

# ISO/IEC JTC 1/SC 32 N 0490

Date: 2000-06-30

REPLACES: --

## ISO/IEC JTC 1/SC 32

### Data Management and Interchange

Secretariat: United States of America (ANSI)  
Administered by Pacific Northwest National Laboratory on behalf of ANSI

DOCUMENT TYPE	Text for CD ballot or comment
TITLE	ISO/IEC CD 11179-3 Information technology - Data Management and Interchange - Metadata Registries (MDR) - Part 3: Registry Metamodel (MDR3)
SOURCE	Joe Christensen (Project Editor)
PROJECT NUMBER	1.32.15.02.03.00
STATUS	This CD has been prepared as specified in the New York, NY, USA meeting resolutions in May 2000. This is a three month ballot.
REFERENCES	
ACTION ID.	LB
REQUESTED ACTION	National Bodies are requested to use the comments template that is used by SC 32/WG 3 in returning their comments. This template is also on the SC 32/WG 2 ftp site.
DUE DATE	2000-10-02
Number of Pages	134
LANGUAGE USED	English
DISTRIBUTION	P & L Members SC Chair WG Conveners and Secretaries

Douglas Mann, Secretariat, ISO/IEC JTC 1/SC 32  
Pacific Northwest National Laboratory \*, 901 D Street, SW., Suite 900, Washington, DC, 20024-2115,  
United States of America

Telephone: +1 703 575 2114; Facsimile: +1 703 681 9180; E-mail: [MannD@battelle.org](mailto:MannD@battelle.org)

available from the JTC 1/SC 32 WebSite <http://www.jtc1sc32.org/>

\*Pacific Northwest National Laboratory (PNL) administers the ISO/IEC JTC 1/SC 32 Secretariat on behalf of ANSI



ISO/IEC JTC 1/SC 32 N **490**

Date: 2000-06-30

(Australia)

ISO/IEC CD 11179-3

ISO/IEC JTC 1/SC 32/WG 2

Secretariat: ANSI

## Information Technology - Data Management and Interchange — Metadata Registries (MDR) — Part 3: Registry Metamodel (MDR3)

### EDITOR'S NOTE

In accordance with SC32/WG2 resolutions of May 2000, it is proposed that the name of the overall standard and each part be renamed and revisions/corrigenda to other parts be considered as part of the CD Ballot process for this revision of Part 3.)

### Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International Standard  
Document subtype:  
Document stage: (30) Committee  
Document language: E

### Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

*[Indicate :  
the full address  
telephone number  
fax number  
telex number  
and electronic mail address*

*as appropriate, of the Copyright Manager of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the draft has been prepared]*

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

## Contents

Foreword.....	xi
Introduction .....	xii
<b>1 Scope.....</b>	<b>1</b>
1.1 Basic attributes of administered components.....	1
1.2 Structure of a metadata registry.....	1
<b>2 Normative reference(s).....</b>	<b>2</b>
<b>3 Definition(s) .....</b>	<b>4</b>
3.1 Definitions of Metamodel Constructs .....	4
3.2 Broader Terms Used in this Part of this Standard .....	5
<b>4 Structure of a Metadata Registry.....</b>	<b>9</b>
4.1 Metamodel for the content of a metadata registry .....	9
4.2 Application.....	10
4.3 Extensibility .....	10
4.4 Description of metamodel.....	10
4.5 Administration Region.....	12
4.6 Naming and Identification Region.....	15
4.7 Classification Region.....	17
4.8 Administered data component Concept Region .....	19
4.9 Conceptual and value domain Region.....	22
4.10 Administered data component Region .....	25
4.11 Metamodel specification .....	26
4.12 Metamodel diagram .....	27
4.13 Modeling object description specifications .....	27
4.14 Metadata class, attribute, and relationship field description specification .....	28
4.15 Alphabetical list of metadata objects.....	28
<b>5 Conformance .....</b>	<b>39</b>
5.1 Conformance level.....	40
5.2 Summary of conformance labels .....	42
5.3 MDR3 metadata set conformance .....	43
5.4 MDR3 metadata instance conformance.....	43
5.5 Application conformance.....	44
5.6 Application varieties .....	45
<b>Annex A (Normative) Modelling Notation.....</b>	<b>49</b>
A.1 Modelling symbols.....	49
<b>Annex B (Normative) IDL representation of the metamodel.....</b>	<b>55</b>
<b>Annex C (Informative) IDEF1X alternative representation model .....</b>	<b>57</b>
<b>Annex D (Informative) Object Role Modeling (ORM) - Natural language Information Analysis Method (NIAM) alternative representation model.....</b>	<b>65</b>
D.1 ISO/IEC 11179-3 Metamodel expressed using ORM graphical form.....	65
D.2 ISO/IEC 11179-3 Metamodel Elementary Sentences from the Object Role Model.....	78
<b>Annex E (Informative) XML Encoding for Data Registry Contents.....</b>	<b>99</b>
E.1 Semi-automatic Encoding of 11179 Part 3 Metamodel into XML Schema .....	99
E.2 Schema Generation Rules.....	101
E.3 Example of Registry Content Encoded in XML.....	102

<b>Annex F (Informative) Mapping the ISO/IEC 11179-3:1994 Basic Attributes to the new metamodel .....</b>	<b>105</b>
<b>F.1 Introduction .....</b>	<b>105</b>
<b>F.2 Mapping the Basic Attributes to the Metamodel.....</b>	<b>106</b>
<b>F.3 Definitional attributes (was sub-clause 6.2).....</b>	<b>110</b>
<b>F.4 Relational attributes (was sub-clause 6.3) .....</b>	<b>110</b>
<b>F.5 Representational attributes (was sub-clause 6.4) .....</b>	<b>114</b>
<b>F.6 Administrative attributes (was sub-clause 6.5).....</b>	<b>120</b>
<b>Bibliography .....</b>	<b>123</b>

## Table of Figures

Figure 1: Metamodel regions .....	11
Figure 2: Administration metamodel region .....	12
Figure 3: Naming and identification metamodel region .....	15
Figure 4: Classification metamodel region.....	17
Figure 5: Administered data component concept metamodel region .....	19
Figure 6: Conceptual and value domain metamodel region .....	22
Figure 7: Administered data component administration metamodel region .....	25
Figure 8: High-level metamodel .....	27
Figure A-1: Sample modelling diagram.....	49
Figure A-2: Class modelling representation.....	50
Figure A-3: Association modelling representation .....	50
Figure A-4: Class relationship modelling representation .....	50
Figure A-5: Class relationship cardinality modelling representation.....	50
Figure A-6: Associative class modelling representation .....	51
Figure A-7: Subtype modelling representation.....	51
Figure A-8: Aggregation modelling representation .....	52
Figure A-9: Composite Aggregation modelling representation .....	52
Figure A-10: Class-attribute modelling representation .....	53
Figure C-1 – IDEF1X high-level metamodel .....	57
Figure C-2 – IDEF1X Administration metamodel region.....	58
Figure C-3 – IDEF1X naming metamodel region.....	59
Figure C-4 – IDEF1X classification metamodel region .....	60
Figure C-5 – IDEF1X conceptual and value domain administration metamodel region .....	61
Figure C-6 – IDEF1X administered data component administration metamodel region .....	62
Figure C-7 – IDEF1X administered data component concept administration metamodel region.....	63

Figure D-1: ORM high-level metamodel..... 65

Figure D-2: ORM administration metamodel region ..... 67

Figure D-3: ORM naming and identification metamodel region ..... 67

Figure D-4: ORM classification metamodel region ..... 68

Figure D-5: ORM administered data component concept administration region ..... 69

Figure D-6: ORM conceptual domain and value domain administration region ..... 70

Figure D-7: ORM administered data component administration region..... 71

Figure D-8: ORM administered component ..... 72

Figure D-9: ORM administered components..... 73

Figure D-10: ORM registration authority ..... 74

Figure D-11: ORM value domain ..... 75

Figure D-12: ORM contact ..... 76

Figure D-13: ORM administered data component concept relationship ..... 77

Figure D-14: ORM value domain relationship..... 77

Figure D-15: ORM conceptual domain relationship ..... 78

Figure F-1: Basic Attributes of Administered data components..... 105



## 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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 11179 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 11179-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 32, *Data Management and Interchange*.

This second edition cancels and replaces the first edition (ISO/IEC 11179-3:1994), clause(s) / subclause(s) / table(s) / figure(s) and annex(es) of which have been technically revised.

ISO/IEC 11179 consists of the following parts, under the general title *Information Technology - Data Management and Interchange — Metadata Registries (MDR)*:

- Part 1: Framework for the Specification and Standardization of Data Elements
- Part 2: Classification for data elements
- *Part 3: Registry Metamodel (MDR3)*
- Part 4: Rules and guidelines for the formulation of data definitions
- Part 5: Naming and identification principles for data elements
- Part 6: Registration of data elements

### EDITOR'S NOTE

In accordance with SC32/WG2 resolutions of May 2000, it is proposed that the name of the overall standard and the other parts, and revisions/corrigenda to other parts, be considered as part of the CD Ballot process for this revision of Part 3.

## Introduction

This Part of the International Standard specifies the structure of a metadata registry and defines basic attributes for specifying administered data components.

Data processing and electronic data interchange rely heavily on accurate, reliable, controllable and verifiable data recorded in databases.

