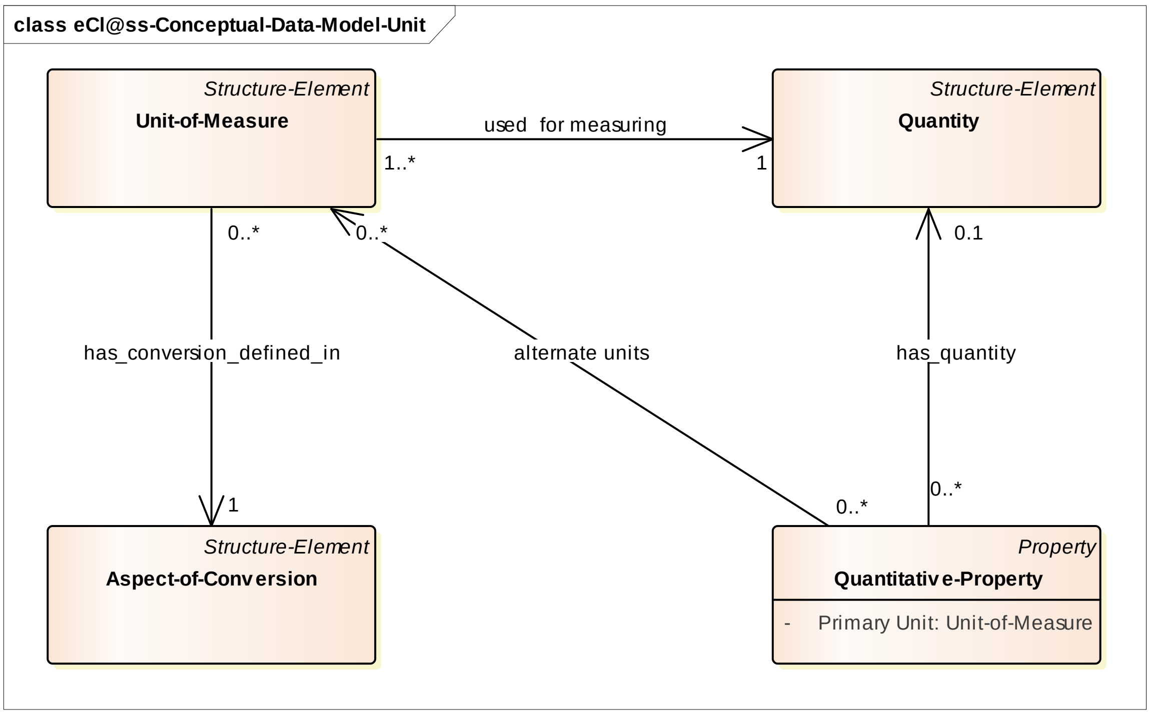

Figure 9: eCl@ss Conceptual Data Model - Unit of Measure

The Aspect-of-conversion relates a Unit of Measure to the base Dimension of the underlying Unit system(s) in terms of exponents of the base Units. Each QUANTITATIVE-PROPERTY is assigned to a QUANTITY and a possible empty set of alternate Unit of Measure which must belong to the same QUANTITY.

The ASPECT OF CONVERSION defines the scales of composed Unit of Measure along the Unit system's Dimensions.

