INTERNATIONAL STANDARD

ISO/IEC 11179-31

First edition
2023-01

Information technology — Metadata registries (MDR) —

Part 31: Metamodel for data specification registration

Technologies de l'information — Registres de métadonnées (RM) — Partie 31: Métamodèle pour l'enregistrement des spécifications de données

Reference number
ISO/IEC 11179-31:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

List of Figures

List of Tables

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC/JTC *1*, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

A list of all parts in the ISO/IEC 11179 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

In ISO/IEC 11179-3, the structure of a Metadata Registry is specified in the form of a conceptual data model. ISO/IEC 11179-3 specifies a metamodel for "registry common facilities", which is intended to be extended by other parts of ISO/IEC 11179 for specific purposes.

This document provides a specification of the extensions to the registry metamodel specified in ISO/IEC 11179-3 to enable the registration of metadata about data elements and associated concepts, such as "data element concepts", "conceptual domains" and "value domains". Generically, these are all referred to as "metadata items". Such metadata are necessary to clearly describe, record, analyse, classify and administer data.

This document is part of the 4th edition modularization of the ISO/IEC 11179 series. It extracts the Data Description package from ISO/IEC 11179-3:2013 to make it more accessible and renames it "Metamodel for data specification registration". At the same time, some enhancements have been made as follows:

— support for externally defined "reference enumerated conceptual domains" (7.4.2.6) and "reference enumerated value domains" (7.4.2.13);

— support for sub-setting of value domains (7.7) and conceptual domains (7.8) within a specified context;

— support for composite data elements and data types (7.9);

— finer-grained conformance options (see 5.3);

— relaxation of some constraints in the standard, while giving registration authorities the ability to enforce them if they wish (see 6.5).

From Clause 5 onwards, this document uses:

— **bold** font to highlight terms which represent metadata objects specified by the metamodel;

— normal font for terms which represent concepts defined in Clause 3.

EXAMPLE **Conceptual_Domain** (7.2.2.2) is a class each instance of which models a conceptual domain.

Information technology — Metadata registries (MDR) —

Part 31:
Metamodel for data specification registration

1 Scope

This document provides a specification for an extension to a Metadata Registry (MDR), as specified in ISO/IEC 11179-3, in which metadata that describes data elements and associated concepts, such as "data element concepts", "conceptual domains" and "value domains" can be registered.

The specification in this document, together with the relevant clauses of the specification in ISO/IEC 11179-3, provides the ability to record metadata about:

a) data elements, units of measure and derivation rules;

b) data element concepts and associated object classes and properties;

c) conceptual domains, conceptual domain subsets and value meanings;

d) value domains, value domain subsets, datatypes and permissible values.

This document is applicable to the formulation of data representations, concepts, meanings and relationships to be shared among people and machines, independent of the organization that produces the data. It is not applicable to the physical representation of data as bits and bytes at the machine level.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11179-3:2023, *Information technology — Metadata registries (MDR) — Part 3: Metamodel for registry common facilities*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 11179-3 and the following apply.

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

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
object class
set of ideas, abstractions or things in the real world that are identified with explicit boundaries and meaning and whose properties and behaviour follow the same rules

3.2
property
quality common to all members of an *object class* (3.1)

3.3
characteristic
abstraction of a *property* (3.2)

EXAMPLE 'Having a cable for connecting with a computer' as a characteristic of the concept 'cord mouse'.

Note 1 to entry: Characteristics are used for describing *concepts* (3.4).

[SOURCE: ISO 1087:2019, 3.2.1]

3.4
concept
unit of knowledge created by a unique combination of *characteristics* (3.3)

Note 1 to entry: Concepts are not necessarily bound to particular natural languages. They are, however, influenced by the social or cultural background which often leads to different categorizations.

Note 2 to entry: A concept is independent of its representation.

[SOURCE: ISO 1087:2019, 3.2.7, modified — Note 2 to entry changed.]

3.5
conceptual domain
CD
concept (3.4) whose meaning is expressed as an enumerated set, a description of subordinate concepts or both, which are *value meanings* (3.10)

3.6
described conceptual domain
conceptual domain (3.5) that is specified by a description or specification, such as a rule, a procedure or a range (i.e. interval)

3.7
enumerated conceptual domain
conceptual domain (3.5) that is specified by a list of all its *value meanings* (3.10)

Note 1 to entry: No ordering of the value meanings is implied.

3.8
local enumerated conceptual domain
enumerated conceptual domain (3.7) whose *value meanings* (3.10) are enumerated locally within the registry

Note 1 to entry: c.f. *reference enumerated conceptual domain* (3.11).

3.9
local enumerated conceptual domain subset
subset of the *value meanings* (3.10) in a *local enumerated conceptual domain* (3.8) used to restrict the value meanings a *data element concept* (3.25) can assume in a particular context

3.10
value meaning
semantic content of a value

Note 1 to entry: The representation of value meanings in a *registry* (3.36) shall be independent of (and shall not constrain) their representation in any corresponding *value domain* (3.13).

3.11
reference enumerated conceptual domain
enumerated conceptual domain (3.7) that is specified by a formal definition

Note 1 to entry: The definition may reference externally enumerated *value meanings* (3.10).

3.12
enumerated conceptual domain definition
formal definition of an *enumerated conceptual domain* (3.7)

Note 1 to entry: The definition may reference externally enumerated *value meanings* (3.10).

3.13
value domain
VD
set of *permissible values* (3.19)

Note 1 to entry: The *value domain* provides representation but has no implication as to what *data element concept* (3.25) the values are associated with nor what the values mean.

Note 2 to entry: The *permissible values* can either be enumerated, expressed via a description, or a combination of the two.

3.14
described value domain
value domain (3.13) that is specified by a description or specification, such as a rule, a procedure or a range (i.e. interval)

3.15
enumerated value domain
value domain (3.13) that is specified by a list of all its *permissible values* (3.19)

Note 1 to entry: No ordering of the permissible values is implied.

3.16
local enumerated value domain
enumerated value domain (3.15) whose *permissible values* (3.19) are stored within the registry

Note 1 to entry: c.f. *reference enumerated value domain* (3.18).

3.17
local enumerated value domain subset
subset of the *permissible values* (3.19) in a *local enumerated value domain* (3.16) used to restrict the value meanings a *data element* (3.23) can assume in a particular context

3.18
reference enumerated value domain
enumerated value domain (3.15) that is specified by reference to an external specification, including externally enumerated *permissible values* (3.19)

3.19
permissible value
designation of a *value meaning* (3.10)

Note 1 to entry: Permissible values may be specified either as part of a *value domain* (3.13) or only associated with a *value meaning* (3.10).

Note 2 to entry: Within a value domain, permissible values can either be enumerated, expressed via a description, or a combination of the two.

Note 3 to entry: Explicit mapping of a single permissible value to a single value meaning is possible only when both the value meaning and permissible value are enumerated, e.g. for code sets. For described permissible values, it is possible for the described meaning to be associated with a range of values, e.g. weight in kilograms.

3.20
data
re-interpretable representation of information in a formalized manner suitable for communication, interpretation or processing

Note 1 to entry: Data can be processed by humans or by automatic means.

[SOURCE: ISO/IEC 2382:2015, 2121272, modified — Notes to entry 2 and 3 deleted.]

3.21
datatype
set of distinct values, characterized by properties of those values and by operations on those values

[SOURCE: ISO/IEC 11404:2007, 3.12]

3.22
datatype scheme
source of the specification of one or more *datatypes* (3.21)

[SOURCE: ISO/IEC 11404:2007, 3.12]

3.23
data element
⟨organization of data⟩ unit of *data* (3.20) that is considered in context to be indivisible

Note 1 to entry: The definition states that a data element is "indivisible" in some context. This means that it is possible that a data element considered indivisible in one context (e.g. telephone number) may be divisible in another context, (e.g. country code, area code, local number).

EXAMPLE The data element "age of a person" with values consisting of all combinations of 3 decimal digits.

[SOURCE: ISO/IEC 2382:2015, 2121599, modified — Example moved to the end without the Note to entry prefix. Other Notes to entry have been replaced.]

3.24
data element collection
one or more *data elements* (3.23) that may be unordered or ordered

Note 1 to entry: Examples of unordered collections are a set or a bag (or multiset). An example of an ordered collection is a list.

3.25
data element concept
concept (3.4) that can be represented in the form of a *data element* (3.23), described independently of any particular representation

Note 1 to entry: A data element concept is implicitly associated with both the property and the object class whose combination it expresses.

Note 2 to entry: A data element concept may also be associated with zero, one or more *conceptual domains* (3.5) each of which expresses its *value meanings* (3.10).

Note 3 to entry: A data element concept may also be associated with zero, one or more *data elements* (3.23) each of which provide representation for the data element concept via its associated *value domain* (3.13).

3.26
data element derivation
application of a *data element derivation rule* (3.28) to one or more input *data elements* (3.23) to derive one or more output data elements

3.27
data element example
representative illustration of a *data element* (3.23)

3.28
data element derivation rule
logical, mathematical, other operations or some combination specifying derivation

3.29
dimensionality
set of equivalent *units of measure* (3.33)

Note 1 to entry: Equivalence between two units of measure is determined by the existence of a quantity preserving one-to-one correspondence between values measured in one unit of measure and values measured in the other unit of measure, independent of context, and where characterizing operations are the same.

Note 2 to entry: The equivalence defined here forms an equivalence relation on the set of all units of measure. Each equivalence class corresponds to a dimensionality. The units of measure "temperature in degrees Fahrenheit" and "temperature in degrees Celsius" have the same dimensionality, because:

a) given a value measured in degrees Fahrenheit there is a value measured in degrees Celsius with the same quantity, and vice-versa, by the well-known correspondences ºC = (5/9)*(ºF - 32) and ºF = (9/5)*(ºC) + 32.

b) the same operations can be performed on both values.

Note 3 to entry: The units of measure "temperature in degrees Celsius" and "temperature in kelvins" do not belong to the same dimensionality. Even though it is easy to convert quantities from one unit of measure to the other (°C = K - 273,15 and K = °C + 273,15), the characterizing operations in kelvins include taking ratios, whereas this is not the case for degrees Celsius. For instance, 20 K is twice as warm as 10 K, but 20 °C is not twice as warm as 10 °C.

Note 4 to entry: Units of measure are not limited to physical categories. Examples of physical categories are: linear measure, area, volume, mass, velocity, time duration. Examples of non-physical categories are: currency, quality indicator, colour intensity.

Note 5 to entry: Quantities may be grouped together into categories of quantities which are mutually comparable. Lengths, diameters, distances, heights, wavelengths and so on would constitute such a category. Mutually comparable quantities have the same dimensionality. ISO 80000-1[13] calls these “quantities of the same kind”.

Note 6 to entry: ISO 80000-1 specifies physical dimensions (e.g. length, mass, velocity). This document also permits non-physical dimensions (e.g. value dimensions such as: currency, quality indicator). The present concept of dimensionality equates to what ISO 80000-1 calls Dimensional Product, rather than to Dimension.

3.30
measure class
set of equivalent *units of measure* (3.33) for association with one or more *dimensionalities* (3.29)

3.31
coordinate
measurement from the origin of a frame of reference

3.32
notation
formal syntax and associated semantics

EXAMPLE UML, MOF, OCL, OWL/RDF, SKOS, CGIF, XCL, XTM or ISO/IEC 11404

Note 1 to entry: Formal syntax is often intended for machine processing.

[SOURCE: ISO/IEC 11179-3:2023, 3.2.36]

3.33
unit of measure
⟨value domain⟩ actual units in which the associated values are measured

Note 1 to entry: ISO 80000-1[13] specifies a system of physical measurement (the International System of Units, SI). Physical measurement is only one type of measurement. Value measurement is another type of measurement. This document permits the use of any appropriate system of measurement.

Note 2 to entry: The *dimensionality* (3.29) of the associated *conceptual domain* (3.5) shall be appropriate for the specified *unit of measure*.

3.34
unit of measure dimensionality
dimensionality (3.29) that specifies the equivalence relation that applies to all values representing a particular unit

3.35
quantity
value associated with a *unit of measure* (3.33)

Note 1 to entry: 32º Fahrenheit and 0º Celsius are quantities, and they are equivalent values in different measuring systems.

3.36
registry
information system for *registration* (3.37)

[SOURCE: ISO/IEC 11179-1:2023, 3.2.34]

3.37
registration
set of rules, operations and procedures for inclusion of an item in a *registry* (3.36)

Note 1 to entry: A detailed description of registration as it applies in ISO/IEC 11179 is found in ISO/IEC 11179-6.

[SOURCE: ISO/IEC 11179-1:2023, 3.2.88]

4 Abbreviated terms

CD	conceptual domain
DE	data element
DEC	data element concept
UML	Unified Modeling Language
URI	Universal Resource Identifier
VD	value domain
XML	eXtensible Markup Language

5 Conformance

5.1 Overview of conformance

Conformance rules for a metadata registry are specified in ISO/IEC 11179-3:2023, Clause 4. The subclause “Degree of conformance” is repeated here for convenience. The subsequent subclauses extend the rules from ISO/IEC 11179-3.

5.2 Degree of conformance

5.2.1 General

The distinction between “strictly conforming” and “conforming” implementations is necessary to address the simultaneous needs for interoperability and extensions. This document describes

specifications that promote interoperability. Extensions are motivated by needs of users, vendors, institutions and industries, and:

a) are not directly specified by this document;

b) are specified and agreed to outside this document;

c) may serve as trial usage for future editions of this document.

A strictly conforming implementation can be limited in usefulness but is maximally interoperable with respect to this document. A conforming implementation can be more useful but can be less interoperable with respect to this document.

5.2.2 Strictly conforming implementations

A strictly conforming implementation:

a) shall support all mandatory, optional and conditional classes, attributes, datatypes and associations;

b) shall not use, test, access or probe for any extension features nor extensions to classes, attributes, datatypes, associations or any combination thereof;

c) shall not recognize, nor act on, nor allow the production of classes, attributes, datatypes, associations or any combination thereof that are dependent on any unspecified, undefined or implementation-defined behaviour.

NOTE The use of extensions to the metamodel can cause undefined behaviour.

5.2.3 Conforming implementations

A conforming implementation:

a) shall support all mandatory, optional and conditional classes, attributes, datatypes and associations;

b) as permitted by the implementation, may use, test, access or probe for extension features or extensions to classes, attributes, datatypes, associations or any combination thereof;

c) may recognize, act on or allow the production of classes, attributes, datatypes, associations or any combination thereof that are dependent on implementation-defined behaviour.

NOTE 1 All strictly conforming implementations are also conforming implementations.

NOTE 2 The use of extensions to the metamodel can cause undefined behaviour.

5.3 Conformance by feature

Conformance claims may be made to the whole of Clause 7 or to specific features within that clause. Clause 7 is dependent upon one or more clauses of ISO/IEC 11179-3, so conformance to all or part of Clause 7 shall be understood to imply conformance also to relevant provisions specified in one or more of the clauses in ISO/IEC 11179-3.

A conformance statement shall specify exactly the features supported and not supported.

5.4 Registry conformance

5.4.1 Standard profiles for edition 4 registries

This document specifies the following standard profiles in addition to those specified in ISO/IEC 11179-3:2023, 4.4.2.

— **Data Specification Registry**: Implements Clause 7 of this document, in addition to all provisions of the "Basic Registry" profile of ISO/IEC 11179-3:2023, 4.4.2.

— **Data Specification Registry with mapping**: Implements Clause 7 of this document, in addition to all provisions of the "Basic Registry with mapping" profile of ISO/IEC 11179-3:2023, 4.4.2.

— **Conceptual and Value Domain Registry:** Implements 7.3 of this document, in addition to all the provisions of the "Basic Registry" profile of ISO/IEC 11179-3:2023, 4.4.2.

— **Conceptual and Value Domain Registry with mapping:** Implements 7.3 of this document, in addition to all the provisions of the "Basic Registry with mapping" profile of ISO/IEC 11179-3:2023, 4.4.2.

5.4.2 Conformance labels

Conformance to the profiles specified in 5.4.1 may be claimed using the following labels, respectively:

— ISO/IEC 11179-31:2023 Data Specification Registry

— ISO/IEC 11179-31:2023 Data Specification Registry with mapping

— ISO/IEC 11179-31:2023 Conceptual and Value Domain Registry

— ISO/IEC 11179-31:2023 Conceptual and Value Domain Registry with mapping.

5.5 Implementation conformance statement (ICS)

An implementation claiming conformance to this document shall include an Implementation Conformance Statement stating:

— whether it conforms or strictly conforms;

— which clauses are or are not supported;

— what extensions, if any, are supported or used.

A standard profile may be referenced, if applicable.

EXAMPLE Product Y conforms to ISO/IEC 11179-31:2023 Data Specification Registry with Mapping, except that it relies on the Item Mapping facility of ISO/IEC 11179-3:2023 in place of implementing the ISO/IEC 11179-31:2023 Data Element Derivation feature (7.6.2.5, 7.6.2.6, 7.6.3.4, 7.6.3.5, 7.6.3.6).