A prerequisite for correct and proper use and interpretation of data is that both users and owners of data have a common understanding of the meaning and representation of administered components such as administered data components. To facilitate an understandable shared view, a number of characteristics, or attributes, have to be defined. A set of attributes required for effective interchange of metadata is described in sub-clause 4.15.

EDITOR'S NOTE: Would the above paragraph be better positioned after the next two?
---

This Part of the standard also addresses the broader administration and retrieval of data by specifying the structure of a metadata registry (Clause 4). A metadata registry is a place to keep facts about characteristics of data that are necessary to clearly describe, record, analyse, and classify data. Clause 4 provides a structure for the recording of metadata that supports the administration and retrieval of registered data including the attribution, classification, definition, naming, identification and registration of administered data components described elsewhere in this standard. This assists users of shared data to have a common understanding of a unit of data's meaning, representation, and identification. A metadata registry supports data sharing with cross-system and cross-organization descriptions of common units of data.

The structure of a metadata registry is stated in the form of a conceptual data model.

This standard is of interest to information developers, information managers, data administrators, and others who are responsible for making data understandable and shareable. It is also of interest to manufacturers of metadata registry "metadata repository" and CASE tool products.

# Information Technology - Data Management and Interchange — Metadata Registries (MDR) — Part 3: Registry Metamodel (MDR3)

## 1 Scope

### 1.1 Basic attributes of administered components

This clause of the International Standard applies to activities including:

- a) the definition, specification and contents of administered data component dictionaries or data registries, including interchanging or referencing among various collections of administered data components
- b) the design and specification of application-oriented data models, databases and message types for data interchange;
- c) the actual use of data in communications and information processing systems;
- d) interchanging or referencing among various collections of administered data components.

This clause of the International Standard is limited to a set of required attributes independent of their machine language description in application systems, databases, data interchange messages etc.

This standard 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. The standard does not directly apply to the actual use of data in communications and information processing systems.

This set of required attributes will have to be extended with additional attributes to enable the performance of a comprehensive data management function. They provide a broad logical structure of the data, but no particular physical structure is implied in this Part of the International Standard.

### 1.2 Structure of a metadata registry

Clauses 8 and 9 in this Part of the International Standard specify the structure of a metadata registry. The structure is stated in the form of a conceptual data model.

A comprehensive metadata registry management function requires a set of rules and procedures. These rules and procedures are set out in the following clauses (and their Annexes) and are complemented elsewhere in this Standard by:

- a) the description of attributes in Clauses 5 to 7 of Part 3 of the standard
- b) rules and guidelines for classifying administered data components in Part 2
- c) rules and guidelines for the formulation of definitions in Part 4
- d) naming and identifying principles for administered data components in Part 5 and
- e) rules and guidelines for registering administered data components in Part 6 of the International Standard.

## **2 Normative reference(s)**

The following standards contain provisions, which, through reference in the text, constitute provisions for this Part of the International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Part of the International Standard are encouraged to investigate the possibility of applying the most recent editions of standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO Standards Handbook 10, Data Processing - Vocabulary, 1982

ISO 646:1983, Information Interchange - ISO 7-bit coded character set for information interchange

ISO 704:1987, Principles and methods of terminology

ISO FDIS 704 Terminology work - Principles and methods, 1999-10-30

ISO DIS 1087-1: Terminology work - Vocabulary - Part 1: Theory and Application 1998-01-21

ISO/IEC DIS 2382-4:1987, Information technology - Vocabulary - Part 4 Organization of Data

ISO/IEC 2382-17:1993, Information technology - Vocabulary Part 17: Databases

ISO 3166:1988, Code for the representation of names of countries

ISO 5127-1: Documentation and Information - Vocabulary Part 1: Basic concepts

ISO 6093:1985, Information processing - Representation of numerical values in character strings for information interchange

ISO TR 9007:1987 Information processing systems - Concepts and terminology for the conceptual schema and the information

ISO 10241:1992, International terminology standards - preparation and layout

ISO/IEC 11179-1:1999, Information technology – Specification and Standardization of data elements Part 1 : Framework for the specification and standardization of data elements

ISO/IEC 11179-2:1999, Information technology – Specification and Standardization of data elements Part 2 : Classification of data elements

ISO/IEC 11179-4:1995, Information technology – Specification and Standardization of data elements Part 4 : Rules and guidelines for the formulation of data definitions

ISO/IEC 11179-5:1995, Information technology – Specification and Standardization of data elements Part 5 : Naming and identification principles for data elements

ISO/IEC 11179-6:1997, Information technology – Specification and Standardization of data elements Part 6 : Registration of data elements

ISO/IEC 11404: 1996, Information technology – Language-Independent Datatypes

ISO CD 15046-3, Geographic information - Part 3: Conceptual schema language, 1999-07-23

ISO/IEC TR 15452:2000, Information technology – Specification of data value domains

### 3 Definition(s)

EDITOR'S NOTE: The definitions in clause 3.1, 3.2, and 4.15 need to be merged into a revised Part 1. Once such a revised part 1 reaches the same stage of progression as this part, it may be possible to reference the definitions of terms from there rather than include them here. However, the document is easier to understand if all definitions are readily available.

For the purposes of this International Standard, the following definitions apply.

Clause 3.1 sets out definitions of metamodel constructs.

Clause 3.2 sets out definitions of broader terms used this part of this standard that are not included in either clause 3.1 or clause 4.15.

Terms used in the metamodel itself are set out in clause 4.15.

#### 3.1 Definitions of Metamodel Constructs

##### 3.1.1 attribute

A characteristic of an object or entity.

##### 3.1.2 attribute capsule

<definition to be supplied>

##### 3.1.3 attribute value

A specific occurrence of an attribute.

Note: See ISO 2382. Part 17.

##### 3.1.4 class

<definition to be supplied>

##### 3.1.5 component

A collective term used to refer to one or more object classes in this model.

##### 3.1.6 conditional MDR3 administered data component attribute

A mandatory MDR3 administered data component attribute, if certain conditions are satisfied.

##### 3.1.7 definition

A statement which describes a concept and permits its differentiation from other concepts within a system of concepts. (Note: See ISO 1087.)

##### 3.1.8 designation

Representation of a concept by a sign which denotes it.

##### 3.1.9 extended MDR3 administered data component attribute

A administered data component attribute that is not defined by this Standard.

Alternative form: extensions to administered data component attributes.

### **3.1.10 identifier**

A linguistically neutral sequence of characters, capable of uniquely identifying that with which it is associated, within a specified context.

### **3.1.11 language**

System of signs for communication, usually consisting of a vocabulary and rules. (From: ISO 5127-1)

### **3.1.12 mandatory MDR3 administered data component attribute**

A administered data component attribute that is defined and required by this Standard.

### **3.1.13 metadata**

Data that defines and describes other data.

### **3.1.14 MDR3 administered data component attribute**

An attribute, as defined by this Standard, of an administered component.

### **3.1.15 name**

The designation of an object by a linguistic expression.

### **3.1.16 optional MDR3 administered data component attribute**

A administered data component attribute that is defined but not required by this Standard.

### **3.1.17 relationship**

A mental link between two or more concepts.

### **3.1.18 required MDR3 administered data component attribute**

See "mandatory MDR3 administered data component attribute".

### **3.1.19 term**

<definition to be supplied>

## **3.2 Broader Terms Used in this Part of this Standard**

### **3.2.1 basic attribute**

An attribute of an administered data component frequently needed to specify an administered data component

### **3.2.2 binding**

<definition to be supplied>

### **3.2.3 classified Component**

An administered component classified in one or more classification schemes.

**3.2.4 concept**

A unit of thought constituted through abstraction on the basis of characteristics common to a set of objects.

**3.2.5 conceptual data model:**

A data model that describes how relevant information is structured in the natural world. It is how the human mind is accustomed to thinking of the information.

**3.2.6 consume**

To read information and then to process it to the extent that some lexical or coding boundaries are discovered. A consumer performs a limited number translation phases.

*Other Forms:* consumer, consumption.

*See Also:* interpret; produce.

*Note:* MDR3 administered data component attributes are consumed before they are interpreted.

EDITOR'S NOTE

Do we want examples included with definitions? Is there a better place to put them?

*Example:* In the following octet stream:

```
<R>
  <A>123.45</A>
  <B>PQR</B>
  <C X="Y">Z</C>
</R>
<R>
  <D>JKL</D>
  <E>
    <F>XXX</F>
    <G>YYY</G>
  </E>
</R>
```

a consumer (as specified in a particular standards binding) might recognize:

- there are two records, both tags are "R"
- the first "R" record contains three records with tags "A", "B", "C"
- the second "R" record contains two records with tags "D" and "E"

However, the consumer:

- might not understand the meanings of tags: what does "<B>...</B>" mean?
- might not validate the tags: is "<C>" permitted to have the attribute "X"?
- might not validate the contents of the records: within record "A", is the contents "123.45" a valid value?



- might limit the depth of its analysis: "R" is only explored one level deep to discover tags "D" and "E", but only a limited analysis (e.g., finding balanced tags) of the contents of "E" such that tags "F" and "G" are not analyzed or discovered.

### 3.2.7 data

A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation or processing by humans or by automatic means. (Note: See ISO 2382, Part 4.)

### 3.2.8 administered data component dictionary

An information resource that specifies, defines and lists a set of administered data components relevant in a specific field of interest.

### 3.2.9 data model

A description of the organization of data in a manner that reflects an information structure.

### 3.2.10 entity

Any concrete or abstract thing of interest, including associations among things.

### 3.2.11 generate

To transform information from its meaning to some form suitable for information interchange.