4.10 Enumeration Constraint

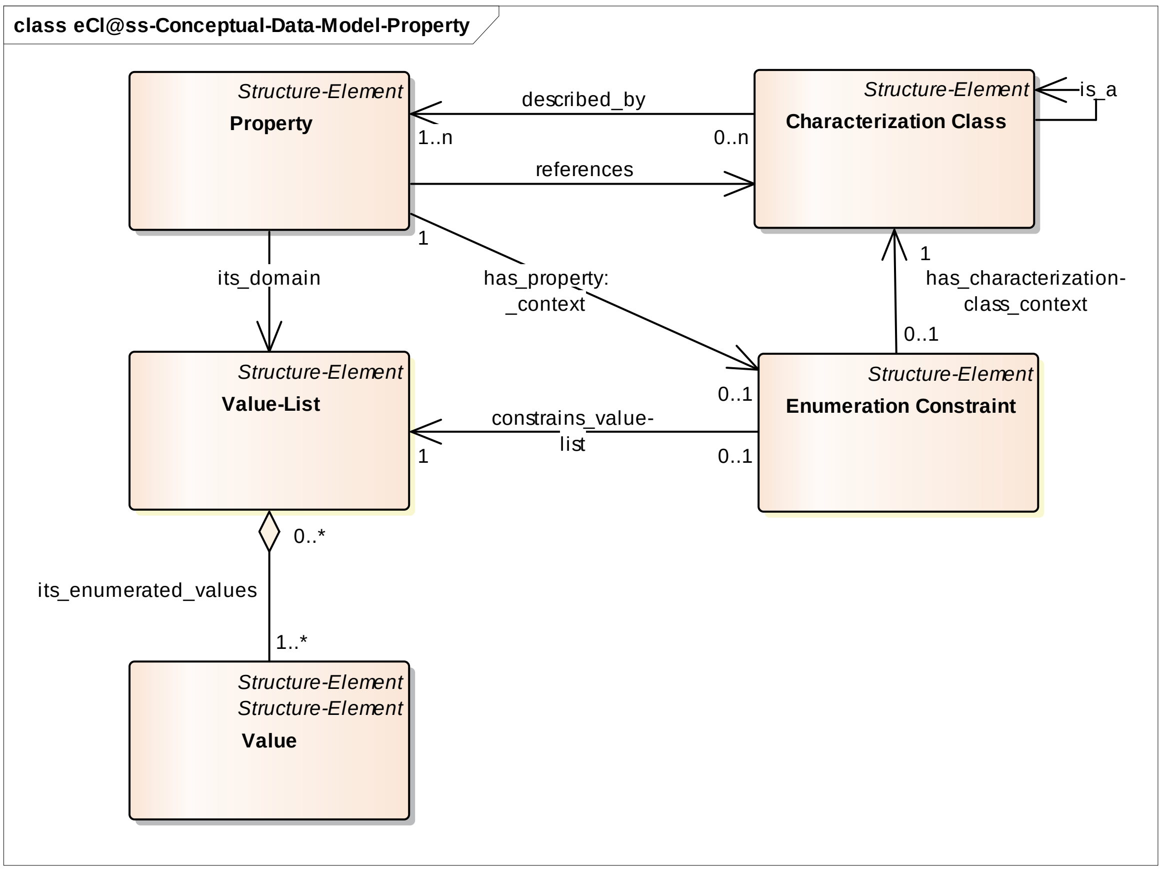

Figure 10: eCl@ss Conceptual Data Model - Enumeration Constraint

VALUE-LIST (S) of a PROPERTY can be restricted by an ENUMERATION CONSTRAINT in context of a CHARACTERIZATION-CLASS, leading to less possible VALUE (S).

5 Structure elements

5.1 Advanced Representation

By introducing eCl@ss 7.0 Advanced Representations, the eCl@ss-Association offers an ISO13584 / IEC 61360 conformant data structure.

5.2 Application Class

An Application Class is a representation of a type or family of product (device, equipment, software or service) which can be described by a generic set of PROPERTY (IES).

Note:

Property lists are compilations of individual features to describe individual commodities. In eCl@ss, each subgroup (Classification Class at the fourth level) is described with the help of Properties. The Properties are not directly assigned to a subgroup but to a so-called Application Class which is connected to a subgroup. In the versions before eCl@ss 6.0, eCl@ss has used sets of Properties that were directly assigned to a Classification Class - as it is still shown in the BASIC version. Application Classes are only visible in the ADVANCED version.

See also:

Set of components of which each component can be described by the same group of data element types.

[reference IEC 61360-1: 2002-02, 2.14]

5.3 Aspect

Aspect is the approach, method of approach, point of view of a product or service or a part of a product or service

- in relation to a given situation,
- from a certain point, from a certain direction
- or at a certain angle

Note:

An Aspect does not describe the product specific Properties itself (like Properties of Block and Application Class Properties), but it includes Properties for that Class under certain conditions or additional Properties for a Class under certain point of views.

Properties that describe the requirements for a device, are a typical Aspect of a device.

Example:

The Aspect “Operating conditions” of a car. It describes under which conditions the car is to be operated: Central Europe, the Arctic or desert.

The resulting Properties, however, such as minimum starting temperature or operational altitude are Properties of the “car” Class and Aspect elements!