5.6 Obligation

Properties and relationships specified in this document are one of: "mandatory", "conditional" or "optional". The obligation is not explicitly stated but is to be inferred from the multiplicity of the property or relationship, and the presence or absence of a condition. In addition, a registration authority can specify additional constraints to be applied to particular **Administered_Item**s (ISO/IEC 11179-3:2023, 9.4.2), using **Constraint_Sets** (ISO/IEC 11179-3:2023, 9.4.5). (See 6.5.)

For the purpose of conformance:

a) mandatory properties and relationships shall exist and shall conform to the provisions of this document;

b) anything specified as Conditional within this document shall be treated as Mandatory if the associated condition is satisfied and shall otherwise be not present;

c) optional properties and relationships are not required to exist, but if they do exist, they shall conform to the provisions of this document.

Such obligation is enforced if and only if the **Registration_Status** of the associated registry items is "Recorded" or higher (see ISO/IEC 11179-3:2023, 3.2.70 and 9.4.4.3; and ISO/IEC 11179-6:2023, 4.3.2).

6 Relationship to ISO/IEC 11179-3

6.1 Metamodel for a metadata registry

A metamodel is a model that describes other models. A metamodel provides a mechanism for understanding the precise structure and components of the specified models, which are needed for the successful sharing of the models by users, software facilities or both.

ISO/IEC 11179-3 uses a metamodel to describe the information model of a metadata registry. The registry in turn will be used to describe and model other data, for example about enterprise, public administration or business applications. The registry metamodel is specified as a conceptual data model, i.e. one that describes how relevant information is structured in the natural world. In other words, it is how the human mind is accustomed to thinking of the information.

6.2 Specification of the metamodel

The conventions used in specifying the metamodel are described in ISO/IEC 11179-3:2023, 5.3. Many of the classes specified in the Data Specification package (see Clause 7) inherit from **Item**, which is specified in the Core model region in ISO/IEC 11179-3:2023, 6.4.2.1. As **Item**s, instances of these classes may be identified, registered, administered, named, defined and classified.

6.3 Use of UML Class diagrams and textual description

This document uses both text and UML class diagrams (References [10], [11]) to describe the metamodel. Both are normative and are intended to be complementary. However, if a conflict exists between what is specified in UML and what is specified in text, the text takes precedence until a correction is made to make them consistent. Further, if a conflict exists between a formal definition and other normative text, the formal definition takes precedence until a correction is made to make them consistent.

A consolidated UML class hierarchy is included as Annex A.

While the model diagrams are presented in UML notation, this document does not assume nor endorse any specific system environment, database management system, database design paradigm, system development methodology, data definition language, command language, system interface, user interface, computing platform or any technology required for implementation.

6.4 Package dependencies

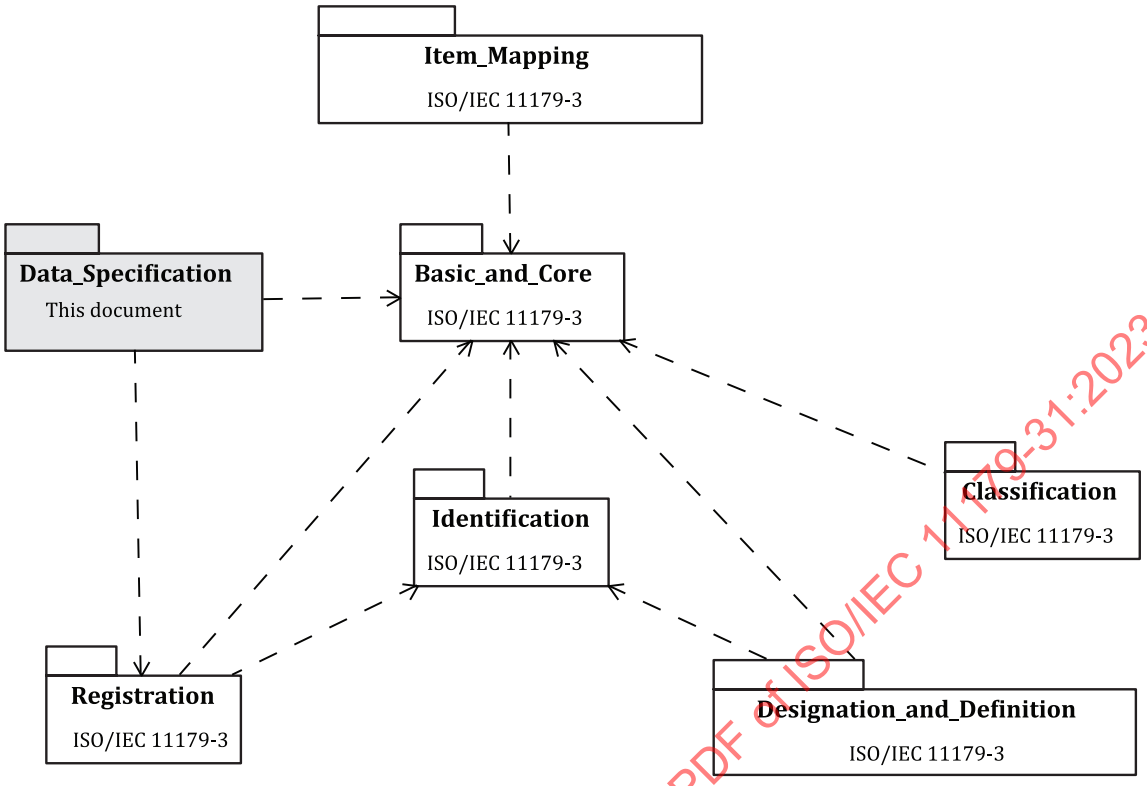


Figure 1 — Package dependencies

Figure 1 illustrates the dependencies among the packages. The lines in Figure 1 illustrate dependencies in the direction of the arrow. The Data Specification package is specified in Clause 7. The other packages shown in the figure are specified in ISO/IEC 11179-3. In order to implement a package that has dependencies, the packages on which it is dependent shall also be implemented. The dependencies are of three types:

a) Subclassing from classes in another package, e.g. **Conceptual_Domain** (7.3.2.4), **Data_Element_Concept** (7.3.2.3), **Object_Class** (7.3.2.1) and **Property** (7.3.2.2) in the Data Element Concept region (7.3) of the Data_Specification package (Clause 7) are all subclassed from the **Concept** class in the Basic_and_Core package specified in ISO/IEC 11179-3:2023, Clause 6.

b) Association between classes, e.g. **Registered_Item** in the Registration package in ISO/IEC 11179-3:2023, 9.4.1 has an association with **Reference_Document** in the Basic_and_Core package specified in ISO/IEC 11179-3:2023, Clause 6.

c) Some attributes use a predefined datatype or a class from another package as a datatype e.g. the **contact** attribute of the **Stewardship_Record** class in the Registration package in ISO/IEC 11179-3:2023, 9.4.8 uses the **Contact** class of the Basic_and_Core package specified in ISO/IEC 11179-3:2023, Clause 6 as a datatype.

Conformance options are specified in Clause 5 and standard conformance profiles in 5.4.1.

6.5 Subclassing the Constraint_Set class

This document extends the **Constraint_Set** class (ISO/IEC 11179-3:2023, 9.4.5) by specifying subclasses to support constraints specified in this document. See Figure 2.

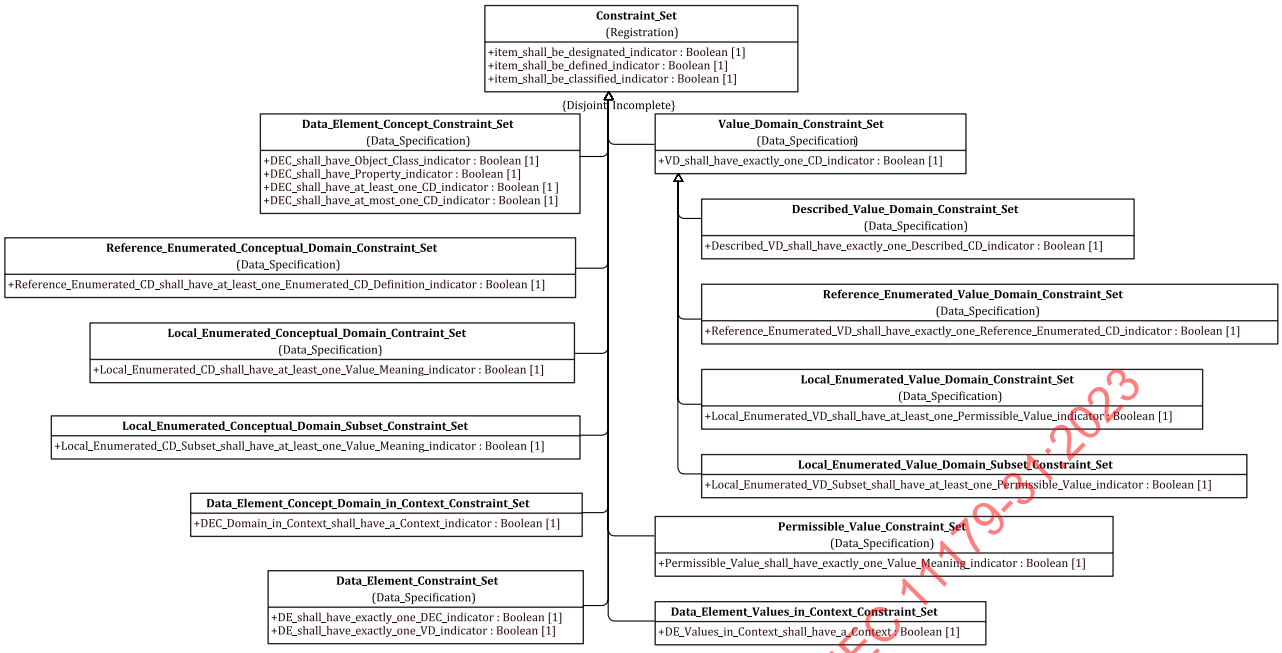


Figure 2 — Subclasses of Constraint_Set

For details, see the following subclauses:

— **Data_Element_Concept_Constraint_Set** class (7.3.2.5);

— **Reference_Enumerated_CD_Constraint_Set** class (7.4.2.16);

— **Local_Enumerated_CD_Constraint_Set** class (7.4.2.17);

— **Local_Enumerated_CD_Subset_Constraint_Set** class (7.8.2.8);

— **Data_Element_Constraint_Set** class (7.6.2.7);

— **Value_Domain_Constraint_Set** class (7.4.2.18);

— **Described_Value_Domain_Constraint_Set** class (7.4.2.19);

— **Reference_Enumerated_VD_Constraint_Set** class (7.4.2.20);

— **Local_Enumerated_VD_Constraint_Set** class (7.4.2.21);

— **Local_Enumerated_VD_Subset_Constraint_Set** class (7.7.2.8);

— **Permissible_Value_Constraint_Set** class (7.4.2.22).

Any registry implementation shall provide a mechanism to enforce these constraints.

7 Data_Specification package

7.1 Overview of the Data_Specification package

This package extends the Basic_and_Core package specified in ISO/IEC 11179-3:2023, Clause 6 by sub-classing classes in that package, both **Item** (ISO/IEC 11179-3:2023, 6.4.2.1) and **Concept** (ISO/IEC 11179-3:2023, 6.4.2.2).

The Data Specification package consists of eight metamodel regions, or views of the model:

— the High-level Data Specification metamodel region (7.2);

— the Data Element Concept metamodel region (7.3);

— the Conceptual and Value Domain metamodel region (7.4);

— the Measurement metamodel region (7.5);

— the Data Element metamodel region (7.6);

— the Value Domain Subset metamodel region (7.7);

— the Conceptual Domain Subset metamodel region (7.8);

— the Composite Data Element and Datatypes metamodel region (7.9).

7.2 High-level Data Specification metamodel region

7.2.1 Overview of the high-level metamodel region

A high-level overview of the metamodel for this region can be found in Figure 3. It shows four classes: **Conceptual_Domain** (7.2.2.2), **Value_Domain** (7.2.2.3), **Data_Element** (7.2.2.4) and **Data_Element_Concept** (7.2.2.5). The Figure also shows four associations among the four classes: **value_domain_meaning** (7.2.3.1), **data_element_domain** (7.2.3.2), **data_element_meaning** (7.2.3.3) and **data_element_concept_domain** (7.2.3.4).

The following text describes the classes and associations shown in the Figure. It also describes a constraint on the high level metamodel not visible in the UML diagram. More detailed descriptions, e.g. of the class attributes, follow in subsequent subclauses.

Figure 3 can be partitioned into two horizontal parts, one upper part comprised of **Data_Element_Concept** and **Conceptual_Domain** and a second lower part comprised of **Data_Element** and **Value_Domain**. This view effectively splits the metamodel between a conceptual (or semantic) level (at the top) and a representational level (below). The representational level describes the information artifacts (in contrast to the semantic constructs of the upper level).

This high-level metamodel omits many details, e.g. attributes and some associations, in the interest of clarity of exposition. For a complete characterization of the metamodel see 7.3 through 7.9. A partial consolidated detailed metamodel is presented in Annex C.

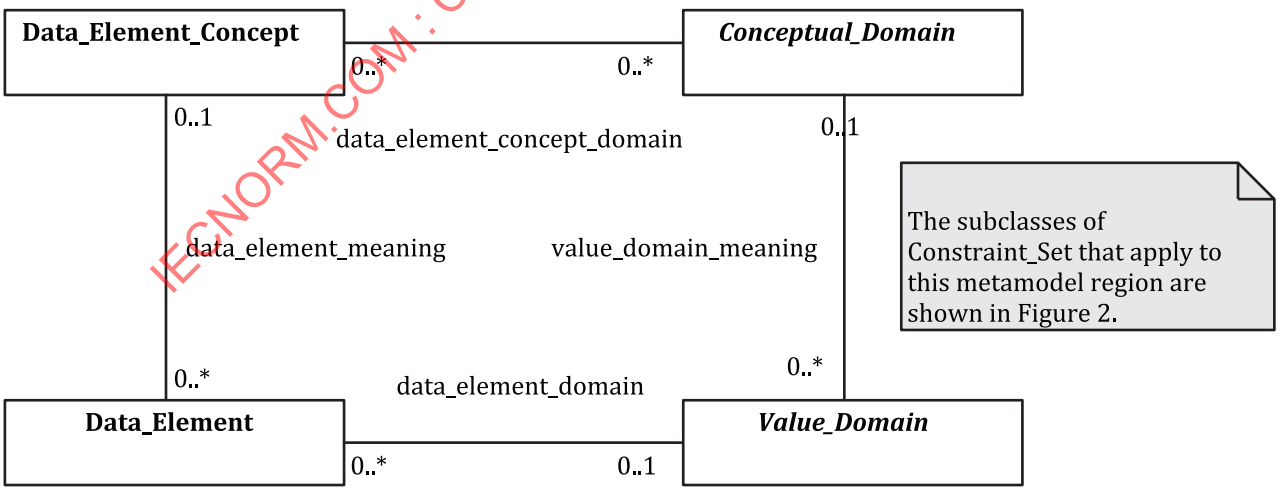


Figure 3 — High-level Data Specification metamodel

ISO/IEC 11179-3:2003[4] required a data element concept to be associated with exactly one conceptual domain. This association was relaxed in ISO/IEC 11179-3:2013[5] and in this document because of feedback from organizations using ISO/IEC 11179-3:2003:

— The lower bound has been relaxed from one to zero to allow a data element concept to be recorded without an associated conceptual domain.

— The upper bound has been increased from one to many to avoid a proliferation of similar data element concepts. For example, it is possible for the data element concept "Person's sex" in one context to be associated with a conceptual domain of { male , female } and in another context with a conceptual domain of { male , female , unknown , not disclosed }. ISO/IEC 11179-3:2003 required that separate data element concepts be specified. ISO/IEC 11179-3:2013 and this document allow an organization to choose whether to specify separate data element concepts or to share a single data element concept. Whichever approach is chosen for data element concepts, it is best practice for separate data elements to be defined and associated with the appropriate value domains.

A registration authority can reapply the constraints which have been relaxed. See 7.2.4.2.

7.2.2 Classes in the High-level Data Specification metamodel region

7.2.2.1 Overview

The classes shown in Figure 3 are described in this subclause, starting with **Conceptual_Domain** (7.2.2.2) and proceeding clockwise around the figure.

7.2.2.2 Conceptual_Domain class

7.2.2.2.1 Direct superclass

Conceptual_Domain is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.2.2 Description of Conceptual_Domain

Conceptual_Domain is a class each instance of which models a conceptual domain, a concept whose meaning is expressed as an enumerated set, a description of subordinate concepts or both, which are value meanings.

For example, one possible conceptual domain is "countries of the world". It can, for example, be associated with two value domains: "three letter country codes" and "full country names". The conceptual domain can be used in several data element concepts, e.g. "person's country of residence", "person's country of birth", "person's country of citizenship".

A conceptual domain can facilitate the mapping of equivalent values of two or more value domains that share the conceptual domain.

Conceptual_Domain is further described in 7.4.2.1.