*Other Forms:* generation.

*See Also:* interpret; produce.

*Note:* MDR3 administered data component attributes are generated before they are produced.

*Example:* To serialize information according to a conceptual model, but not yet render the information in a specific coding or encoding.

### 3.2.12 information model

A high level description of the organization of information in a manner that reflects its structure. It takes the form of logical groupings of entities and levels of sub-entities, without showing any relationships between entities other than the hierarchies of sub-entities.

### 3.2.13 interpret

To process information to discover its meaning, to the extent required by this Standard.

*Other Forms:* interpreter, interpretation.

*See Also:* consume; generate.

*Note:* MDR3 administered data component attributes are consumed before they are interpreted.

## EDITOR'S NOTE

Do we want examples included with definitions? Is there a better place to put them?

*Example:* In the following octet stream:

```
<R>
  <A>123.45</A>
  <B>PQR</B>
  <C X="Y">Z</C>
</R>
<R>
  <D>JKL</D>
  <E>
    <F>XXX</F>
    <G>YYY</G>
  </E>
</R>
```

a consumer (as specified in a particular standards binding) might recognize:

- there are two records, both tags are "R"
- the first "R" record contains three records with tags "A", "B", "C"
- the second "R" record contains two records with tags "D" and "E"

Because only these tags are recognized, only these tags are candidates for interpretation. Assuming tag "E" represents an extension, an interpreter might only recognize the standardized tags "A", "B", "C", and "D".

Based on (1) the separation of the "consume" and "interpret" phases of translation; and (2) a particular standards binding (XML-like in this case); then an application might only interpret the standardized features A, B, C, and D.

An MDR3 application that combines consumption and interpretation, but only interprets standardized MDR3 administered data component attributes might be strictly conforming MDR3 reader.

### 3.2.14 metadata entry application

An application used for the recording of metadata, either manually, or through its extraction from existing databases

### 3.2.15 metadata reader application

An application used for the consumption and interpretation of metadata.

### 3.2.16 metadata registry

A system for managing structured metadata describing the semantic content of sharable data and metadata. a metadata set is an instance of a set instances of metadata

### 3.2.17 metadata set

An instance of a set instances of metadata

**3.2.18 minimum-maxima**

The smallest maximum value.

*Example:* "The minimum-maxima length of element X shall be 25".

**3.2.19 produce**

To process data to extent that lexical or coding boundaries are defined and then to write the resultant information.

*Other Forms:* producer, production.

*See Also:* generate; consume.

*Note:* MDR3 administered data component attributes are generated before they are produced.

**3.2.20 schema**

A formalism for representing knowledge about a simple concept, an entity, or a class of objects by means of its possible values.

**3.2.21 shareable data**

Data that has precise identifiers, meaning, structures, and values.

**3.2.22 syntax**

<definition to be supplied>

**4 Structure of a Metadata Registry****4.1 Metamodel for the content of a metadata registry**

This standard 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.

A metamodel is like any other data model except it describes the structure of data that is about data. Since this standard is specified as a conceptual data model, there can be no exact mapping to an implementation of a metadata registry. A conceptual data model 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. As a conceptual data model, there is no one-to-one match between attributes and fields, columns, objects, et cetera in a database. There may be more than one field per attribute and some entities and relationships may be implemented as fields. There is no intent that an implementation should have a table for each relationship or entity.

It is assumed that an implementor will use this conceptual data model to develop a more specific logical data model of the identical sphere of interest. A logical data model describes the same data, but as structured in an information system. It is often referred to as a Model of the Information System. A logical data model can be directly used for database design.

## 4.2 Application

A metamodel is necessary for coordination of data representation between environments that store, manipulate and exchange data. The metamodel enables systems tools and information repositories to store, manipulate and exchange the metadata for data attribution, classification, definition, naming, identification, and registration. In this manner, consistency of data content is assured among systems tools and information repositories.

Using the metamodel, mappings to the schema of each tool set can be developed. The metamodel constructs can be translated into the language of each tool set, preserving the concepts represented in the original model.

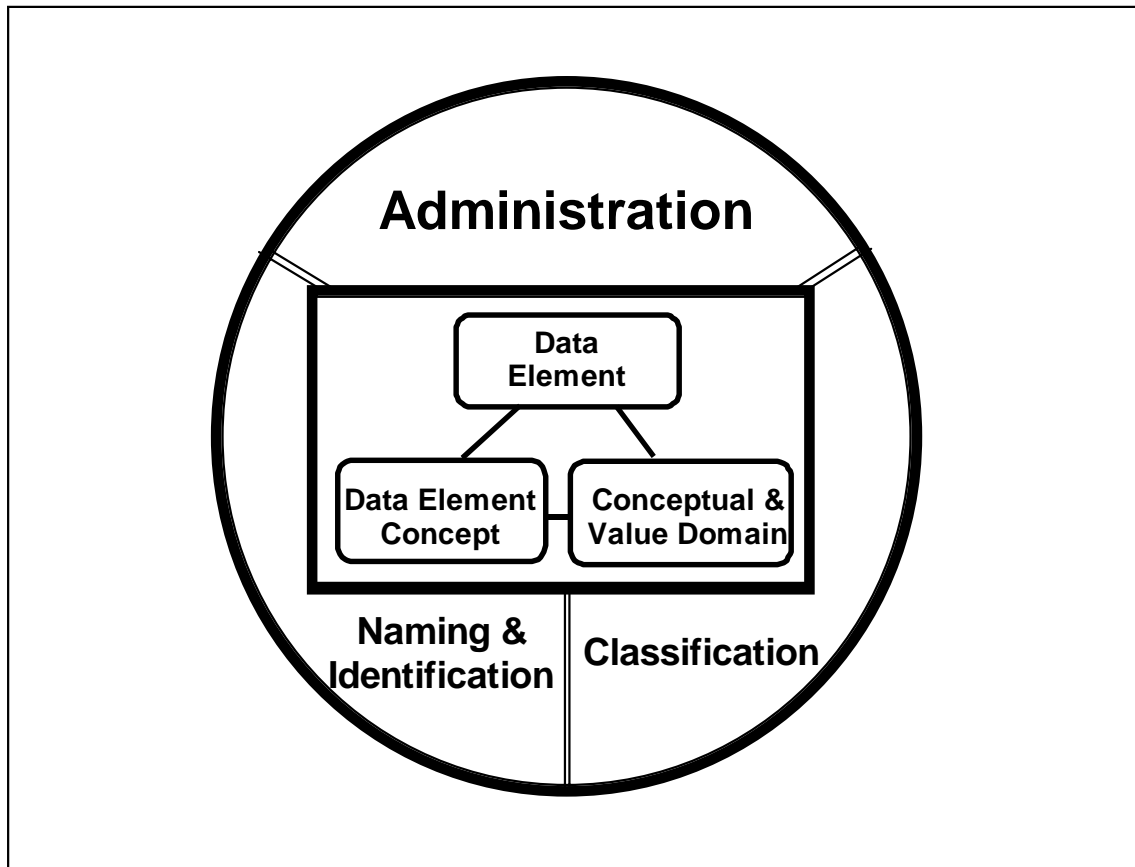
## 4.3 Extensibility

It is not expected that this schema will completely accommodate all users. For example, scientific data requires metadata attributes not addressed in this standard. Each industry or each application may find a need to extend this schema. Such extensions shall be considered conformant if they do not violate any of the rules inherent in the structure and content as specified by the metamodel in this standard. Entities, relationships, and attributes may be added to this conceptual data model, but the core schema shall not be altered.

## 4.4 Description of metamodel

The metamodel can be divided into six functional regions for descriptive purposes, as shown in Figure 1: Metamodel regions. The primary region is the Administration region that supports the administrative components in a registry. It is from this region that the other regions extend. The other five regions are Naming and Identification, Classification, Administered data component, Administered data component Concept, and Conceptual and Value Domain.

The Naming and Identification region is used to manage names of components in the registry. The Classification region is used to manage classification schemes and the registry components that are in the classification schemes. Administered data components, administered data component concepts, conceptual domain, and value domains are administered in their respective regions under the Administration, naming, and classification support regions.



**Figure 1: Metamodel regions**

Is the remaining text in Clause 4.4 supposed to be deleted?

The structure described by this metamodel may be distributed over several implementations. These implementations may be database, registries, repositories, dictionaries, etc. The importance of this metamodel is the exposure for understanding, sharing, and reusing of the contents of implementations.

The following subclauses detail the six regions. The complete metamodel diagram is represented graphically in Figure 8: High-level metamodel in clause 4.12. Annex A contains an explanation of the symbols used in the following model diagrams. Clause 4.15 contains descriptions detailing each entity, relationship, and attribute in the diagrams.

The following is a narrative and diagram description of the regions shown in Figure 1. In the text of the clause, labels from in the diagrams are shown in *italics*. References to an object in one diagram that are detailed in another diagram are shown with a grey background.

### 4.5 Administration Region

The primary region is the Administration region that supports the administrative aspects of components in a registry. It is from this region that the other regions extend. This region addresses the *organizations* that are responsible for and/or have submitted components to the registry. The supporting documentation is also addressed in this region. Figure 2 represents the Administration region.