5.4 Aspect of Conversion

An Aspect of Conversion relates a UNIT OF MEASURE to the base of the underlying unit system(s) in terms of exponents of the base units.

5.5 Basic Representation

eCl@ss Basic Representation presents a version variant with all essential new contents in the usual format. Consequently, the Basic version is downloadable in a CSV and an XML serialization.

5.6 Block

A Block is a subset of Properties within an Application Class describing an abstraction of a feature of the product or of a part of a composite device.

Note:

If all Properties of a device type are arranged with equal importance on one single level, the list will become less understandable while more Properties are added. Clarity can be achieved by structuring the Properties into sub-concepts, which are described by Properties. These sub-concepts are called Block (of Properties).

Every Block of Properties has a name and definition as defined in IEC 61360 Part 2 ed1 / ISO13584-42 ed1 for Properties. Blocks are structured in a similar way to Properties.

A Block - in contrary to the related concept of an Aspect - describes characteristics of the product itself.

A Block of Properties consists of one or more Properties describing an abstraction of a feature of an object or of a part of a composite device. A Block of Properties may contain other Blocks of Properties nested to the necessary level as dictated by the technical requirements. At the lowest level, a Block will contain only Properties.

5.7 Classification Class

Classification Class(es) divide products into certain categories of similar product(s), the product-groups.

Every Classification Class has the following main attributes

- a unique identifier (IRDI),
- preferred name
- a coded name that represents the classification structure, e.g. 27-01-01-01

Note:

eCl@ss is a monohierarchical classification system, i.e. every product group is to be found only once in the hierarchical tree structure.

With the development of eCl@ss 7.1, the definition of a classification class was defined as a mandatory field, i.e. starting with eCl@ss 7.1 every new classification class must have a definition. That does not imply that all Classification Classes that were created before 7.1 will have a definition as well.

5.8 Characterization Class

Concept which represents a set of instance items (e.g. Products) which may be described by the same set of PROPERTY (IES).

5.9 Condition

PROPERTY which acts as condition for a DEPENDING PROPERTY.

Note:

There are cases, where a Property's value makes only sense if another Property's Value is mandatory existing.

In this case the Property is a Depending Property which depends on a non-empty set of Properties, its CONDITIONS.

5.10 Coded Value

VALUE represented by code and thus might be translated.

Note:

- A Value is a STRUCTURE-ELEMENT
- A Coded-Value can be looked up in the DICTIONARY

5.11 Data Type

A Data Type defines what kind of information a Property or Value may transport.

List of Enumeration Values:

Name	Explanation
BOOLEAN	Represents Boolean algebraic values In serializations, the format (YES | NO) could be used. Example: Yes
DATE	Represents values which are containing date. In serializations, the format yyyy-mm-dd according ISO 8601-1:2019 could be used. Example: 1979-01-15
STRING	A finite sequence of symbols that are chosen from a set or alphabet […] a sequence of characters

	(http://en.wikipedia.org/wiki/String_(computer_science)). Cannot be translated into other languages. Example: DN 700 ; 10 Mbps
STRING_TRANSLATABLE	A finite sequence of symbols that are chosen from a set or alphabet […] a sequence of characters (http://en.wikipedia.org/wiki/String_(computer_science). Values for translatable string type Properties are required to each have an ISO 639-1 language assigned and may be further specialized by adding country identification. Examples: "red" (language: English) ; "grey" (language: English, country: United Kingdom) ; "Aluminum" (language: German). Note: ISO 639-2 defines a couple of exceptions that can be used in edge cases: • "undetermined" language (code: und) • "no linguistic content" language (code: zxx) • "multiple languages" language (code: mul)
INTEGER_COUNT	Data Type which represents some finite subset of the mathematical integers. These are also known as integral Data Types. Used only for counting. (http://en.wikipedia.org/wiki/Integer_(computer_science). Example: 1 ; 10 ; 111
INTEGER_MEASURE	Data Type which represents some finite subset of the mathematical integers. These are also known as integral Data Types. Used for measuring in a specific Unit of Measure. (http://en.wikipedia.org/wiki/Integer_(computer_science). Example: 1 ; 10 ; 111
RATIONAL	To represent rational numbers like 1/3 and -11/17 without rounding (http://en.wikipedia.org/wiki/Rational_data_type). Example: 1/3 ; 1 2/3
RATIONAL_MEASURE	To represent rational numbers like 1/3 and -11/17 without rounding (http://en.wikipedia.org/wiki/Rational_data_type).