7.2.2.2.3 Associations of Conceptual Domain

As a subclass of **Concept**, **Conceptual_Domain** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Conceptual_Domain** has the following additional associations:

— **data_element_concept_domain** (7.2.3.4);

— **value_domain_meaning** (7.2.3.1).

More associations of **Conceptual_Domain** are described in 7.4.2.1.3.

7.2.2.2.4 Attributes of Conceptual_Domain

Attributes of **Conceptual_Domain** are described in 7.4.2.1.4.

7.2.2.3 Value_Domain class

7.2.2.3.1 Direct superclass

Value_Domain is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.3.2 Description of Value_Domain

Value_Domain is a class each instance of which models a value domain, a collection of permissible values. A value domain provides representation but has no implication as to what data element concept the values are associated with, nor what the values mean. The binding of permissible values to their corresponding value meanings is described in 7.4.3.2.2.

Instances of the **Value_Domain** class are each associated with zero or one instance of the **Conceptual_Domain** class (7.2.2.2) through instances of the association **value_domain_meaning** (7.2.3.1). Through this association, a value domain provides a representation for the conceptual domain and the conceptual domain provides meaning for the value domain. Although the association is shown as optional in both directions, without such an association, a value domain has no meaning and a conceptual domain has no representation.

An example of a conceptual domain and a set of value domains is ISO 3166[1] Codes for the representation of names of countries. For instance, ISO 3166 describes the set of seven value domains:

— short name in English;

— official name in English;

— short name in French;

— official name in French;

— alpha-2 code;

— alpha-3 code;

— numeric code.

Additional examples of value domains are:

— Sex, which contains two designations (permissible values), M -> Male and F-> Female;

— Parent, which contains two designations (permissible values), M -> Mother and F -> Father.

NOTE These two value domains are defined over the same set of values, but they are mapped to separate conceptual domains.

Value_Domain is further described in 7.4.2.8.

Value domains may be reused for multiple data elements. See the discussion of countries of the world in 7.2.2.2.2.

7.2.2.3.3 Associations of Value_Domain

As a subclass of **Item**, **Value_Domain** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). **Value_Domain** has the following additional associations:

— **value_domain_meaning** (7.2.3.1);

— **data_element_domain** (7.2.3.2).

7.2.2.3.4 Attributes of Value_Domain

Attributes of **Value_Domain** are described in 7.4.2.8.4.

7.2.2.4 Data_Element class

7.2.2.4.1 Direct superclass

Data_Element is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.4.2 Description of Data_Element

Data_Element is a class each instance of which models a data element, a unit of data that is considered in context to be indivisible. A data element is a basic unit of data of interest to an organization, for which the definition, identification, representation and permissible values are specified by means of a set of attributes. Examples of data element include: a column in a table of a database managed using the SQL database language, a field in a record or form, an XML element, the attribute of a Java class or a variable in a program. The description of data elements was the original major purpose of the metadata registries described in ISO/IEC 11179.

Data_Element is further described in 7.6.2.1.

7.2.2.4.3 Associations of Data_Element

As a subclass of **Item**, **Data_Element** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). **Data_Element** has the following additional associations:

— **data_element_meaning** (7.2.3.3);

— **data_element_domain** (7.2.3.2).

Additional associations of **Data_Element** are described in 7.6.2.1.3.

7.2.2.4.4 Attributes of Data_Element

Attributes of **Data_Element** are described in 7.6.2.1.4.

7.2.2.5 Data_Element_Concept class

7.2.2.5.1 Direct superclass

Data_Element_Concept is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.2.2.5.2 Description of Data_Element_Concept

Data_Element_Concept is a class each instance of which models a data element concept, a concept that is an association of a property with an object class. A data element concept is a concept that can be represented in the form of a data element, described independently of any particular representation.

A data element concept is a usage of a conceptual domain, e.g. "person's country of residence" vs. "country", which effectively narrows the meaning of the conceptual domain.

A data element concept is an abstraction of one or more data elements. Each data element addresses issues of concrete representation, e.g. codes, measurement units, etc. A data element concept may be represented by multiple data elements, which may vary in their value domains.

A data element concept can facilitate the mapping of equivalent values of two or more data elements that share the data element concept.

Data_Element_Concept is further described in 7.3.2.3.

7.2.2.5.3 Associations of Data_Element_Concept

As a subclass of **Concept**, **Data_Element_Concept** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Data_Element_Concept** has the following additional associations:

— **data_element_concept_domain** (7.2.3.4);

— **data_element_meaning** (7.2.3.3).

Other associations of **Data_Element_Concept** are described in 7.3.2.3.3.

7.2.2.5.4 Attributes of Data_Element_Concept

As a subclass of **Concept**, **Data_Element_Concept** inherits **Concept**'s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4).

7.2.3 Associations of the High-Level Data Specification metamodel region

7.2.3.1 value_domain_meaning association

The **value_domain_meaning** association records the binding of zero, one or more instances of the **Value_Domain** class (7.2.2.3) to zero or one instances of the **Conceptual_Domain** class (7.2.2.2). The association records the value domains which provide representation to the conceptual domain, which in turn provides meaning to the value domains.

It is best practice for a value domain to be associated with exactly one conceptual domain. A registration authority can enforce this using the **Value_Domain_Constraint_Set.VD_shall_have_exactly_one_CD** constraint (7.4.2.18.4).

A value domain is a collection of permissible values which provide values for associated value meanings. The existence of a **value_domain_meaning** association between an instance of the **Conceptual_Domain** class and an instance of the **Value_Domain** class implies the existence of associations between the corresponding individual value meanings and permissible values. These associations are recorded in this metamodel as **Permissible_Values** (7.4.2.12) for enumerated value domains. For described value domains, the associations are implicit.

NOTE In this metamodel, value domains are constrained to have a unique set of meanings (the associated conceptual domain), i.e. a value domain is a function from permissible values to value meanings. If for some reason one wanted to reuse a value domain (and the associated values, e.g. a code set) for more than one meaning (e.g. see the additional examples in 7.2.2.3), one is forced to create another value domain and another set of permissible values. This constraint is enforced so that within a value domain, one can unambiguously determine the value meanings (in the conceptual domain) for the permissible values (in the value domain) associated with a data element. See discussion under Constraints in 7.2.4.1.

7.2.3.2 data_element_domain association

The **data_element_domain** association binds zero, one or more instances of the **Data_Element** class (7.2.2.4) to zero or one instances of the **Value_Domain** class (7.2.2.3) which specifies the permissible values which may be stored in the data element. The value domain provides a domain of permissible values for the data element, which in turn provides usages of the value domain.

It is best practice for a data element to be associated with exactly one value domain. A registration authority can enforce this using the constraint **Data_Element_Constraint_Set.DE_shall_have_exactly_one_VD** (7.6.2.7.4).

7.2.3.3 data_element_meaning association

The **data_element_meaning** association binds zero, one or more instances of the **Data_Element** class (7.2.2.4) to zero or one instance of the **Data_Element_Concept** class (7.2.2.5). The data element concept provides meaning to the data elements, which in turn provide representation to the data element concept.

It is best practice for a data element to be associated with exactly one data element concept. A registration authority can enforce this using the **Data_Element_Constraint_Set.DE_shall_have_exactly_one_DEC** constraint (7.6.2.7.4).

7.2.3.4 data_element_concept_domain association

The **data_element_concept_domain** association binds zero, one or more instances of the **Data_Element_Concept** class (7.2.2.5) to zero, one or more instances of the **Conceptual_Domain** class (7.2.2.2). The conceptual domains provide the domain for each data element concept, which in turn provide usages for the conceptual domain.

It is best practice for a data element concept to be associated with exactly one conceptual domain. A registration authority can enforce this using the constraints **Data_Element_Concept_Constraint_Set.DEC_shall_have_at_least_one_CD** and **Data_Element_Concept_Constraint_Set.DEC_shall_have_at_most_one_CD** (7.3.2.5.4).

The **data_element_concept_domain** association narrows the scope (meaning) of a conceptual domain to that of the data element concept.

EXAMPLE "person's country of birth" (data element concept) vs. "country" (conceptual domain).

NOTE ISO/IEC 11179-3:2003[4] required a data element concept to be associated with exactly one conceptual domain. This association was relaxed in ISO/IEC 11179-3:2013[5] and in this document because of feedback from organizations using ISO/IEC 11179-3:2003. The lower bound has been relaxed from one to zero to allow a data element concept to be recorded without an associated conceptual domain. The upper bound has been increased from one to many, to avoid a proliferation of similar data element concepts. For example, it is possible for the data element concept: "Person's marital status" in one context to be associated with a conceptual domain of { single , married }, and in another context with a conceptual domain of { single , married , divorced , widowed }. ISO/IEC 11179-3:2003 required that separate data element concepts be specified to support this. ISO/IEC 11179-3:2013 and this document allow an organization to choose whether to specify separate data element concepts or to share a single data element concept.

Even if a data element concept is associated with multiple conceptual domains, separate data elements should still be defined and associated with the appropriate value domains.

7.2.4 Constraints of the High Level metamodel region

7.2.4.1 Equality of mappings from data element to conceptual domain

There are two paths in the metamodel in Figure 3 from the **Data_Element** class to the **Conceptual_Domain** class. One can either proceed clockwise from **Data_Element** class via the **data_element_meaning** association to **Data_Element_Concept** class and then via **data_element_concept_domain** association to the **Conceptual_Domain** class; or alternatively, one can proceed counterclockwise from the **Data_Element** class via the **data_element_domain** association to the **Value_Domain** class and then via the **value_domain_meaning** association to the **Conceptual_Domain** class.

CONSTRAINT: It shall be the case that starting from a specific instance of the **Data_Element** class that the instance of the **Conceptual_Domain** class reached via the **Value_Domain** shall be one of those reached via the **Data_Element_Concept**.

It is best practice for a data element concept to be related to exactly one conceptual domain, in which case the instance of the **Conceptual_Domain** class reached via the **Data_Element_Concept** shall be the same as the instance of the **Conceptual_Domain** class reached via the **Value_Domain**.

NOTE The possible inverse constraint (starting from **Conceptual_Domain**) is not true, because the associations are not functions (uniquely valued) in the inverse directions.

7.2.4.2 Discretionary constraints

Some constraints can be applied at the discretion of the registration authority responsible for a registered item. These are specified in other detailed metamodel regions, but some apply to this high-level model region. See:

— **Data_Element_Concept_Constraint_Set** class (7.3.2.5);

— **Data_Element_Constraint_Set** class (7.6.2.7);

— **Value_Domain_Constraint_Set** class (7.4.2.18).

7.3 Data Element Concept metamodel region

7.3.1 Overview of the Data Element Concept metamodel region

The Data Element Concept metamodel region is illustrated in Figure 4. The purpose of this region is to maintain the information on the concepts related to data elements. The metadata objects (ISO/IEC 11179-3:2023, 3.2.31) in this region concern semantics. Concepts are independent of any internal or external physical representation.

The metadata objects in this region are: **Conceptual_Domain**s (7.3.2.4), **Data_Element_Concept**s (7.3.2.3), **Object_Classes** (7.3.2.1) and **Properties** (7.3.2.2). **Object_Classes** and **Properties** may be combined to form **Data_Element_Concepts**. All these metadata objects are subclasses of **Concept** (ISO/IEC 11179-3:2023, 6.4.2.2).

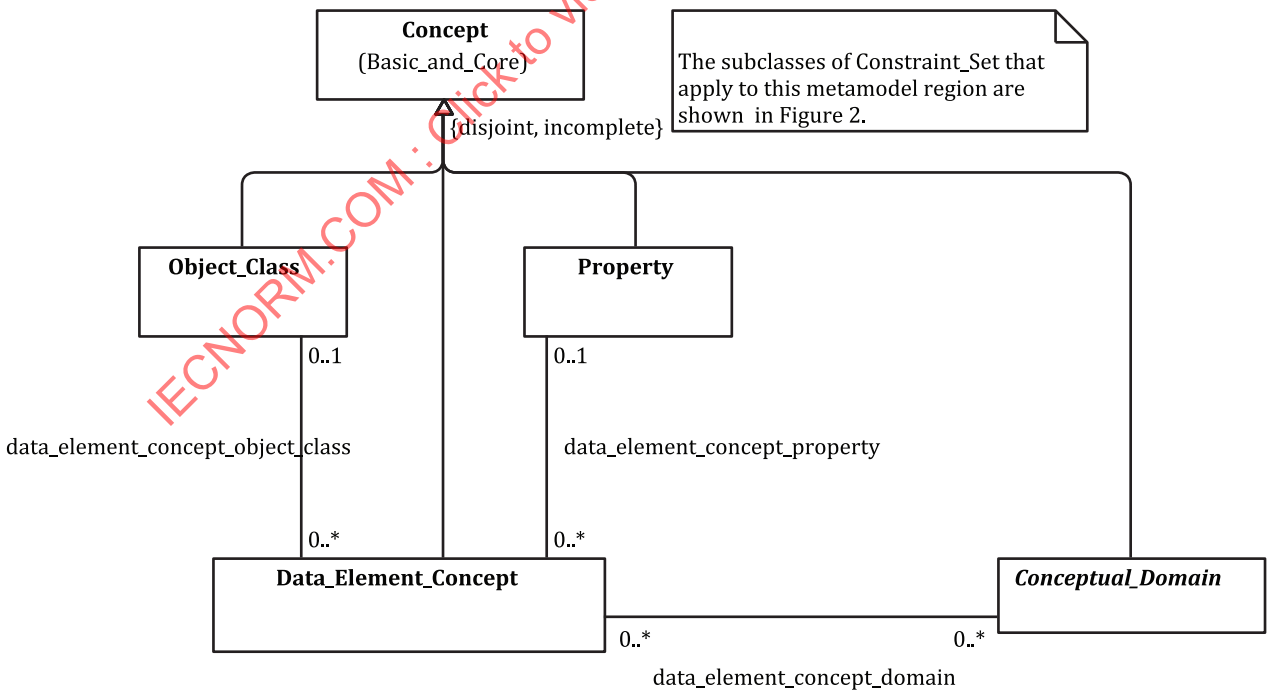


Figure 4 — Data Element Concept metamodel region

7.3.2 Classes in the Data Element Concept metamodel region

7.3.2.1 Object_Class class

7.3.2.1.1 Direct superclass

Object_Class is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.3.2.1.2 Description of Object_Class

Object_Class is a class each instance of which models an object class. An object class is a concept that represents a set of ideas, abstractions or things in the real world that can be identified with explicit boundaries and meaning and whose properties and behavior follow the same rules. It may be either a single or a group of associated concepts, abstractions or things.

7.3.2.1.3 Associations of Object_Class

As a subclass of **Concept**, **Object_Class** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Object_Class** has the following additional association:

— **data_element_concept_object_class** association (7.3.3.1).

7.3.2.1.4 Attributes of Object_Class

As a subclass of **Concept**, **Object_Class** inherits **Concept**'s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4). No additional attributes are specified in this metamodel region.

7.3.2.2 Property class

7.3.2.2.1 Direct superclass

Property is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (also specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.3.2.2.2 Description of Property

Property is a class each instance of which models a property, a quality common to all members of an object class. A property may be any feature that humans naturally use to distinguish one individual object from another. It is the human perception of a single quality of an object class in the real world. It is conceptual and thus has no particular associated means of representation by which the property can be communicated.

7.3.2.2.3 Associations of Property

As a subclass of **Concept**, **Property** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Property** has the following additional association:

— **data_element_concept_property** association (7.3.3.2).

7.3.2.2.4 Attributes of Property

As a subclass of **Concept**, **Property** inherits **Concept**'s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4). No additional attributes are specified in this metamodel region.

7.3.2.3 Data_Element_Concept class

7.3.2.3.1 Direct superclass

Data_Element_Concept is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.3.2.3.2 Description of Data_Element_Concept

Data_Element_Concept is a class each instance of which models a data element concept. A data element concept is a specification of a concept independent of any particular representation. A data element concept can be represented by a data element.

The combination of a property and an object class provides significance beyond either that of the property or the object class. An instance of **Data_Element_Concept** thus has a **Definition** (ISO/IEC 11179-3:2023 , 8.2.3) independent of the **Definition** of the corresponding instances of **Object_Class** or **Property**.

Data_Element_Concept is further described in 7.2.2.5.

7.3.2.3.3 Associations of Data_Element_Concept

As a subclass of **Concept**, **Data_Element_Concept** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Data_Element_Concept** has the following additional associations:

— **data_element_concept_object_class** association (7.3.3.1);

— **data_element_concept_property** association (7.3.3.2);

— **data_element_concept_domain** association (7.3.3.3).

Other associations of **Data_Element_Concept** are specified in 7.2.2.5.3.

7.3.2.3.4 Attributes of Data_Element_Concept

As a subclass of **Concept**, **Data_Element_Concept** inherits **Concept**'s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4). No additional attributes are specified in this metamodel region.

7.3.2.4 Conceptual_Domain class

Conceptual_Domain is described in 7.4.2.1 as part of the Conceptual and Value Domain region (7.4).