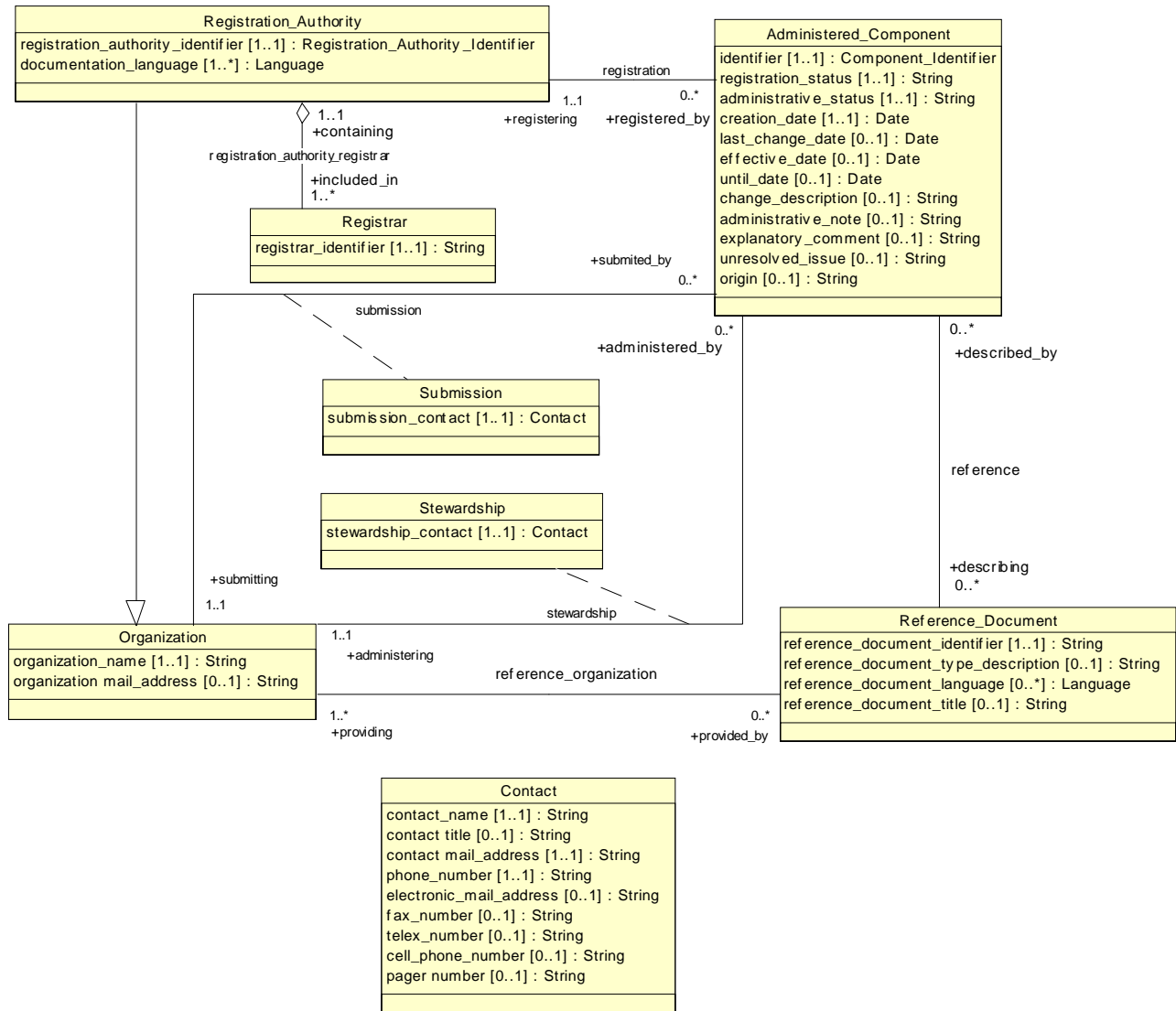


Figure 2: Administration metamodel region

#### 4.5.1 Administration metamodel region components

An *administered component* is a component for which administrative information is recorded. It may be an administered data component or its components (classification scheme, value domain, administered data component concept, conceptual domain, object class, property, etc.) that require definition and specification for reuse and/or sharing in or among enterprises. It is an important component that requires administration for use in and among organizations. An *administered component* is a generalization for an *administered data component*, *classification scheme*, *value domain*, *rule*, or *classified component*. Each of these subtypes has the relationships of the *administered component*.

A *registration authority* is any organization authorized to register metadata. A *registration authority* is a subtype of organization and inherits all of its attributes and relationships. An *administered component* has a *registration authority* that is its owner.

For each *registration authority*, each instance of an *administered component* shall have a unique identifier used to identify it and a *version* that together, distinguishes it from any other *administered component*. Identifiers shall be unique within a *registration authority* for each occurrence of an *administered component*, (i.e., *object class*, *property*, *administered data component*, *administered data component concept*, *conceptual domain*, etc.). Each *administered component* in the owner's metadata registry shall have a (administered component) *registration status* indicating the point in a registration life cycle applying to it. Each *administered component* in the owner's metadata registry shall also have a (administered component) *administrative status* indicating the point in the registration processing process. A *registration authority* may register many *administered components*. In turn, many *registration authorities* may register an *administered component*.

When an *administered component* is submitted to the metadata registry for registration, the responsible and submitting organizations shall be identified as well as the *contact person* for each organization.

A *registration authority* is composed of one or more *registrars* — these are the persons who perform the administrative steps to register *administered components* in a metadata registry.

Each *administered component* has a *record* of administrative information treated as a unit. When an *administered component* is modified, it is a new *version* and is thus a new administered component. The *administered component - creation date*, the reason for change (*administered component - change*), the *contact persons* for the responsible and submitting organizations, registration authority, and the *registrar* shall be provided for this new *administered component*. The *registrar* may collect history by retaining the old *administered component*.

An *administered component* may also have one or more *reference documents*. For each *reference document*, the organization that authored the reference document must be identified.

#### 4.5.2 Administration metamodel region attributes

The attributes in the Administration metamodel region shall be as follows:

<u>Attribute</u>	<u>Allowed Occurrences</u>
<i>Contact person - mail address</i>	One for each <i>component responsible organization</i> and for each <i>component submitting organization</i>
<i>Contact person - name</i>	One for each <i>component responsible organization</i> and for each <i>component submitting organization</i>
<i>Contact person - title</i>	One optional for each <i>component responsible organization</i> and for each <i>component submitting organization</i>

<u>Attribute</u>	<u>Allowed Occurrences</u>
<i>Contact person - phone number</i>	Required
<i>Cell phone number</i>	Optional
<i>Pager number</i>	Optional
<i>Electronic mail address</i>	Optional
<i>Fax number</i>	Optional
<i>Telex number</i>	Optional
<i>Administered component - administrative status</i>	One per <i>administered component</i>
<i>Administered component - registration status</i>	One per <i>administered component</i>
<i>Administered component - identifier</i>	One per <i>administered component</i>
<i>Administered component - version</i>	One per <i>administered component</i>
<i>Organization - label</i>	One for each <i>component responsible organization</i> and for each <i>component submitting organization</i>
<i>Organization- mail address</i>	One optional for each <i>component responsible organization</i> and for each <i>component submitting organization</i>
<i>Administered component - change</i>	One conditional per <i>administered component</i>
<i>Administered component- creation date</i>	One per <i>administered component</i>
<i>Administered component - administrative note</i>	One optional per <i>administered component</i>
<i>Administered component - origin</i>	One optional per <i>administered component</i>
<i>Administered component - unresolved issue</i>	One optional per <i>administered component</i>
<i>Reference document - label</i>	One for each <i>reference document</i>
<i>Reference document - type</i>	One optional for each <i>reference document</i>
<i>Registrar - label</i>	One for each <i>registrar</i> in a <i>registration authority</i>
<i>Registration authority - documentation language</i>	From one to many per <i>administered component</i>
<i>Registration authority - identifier</i>	From one to many per <i>administered component</i>



## 4.6 Naming and Identification Region

The Naming and Identification region is used to manage the names of administered components and the `name` contexts that provide the sphere for the names. It is recognized that a component may have many names that will vary depending on discipline, locality, technology, etc. Figure 3 represents the Naming and Identification region.