	Used for elements that are measures of type RATIONAL (i.e. expressed in a specific Unit of Measure) Example: 1/3 ; 1 2/3
REAL_CURRENCY	A rational number expressed in decimal representation (http://en.wikipedia.org/wiki/Real_number). Used for measuring in a specific currency. Example: 1,5 ; 102,35
REAL_MEASURE	A rational number expressed in decimal representation (http://en.wikipedia.org/wiki/Real_number). Used for measuring in a specific Unit of Measure. Example: 1,5 ; 102,35
REFERENCE	Represents a reference to a Block or Application Class
URL	Represents Values according to ISO 13584-24:2003 Example: http://www.eclass-cdp.com
TIME	Represents Values which are containing time In serializations, the format hh:mm according ISO 8601-1:2019 could be used. Example: 12:45
TIMESTAMP	Represents Values which are containing timestamp (i.e. date and time) In serializations, the format yyyy-mm-dd hh:mm according ISO 8601-1:2019 Example: 1979-01-15T12:45:00Z
AXIS1	According to ISO 10303 axis1_placement is the direction and location in three-dimensional space of a single axis and is defined by a point (first three numbers) and an axis direction (last three numbers).
AXIS2	According to ISO 10303 axis2_placement_2d is the location and orientation in two-dimensional space of two mutually

	perpendicular axes defined by a point (first two numbers) and an axis (last two numbers).
AXIS3	According to ISO 10303 axis2_placement_3d is the location and orientation in three-dimensional space of two mutually perpendicular axes defined by a point (first three numbers) and two axes (middle and last three numbers).

5.12 Depending Property

There are cases, where a Property's Value makes only sense if another Property's Value is mandatory existing.

In this case the property is a DEPENDING PROPERTY which depends on a non-empty-set of Properties, its CONDITIONS.

5.13 Dictionary

Consistent ontological structure, composed by a set of Structure-Elements, which are checked for consistency, eCl@ss rules and which are release managed together.

Note: The dictionary contains the classification and product description.

5.14 Enumeration Constraint

An Enumeration Constraint is assigned to the CHARACTERIZATION-CLASS and enumerates the possible Values (within the Property's Value List) of a Property in context of this Characterization-Class.

Example:

Property PRO123 has a Value List VLI123 that contains the Values [VAL001; VAL002, VAL003; VAL004].

By applying EC001 to Class CCL001 only the following Values are valid for PRO123: [VAL001; VAL002, VAL003]

By applying EC002 to Class CCL002 only the following Values are valid for PRO123: [VAL001; VAL002]

By applying EC003 to Class CCL003 only the following Value is valid for PRO123: [VAL004]

Note:

The description of a suggested Value List including constraints in the BASIC Release Notes is as follows (Update: Release 8.0): Value List - constraint

5.15 Explicit Value

Explicit Value which are Values representing commonly used concepts, like numbers (e.g. 10,25) which do not need definition in dictionaries.

5.16 Keyword

Keywords aid the search for Classification Classes within the eCl@ss Class structure.

As only one name can be defined for a Classification Class, eCl@ss offers the possibility, to assign more alternative names for a Class with the help of Keywords.

Example:

The eCl@ss 8.1 Class 23-11-01-04 "Hammer head bolt" has the Keyword "T-head bolt" as an alternative name for the same product group, so that users who speak of a "T-head bolt" can find the same Class that is officially named "Hammer head bolt".

Keywords must not have the same meaning as the preferred name of the Classification Class (e.g. Class: pliers, keyword: pipe pliers). Keywords must not match the preferred name of the Class they belong to. Keywords must not be too general in character (the Keyword ‘school’ would be too vague for a Class named ‘ballpoint pen’). Keywords are preferably to be entered in singular form. Keywords cannot be translated 1:1 and therefore vary in the eCl@ss language versions.