7.3.2.5 Data_Element_Concept_Constraint_Set class

7.3.2.5.1 Direct superclass

Data_Element_Concept_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.3.2.5.2 Description of Data_Element_Concept_Constraint_Set

Data_Element_Concept_Constraint_Set extends **Constraint_Set** (specified in ISO/IEC 11179-3:2023, 9.4.5) with constraints applicable to the **Data_Element_Concept** class and its associations. **Data_Element_Concept_Constraint_Set** is illustrated in Figure 2.

7.3.2.5.3 Associations of Data_Element_Concept_Constraint_Set

As a subclass of **Constraint_Set**, **Data_Element_Concept_Constraint_Set** inherits **Constraint_Set**'s associations.

7.3.2.5.4 Attibutes of Data_Element_Concept_Constraint_Set

See Table 1.

Table 1 — Attributes of Data_Element_Concept_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
DEC_shall_have_Object_Class_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether this data element concept is required to have a linked object class. In the metamodel this link is represented by an instance of the **data_element_concept_object_class** association (7.3.3.1). Best practice is to set this indicator to TRUE.
DEC_shall_have_Property_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether this data element concept is required to have a linked property. In the metamodel this link is represented by an instance of the **data_element_concept_property** association (7.3.3.2). Best practice is to set this indicator to TRUE.
DEC_shall_have_at_least_one_CD_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether this data element concept is required to have at least one linked conceptual domain. In the metamodel this link is represented by an instance of the **data_element_concept_domain** association (7.3.3.3). Best practice is to set this indicator to TRUE.
DEC_shall_have_at_most_one_CD_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether this data element concept is required to have at most one linked conceptual domain. In the metamodel this link is represented by an instance of the **data_element_concept_domain** association (7.3.3.3). Best practice is to set this indicator to TRUE.

7.3.3 Associations in the Data Element Concept metamodel region

7.3.3.1 data_element_concept_object_class association

The **data_element_concept_object_class** association binds zero, one or more instances of the **Data_Element_Concept** class (7.3.2.3) to zero or one instances of the **Object_Class** class (7.3.2.1) that represents a particular set of ideas, abstractions or things in the real world whose properties and behaviour follow a set of rules as represented by the **Data_Element_Concept**.

EXAMPLE Associating the **Object_Class** "Person" with the **Data_Element_Concept** "Person height".

It is best practice for a data element concept to be associated with exactly one object class. A registration authority can enforce this using the constraint **Data_Element_Concept_Constraint_Set. DEC_shall_have_Object_Class** (7.3.2.5.4).

7.3.3.2 data_element_concept_property association

The **data_element_concept_property** association binds zero, one or more instances of the **Data_Element_Concept** class (7.3.2.3) to zero or one instance of the **Property** class (7.3.2.2).

EXAMPLE Associating the **Property** "height" with the **Data_Element_Concept** "Person height".

It is best practice for a data element concept to be associated with exactly one property. A registration authority can enforce this using the constraint **Data_Element_Concept_Constraint_Set. DEC_shall_have_Property** (7.3.2.5.4).

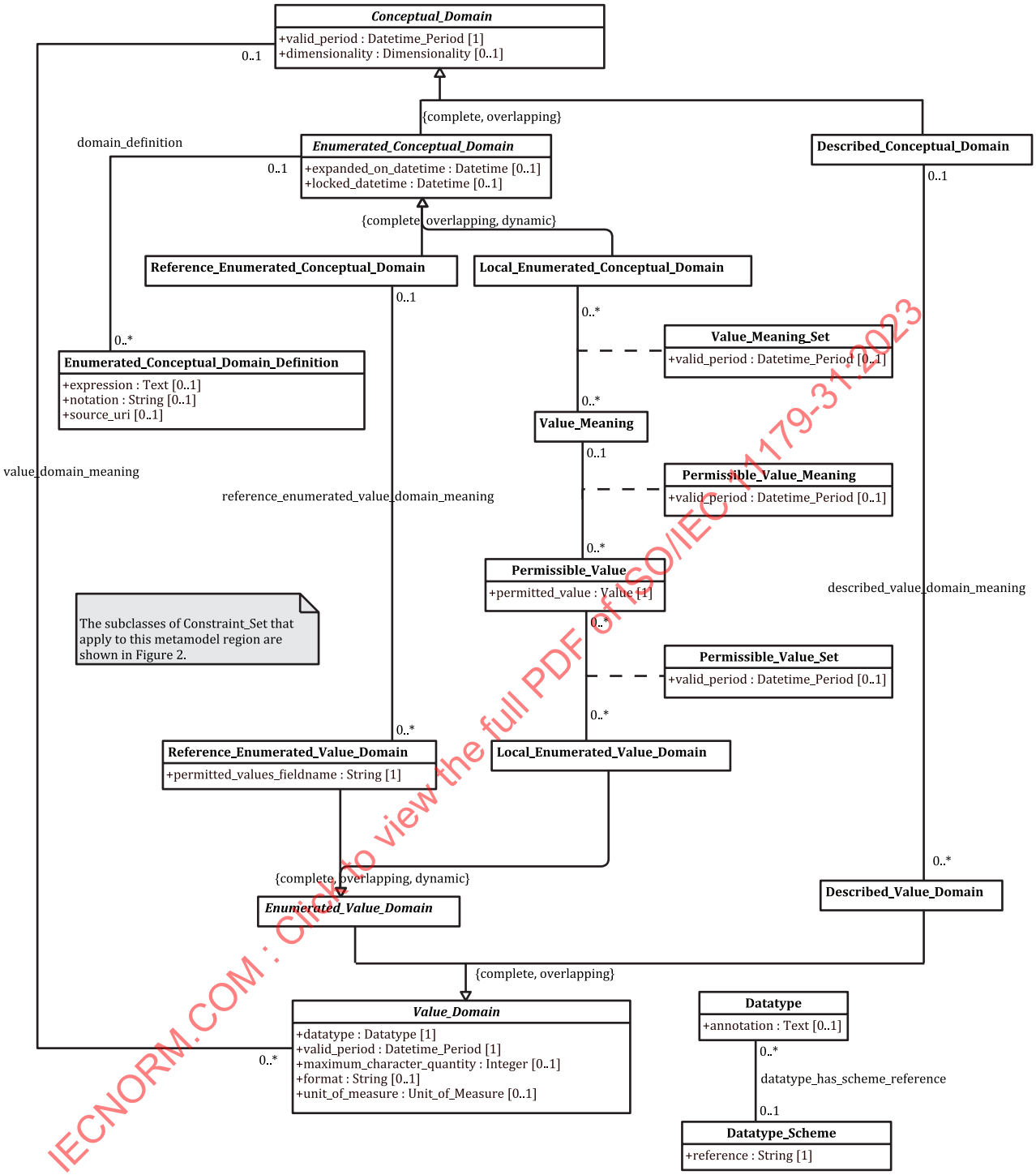
7.3.3.3 data_element_concept_domain association

The **data_element_concept_domain** association is specified in (7.2.3.4). It is best practice for a data element concept to be associated with exactly one conceptual domain. A registration authority can enforce this using the constraints **Data_Element_Concept_Constraint_Set.DEC_shall_have_at_least_one_CD** and **Data_Element_Concept_Constraint_Set.DEC_shall_have_at_most_one_CD** (7.3.2.5.4).

7.4 Conceptual Domain and Value Domain metamodel region

7.4.1 Overview of the Conceptual Domain and Value Domain metamodel region

This metamodel region addresses the administration within the registry of conceptual domains, represented by the **Conceptual_Domain** class (7.4.2.1), and value domains, represented by the **Value_Domain** class (7.4.2.8). These domains can be viewed as logical code sets and physical code sets respectively. The region is illustrated in Figure 5.



NOTE In this figure, the use of *italics* in the names of **Conceptual_Domain**, **Enumerated_Conceptual_Domain**, **Value_Domain** and **Enumerated_Value_Domain** indicates that they are abstract classes, meaning that only their concrete subclasses can be instantiated.

Figure 5 — Conceptual Domain and Value Domain metamodel region

7.4.2 Classes in the Conceptual Domain and Value Domain metamodel region

7.4.2.1 Conceptual_Domain class

7.4.2.1.1 Direct superclass

Conceptual_Domain is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.1.2 Description of Conceptual_Domain

Conceptual_Domain is a class each instance of which models a conceptual domain, a concept whose meaning is expressed as either an enumerated set or a description of subordinate concepts or both, which are value meanings.

Conceptual_Domain is an abstract class, which has two subclasses:

— **Described Conceptual_Domain** (7.4.2.2);

— **Enumerated_Conceptual_Domain** (7.4.2.3).

Every **Conceptual_Domain** instance shall be one of the above subclasses or a combination of both.

7.4.2.1.3 Associations of the Conceptual_Domain

As a subclass of **Concept**, **Conceptual_Domain** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Conceptual_Domain** has the following additional association in this metamodel region:

— **value_domain_meaning** (7.4.4.1).

The High-level Data Specification metamodel region specifies the following additional association:

— **data_element_concept_domain** (7.2.3.4).

7.4.2.1.4 Attributes of Conceptual_Domain

As a subclass of **Concept**, **Conceptual_Domain** inherits **Concept**'s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4). **Conceptual_Domain** has additional attributes as specified in Table 2.

Table 2 — Attributes of Conceptual_Domain class

Attribute name	Multiplicity	Datatype	Description
valid_period	1	**Datetime_Period** (ISO/IEC 11179-3:2023, 6.3.10)	Definition: datetime period for which this conceptual domain is, was or will become, a valid conceptual domain. A registration authority may determine whether these datetimes are the datetimes the conceptual domain becomes valid in a registry, or the datetimes the conceptual domain becomes part of the source domain or some other datetimes. The absence of the **Datetime_Period.end_datetime** indicates that the conceptual domain is still valid. The prescence of the **Datetime_Period.end_datetime** indicates when the conceptual domain became or will become invalid.

Table 2 *(continued)*

Attribute name	Multiplicity	Datatype	Description
dimensionality	0..1	**Dimensionality** (7.4.2.3)	Definition: expression of measurement without units This **dimensionality** attribute specifies the dimensionality as elaborated in the discussion of the **Dimensionality** class in 7.4.2.3. ISO 80000-1[13] specifies physical dimensions (e.g. length, mass, velocity). This document also permits non-physical dimensions (e.g. value dimensions such as: currency, quality indicator).

7.4.2.1.5 Constraints on Conceptual_Domain

7.4.2.1.5.1 Constraint 1

valid_period shall have a **Datetime_Period.start_datetime**.

7.4.2.1.5.2 Constraint 2

When a **dimensionality** is specified, then any unit of measure specified for any value domain that is based on this conceptual domain, shall be consistent with this dimensionality.

7.4.2.2 Described_Conceptual_Domain class

7.4.2.2.1 Direct superclass

Conceptual_Domain (7.4.2.1)

7.4.2.2.2 Description of Described_Conceptual_Domain

Described_Conceptual_Domain is a class each instance of which models a described conceptual domain, a conceptual domain that is specified by a description or specification, such as a rule, a procedure or a range (i.e. interval), because it cannot be expressed as a finite set of value meanings.

7.4.2.2.3 Associations of Described_Conceptual_Domain

As a subclass of **Conceptual_Domain**, **Described_Conceptual_Domain** inherits **Conceptual_Domain**'s associations (7.4.2.1.3). **Described_Conceptual_Domain** has the following additional association:

— **described_value_domain_meaning** (7.4.4.2).

7.4.2.2.4 Attributes of Described_Conceptual_Domain

As a subclass of **Conceptual_Domain**, **Described_Conceptual_Domain** inherits **Conceptual_Domain**'s attributes (7.4.2.1.4). **Described_Conceptual_Domain** has no additional attributes.

7.4.2.2.5 Constraint on Described_Conceptual_Domain

Each instance of **Described_Conceptual_Domain** shall be defined using the **item_definition** association of the **Item** superclass. Thus, **Definition.text** (ISO/IEC 11179-3:2023, 8.4.2.3) shall be specified. This definition constitutes the description of the conceptual domain.

This constraint can be enforced by setting **Constraint_Set.item_shall_be_defined_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

7.4.2.3 Enumerated_Conceptual_Domain class

7.4.2.3.1 Direct superclass

Conceptual_Domain (7.4.2.1).

7.4.2.3.2 Description of Enumerated_Conceptual_Domain

Enumerated_Conceptual_Domain is a class each instance of which models an enumerated conceptual domain, a conceptual domain that is specified by a list of all its value meanings.

EXAMPLE The notion of countries that is specified in ISO 3166[1], Codes for the representation of names of countries.

Enumerated_Conceptual_Domain is an abstract class, which has two subclasses:

— **Local_Enumerated_Conceptual_Domain** (7.4.2.4);

— **Reference_Enumerated_Conceptual_Domain** (7.4.2.6).

Every instance of **Enumerated_Conceptual_Domain** shall be one of the above subclasses or a combination of both. A reference enumerated conceptual domain can be extended by defining additional value meanings in a local enumerated conceptual domain. Also, a reference enumerated conceptual domain can be converted to a local enumerated conceptual domain by importing its value meanings from the external source.

7.4.2.3.3 Associations of Enumerated_Conceptual_Domain

As a subclass of **Conceptual_Domain**, **Enumerated_Conceptual_Domain** inherits **Conceptual_Domain**'s associations (7.4.2.1.3). **Enumerated_Conceptual_Domain** has the following additional association:

— **domain_definition** (7.4.4.3).

7.4.2.3.4 Attributes of Enumerated_Conceptual_Domain

As a subclass of **Conceptual_Domain**, **Enumerated_Conceptual_Domain** inherits **Conceptual_Domain**'s attributes (7.4.2.1.4). **Enumerated_Conceptual_Domain** has additional attributes as specified in Table 3.

Table 3 — Attributes of Enumerated_Conceptual_Domain class

Attribute name	Multiplicity	Datatype	Description
expanded_on_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: datetime at which the set of value meanings associated with the reference enumerated conceptual domain was expanded and imported into the registry as a local enumerated conceptual domain.
locked_datetime	0..1	**Datetime** (ISO/IEC 11179-3:2023, 6.2.3)	Definition: datetime on which the set of value meanings associated with the reference enumerated conceptual domain was or will be locked, meaning subsequent changes should not be used It is possible that the value meanings have been imported as a local enumerated conceptual domain.

7.4.2.4 Local_Enumerated_Conceptual_Domain class

7.4.2.4.1 Direct superclass

Enumerated_Conceptual_Domain (7.4.2.3).

7.4.2.4.2 Description of Local_Enumerated_Conceptual_Domain

Local_Enumerated_Conceptual_Domain is a class each instance of which models a local enumerated conceptual domain, an enumerated conceptual domain whose value meanings are enumerated within the same registry as the enumerated conceptual domain.

7.4.2.4.3 Associations of Local_Enumerated_Conceptual_Domain

As a subclass of **Enumerated_Conceptual_Domain**, a **Local_Enumerated_Conceptual_Domain** inherits **Enumerated_Conceptual_Domain**'s associations (7.4.2.3.3). This metamodel region adds the following association class:

— **Value_Meaning_Set** association class (7.4.3.1).

The **Conceptual_Domain_Subset** metamodel region adds the following association:

— **local_enumerated_conceptual_domain_has_subset** (7.8.4.5).

7.4.2.4.4 Attributes of Local_Enumerated_Conceptual_Domain

As a subclass of **Enumerated_Conceptual_Domain**, a **Local_Enumerated_Conceptual_Domain** inherits **Enumerated_Conceptual_Domain**'s attributes (7.4.2.3.4). **Local_Enumerated_Conceptual_Domain** has no additional attributes.

7.4.2.5 Value_Meaning class

7.4.2.5.1 Direct superclass

Value_Meaning is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.5.2 Description of Value_Meaning

Value_Meaning is a class each instance of which models a value meaning, which provides semantic content of a possible value.

Each member of a local enumerated conceptual domain has a value meaning that provides its distinction from other members. In the example of ISO 3166[1], the notion of each country as specified would be the value meanings. The representation of value meanings in a registry shall be independent of (and shall not constrain) their representation in any corresponding value domain. A particular value meaning may have more than one means of representation by permissible values— each from a distinct local enumerated value domain.

The description of an instance of the **Value_Meaning** class is specified by associating a **Definition** (ISO/IEC 11179-3:2023, 8.2.3) with the grand-parent **Item** (ISO/IEC 11179-3:2023, 6.4.2.1).

7.4.2.5.3 Associations of Value_Meaning

As a subclass of **Concept**, **Value_Meaning** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). This metamodel region adds the following association classes:

— **Value_Meaning_Set** association class (7.4.3.1);

— **Permissible_Value_Meaning** association class (7.4.3.2).

The **Conceptual_Domain_Subset** metamodel region adds the following association class:

— **Subset_Value_Meanings** association class (7.8.3.2).

7.4.2.5.4 Attributes of Value_Meaning

As a subclass of **Concept**, **Value_Meaning** inherits **Concept**'s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4). **Value_Meaning** has no additional attributes.

7.4.2.6 Reference_Enumerated_Conceptual_Domain

7.4.2.6.1 Direct superclass

Enumerated_Conceptual_Domain (7.4.2.3).