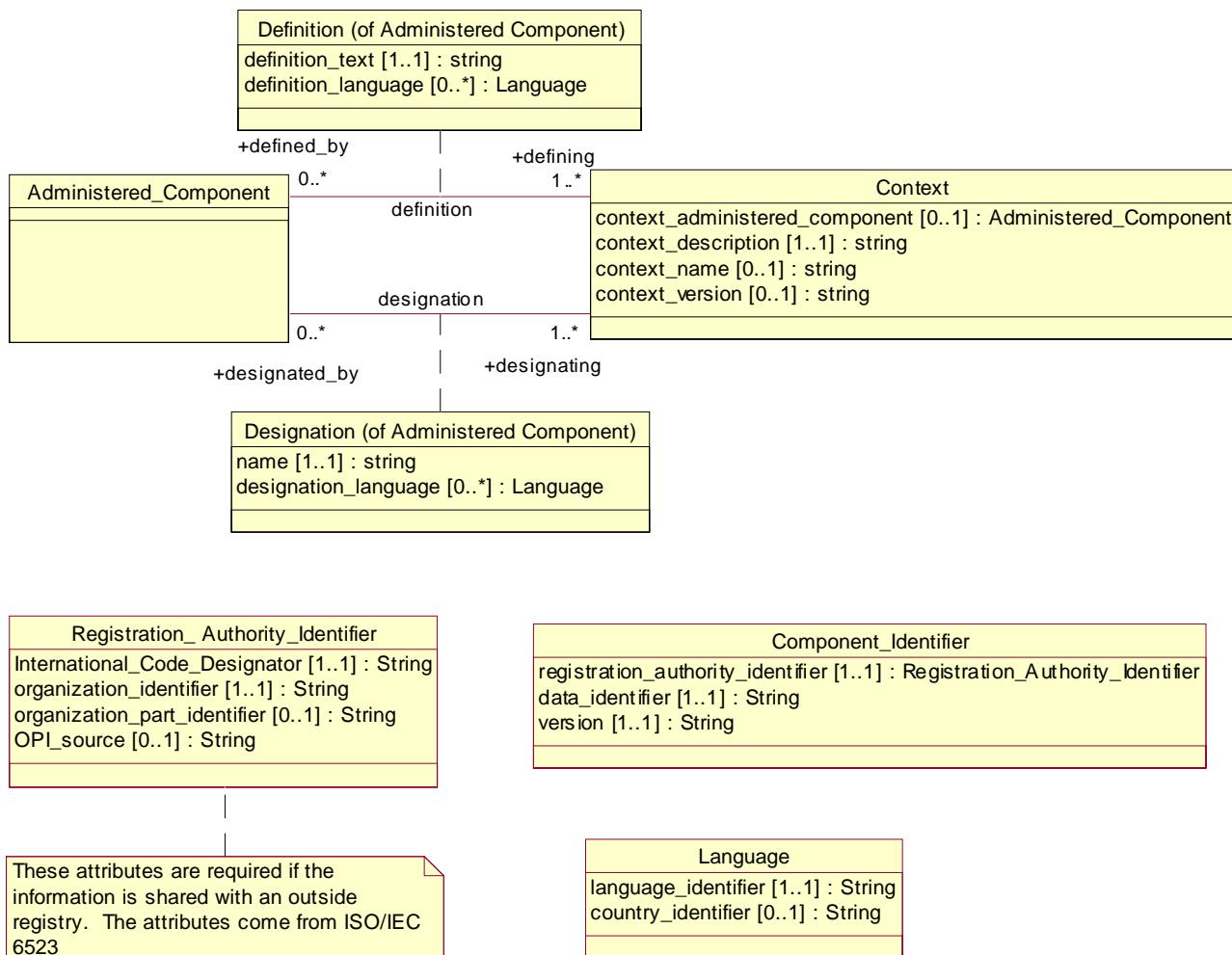


Figure 3: Naming and identification metamodel region

ISO/IEC 11179-5 *Naming and identification principles for data elements* provides rules and guidelines for naming and identification of administered components within a context.

### 4.6.1 Naming and identification metamodel region components

Each *administered component* has one or more names. Each name has a *context* (the system, database, file, data model, standard document, or other environment) in which the subject data has meaning and is known by that name. Names shall be unique within a context and within each subtype of *administered component* (e.g. the name of an *administered data component* and a *value domain* cannot be the same within the same *context*). A *context* may be used by many *administered components*.

#### 4.6.2 Naming and identification metamodel region attributes

The attributes in the Naming and Identification metamodel region shall be as follows:

<u>Attribute</u>	<u>Allowed Occurrences</u>
<i>Designation name</i>	One per <i>context</i>
<i>Context - description</i>	One or more per <i>administered component</i>
<i>Context - label</i>	One or more optional per <i>administered component</i>
<i>Context - language</i>	One or more per <i>administered component</i>
<i>Context - version</i>	One or more optional per <i>administered component</i>

## 4.7 Classification Region

The Classification region is used to manage *classification schemes* and the registry components that are in the classification schemes. *Administered components* may require classification under one or more schemes. The *classification scheme* may be a taxonomy, a network, an ontology, or any other system for systematizing where the categories are mutually exclusive. The classification may also be just a list of controlled vocabulary of property words (or terms). The list might be taken from the "leaf level" of taxonomy. Figure 4 represents the Classification region.

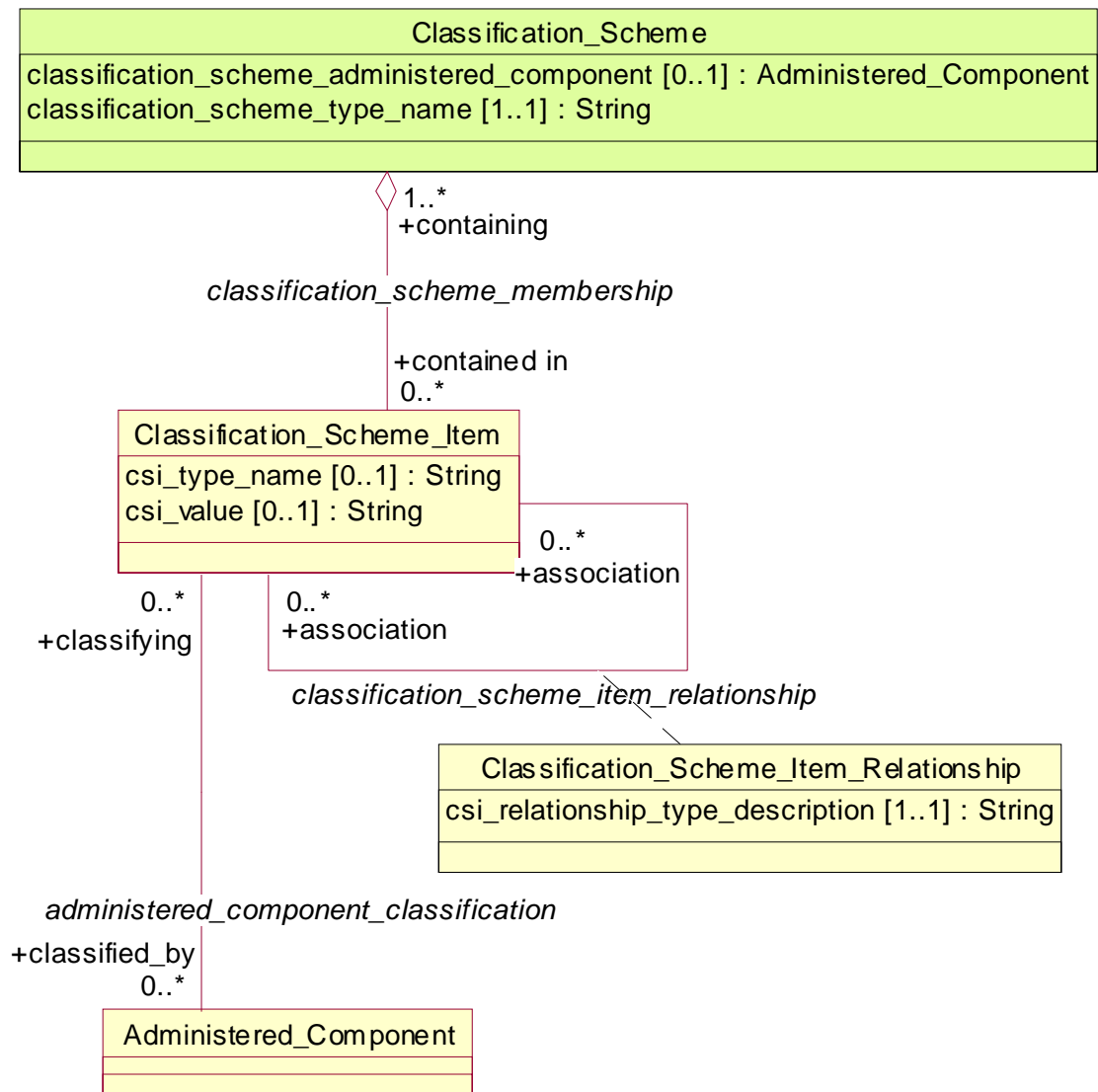


Figure 4: Classification metamodel region

### 4.7.1 Classification metamodel region components

A *classified component* is the descriptive information for a classified item in a classification scheme. An administered component may be classified in one or more classification schemes.

A *classified component* is a generalization for an *object class* (*concept* or *concept application association*), *property class*, *representation class*, *administered data component concept*, or *conceptual domain*. Since it is one of the subtypes of the *administered component*, the *classified component* is thus an *administered component* that is extended to have a definition and an optional *classification scheme*.

A *classified component* may be arranged in one or more *classification schemes*. *Classification schemes* may coexist and any *classified component* may have a different name in each one since they are each from a different *context*. Conversely, a *classified component* may exist in this structure without being associated with any other *classified component*. In essence, having a *classification scheme* is optional.

*Classification schemes* that associate one component to one or more others (*component classification association*) assist navigation through a large number of *object classes* or other *classified components*. *Classification schemes* with inheritance support understanding by contributing the definition of the one or more parent components.

Each subtype of *classified component* shall be a) comprehensive, b) have no implied representation, and c) provide enough categories to make classification useful.

#### 4.7.2 Classification metamodel region attributes

The attributes in the Classification metamodel region shall be as follows:

<u>Attribute</u>	<u>Allowed Occurrences</u>
<i>Classification scheme- type</i>	One for each <i>classification scheme</i>
<i>Component classification association - type description</i>	One for each <i>classified component</i> in a <i>classification scheme</i> where the components are associated with each other.
<i>Definition - context</i>	One for each <i>definition</i>
<i>Definition - text</i>	One for each <i>definition</i>

In addition, *classified component* and *classification scheme* shall have attributes inherited through *administered component*.