5.17 Major Release

An eCl@ss Major Release is a Release type that includes all possible modifications of existing Structure-Elements (including structural modifications) and the addition of new elements that:

- Include every kind of change including structural changes, e.g. structural changes of the Class hierarchy, changes of existing relations between structural elements, MOVEs, JOINs and SPLITs. For an overview please have a look at the list of valid Change Requests per Release.
- Are not downwards-compatible to previous Releases due to possible structural changes.
- Cannot be integrated into a system as upgrade of the previous version automatically. This process is supported though not fully automated by available transaction update files and Release Update Files. A manual re-mapping process of one's data might still be necessary.
- Are published every 2 to 4 years (see Release Roadmap).
- Are coded with the help of a Major Release Number and a zero Minor Release Number (X.0, e.g. 11.0 the next Major Release after 10.0.1)

5.18 Minor Release

An eCl@ss Minor Release is a Release type that includes the modification of certain attributes of existing Structure-Elements (e.g. textual changes) and the addition of new elements that:

- Include additions (possible on all hierarchical levels), the removal of erroneous Keywords and suggested Property Values and, if necessary, the correction of them. For an overview please have a look at the list of valid Change Requests per Release

- Are downwards-compatible within the same Major Release Number (e.g. all 9.x Releases with each other, all 10.x Releases with each other etc.)
- Can automatically be integrated into a system as an upgrade of the previous version as only version changes of existing structural elements and additions of new structural elements are made.
- Are planned to be published once a year (see Release Roadmap)
- Are coded with the help of a Major Release Number and a Minor Release Number (n.X, e.g. 9.1 - the first Minor Release after Major Release 9.0)

5.19 Non-Depending Property

A Property of which the (instantiated) VALUE is not depending on any other Value.

See also: DEPENDING-PROPERTY

5.20 International Registration Data Identifier (IRDI)

eCl@ss uses globally unique identifiers for every object included in the eCl@ss standard.

It consists of several parts:

- An International Code Designator (ICD) according to ISO 6523-1, followed by an Organization Identifier (OI) that globally identifies eCl@ss as the publishing organization (eCl@ss: 0173, other registered organizations include ISO (0112), ODETTE (0177), SIEMENS (0175), GTIN (0160))
- A Code Space Identifier (CSI) that identifies the type of object (e.g. 01 for Class, 02 for P etc.)

Note:

The International Registration Data Identifier is based on the international standards ISO/IEC 11179-6, ISO TS 29002-5 and ISO 6532.

See also: http://wiki.eclass.eu/wiki/IRDI

5.21 Property

A Property describes a characteristic of a product.

Each Property has an associated Data Type.

If the Property has a numeric Data Type and is used for measuring, the Property must have a Unit of Measure.

Note:

Every Property has a unique identifier (IRDI), a preferred name, a definition and a Value domain - among other attributes.

An attribute of the Property is the attribute that defines, that the Property uses Property-Levels.

5.22 Property-Level

Property Levels define a Property and its levels out of min, max, typ, nom.

The values are a vector, which includes the minimal, maximal, typical and nominal value (min, max, typ, nom) concurrently.

Note:

Property-Level's are sometimes referred as Level Type.

5.23 Quantity

A Quantity is a characteristic of an object that can be quantified by measurement.

A Quantity can be expressed as the combination of Unit of Measure and a number, where the number is the magnitude.

Example: A length or mass are physical Quantities.

5.24 Quantitative Property

Property with a numerical Value representing a physical Quantity.

Data Types INTEGER_MEASURE, REAL_MEASURE or RATIONAL_MEASURE

are mandatory Quantitative-Properties.

Each Quantitative-Property has exactly one Unit of Measure, its primary Unit, assigned.

5.25 Release

Release is the process of assigning either unique version numbers to unique states of eCl@ss.

Within a given version number category (Major, Minor), these numbers are generally assigned in increasing order and correspond to new developments.

Release control is often used for keeping track of incrementally different versions of information.

5.26 Structure Element

eCl@ss Business Objects are called Structure Elements.

eCl@ss uses globally unique identifiers for every Structure Element included in the eCl@ss standard.

This globally unique identifiers are called IRDI (International Registration Data Identifier)

The International Registration Data Identifier is based on the international standards ISO/IEC 11179-6, ISO TS 29002-5 and ISO 6532.