7.4.2.6.2 Description of Reference_Enumerated_Conceptual_Domain

Reference_Enumerated_Conceptual_Domain is a class each instance of which models a reference enumerated conceptual domain, an enumerated conceptual domain that is specified by a formal definition.

7.4.2.6.3 Associations of Reference_Enumerated_Conceptual_Domain

As a subclass of **Enumerated_Conceptual_Domain** (7.4.2.3), **Reference_Enumerated_Conceptual_Domain** inherits **Enumerated_Conceptual_Domain**'s associations (7.4.2.3.3). **Reference_Enumerated_Conceptual_Domain** has the following additional association:

— **reference_enumerated_value_domain_meaning** association (7.4.4.4).

7.4.2.6.4 Attributes of Reference_Enumerated_Conceptual_Domain

As a subclass of **Enumerated_Conceptual_Domain** (7.4.2.3), **Reference_Enumerated_Conceptual_Domain** inherits **Enumerated_Conceptual_Domain**'s attributes (7.4.2.3.4). **Reference_Enumerated_Conceptual_Domain** has no additional attributes.

7.4.2.7 Enumerated_Conceptual_Domain_Definition

7.4.2.7.1 Direct superclass

Enumerated_Conceptual_Domain_Definition is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.7.2 Description of Enumerated_Conceptual_Domain_Definition

Enumerated_Conceptual_Domain_Definition is a clas, each instance of which models an enumerated conceptual domain definition. This definition applies to externally defined enumerated conceptual domains, which may be referenced via a reference enumerated conceptual domain or imported into a local enumerated conceptual domain.

7.4.2.7.3 Associations of Enumerated_Conceptual_Domain_Definition

As a subclass of **Item**, **Enumerated_Conceptual_Domain_Definition** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). **Enumerated_Conceptual_Domain_Definition** has the following additional association:

— **domain_definition** association (7.4.4.3).

7.4.2.7.4 Attributes of Enumerated_Conceptual_Domain_Definition

See Table 4.

Table 4 — Attributes of Enumerated_Conceptual_Domain_Definition class

Attribute name	Multiplicity	Datatype	Description
expression	0..1	**Text** (ISO/ IEC 11179-3:2023, 6.2.12)	Definition: formal definition of the enumerated conceptual domain expressed in some formal *notation*
notation	0..1	**String** (ISO/ IEC 11179-3:2023, 6.2.11)	Conditional. Mandatory if **expression** is present. Definition: name of the notation used in the **expression** Examples: OWL-DL, SQWRL
source_uri	0..1	**String** (ISO/ IEC 11179-3:2023, 6.2.11)	Conditional. Mandatory if **expression** is absent. Definition: URI for the source of the enumerated conceptual domain definition

7.4.2.8 Value_Domain class

7.4.2.8.1 Direct superclass

Value_Domain is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.8.2 Description of Value_Domain

Value_Domain is a class each instance of which models a value domain, a set of permissible values. A value domain provides representation but has no implication as to the data element concept with which the values are associated, nor what the values mean.

A **Value_Domain** is an abstract class which is used to denote a collection of permissible values. A **Value_Domain** has two possible subclasses:

— **Enumerated_Value_Domain** (7.4.2.10);

— **Described_Value_Domain** (7.4.2.9).

Every **Value_Domain** instance shall be one of the above subclasses or a combination of both.

7.4.2.8.3 Associations of Value_Domain

As a subclass of **Item**, **Value_Domain** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). **Value_Domain** has the following additional association in this metamodel region:

— **value_domain_meaning** (7.4.4.1);

and the following additional association in the Data Element metamodel region (7.6):

— **data_element_domain** association (7.6.3.1).

NOTE ISO/IEC 11179-3:2013 had a recursive value_domain_subset association on **Value_Domain**. In this edition, the **Item_Mapping** of ISO/IEC 11179-3:2023, 11.3.1 can be used. This edition also has the **Local_Enumerated_Value_Domain_Subset** class (7.7.2.7) which provides a subsetting of values in a **Local_ Enumerated_Value_Domain** (7.4.2.10) for use in a particular context.

7.4.2.8.4 Attributes of Value_Domain

See Table 5.

Table 5 — Attributes of Value_Domain class

Attribute name	Multiplicity	Datatype	Description
datatype	1	**Datatype** (7.4.2.14)	Definition: datatype used in this value domain It applies to all values in the value domain.
valid_period	1	**Datetime_Period** (ISO/IEC 11179-3:2023, 6.3.10)	Definition: datetime period for which this value domain is, was or will become a valid value domain A registration authority may determine whether these datetimes are the datetimes the value domain becomes valid in a registry, or the datetimes the value domain becomes part of the source domain or some other datetimes. The absence of the **Datetime_Period.end_datetime** indicates that the value domain is still valid. The prescence of the **Datetime_Period.end_datetime** indicates when the value domain became or will become invalid. By imputation, the **Datetime_Period.start_datetime** is also considered to be the datetime at which the associated permissible values (if applicable) were bound to the associated value meanings, unless explicitly overridden for a particular permissible value. Also by imputation, the **Datetime_Period.end_datetime** is considered to be the datetime on which the associated permissible values (if applicable) ceased or will cease to be bound to their associated value meanings, unless explicitly overridden for a particular permissible value.
format	0..1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: template for the structure of the presentation of the value(s) Example: YYYY-MM-DD for a date
maximum_character_quantity	0..1	**Integer** (ISO/IEC 11179-3:2023, 6.2.5)	Definition: maximum number of characters available to represent the data element value Applicable only to character datatypes.
unit_of_measure	0..1	**Unit_of_Measure** (7.5.2.1)	Definition: unit of measure used in a value domain Applies to all values in the value domain. Constraints on the unit of measure dimensionality are specified in 7.4.5.3.

7.4.2.8.5 Constraint on Value_Domain

valid_period shall have a **Datetime_Period.start_datetime**.

7.4.2.9 Described_Value_Domain class

7.4.2.9.1 Direct superclass

Value_Domain (7.4.2.8).

7.4.2.9.2 Description of Described_Value_Domain

Described_Value_Domain is a class each instance of which models a described value domain, a value domain that is specified by a description or specification, such as a rule, a procedure or a range (i.e. interval), rather than as an explicit set of permissible values.

7.4.2.9.3 Associations of Described_Value_Domain

As a subclass of **Value_Domain**, **Described_Value_Domain** inherits **Value_Domain**'s associations (7.4.2.8.3). **Described_Value_Domain** has the following additional association:

— **described_value_domain_meaning** (7.4.4.2).

7.4.2.9.4 Attributes of Described_Value_Domain

As a subclass of **Value_Domain**, a **Described_Value_Domain** inherits **Value_Domain**'s attributes (7.4.2.8.4). **Described_Value_Domain** has no additional attributes.

7.4.2.9.5 Constraints on Described_Value_Domain

Each instance of **Described_Value_Domain** shall be defined using the **item_definition** association of the grand-parent **Item** superclass. Thus, **Definition.text** (ISO/IEC 11179-3:2023, 8.4.2.3) shall be specified. This definition constitutes the description of the value domain.

This constraint can be enforced by setting **Constraint_Set.item_shall_be_defined_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

7.4.2.10 Enumerated_Value_Domain class

7.4.2.10.1 Direct superclass

Value_Domain (7.4.2.8).

7.4.2.10.2 Description of Enumerated_Value_Domain

Enumerated_Value_Domain is a class each instance of which models an enumerated value domain, a value domain that is specified by a list of all its permissible values.

An **Enumerated_Value_Domain** is an abstract class which has two possible concrete subclasses:

— **Local_Enumerated_Value_Domain** (7.4.2.11);

— **Reference_Enumerated_Value_Domain** (7.4.2.13).

Every **Enumerated_Value_Domain** instance shall be one of the above subclasses or a combination of both. A reference enumerated value domain can be extended by defining additional permissible values in a local enumerated value domain. Also, a reference enumerated value domain can be converted to a local enumerated value domain by importing its permissible values from the external source.

7.4.2.10.3 Associations of Enumerated_Value_Domain

As a subclass of **Value_Domain**, **Enumerated_Value_Domain** inherits **Value_Domain**'s associations (7.4.2.8.3). The concrete subclasses of **Enumerated_Value_Domain** add other associations.

7.4.2.10.4 Attributes of Enumerated_Value_Domain

As a subclass of **Value_Domain**, an **Enumerated_Value_Domain** inherits **Value_Domain**'s attributes (7.4.2.8.4). **Enumerated_Value_Domain** has no additional attributes.

7.4.2.11 Local_Enumerated_Value_Domain class

7.4.2.11.1 Direct superclass

Enumerated_Value_Domain (7.4.2.10).

7.4.2.11.2 Description of Local_Enumerated_Value_Domain

Local_Enumerated_Value_Domain is a class each instance of which models a local enumerated value domain, an enumerated value domain whose permissible values are enumerated within the same registry as the value domain.

7.4.2.11.3 Associations of Local_Enumerated_Value_Domain

As a subclass of **Enumerated_Value_Domain**, **Local_Enumerated_Value_Domain** inherits **Enumerated_Value_Domain**'s associations (7.4.2.10.3). **Local_Enumerated_Value_Domain** has the following additional association class:

— **Permissible_Value_Set** association class (7.4.3.3).

7.4.2.11.4 Attributes of Local_Enumerated_Value_Domain

As a subclass of **Enumerated_Value_Domain**, **Local_Enumerated_Value_Domain** inherits **Enumerated_Value_Domain**'s attributes (7.4.2.10.4). **Local_Enumerated_Value_Domain** has no additional attributes.

7.4.2.12 Permissible_Value class

7.4.2.12.1 Direct superclass

Permissible_Value is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.12.2 Description of Permissible_Value

Permissible_Value is a class each instance of which models a permissible value, the designation of a value meaning. A permissible value is an expression of a value meaning within zero, one or more local enumerated value domains.

7.4.2.12.3 Associations of Permissible_Value

As a subclass of **Item**, **Permissible_Value** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). The following association classes are added in this metamodel region:

— **Permissible_Value_Meaning** association class (7.4.3.2);

— **Permissible_Value_Set** association class (7.4.3.3).

The **Value_Domain_Subset** metamodel region adds another association class:

— **Subset_Values** association class (7.7.3.1).

7.4.2.12.4 Attributes of *Permissible_Value*

See Table 6.

Table 6 — Attributes of Permissible_Value class

Attribute name	Multiplicity	Datatype	Description
permitted_value	1	**Value** (ISO/ IEC 11179-3:2023, 6.2.13)	Definition: the actual value of the permissible value

NOTE **permitted_value** is required even though the parent **Item** can have one or more **Designation.signs** so that a single value is specified.

7.4.2.12.5 Example of Permissible_Values

The following example from ISO 3166-3 illustrates how permissible values can be valid for specified period of time only.

ISO 3166-3 NEWSLETTER No. I-1 Date: 2002-11-15 announced a change of country name and associated codes from:

Former country name: East Timor

Former alpha-2 code: TP

Former alpha-3 code: TMP

Former numeric code: 626

Validity start date: 1974-01-01

Validity end date: 2002-11-14

To:

New country name: Timor Leste

New alpha-2 code: TL

New alpha-3 code: TLS

New numeric code: 626

Validity start date: 2002-11-15

Validity end date: (null)

7.4.2.13 Reference_Enumerated_Value_Domain class

7.4.2.13.1 Direct superclass

Enumerated_Value_Domain (7.4.2.10).

7.4.2.13.2 Description of Reference_Enumerated_Value_Domain

Reference_Enumerated_Value_Domain is a class each instance of which models a reference enumerated value domain, a value domain that is specified by a reference to an external source for all its permissible values.

A **Reference_Enumerated_Value_Domain** is like a **Local_Enumerated_Value_Domain**, except the enumerated values are to be referenced from the external source that is specified by the associated

Reference_Enumerated_Conceptual_Domain (7.4.2.6), in conjunction with the **fieldname_for_permitted_value** attribute. A **Reference_Enumerated_Value_Domain** may be converted to a **Local_Enumerated_Value_Domain** by importing the permissible values as a set.

7.4.2.13.3 Associations of Reference_Enumerated_Value_Domain

As a subclass of **Enumerated_Value_Domain**, **Reference_Enumerated_Value_Domain** inherits **Enumerated_Value_Domain**'s associations. **Reference_Enumerated_Value_Domain** has the following additional association:

— **reference_enumerated_value_domain_meaning** (7.4.4.4).

7.4.2.13.4 Attributes of Reference_Enumerated_Value_Domain

As a subclass of **Enumerated_Value_Domain**, a **Reference_Enumerated_Value_Domain** inherits **Enumerated_Value_Domain**'s attributes (7.4.2.10.4). **Reference_Enumerated_Value_Domain** has an additional attribute as specified in Table 7.

Table 7 — Attributes of Reference_Enumerated_Value_Domain class

Attribute name	Multiplicity	Datatype	Description
permitted_values_fieldname	1	**String** (ISO/IEC 11179-3:2023, 6.2.11)	Definition: the name of the field at the external source where the permitted values can be accessed

7.4.2.14 Datatype class

7.4.2.14.1 Direct superclass

Datatype is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.14.2 Description of Datatype

Datatype is a class each instance of which models a datatype, a set of distinct values characterized by properties of those values and by operations on those values. For example, the category used for the collection of letters, digits, symbols or some combination thereof to depict values of a **Data_Element** (7.6.2.1) determined by the operations that may be performed on the **Data_Element**.

This document is intended to accommodate datatypes from datatype schemes specified in external standards. It is also intended to accommodate other non-standard datatype schemes. Possible standardized datatype schemes include:

— C programming language datatypes (see Reference [3]);

— General Purpose Datatypes (see Reference [9]);

— Health Informatics datatypes (see Reference [12]);

— SQL datatypes (see Reference [2]);

— XML Schema datatypes (see Reference [14]);

etc.

7.4.2.14.3 Associations of Datatype

As a subclass of **Item**, **Datatype** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2).

Datatype has the following additional association:

— **datatype_has_scheme_reference** association (7.4.3.3).

7.4.2.14.4 Attributes of Datatype

See Table 8.

Table 8 — Attributes of Datatype class

Attribute name	Multiplicity	Datatype	Description
annotation	0..1	**Text** (ISO/IEC 11179-3:2023, 6.2.12)	Definition: specifying information to further define the **Datatype**

7.4.2.14.5 Constraints on Datatype

A datatype shall be designated and defined using the **item_designation** and **item_definition** associations of the **Item** superclass.

— **Designation.sign** (ISO/IEC 11179-3:2023, 8.4.1.3) shall be specified. This constraint can be enforced by setting **Constraint_Set.item_shall_be_designated_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

— **Definition.text** (ISO/IEC 11179-3:2023, 8.4.2.3) shall be specified. This constraint can be enforced by setting **Constraint_Set.item_shall_be_defined_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

7.4.2.14.6 Examples of Datatypes

Any applicable datatype may be specified, e.g.

EXAMPLE 1

Table 9 — Example 1 of Datatype class

Classname.Attributename	Value
Designation.sign	integer
Definition.text	mathematical datatype comprising the exact integral values.
Datatype_Scheme.scheme_reference	ISO/IEC 11404:2007[9]
Datatype.annotation	integer values can be positive or negative.

EXAMPLE 2

Table 10 — Example 2 of Datatype class

Classname.Attributename	Value
Designation.sign	BL
Definition.text	the values of two-valued logic
Datatype_Scheme.scheme_reference	ISO 21090[12]
Datatype.annotation	A BL value can be either true or false or can have a nullFlavor.

7.4.2.15 Datatype_Scheme class

7.4.2.15.1 Direct superclass

Datatype_Scheme is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.15.2 Description of Datatype_Scheme

Datatype_Scheme is a class each instance of which models a datatype scheme, a source of the specification of one or more datatypes.

This document is intended to accommodate datatypes from datatype schemes specified in external standards. It is also intended to accommodate other non-standard datatype schemes. Possible standardized datatype schemes include:

— C programming language datatypes (see Reference [3]);

— General Purpose Datatypes (see Reference [9]);

— Health Informatics datatypes (see Reference [12]);

— SQL datatypes (see Reference [2]);

— XML Schema datatypes (see Reference [14]);

etc.

7.4.2.15.3 Associations of Datatype_Scheme

As a subclass of **Item**, **Datatype_Scheme** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2).

Datatype_Scheme has the following additional association:

— **datatype_has_scheme_reference** association (7.4.3.3).

7.4.2.15.4 Attributes of Datatype_Scheme

See Table 11.

Table 11 — Attributes of Datatype_Scheme class

Attribute name	Multiplicity	Datatype	Description
scheme_reference	0..1	**Reference_Document** (ISO/IEC 11179-3:2023, 6.23.7)	Definition: reference to the **Datatype_Scheme** specification In this document, the manner of reference is specified by the registration authority.

7.4.2.16 Reference_Enumerated_CD_Constraint_Set

7.4.2.16.1 Direct superclass

Reference_Enumerated_CD_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.16.2 Description of Reference_Enumerated_CD_Constraint_Set

Reference_Enumerated_CD_Constraint_Set extends **Constraint_Set** with constraints applicable to the **Reference_Enumerated_Conceptual_Domain** (7.4.2.6) class and its associations.