### 4.8 Administered data component Concept Region

The purpose of the administered data component concept region is to maintain the information on the concepts upon which the administered data components are developed. The components of this region concentrate on semantics. The concepts are independent of any internal or external physical representation. The major components of this region are concepts and properties, which are combined to form administered data component concepts.

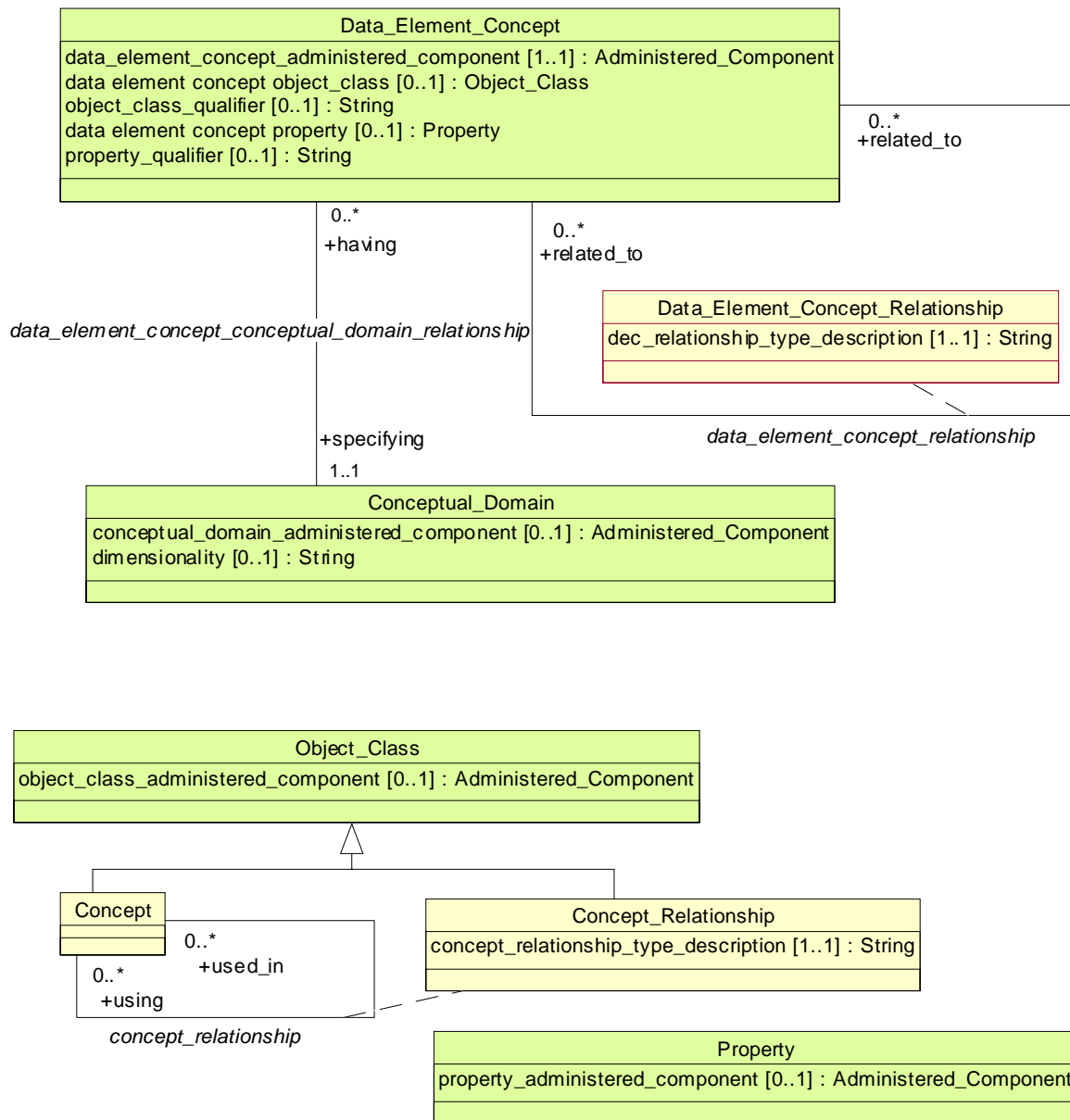


Figure 5: Administered data component concept metamodel region

#### 4.8.1 Object Class

An *object class* is 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. An *object class* may be a single unit of thought (i.e., *concept*) or a set of *concepts* in a relationship with each other to form a more complex concept (i.e., *concept application association*).

An *object class* is a subtype of a *classified component*, which is a subtype of an *administered component* and thus inherits its definition attribute and all of the relationships of both of these supertypes. A *concept* and a *concept application association* are subtypes of an *object class*.

As a *classified component*, an *object class* may exist within a *classification scheme*. An *object class* may reside in this structure but may not necessarily have a *property*.

#### 4.8.2 Property

A *property* is a characteristic common to all members of an object class. It may be any feature that humans naturally use to distinguish one individual object from another. It is the human perception of a single characteristic 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.

A *property* is a subtype of a *classified component*, which is a subtype of an *administered component* and thus inherits its definition attribute and all of the relationships of both of these supertypes.

As a *classified component*, a *property* may exist within a *classification scheme*. A *property* may reside in this structure but may not necessarily be associated with an *object class* or may not have a representational form (i.e., *value domain*) stored. To support a controlled vocabulary of property words, a *property* may exist in this structure without being associated with any other *property*. In other words, having a *classification scheme* of *properties* is optional.

#### 4.8.3 Administered data component concept

An *administered data component concept* is a concept that can be represented in the form of an administered data component, described independently of any particular representation. An *administered data component concept* shall have exactly one *object class* and one *property*. It cannot exist in this metadata registry without both. The union of a *property* and an *object class* provides significance beyond either that of the *property* or the *object class*. An *administered data component concept* thus has a *definition* independent from the *definition* of the *object class* or the *property*.

Through the *administered data component concept association*, an *administered data component concept* may be associated with one or more other *administered data component concepts* that it modifies, is modified by, or is otherwise linked.

A *administered data component concept* is a subtype of a *classified component*, which is a subtype of an administered component and thus inherits its definition attribute and all of the relationships of both of these supertypes.

As a *classified component*, an *administered data component concept* may exist within a *classification scheme*. An *administered data component concept* may reside in this structure but may not necessarily have an *administered data component* representation.

A *administered data component concept* shall have one *conceptual domain*. The *conceptual domain* is all permissible value meanings of an administered data component concept without a specified representation. The *conceptual domain* is detailed in the Conceptual and Value Domain Administration clause that follows this clause.

#### 4.8.4 Administered data component concept metamodel administration region attributes

The attributes in the Administered data component concept metamodel region shall be as follows:

<u>Attribute</u>	<u>Allowed Occurrences</u>
<i>Administered data component concept - object class qualifier</i>	One optional per <i>administered data component concept</i>
<i>Administered data component concept - property qualifier</i>	One optional per <i>administered data component concept</i>
<i>Administered data component concept association - type description</i>	From none to many per <i>administered data component concept</i>

In addition, *object class*, *property*, *concept*, *concept application association*, *administered data component concept* and *conceptual domain* shall have attributes inherited through *classified component*.

### 4.9 Conceptual and value domain Region

This region of the metamodel addresses the administration of conceptual domains and value domains. These domains can be viewed as logical code sets and physical code sets. Conceptual domains support administered data component concepts and value domains support administered data components.