Note:

The IRDI contains of

- 0173-1 = eCl@ss (as data supplier)
- Three letters followed by three numbers, e.g. AAA123
- Running number padded to length 3 with leading zeroes

5.27 Synonym

Synonyms are terms which have the same meaning as the preferred names of Structure Elements.

As a rule, Synonyms must not match the preferred name of the Structure Element.

Note:

Before Release 7.0, Synonyms were managed as the Property attribute "alias name". Since Release 7.0 they are distinct structural elements.

5.28 Terminological Information

eCl@ss is multilingual and describes its Structure Elements with the following translatable attributes:

- Preferred Name
- Short Name
- Definition
- Language

5.29 Template

An eCl@ss Template is a bilaterally agreed data requirement specification, i.e. it underlies the communication between an information requestor and an information provider.

It defines which data from an Application Class from the eCl@ss dictionary shall occur in transactional data (messages) or catalogs and how this data is expected to be presented. Moreover, it can clarify the usage of individual Properties by narrowing down the options e.g. for a Property being multivalued or having a restricted set of proposed Values.

The data dictionary defines how products and services are classified and can be described.

In the Template, there is modelled how the expected response to a query looks like; in other words: The Template is used to structure the catalog and gives hints at filling out the catalog, but to understand the meaning of the catalog one does only need the dictionary.

This means that the following referencing requirements apply:

- A Template does not contain its own set of identifiers (except the Template identifier itself).
- A catalogue references concept identifiers from the dictionary, not from the Template.

- A Template is not needed to decode a catalogue.
- A Template is needed to create a catalogue that meets a data recipient's requirements.

Note: eCl@ss is not providing Templates (as by eCl@ss 11.1), however Templates are contained in the eCl@ss conceptual model and might be defined by any eCl@ss user.

5.30 Unit of Measure

A Unit of Measurement is a definite magnitude of a Quantity, defined and adopted by convention or by law, that is used as a standard for measurement of the same kind of Quantity.

Any other Quantity of that kind can be expressed as a multiple of the Unit of Measurement.

Note:

- The metre is a Unit of length that represents a definite predetermined length. When we say 10 metres (or 10 m), we actually mean 10 times the definite predetermined length called "metre".
- Measurement is a process of determining how large or small a physical Quantity is as compared to a basic reference Quantity of the same kind.

5.31 Value

Specifies one possible content of a Property compliant with the Domain of the Property.

A Value has a Data Type assigned.

5.32 Value List

Defines a set of permissible Value(s) being implicit by the Data Type or explicit by listing the possible Value(s).

A Value List has a unique identifier and describes the relation between a Property and a Value.

A Value List according to ISO 13584 is a set of restrictive Values, i.e. lists of Values that are complete and therefore only these Values are valid for a Property. In eCl@ss, only the BOOLEAN list (comprising the Values YES and NO) and Value Lists that form a Polymorphism are considered to be restrictive, because they are exhaustive, and no other Value is possible. This relation is therefore true in any given context.

However, within eCl@ss a flag allows deciding if a Value List is only a suggestion. A suggested Value List, also known as open Value list, consists of suggested Values, i.e. Values that are proposals and are not restrictive. The exhaustiveness of these listed Values cannot be guaranteed as more Values might be valid in a certain context. E.g. for a Property colour Value can only be interpreted as being suggestions - more colours can always be possible. The suggested Value List (having a unique IRDI itself) therefore describes the relation between a Class, a Property and a Value, because in theory, a Property can have different valid Values in the context of different Classes, see CONSTRAINT.

Note: With eCl@ss the default for Value Lists are suggested Value Lists.

Note: Usage of Value List

Prior to Release 8.0, eCl@ss interpreted its Value Lists as open, i.e. as suggestions that were never intended to be exhaustive. As the ISO defines Value Lists as restrictive and exclusive eCl@ss had to change its structure to be ISO-compliant. Therefore eCl@ss now distinguishes between "restrictive Value Lists" (ISO view) and "suggested Value Lists" (proposed "open" lists that are not exhaustive). Because of this, most Value that were noted until Release 6.2 are now marked as CLOSED as they are to be interpreted as restrictive. Correspondingly, all relations between Properties and Values in the context of a Class listed in the Value proposal file (eClass7_0_PR_VA_suggested_en_02.csv) are new, i.e. all relations listed in that file are interpreted as NEW and not published here again.