Reference_Enumerated_CD_Constraint_Set is illustrated in Figure 2.

7.4.2.16.3 Associations of Reference_Enumerated_CD_Constraint_Set

As a subclass of **Constraint_Set, Reference_Enumerated_CD_Constraint_Set** inherits **Constraint_Set**'s associations.

7.4.2.16.4 Attributes of Reference_Enumerated_CD_Constraint_Set

See Table 12.

Table 12 — Attributes of Reference_Enumerated_CD_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
Reference_Enumerated_CD_ shall_have_at_least_one_ Enumerated_CD_ Definition_indicator	1	**Boolean** (ISO/ IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether this reference enumerated conceptual domain is required to have at least one linked enumerated conceptual domain definition. In the metamodel this link is represented by an instance of the **domain_definition** association (7.4.4.3). Best practice is to set this indicator to TRUE.

7.4.2.17 Local_Enumerated_CD_Constraint_Set

7.4.2.17.1 Direct superclass

Local_Enumerated_CD_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.17.2 Description of Local_Enumerated_CD_Constraint_Set

Local_Enumerated_CD_Constraint_Set extends **Constraint_Set** with constraints applicable to the **Local_Enumerated_Conceptual_Domain** (7.4.2.4) class and its associations.

Local_Enumerated_CD_Constraint_Set is illustrated in Figure 2.

7.4.2.17.3 Associations of Local_Enumerated_CD_Constraint_Set

As a subclass of **Constraint_Set**, **Local_Enumerated_CD_Constraint_Set** inherits **Constraint_Set**'s associations.

7.4.2.17.4 Attributes of Local_Enumerated_CD_Constraint_Set

See Table 13.

Table 13 — Attributes of Local_Enumerated_CD_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
Local_Enumerated_CD_shall_have_at_least_one_Value_Meaning_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether a local enumerated conceptual domain is required to have at least one value meaning. In the metamodel this link is represented by an instance of the **Value_Meaning_Set** association class (7.4.3.1). Best practice is to set this indicator to TRUE.

7.4.2.18 Value_Domain_Constraint_Set

7.4.2.18.1 Direct superclass

Value_Domain_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.18.2 Description of Value_Domain_Constraint_Set

Value_Domain_Constraint_Set extends **Constraint_Set** with constraints applicable to the **Value_Domain** class (7.4.2.8) and its associations. **Value_Domain_Constraint_Set** is illustrated in Figure 2.

7.4.2.18.3 Associations of Value_Domain_Constraint_Set

As a subclass of **Constraint_Set**, **Value_Domain_Constraint_Set** inherits **Constraint_Set**'s associations.

7.4.2.18.4 Attributes of Value_Domain_Constraint_Set

See Table 14.

Table 14 — Attributes of Value_Domain_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
VD_shall_have_exactly_one_CD_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether a value domain is required to have exactly one conceptual domain. In the metamodel this link is represented by an instance of the **value_domain_meaning** association (7.4.4.1) and indirectly via the **Permissible_Value_Set** (7.4.3.3), **Permissible_Value_Meaning** (7.4.3.2) and **Value_Meaning_Set** (7.4.3.1) association classes. Best practice is to set this indicator to TRUE.

7.4.2.19 Described_Value_Domain_Constraint_Set

7.4.2.19.1 Direct superclass

Described_Value_Domain_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.19.2 Description of Described_Value_Domain_Constraint_Set

Described_Value_Domain_Constraint_Set extends **Constraint_Set** with constraints applicable to the **Described_Value_Domain** class (7.4.2.9) and its associations. **Described_Value_Domain_Constraint_Set** is illustrated in Figure 2.

7.4.2.19.3 Associations of Described_Value_Domain_Constraint_Set

As a subclass of **Constraint_Set**, **Described_Value_Domain_Constraint_Set** inherits **Constraint_Set**'s associations.

7.4.2.19.4 Attributes of Described_Value_Domain_Constraint_Set

See Table 15.

Table 15 — Attributes of Described_Value_Domain_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
Described_VD_shall_have_exactly_one_Described_CD_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether a described value domain is required to have exactly one described conceptual domain. In the metamodel this link is represented by an instance of the **described_value_domain_meaning** association (7.4.4.2). Best practice is to set this indicator to TRUE.

7.4.2.20 Reference_Enumerated_VD_Constraint_Set

7.4.2.20.1 Direct superclass

Reference_Enumerated_VD_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.20.2 Description of Reference_Enumerated_VD_Constraint_Set

Reference_Enumerated_VD_Constraint_Set extends **Constraint_Set** with constraints applicable to the **Reference_Enumerated_Value_Domain** class (7.4.2.13) and its associations. **Reference_Enumerated_VD_Constraint_Set** is illustrated in Figure 2.

7.4.2.20.3 Associations of Reference_Enumerated_VD_Constraint_Set

As a subclass of **Constraint_Set**, **Reference_Enumerated_VD_Constraint_Set** inherits **Constraint_Set**'s associations.

7.4.2.20.4 Attributes of Reference_Enumerated_VD_Constraint_Set

See Table 16.

Table 16 — Attributes of Reference_Enumerated_VD_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
Reference_Enumerated_VD_ shall_have_exactly_one_ Reference_Enumerated_ CD_indicator	1	**Boolean** (ISO/ IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether a reference enumerated value domain is required to have exactly one reference enumerated conceptual domain. In the metamodel this link is represented by an instance of the **reference_enumerated_value_ domain_meaning** association (7.4.4.4). Best practice is to set this indicator to TRUE.

7.4.2.21 Local_Enumerated_VD_Constraint_Set

7.4.2.21.1 Direct superclass

Local_Enumerated_VD_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.21.2 Description of Local_Enumerated_VD_Constraint_Set

Local_Enumerated_VD_Constraint_Set extends **Constraint_Set** with constraints applicable to the **Local_Enumerated_Value_Domain** class (7.4.2.11) and its associations. **Local_Enumerated_VD_ Constraint_Set** is illustrated in Figure 2.

7.4.2.21.3 Associations of Local_Enumerated_VD_Constraint_Set

As a subclass of **Constraint_Set**, **Local_Enumerated_VD_Constraint_Set** inherits **Constraint_Set**'s associations.

7.4.2.21.4 Attributes of Local_Enumerated_VD_Constraint_Set

See Table 17.

Table 17 — Attributes of Local_Enumerated_VD_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
Local_Enumerated_VD_shall_ have_at_least_one_ Permissible_Value_ indicator	1	**Boolean** (ISO/ IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether a local enumerated value domain is required to have at least one permissible value. In the metamodel this link is represented by an instance of the **Permissible_Value_Set** association class (7.4.3.3). Best practice is to set this indicator to TRUE.

7.4.2.22 Permissible_Value_Constraint_Set

7.4.2.22.1 Direct superclass

Permissible_Value_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.2.22.2 Description of Permissible_Value_Constraint_Set

Permissible_Value_Constraint_Set extends **Constraint_Set** with constraints applicable to the **Permissible_Value** class (7.4.2.12) and its associations. **Permissible_Value_Constraint_Set** is illustrated in Figure 2.

7.4.2.22.3 Associations of Permissible_Value_Constraint_Set

As a subclass of **Constraint_Set**, **Permissible_Value_Constraint_Set** inherits **Constraint_Set**'s associations.

7.4.2.22.4 Attributes of Permissible_Value_Constraint_Set

See Table 18.

Table 18 — Attributes of Permissible_Value_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
Permissible_Value_shall_have_exactly_one_Value_Meaning_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether a permissible value is required to have exactly one value meaning. In the metamodel this link is represented by an instance of the **Permissible_Value_Meaning** association class (7.4.3.2). Best practice is to set this indicator to TRUE.

7.4.3 Association Classes in the Conceptual Domain and Value Domain metamodel region

7.4.3.1 Value_Meaning_Set association class

7.4.3.1.1 Direct superclass

Value_Meaning_Set is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.3.1.2 Description of Value_Meaning_Set

Value_Meaning_Set is an association that binds zero, one or more instances of the **Value_Meaning** class (7.4.2.5) to zero, one or more instances of the **Local_Enumerated_Conceptual_Domain** class (7.4.2.3).

Value_Meaning_Set is also a class, with attributes as described in 7.4.3.1.3.

7.4.3.1.3 Attributes of Value_Meaning_Set

See Table 19.

Table 19 — Attributes of Value_Meaning_Set association class

Attribute name	Multiplicity	Datatype	Description
valid_period	0..1	**Datetime_Period** (ISO/IEC 11179-3:2023, 6.3.10)	Definition: datetime period for which the associated value meaning became, or will become, a valid value meaning within the local enumerated conceptual domain If present, this attribute specifies a **Datetime_Period.start_datetime**, a **Datetime_Period.end_datetime** or both. A registration authority may determine whether these datetimes are the datetimes the value meaning becomes valid in a registry, or the datetimes the value meaning becomes part of the source domain or some other datetimes. If no **Datetime_Period.start_datetime** is specified, the **Datetime_Period.start_datetime** of the associated instance of **Conceptual_Domain** (7.4.2.1.4) applies. The absence of the **Datetime_Period.end_datetime** indicates that the value meaning is still valid unless there is a **Datetime_Period.end_datetime** on the associated instance of **Conceptual_Domain** (7.4.2.1.4). It is best practice that, if the **valid_period** is specified, it be within the **valid_period** of the associated instance of the **Conceptual_Domain** class (7.4.2.1.4).

7.4.3.2 Permissible_Value_Meaning association class

7.4.3.2.1 Direct superclass

Permissible_Value_Meaning is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.3.2.2 Description of Permissible_Value_Meaning

Permissible_Value_Meaning is an association that binds zero, one or more instances of the **Permissible_Value** class (7.4.2.12) with zero or one instance of the **Value_Meaning** class (7.4.2.5), providing meaning for the instances of **Permissible_Value**, which in turn provide representation for the instance of **Value_Meaning**.

Permissible_Value_Meaning is also a class, with attributes as specified in 7.4.3.2.3.

7.4.3.2.3 Attributes of Permissible_Value_Meaning

See Table 20.

Table 20 — Attributes of Permissible_Value_Meaning association class

Attribute name	Multiplicity	Datatype	Description
valid_period	0..1	**Datetime_Period** (ISO/IEC 11179-3:2023, 6.3.10)	Definition: datetime period for which the associated permissible value became, or will become, linked with the associated value meaning If present, this attribute specifies a **Datetime_Period.start_datetime**, a **Datetime_Period.end_datetime** or both. A registration authority may determine whether these datetimes are the datetimes the permissible value became linked with the value meaning in a registry, or the datetimes the permissible value became linked with the value meaning in the source domain or some other datetimes. If no **Datetime_Period.start_datetime** is specified, the **Datetime_Period.start_datetime** of the associated instance of **Value_Meaning_Set** (7.4.3.1.3) or **Conceptual_Domain** (7.4.2.1.4) applies. The absence of the **Datetime_Period.end_datetime** indicates that the permissible value is still linked with the value meaning unless there is a **Datetime_Period.end_datetime** on the associated instance of **Value_Meaning_Set** (7.4.3.1.3), if present, else of **Conceptual_Domain** (7.4.2.1.4). It is best practice that, if the **valid_period** is specified, it be within the **valid_period** of the associated instance of **Value_Meaning_Set** (7.4.3.1.3), if present, else of **Conceptual_Domain** (7.4.2.1.4) and further that it be within the **valid_period** of at least one associated instance of **Permissible_Value_Set** (7.4.3.3.3), if present, or of the associated **Value_Domain** (7.4.2.8.4)

7.4.3.3 Permissible_Value_Set association class

7.4.3.3.1 Direct superclass

Permissible_Value_Set is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.4.3.3.2 Description of Permissible_Value_Set

Permissible_Value_Set is an association that binds zero, one or more **Permissible_Values** (7.4.2.12) to zero, one or more **Local_Enumerated_Value_Domain** (7.4.2.11). **Permissible_Value_Set** is a weak containment association, i.e. deletion of the containing domain does not cause a cascading delete of the members (**Permissible_Values**). Thus, it is possible to associate **Permissible_Values** with **Value_Meanings** without defining a complete **Value_Domain**.

Permissible_Value_Set is also a class, with attributes as specified in 7.4.3.3.3.

7.4.3.3.3 Attributes of Permissible_Value_Set

See Table 21.

Table 21 — Attributes of Permissible_Value_Set association class

Attribute name	Multiplicity	Datatype	Description
valid_period	0..1	**Datetime_Period** (ISO/IEC 11179-3:2023, 6.3.10)	Definition: datetime period for which this permissible value became, or will become, a valid permissible value within the local enumerated value domain. If present, this attribute specifies a **start_datetime**, an **end_datetime** or both. A registration authority may determine whether these datetimes are the datetimes the value meaning becomes valid in a registry, or the datetimes the value meaning becomes part of the source domain or some other datetimes. If no **start_datetime** is specified, the **start_datetime** of the associated **Value_Domain** applies (7.4.2.8.4). The absence of the **end_datetime** indicates that the value meaning is still valid unless there is an **end_datetime** on the associated **Value_Domain** (7.4.2.8.4). It is best practice that the **valid_period**, if specified, be within the **valid_period** of the associated instance of the **Value_Domain** class (7.4.2.8.4).

7.4.4 Associations in the Conceptual Domain and Value Domain metamodel region

7.4.4.1 value_domain_meaning association

The **value_domain_meaning** association binds zero, one or more instances of the **Value_Domain** class (7.4.2.8) to zero or one instance of the **Conceptual_Domain** class (7.4.2.1) that provides the meanings for each of the values within the **Value_Domain**, which in turn provide representation to the meanings.

This association is intended primarily to allow an instance of **Value_Domain** to be associatied with an instance of **Conceptual_Domain** before detailed **Permissible_Values** and **Value_Meanings** are associated.

It is best practice to remove the high-level association once the detailed associations are specified. However, if the high-level association is not removed, the constraints on this association specified in 7.4.5.2 shall be enforced.

7.4.4.2 described_value_domain_meaning association

The **described_value_domain_meaning** association binds zero, one or more instances of the **Described_Value_Domain** class (7.4.2.13) to zero or one instance of the **Described_Conceptual_Domain** class (7.4.2.2), providing meaning for the **Described_Value_Domains**, which in turn provide representation for the **Described Conceptual_Domain**.

It is best practice for an instance of **Described_Value_Domain** to be linked with exactly one instance of **Described_Conceptual_Domain**. A registration authority can enforce this using the constraint **Described_Value_Domain_Constraint_Set.Described_VD_shall_have_exactly_one_Described_CD**.

7.4.4.3 domain_definition association

The **domain_definition** association binds zero, one or more instances of the **Enumerated_Conceptual_Domain_Definition** class (7.4.2.7) to zero or one instance of the **Reference_Enumerated_Conceptual_Domain** class (7.4.2.6), providing definition for the **Reference_Enumerated_Conceptual_Domain**.

Where multiple **Enumerated_Conceptual_Domain_Definition**s are specified, the **expression** (7.4.2.7.4) of each shall be equivalent, each specified using a different **notation** (7.4.2.7.4).

7.4.4.4 reference_enumerated_value_domain_meaning association

The **reference_enumerated_value_domain_meaning** association binds zero, one or more instances of the **Reference_Enumerated_Value_Domain** class (7.4.2.13) to zero or one instance of the **Reference_Enumerated_Conceptual_Domain** class (7.4.2.6), which provides meaning for the **Reference_Enumerated_Value_Domain**(s).

It is best practice for an instance of **Reference_Enumerated_Value_Domain** to be linked with exactly one instance of **Reference_Enumerated_Conceptual_Domain**. A registration authority can enforce this using the constraint **Reference_Enumerated_VD_Constraint_Set.Ref_Enum_VD_shall_have_exactly_one_Ref_Enum_CD** (7.4.2.20.4).

7.4.4.5 datatype_has_scheme_reference association

The **datatype_has_scheme_reference** association binds zero or one instances of the **Datatype** class (7.4.2.14) to zero or one instance of the **Datatype_Scheme** class (7.4.2.15), providing the datatype scheme for the datatype.

7.4.5 Additional Constraints of the Conceptual Domain and Value Domain metamodel region

7.4.5.1 Overview of additional contraints

This subclause specifies additional constraints that are not included in the UML diagram.

7.4.5.2 value_domain_meaning association constraints

7.4.5.2.1 Constraint #1: Consistency of Enumeration, Description or combination for Conceptual and Value_Domains

Suppose that *r* is an instance of the class **Value_Domain** (7.4.2.8) and *s* is an instance of the class **Conceptual_Domain** (7.4.2.1), such that *s* is the meaning of *r* according to the **value_domain_meaning** association (7.4.4.1). Then it is the case that:

a) *r* is an instance of **Local_Enumerated_Value_Domain** (7.4.2.11) and *s* is an instance of **Local_Enumerated_Conceptual_Domain** (7.4.2.4); or

b) *r* is an instance of **Reference_Enumerated_Value_Domain** (7.4.2.13) and *s* is an instance of **Reference_Enumerated_Conceptual_Domain** (7.4.2.6); or

c) *r* is an instance of **Described_Value_Domain** (7.4.2.9) and *s* is an instance of **Described_Conceptual_Domain** (7.4.2.2).