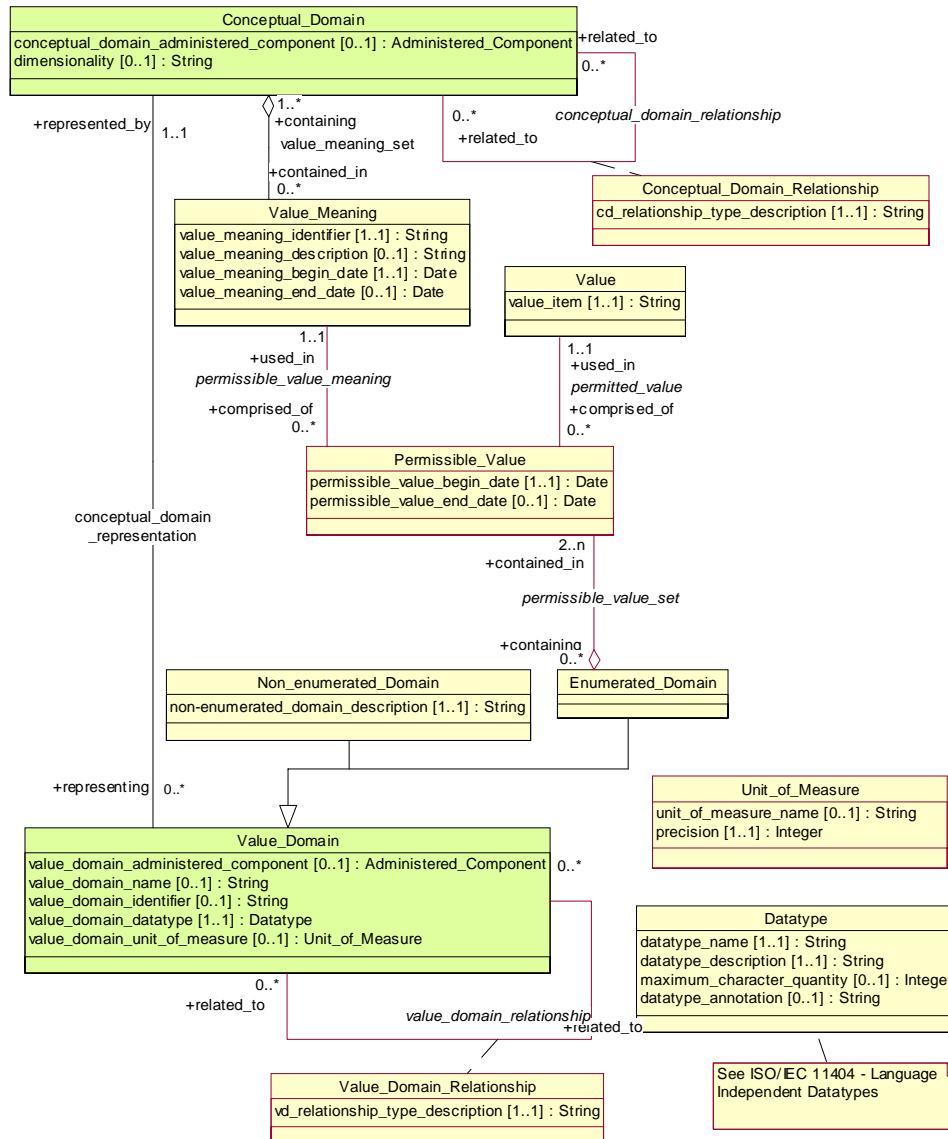


Figure 6: Conceptual and value domain metamodel region

#### 4.9.1 Conceptual domain

The *conceptual domain* is a set of value meanings of an administered data component concept, expressed without representation. An *administered data component concept* shall have one *conceptual domain*. A *conceptual domain* may be composed of other *conceptual domains* or may be a member (component) of a larger *conceptual domain*.



A *conceptual domain* is a subtype of a *classified component*, which is a subtype of an administered component and thus inherits its definition and administrative attributes and all of the relationships of both of these supertypes.

A *conceptual domain* sometimes contains a finite allowed inventory of notions that can be categorized. Each member has a *value meaning* that provides its distinction from other members. An example of a *conceptual domain* is the notion of countries that is specified in ISO standard 3166, Codes for the representation of names of countries. The notion of each country as specified would be *the value meanings*.

#### 4.9.2 Value domain

One of the key components of a representation is the *value domain*. A *value domain* provides representation, but has no implication as to what *administered data component concept* the values are associated nor what the values mean.

A *value domain* is associated with a *conceptual domain*. A *value domain* provides a representation for a *conceptual domain*. An example of a *conceptual domain* and a set of *value domains* is ISO standard 3166, Codes for the representation of names of countries. For instance, ISO 3166 describes the set of by five *value domains*: short name in English, official name in English, alpha-2 code, alpha-3 code, and numeric code.

A *value domain* may be dependent upon one or more other *value domains* that it modifies, is modified by, or is otherwise linked. In other words, the permissible values of one *value domain* may dependent upon the particular *value instance* of another *administered data component*.

A *value domain* is subtype of an *administered component*. It inherits the Administration attributes from the *administered component*. A *value domain* may be expressed as an *non-enumerated domain* such as a rule, a procedure, or a range (i.e., interval), or it may be expressed as an *enumerated domain*. Since these are subtypes of *value domain* they inherit all the relationships and attributes of the latter.

If it is an *enumerated domain*, the *value domain* shall have a listed set of two or more *permissible values*. Each *permissible value* is associated with a *value meaning*. A single *value meaning* may have more than one means of representation by *permissible values* — one from each appropriate *enumerated domain*.

A *value domain* is associated with 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, and/or symbols to depict values of an *administered data component* determined by the operations that may be performed on the *administered data component*. If meaningful, a *value domain* may also be associated with a *character set* — the collective symbols of a formalized writing system for a language used to intelligibly communicate data. In addition, if meaningful, a *value domain* may also be associated with a *unit of quantity* — a standard unit used when a representation class indicates that a *value domain* is a one of the representational forms of quantity.

#### 4.9.3 Conceptual and value domain administration metamodel region attributes

The attributes in the Conceptual and Value Domain Administration metamodel region shall be as follows:

<u>Attribute</u>	<u>Allowed Occurrences</u>
<i>character set - name</i>	One per <i>value domain</i> that has a character-based <i>datatype</i>
<i>datatype - description</i>	One per <i>value domain</i>
<i>datatype - label</i>	One per <i>value domain</i>
<i>non-enumerated domain - description</i>	One per <i>value domain</i> that is a <i>non-enumerated domain</i>
<i>permissible value - item</i>	One per each <i>permissible value</i> of a <i>value domain</i> that is an <i>enumerated domain</i>

<i>unit of quantity - name</i>	One per <i>value domain</i> with a quantity <i>representation class</i>
<i>unit of quantity - precision</i>	One per <i>value domain</i> with a quantity <i>representation class</i>
<i>value domain - format</i>	One optional per <i>value domain</i>
<i>value domain - label</i>	One optional per <i>value domain</i>
<i>value domain - maximum character quantity</i>	One per <i>value domain</i>
<i>value domain - minimum character quantity</i>	One optional per <i>value domain</i>
<i>value domain association- type description</i>	From none to many per <i>value domain</i>
<i>value meaning - description</i>	One optional per <i>value meaning</i>
<i>value meaning - identifier</i>	One per <i>value meaning</i>

### 4.10 Administered data component Region

The Administered data component metamodel region is used to address the administration of *administered data components*. *Administered data components* provide the formal representations for some information (such as a fact, a proposition, an observation, etc.) about some concrete or abstract thing. *Administered data components* are reusable and shareable representations of *administered data component concepts*.

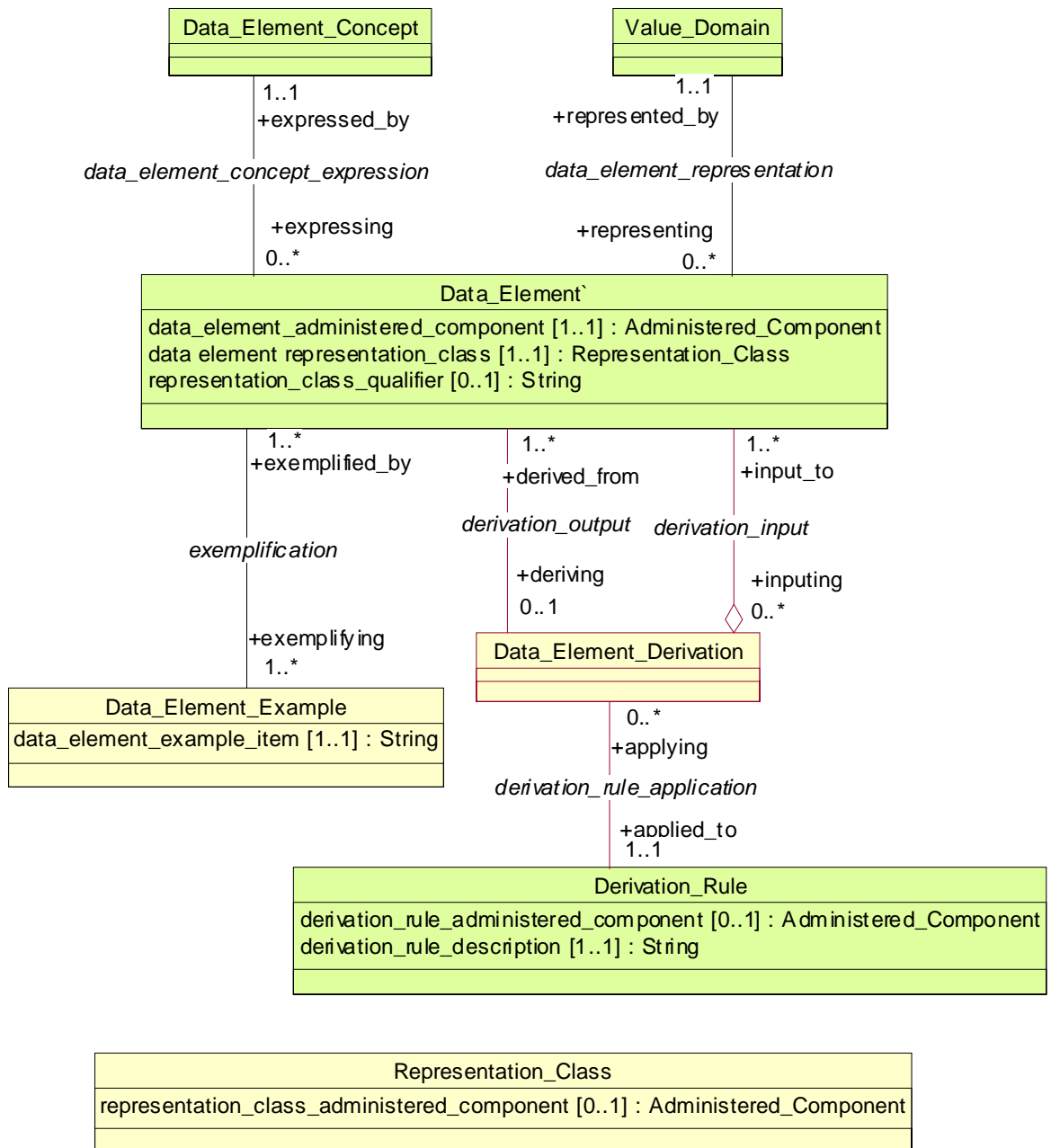


Figure 7: Administered data component administration metamodel region

#### 4.10.1 Administered data component administration metamodel region components

A *administered data component* 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. In the simplest way of thinking about it, an *administered data component* links a *value domain* with a meaning. From a different and more complete prospective, *administered data components* are formed when an *administered data component concept* is assigned a representation. One of the key components of a representation is the *value domain*, i.e., restricted permissible values. A *value domain* provides representation, but has no implication as to what *administered data component concept* the values are associated nor what the values mean. The association between an *administered data component concept*, a *representation class*, and a relevant *value domain* is an *administered data component*. (In general usage, the term *administered data component* and *administered data component type* are used interchangeably. In this standard, the shorter term *administered data component* is used.) The *administered data component concept* may be associated with several *value domains* resulting in a different *administered data component* for each association. A *value domain* may not reside in this structure without being the representation of an identified *administered data component concept*. Each *value domain* must be assigned to exactly one *representation class*. A *representation class* may reside in this structure without being associated with either an *administered data component concept* or a *value domain*.