Since neither **Value_Domain**s, nor **Conceptual_Domains** are disjoint with respect to the enumerated and described subclasses it is possible that *r* and *s* are some combination of enumerated and described value domains or conceptual domains.

7.4.5.2.2 Constraint #2: Consistency of meanings reached by meaning associations

Suppose that there exists an instance *x* of the class **Described_Value_Domain** (7.4.2.13), such that the instance *y* is the meaning of *x* according to the **value_domain_meaning** association (7.4.4.1) [since

every instance of a **Described_Value_Domain** is also a **Value_Domain** (7.4.2.8)] where *y* is some instance of a **Conceptual_Domain** (7.4.2.1) [either a **Described_Conceptual_Domain** (7.4.2.2) or an **Enumerated_Conceptual_Domain** (7.4.2.3)].

If via the **described_value_domain_meaning** association (7.4.4.2) there also exists an instance *z* of the **Described_Conceptual_Domain** such that *z* is the meaning of *x*. Then it shall be the case that *z* is equal to *y*, i.e. the meaning of *x* shall be same according to both the **value_domain_meaning** and **described_value_domain_meaning** associations.

7.4.5.3 Consistent dimensionalities

The dimensionality of the unit of measure of a value domain shall be the same as the dimensionality of the conceptual domain which provides the meaning of the value domain.

7.5 Measurement metamodel region

7.5.1 Overview of the Measurement metamodel region

The Measurement metamodel region is illustrated in Figure 6.

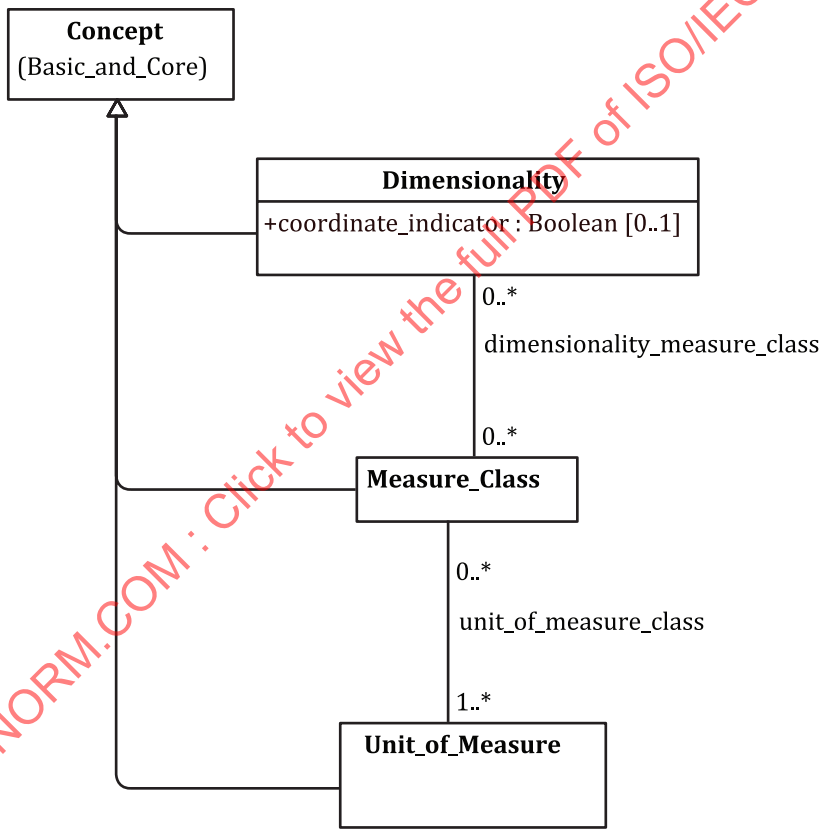


Figure 6 — Measurement metamodel region

7.5.2 Classes in the Measurement metamodel region

7.5.2.1 Unit_of_Measure class

7.5.2.1.1 Direct superclass

Unit_of_Measure is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.5.2.1.2 Description of Unit_of_Measure

Unit_of_Measure is a class each instance of which models a unit of measure, the units in which associated values are measured. If appropriate, an instance of **Value_Domain** (7.4.2.8) (not shown in Figure 6) can be linked to an instance of **Unit_of_Measure** (see 7.4.2.8.4) to specify the units in which the values of any linked instance of **Data_Element** (7.6.2.1) are measured.

7.5.2.1.3 Associations of Unit_of_Measure

As a subclass of **Concept**, **Unit_of_Measure** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Unit_of_Measure** has the following additional association:

— **unit_of_measure_class** (7.5.3.2).

7.5.2.1.4 Attributes of Unit_of_Measure

As a subclass of **Concept**, **Unit_of_Measure** inherits **Concept**'s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4). **Unit_of_Measure** has no additional attributes.

7.5.2.1.5 Constraints on Unit_of_Measure

An instance of the **Unit_of_Measure** class shall be designated and defined using the **item_designation** and **item_definition** associations of the **Item** superclass.

— **Designation.sign** (ISO/IEC 11179-3:2023, 8.4.1.3) shall be specified. This constraint can be enforced by setting **Constraint_Set.item_shall_be_designated_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

— **Definition.text** (ISO/IEC 11179-3:2023, 8.4.2.3) shall be specified. This constraint can be enforced by setting **Constraint_Set.item_shall_be_defined_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

7.5.2.2 Measure_Class class

7.5.2.2.1 Direct superclass

Measure_Class is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.5.2.2.2 Description of Measure_Class

Measure_Class is a class each instance of which models a measure class, a set of equivalent units of measure that can be shared across multiple dimensionalities. **Measure_Class** allows a grouping of units of measure to be specified once and reused by multiple dimensionalities.

EXAMPLE Define the **Measure_Classes**: Metric Linear Distance and Imperial Linear Distance, each associated with the appropriate **Units_of_Measure**, and associate them with **Dimensionalities**: Height, Width and Depth to model the three spatial dimensions.

7.5.2.2.3 Associations of Measure_Class

As a subclass of **Concept**, **Measure_Class** inherits **Concept**'s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Measure_Class** has the following additional associations:

— **dimensionality_measure_class** (7.5.3.1);

— **unit_of_measure_class** (7.5.3.2).

7.5.2.2.4 Attributes of Measure_Class

As a subclass of **Concept**, **Measure_Class** inherits **Concept**'s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4). **Measure_Class** has no additional attributes.

7.5.2.2.5 Constraints on Measure_Class

An instance of the **Measure_Class** class shall be designated and defined using the **item_designation** and **item_definition** associations of the **Item** superclass.

— **Designation.sign** (ISO/IEC 11179-3:2023, 8.4.1.3) shall be specified. This constraint can be enforced by setting **Constraint_Set.item_shall_be_designated_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

— **Definition.text** (ISO/IEC 11179-3:2023, 8.4.2.3) shall be specified. This constraint can be enforced by setting **Constraint_Set.item_shall_be_defined_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

7.5.2.3 Dimensionality class

7.5.2.3.1 Direct superclass

Dimensionality is a subclass of **Concept** (specified in ISO/IEC 11179-3:2023, 6.4.2.2). **Concept** in turn is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.5.2.3.2 Description of Dimensionality

Dimensionality is a class each instance of which models a dimensionality, a set of measure classes each of which in turn groups a set of equivalent units of measure, where equivalence between two units of measure is determined by the existence of a quantity-preserving one-to-one correspondence between values measured in one unit of measure and values measured in the other unit of measure, independent of context, and where the characterizing operations are the same.

A very common example is the use of temperature to measure the absolute temperature of a point, or to measure the size of a temperature interval, e.g. the temperature difference across the wall of a furnace. Aside from the semantic difference, the function for converting units of measure, e.g. temperature, depends on whether it is a coordinate or an interval measure. For example, when converting degrees Celsius to Kelvins, one adds 273.16 for temperature coordinates, but not for temperature interval measures.

In the **Dimensionality** class, the frame of reference for the dimensionality is not explicitly specified. For some units of measure, such as temperature in Kelvins or degrees Celsius, the frame of reference is implicit in the units of measure. Additional examples of coordinate dimensionalities would include longitude and latitude. However, in many cases the frame of reference for a coordinate measurement is specified as part of the **Data_Element** (7.6.2.1). This is quite common in computer aided design applications.

Dimensionalities which use the same units of measure can apply to very different concepts.

EXAMPLE 1 Inches, feet, metres and centimetres are all units of measure whose dimensionality is length. Other common dimensionalities include: mass, time, area, volume, etc.

NOTE 1 The equivalence defined here forms an equivalence relation on the set of all units of measure. Each equivalence class corresponds to a dimensionality. The units of measure "temperature in degrees Fahrenheit" and "temperature in degrees Celsius" have the same dimensionality, because given a value measured in degrees Fahrenheit there is a value measured in degrees Celsius that is the same quantity, and vice-versa. Quantity preserving one-to-one correspondences are the well-known equations °C = (5/9)*(°F - 32) and °F = (9/5)*(°C) + 32. (Temperature coordinates are assumed here. There is no offset when converting among temperature interval measures, e.g. the temperature difference between the coldest and hottest temperature on a day.)

NOTE 2 Units of measure are not limited to physical categories. Examples of physical categories are: linear measure, area, volume, mass, velocity, time duration. Examples of non-physical categories are: currency, quality indicator, colour intensity.

NOTE 3 Quantities can be grouped together into categories of quantities which are mutually comparable. Lengths, diameters, distances, heights, wavelengths and so on would constitute such a category. Mutually comparable quantities usually have the same dimensionality (see also NOTE 4) ISO 80000-1[13] calls these "quantities of the same kind".

NOTE 4 The requirement of common “characterizing operations” for all units of measure with the same dimensionality is a stronger requirement than that commonly adopted in conventional dimensional analysis (where comparability and transformability usually suffice). Thus, with respect to temperature, absolute temperature coordinates (e.g. Kelvins) are here considered to be a different dimensionality than “offset” temperature coordinates (e.g. degrees Celsius or Fahrenheit). It is meaningful to take the ratio of absolute temperature coordinates, but not of “offset” temperature coordinates, wherein the arbitrary translation of zero renders ratios meaningless. The notion of characterizing operations used here has been adapted from the statistics literature where distinctions are commonly made among categorical, ordered, interval and ratio measures.

NOTE 5 Dimensionalities for physical units of measurement are commonly specified as the products or quotients of powers of basis dimensions: mass, length, time, etc. However, this metamodel does not dictate the specification of dimensionalities, only their names and coordinate status.

EXAMPLE 2 Angular velocity and frequency can be defined as two distinct dimensionalities sharing a measure class grouping all units of measure that represent T^{-1} (where T represents Time).

7.5.2.3.3 Associations of Dimensionality

As a subclass of **Concept**, **Dimensionality** inherits **Concept**’s associations (ISO/IEC 11179-3:2023, 6.4.2.2.3). **Dimensionality** has the following additional association:

— **dimensionality_measure_class** (7.5.3.1).

7.5.2.3.4 Attributes of Dimensionality

As a subclass of **Concept, Measure_Class** inherits **Concept**’s attributes (ISO/IEC 11179-3:2023, 6.4.2.2.4). **Dimensionality** has additional attributes as shown in Table 22.

Table 22 — Attributes of Dimensionality class

Attribute name	Multiplicity		Datatype	Description
coordinate_indicator	0...1		**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Conditional. Definition: predicate on a dimensionality whose value is TRUE if the dimensionality is a coordinate. Condition: The indicator shall be specified if the dimensionality represents physical units. If the dimensionality refers to an interval measure, the value of the coordinate_indicator shall be FALSE. If the dimensionality does not represent physical units, the coordinate_indicator shall be NULL. Thus, NULL shall be interpreted as 'Not applicable'. EXAMPLE: There might be two dimensionalities concerned with length: one a measure of the size of an object (hence an interval measure), the other a measure of the location of an object (hence a coordinate).

7.5.2.3.5 Constraints on Dimensionality

An instance of the **Dimensionality** class shall be designated and defined using the **item_designation** and **item_definition** associations of the **Item** superclass.

— **Designation.sign** (ISO/IEC 11179-3:2023, 8.4.1.3) shall be specified. This constraint can be enforced by setting **Constraint_Set.item_shall_be_designated_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

— **Definition.text** (ISO/IEC 11179-3:2023, 8.4.2.3) shall be specified. This constraint can be enforced by setting **Constraint_Set.item_shall_be_defined_indicator** (ISO/IEC 11179-3:2023, 9.4.5.4) to TRUE.

7.5.3 Associations in the Measurement metamodel region

7.5.3.1 dimensionality_measure_class association

The **dimensionality_measure_class** association binds zero, one or more instances of the **Dimensionality** class (7.5.2.3) with zero, one or more instances of the **Measure_Class** class (7.5.2.2) which group together the units of measure that apply to the dimensionality.

NOTE While units of measure are commonly physical units of measure, they can also measure artificial constructs such as currency (in which the corresponding dimensionality would be "Money", or something similar).

7.5.3.2 unit_of_measure_class association

The **unit_of_measure_class** association binds one or more instances of the **Units_of_Measure** class (7.5.2.1) with zero, one or more instances of the **Measure_Class** (7.5.2.2). The association records the grouping of similar units of measure.

NOTE While units of measure are commonly physical units of measure, they can also measure artificial constructs such as currency (in which the corresponding dimensionality would be "Money" or something similar).

7.6 Data Element metamodel region

7.6.1 Overview of the Data Element metamodel region

The Data Element metamodel region, illustrated in Figure 7, is used to address the administration of data element, represented in the metamodel by the **Data_Element** class (7.6.2.1). Data elements provide the formal representations for some information (such as a fact, a proposition, an observation, etc.) about some concrete or abstract thing. Data elements are reusable and shareable representations of data element concepts.

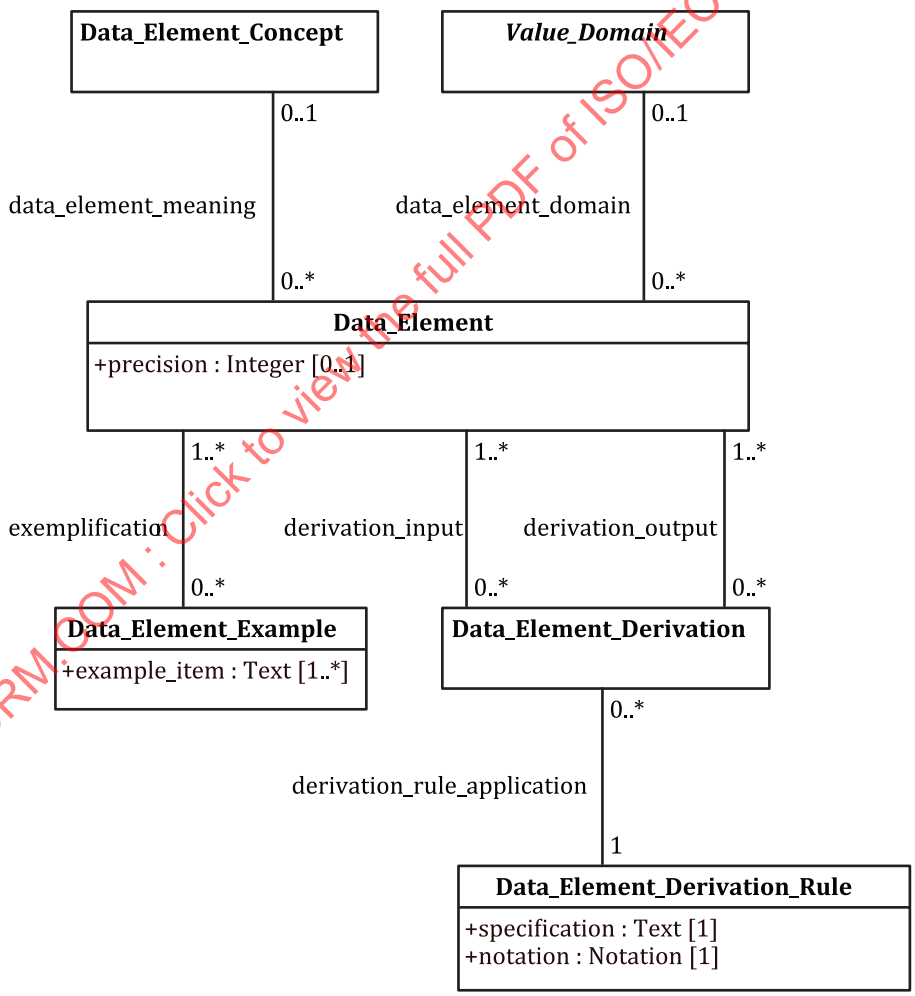


Figure 7 — Data Element metamodel region

NOTE ISO/IEC 11179-3:2003[4] contained an explicit **Representation_Class** to allow typing of **Data_Elements** and **Value_Domains**. In ISO/IEC 11179-3:2013[5] and this document, **Representation_Class** is treated as just an instance of **Classification_Scheme**, described in ISO/IEC 11179-3:2023, Clause 10.

7.6.2 Classes in the Data Element metamodel region

7.6.2.1 Data_Element class

7.6.2.1.1 Direct superclass

Data_Element is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.6.2.1.2 Description of Data_Element

Data_Element is a class each instance of which models a data element, a unit of data that is considered in context to be indivisible.

A data element is considered to be a basic unit of data of interest to an organization. It is a unit of data for which the definition, identification, representation and permissible values are specified by means of a set of attributes.

A data element is formed when a data element concept is assigned a representation. One of the key components of a representation is the value domain, i.e. restricted valid values.

7.6.2.1.3 Associations of Data_Element

As a subclass of **Item**, **Data_Element** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). A **Data_Element** has the following additional associations:

— **data_element_domain** (7.6.3.1);

— **data_element_meaning** (7.6.3.2);

— **exemplification** (7.6.3.3);

— **deriviation_input** (7.6.3.4);

— **derivation_output** (7.6.3.5).

7.6.2.1.4 Attributes of Data_Element

See Table 23.

Table 23 — Attributes of Data_Element class

Attribute name	Multiplicity	Datatype	Description
precision	0..1	**Integer** (ISO/IEC 11179-3:2023, 6.2.5)	Definition: number of decimal places permitted in any associated data element values

7.6.2.2 Data_Element_Concept class

Data_Element_Concept is described under the Data Element Concept metamodel region in 7.3.2.3. This metamodel region specifies the following additional association:

— **data_element_meaning** association (7.6.3.2).

7.6.2.3 Value_Domain class

Value_Domain is described under the Conceptual Domain and Value Domain metamodel region in 7.4.2.8. This metamodel region specifies the following additional association:

— **data_element_domain** association (7.6.3.1).

7.6.2.4 Data_Element_Example class

7.6.2.4.1 Direct superclass

Data_Element_Example is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.6.2.4.2 Description of Data_Element_Example

Data_Element_Example is a class each instance of which models a data element example, a representative illustration of a data element.

7.6.2.4.3 Associations of Data_Element_Example

As a subclass of **Item**, **Data_Element_Example** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). A **Data_Element_Example** has the following additional association:

— **exemplification** association (7.6.3.3).

7.6.2.4.4 Attributes of Data_Element_Example

See Table 24.

Table 24 — Attributes of Data_Element_Example class

Attribute name	Multiplicity	Datatype	Description
example_item	1..*	**Text** (ISO/IEC 11179-3:2023, 6.2.12)	Definition: actual illustrative case of the data element

7.6.2.5 Data_Element_Derivation_Rule class

7.6.2.5.1 Direct superclass

Data_Element_Derivation_Rule is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.6.2.5.2 Description of Data_Element_Derivation_Rule

Data_Element_Derivation_Rule is a class each instance of which models a data element derivation rule, logical, mathematical, other operations or some combination thereof specifying derivation. The data element derivation rule can range from a simple operation such as subtraction to a very complex set of derivations (**derivation** being defined as a relationship between a **Data_Element_Derivation_Rule** and an input set upon which it acts). **Data_Element_Derivation_Rules** are not limited to arithmetic and logical operations.

7.6.2.5.3 Associations of Data_Element_Derivation_Rule

As a subclass of **Item**, **Data_Element_Derivation_Rule** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). A **Data_Element_Derivation_Rule** has the following additional association:

— **derivation_rule_application** association (7.6.3.6).

7.6.2.5.4 Attributes of Data_Element_Derivation_Rule

See Table 25.

Table 25 — Attributes of Data_Element_Derivation_Rule class

Attribute name	Multiplicity	Datatype	Description
specification	1	**Text** (ISO/IEC 11179-3:2023, 6.2.12)	Definition: text of a specification of this data element derivation rule
notation	1	**Notation** (ISO/IEC 11179-3:2023, 6.2.7)	Definition: notation used to specify the data element derivation rule

7.6.2.5.5 Example of Data_Element_Derivation_Rule

One simple example of a data element derivation rule might be concatenation, where if **notation** is 'EBNF', the **specification** can be specified as:

— output = input1, [input2, [input3...]]

7.6.2.6 Data_Element_Derivation class

7.6.2.6.1 Direct superclass

Data_Element_Derivation is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.6.2.6.2 Description of Data_Element_Derivation

Data_Element_Derivation is a class each instance of which models a data element derivation, the application of a data element derivation rule to one or more input data elements to derive one or more output data elements.

Data_Element_Derivation is a class that associates the **Data_Element**(s) (7.6.2.1) that serve as sources or inputs with a **Data_Element_Derivation_Rule** (7.6.2.5) and the **Data_Element**(s) that are the products or outputs of the **Data_Element_Derivation_Rule**.

Data_Element_Derivation operates on the values of instances of **Data_Elements**. For operations on the **Data_Element** types see 7.9.

7.6.2.6.3 Associations of Data_Element_Derivation

As a subclass of **Item**, **Data_Element_Derivation** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). **Data_Element_Derivation** has the following additional associations:

— **derivation_input** (7.6.3.4);

— **derivation_output** (7.6.3.5);

— **derivation_rule_application** (7.6.3.6).

7.6.2.6.4 Attributes of Data_Element_Derivation

Data_Element_Derivation has no attributes.

7.6.2.7 Data_Element_Constraint_Set class

7.6.2.7.1 Direct superclass

Data_Element_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.6.2.7.2 Description of Data_Element_Constraint_Set

Data_Element_Constraint_Set extends **Constraint_Set** with constraints applicable to the **Data_Element** class (7.6.2.1) and its associations. **Data_Element_Constraint_Set** is illustrated in Figure 2.

7.6.2.7.3 Associations of Data_Element_Constraint_Set

As a subclass of **Constraint_Set**, **Data_Element_Constraint_Set** inherits **Constraint_Set**'s associations.

7.6.2.7.4 Attributes of Data_Element_Constraint_Set

See Table 26.

Table 26 — Attributes of Data_Element_Constraint_Set class

Attribute name	Multiplicity	Datatype	Description
DE_shall_have_exactly_one_DEC_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether a data element is required to have exactly one data element concept In the metamodel this link is represented by an instance of the **data_element_meaning** association (7.6.3.2). Best practice is to set this indicator to TRUE.
DE_shall_have_exactly_one_VD_indicator	1	**Boolean** (ISO/IEC 11179-3:2023, 6.2.2)	Definition: indicator as to whether a data element is required to have exactly one value domain In the metamodel this link is represented by an instance of the **data_element_domain** association (7.6.3.1). Best practice is to set this indicator to TRUE.

7.6.3 Associations in the Data Element metamodel region

7.6.3.1 data_element_domain association

The **data_element_domain** association binds zero, one or more instances of the **Data_Element** class (7.6.2.1) with zero or one instance of the **Value_Domain** class (7.4.2.8) that describes a set of possible values that may be recorded in an instance of the **Data_Element**.

It is best practice for a data element to be associated with exactly one value domain. A registration authority can enforce this using the constraint **Data_Element_Constraint_Set. DE_shall_have_exactly_one_VD** (7.6.2.7.4).

7.6.3.2 data_element_meaning association

The **data_element_meaning** association binds zero, one or more instances of the **Data_Element** class (7.6.2.1) with zero or one instance of the **Data_Element_Concept** class (7.3.2.3) that provides the meaning for the data element.

It is best practice for a data element to be associated with exactly one data element concept. A registration authority can enforce this using the constraint **Data_Element_Constraint_Set. DE_shall_have_exactly_one_DEC** (7.6.2.7.4).

A data element concept can be indirectly associated with several value domains resulting in a different data element for each association.

7.6.3.3 exemplification association

The **exemplification** association binds one or more instances of the **Data_Element** class (7.6.2.1) with zero, one or more instances of the **Data_Element_Example** class (7.6.2.4) that each provides an example instance or use of the data element.

7.6.3.4 derivation_input association

The **derivation_input** association binds one or more instances of the **Data_Element** class (7.6.2.1) with zero, one or more instances of the **Data_Element_Derivation** class (7.6.2.6). The association records the data elements that are input to the data element derivation.

7.6.3.5 derivation_output association

The **derivation_output** association binds zero, one or more instances of the **Data_Element_Derivation** (7.6.2.6) with one or more instances of the **Data_Element** class (7.6.2.1) that are the result of the application of the data element derivation. The association records the data elements that are output to the data element derivation.

Note If a **data element** can have multiples sources, then it can also have multiple **data element derivations**.

7.6.3.6 derivation_rule_application association

The **derivation_rule_application** association binds zero, one or more instances of the **Data_Element_Derivation** class (7.6.2.6) with exactly one instance of the **Data_Element_Derivation_Rule** class (7.6.2.5) that specifies the data element derivation rule to be used for the data element derivation.

7.7 Value Domain Subset metamodel region

7.7.1 Overview of the Value Domain Subset metamodel region

This metamodel region, illustrated in Figure 8, specifies how subsets of a local enumerated value domain can be registered for use by data elements where some of the values in the full local enumerated value domain are not applicable.

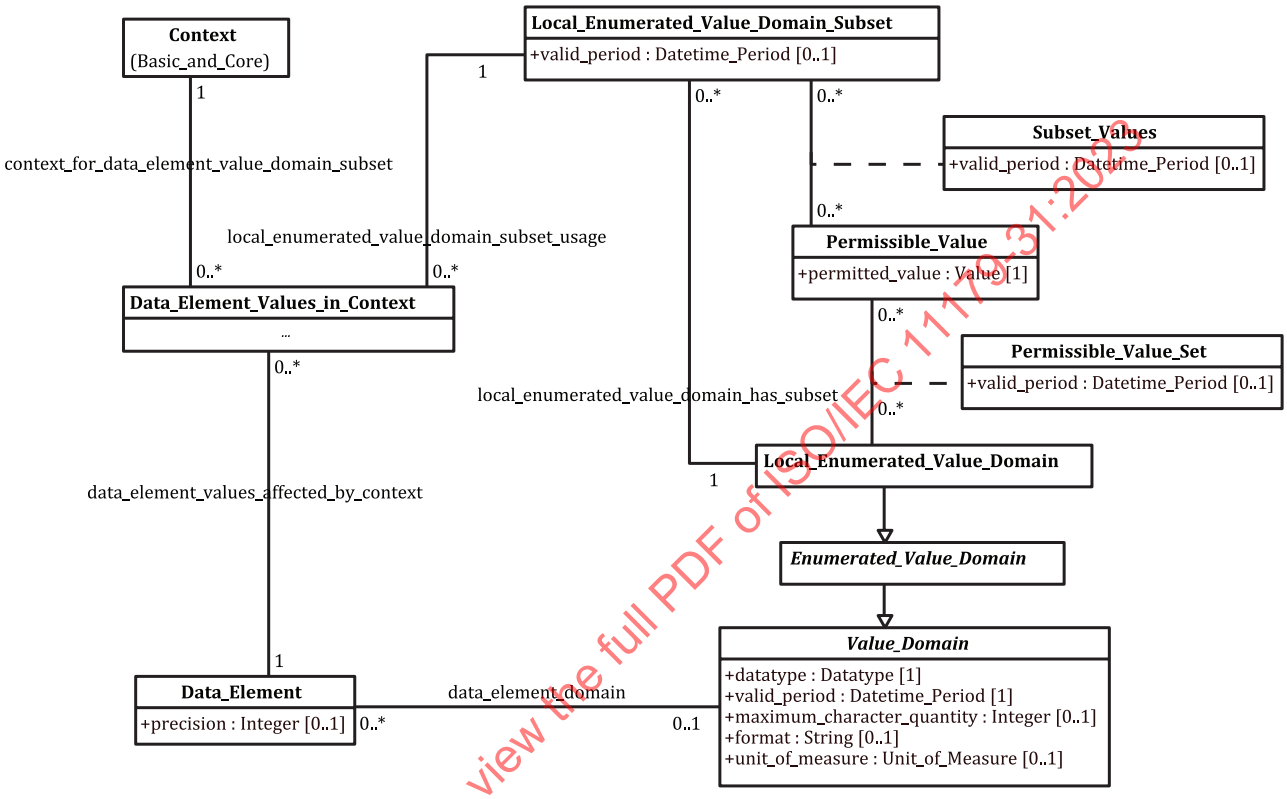


Figure 8 — Value Domain Subset metamodel region

7.7.2 Classes in the Value Domain Subset metamodel region

7.7.2.1 Context class

Context is described in ISO/IEC 11179-3:2023, 6.4.2.3 and ISO/IEC 11179-3:2023, 8.2.2. This metamodel region specifies the following additional association:

— **context_for_data_element_value_domain_subset** association (7.7.4.1).

In this metamodel region, the context specified is that to which only the specified subset of values applies.

7.7.2.2 Data_Element class

Data_Element is described in 7.6.2.1. This metamodel region specifies the following additional association:

— **data_element_values_affected_by_context** association (7.7.4.2).

7.7.2.3 Data_Element_Values_in_Context class

7.7.2.3.1 Direct superclass

Data_Element_Values_in_Context is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.7.2.3.2 Description of Data_Element_Values_in_Context

Data_Element_Values_in_Context is a class each instance of which models the use of a local enumerated value domain subset to represent the permissible values for a data element in a specified context, where some of the permissible values in the full local enumerated value domain are not applicable.

7.7.2.3.3 Associations of Data_Element_Values_in_Context

As a subclass of **Item**, **Data_Element_Values_in_Context** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). **Data_Element_Values_in_Context** has the following additional associations:

— **context_for_data_element_value_domain_subset** (7.7.4.1);

— **data_element_values_affected_by_context** (7.7.4.2);

— **local_enumerated_value_domain_subset_usage** (7.7.4.3).

7.7.2.3.4 Attributes of Data_Element_Values_in_Context

None.

7.7.2.4 Value_Domain class

Value_Domain is described in 7.4.2.8.

7.7.2.5 Enumerated_Value_Domain class

Enumerated_Value_Domain is described in 7.4.2.10.

7.7.2.6 Local_Enumerated_Value_Domain class

Local_Enumerated_Value_Domain is described in 7.4.2.11. This metamodel region specifies the following additional association:

— **local_enumerated_value_domain_has_subset** (7.7.4.4).

7.7.2.7 Local_Enumerated_Value_Domain_Subset class

7.7.2.7.1 Direct superclass

Local_Enumerated_Value_Domain_Subset is a subclass of **Item** (specified in ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.

7.7.2.7.2 Description of Local_Enumerated_Value_Domain_Subset

Local_Enumerated_Value_Domain_Subset is a class each instance of which models a local enumerated value domain subset, a subset of the permissible values in a local enumerated value domain.

7.7.2.7.3 Associations of Local_Enumerated_Value_Domain_Subset

As a subclass of **Item**, **Local_Enumerated_Value_Domain_Subset** inherits **Item**'s associations (ISO/IEC 11179-3:2023, 6.4.2.1.2). **Local_Enumerated_Value_Domain_Subset** has the following additional associations:

— **local_enumerated_value_domain_subset_usage** (7.7.4.3);

— **local_enumerated_value_domain_has_subset** (7.7.4.4);

and the following association class:

— **subset_values** (7.7.3.1).

7.7.2.7.4 Attributes of Local_Enumerated_Value_Domain_Subset

See Table 27.

Table 27 — Attributes of Local_Enumerated_Value_Domain_Subset class

Attribute name	Multiplicity	Datatype	Description
valid_period	1	**Datetime_Period** (ISO/IEC 11179-3:2023, 6.3.10)	Definition: datetime period for which this local enumerated value domain subset became, or will become, a valid local enumerated value domain subset. This attribute specifies a **Datetime_Period.start_datetime** and possibly a **Datetime_Period.end_datetime**. A registration authority may determine whether these datetimes are the datetimes the local enumerated value domain subset becomes valid in a registry, or the datetimes the local enumerated value domain subset becomes part of the source domain or some other datetimes. The absence of the **Datetime_Period.end_datetime** indicates that the local enumerated value domain subset is still valid, unless a **Datetime_Period.end_datetime** is specified on the associated **Value_Domain** (7.4.2.8). The presence of the **Datetime_Period.end_datetime** indicates when the local enumerated value domain subset became or will become invalid. By imputation, the **start_datetime** is also considered to be the datetime at which the associated permissible values (if applicable) were bound to the associated value meanings, unless explicitly overridden for a particular permissible value. Also by imputation, the **end_datetime** is considered to be the datetime on which the associated permissible values (if applicable) ceased or will cease to be bound to their associated value meanings, unless explicitly overridden for a particular permissible value. It is best practice for the **valid_period** to be within the **valid_period** of the associated **Value_Domain**.

7.7.2.7.5 Constraint on Local_Enumerated_Value_Domain_Subset

valid_period shall have a **start_datetime**.

7.7.2.8 Local_Enumerated_VD_Subset_Constraint_Set class

7.7.2.8.1 Direct superclass

Local_Enumerated_VD_Subset_Constraint_Set is a subclass of **Constraint_Set** (ISO/IEC 11179-3:2023, 9.4.5). **Constraint_Set** in turn is a subclass of **Item** (ISO/IEC 11179-3:2023, 6.4.2.1), allowing instances to be identified, registered, administered, named, defined and classified.