A *administered data component* shall have an *administered data component concept*, and a *value domain*, and a *representation class*. It cannot exist in this metadata registry without these.

A *administered data component* has one or more *examples* that are used to provide representative samples of the *administered data component*. An *administered data component* may have a *rule* that is a specification of derivation for the *administered data component*. The *rule* may range from a simple operation such as subtraction to a very complex set of derivations (derivation being defined as a relationship between a rule and an input set upon which it acts). *Rules* are not limited to arithmetic and logical operations. A derivation

A *administered data component* and a *rule* are subtypes of an *administered component* and thus inherit all of the relationships and attributes from the *administered component*.

#### 4.10.2 Administered data component administration metamodel region attributes

The attributes in the Administered data component Representation metamodel region shall be as follows:

<u>Attribute</u>	<u>Allowed Occurrences</u>
<i>administered data component -representation class qualifier</i>	One optional per <i>administered data component</i>
<i>Example - item</i>	One or more per <i>administered data component</i>
<i>rule - description</i>	One per <i>rule</i>

#### 4.11 Metamodel specification

The high-level metamodel for the management of shareable data is shown in Figure 8. Clause 4.12 is the complete diagram for the metamodel. Clause 4.13 contains the specifications that for describing the entities, relationships, and attributes that comprises the metamodel. Clause 4.15 contains an alphabetical list of specifications of the metamodel objects.



#### 4.14 Metadata class, attribute, and relationship field description specification

<p>EDITOR'S NOTE</p> <p>Previous versions of this table included columns identifying whether the entry was for a Class, a Relationship, or an Attribute, and for Attributes, what is the Obligation / Condition associated with it. Are we happy to have deleted this?</p>
--

The following structure is used to describe the *classes, attributes, and relationships* of the metamodel.

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
The fully qualified term. Attribute names are preceded by the name of the class in which they appear. Class names and Relationship names are not qualified.	The unqualified name of the term.	A statement which describes a concept and permits its differentiation from other concepts within a system of concepts.	General notes or comments not recorded in other fields.

#### 4.15 Alphabetical list of metadata objects

The following table is sorted on the column "Term (without class)".

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Administered Component	Administered Component	A component for which administrative information is recorded.	
administered component classification	administered component classification	The relationship where an administered component is classified.	
Administered Component – administrative note	administrative note	Any general notes about the administered component.	
Administered Component – administrative status	administrative status	A designation of the position in the processing life-cycle of a registration authority for handling registration requests.	
Conceptual Domain Relationship - cd relationship type description	cd relationship type description	A description of the type of relationship between a conceptual domain and one or more other conceptual domains.	
Contact - cell phone number	cell phone number	A cell phone number for communication with the contact.	
Administered Component - change description	change description	The description of what has changed in the administered component since the prior version of the administered component.	

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Classification Scheme	Classification Scheme	The descriptive information for an arrangement or division of objects into groups based on characteristics, which the objects have in common.	For example, origin, composition, structure, application, function, etc.; See ISO/IEC 11179, Part 2.
Classification Scheme - classification scheme administered component	classification scheme administered component	The administered component information for a classification scheme.	
Classification Scheme Item	Classification Scheme Item	An item of content in a classification scheme.	Note: This may be a node in a taxonomy or ontology, a term in a thesaurus, etc.
Classification Scheme Item Relationship	Classification Scheme Item Relationship	The relationship among items in a classification scheme.	
classification scheme membership	classification scheme membership	The relationship of a classification scheme with its items.	
Classification Scheme - classification scheme type name	classification scheme type name	The name of the type of classification scheme.	
Component Identifier	Component Identifier	An identifier for a component.	
Concept	Concept	A unit of thought constituted through abstraction based on characteristics common to a set of objects.	NOTE: Concepts are not bound to a particular language. They are, however, influenced by the social or cultural background. See ISO 1087.
Concept Relationship	Concept Relationship	A semantic link between concepts.	
Concept Relationship - concept relationship type description	concept relationship type description	A description of the type of relationship between the concepts.	
Conceptual Domain	Conceptual Domain	A set of possible valid value meanings of an administered data component concept, expressed without representation.	
Conceptual Domain - conceptual domain administered component	conceptual domain administered component	The administered component information for a conceptual domain.	
Conceptual Domain Relationship	Conceptual Domain Relationship	A relationship between two conceptual domains.	
conceptual domain representation	conceptual domain representation	A relationship between a conceptual domain and a value domain.	

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Contact	Contact	An instance of a role of an individual or an organization (or organization part or organization person) to whom an information item(s), a material object(s) and/or person(s) can be sent to or from in a specified context.	
Contact - contact mail address	contact mail address	A description of where to send written correspondence to the contact.	EDITOR'S NOTE: Compare this definition with that of "organization mail address". See also note in sub-clause D.2.54 in Annex D.
Contact - contact name	contact name	The name of the contact.	
Contact - contact title	contact title	The name of the position held by the contact.	
Context (for administered component)	Context (for administered component)	A universe of discourse in which a name or definition is used.	
Context - context administered component	context administered component	The administered component information for a context.	
Context - context description	context description	The textual description of the context.	
Context - context name	context name	The context name.	
Context - context version	context version	The unique version identifier of a context.	
Language - country identifier	country identifier	A country identifier further specifying the geographic region associated with the language.	Note: Use the three digit numeric codes from ISO 3166, Part 1, with extensions, if required.
Administered Component - creation date	creation date	The date the administered component was created.	
Classification Scheme Item Relationship - csi relationship type description	csi relationship type description	A description of the type of relationship between a classification scheme item and one or more other classification scheme items in a classification scheme.	
Classification Scheme Item - csi type name	csi type name	The name of the type of the classification scheme item value.	
Classification Scheme Item - csi value	csi value	An instance of a classification scheme item.	
Administered data component	Administered data component	A unit of data for which the definition, identification, representation and permissible values are specified by means of a set of attributes.	



<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Administered data component - administered data component administered component	administered data component administered component	The administered component information for an administered data component.	
Administered data component Concept	Administered data component Concept	A concept that can be represented in the form of an administered data component, described independently of any particular representation.	
Administered data component Concept - administered data component concept administered component	administered data component concept administered component	The administered component information for an administered data component concept.	
administered data component concept conceptual domain relationship	administered data component concept conceptual domain relationship	The relationship between an administered data component concept and its conceptual domain.	
administered data component concept expression	administered data component concept expression	The relationship between an administered data component and an administered data component concept.	
Administered data component Concept - administered data component concept object class	administered data component concept object class	The designation of an object class for an administered data component concept	
Administered data component Concept - administered data component concept property	administered data component concept property	The designation of a property for an administered data component concept	
Administered data component Concept Relationship	Administered data component Concept Relationship	The attribution of a relationship of an administered data component concept with another administered data component concept.	
Administered data component Derivation	Administered data component Derivation	The relationship among a derived administered data component, the rule controlling its derivation, and the administered data component(s) from which it is derived.	
Administered data component Example	Administered data component Example	Representative illustration of the administered data component.	

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Administered data component Example - administered data component example item	administered data component example item	Actual illustrative case of the administered data component.	
administered data component representation	administered data component representation	The relationship between an administered data component and its value domain.	
Administered data component - administered data component representation class	administered data component representation class	The class of representation of an administered data component.	
Component Identifier - data identifier	data identifier	The unique identifier for an administered component within a registration authority.	
Datatype	Datatype	A set of distinct values, characterized by properties of those values and by operations on those values.	Needs further discussion; See ISO 11404.
Value Domain - datatype	datatype	The datatype used in a value domain.	Needs further discussion; See ISO 11404.
Datatype - datatype annotation	datatype annotation	EDITOR'S NOTE: To be added.	Needs further discussion; See ISO 11404.
Datatype - datatype description	datatype description	A description of the parameters of a datatype.	Needs further discussion; See ISO 11404.
Datatype - datatype name	datatype name	A designation for the category of datatype.	Needs further discussion; See ISO 11404.
Administered data component Concept Relationship - dec relationship type description	dec relationship type description	The description of the type of relationship with another administered data component concept that this administered data component concept modifies, is modified by, or is otherwise linked with.	
Definition (of Administered Component)	Definition (of Administered Component)	The definition of an administered component within a context.	
Definition (of Administered Component) - definition language	definition language	The identifier of the language used within the definition.	Note: Includes natural language and special languages, not computer languages.
Definition (of Administered Component) - definition text	definition text	The text of the definition.	
derivation input	derivation input	The relationship specifying the source administered data component(s) for an administered data component derivation.	

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
derivation output	derivation output	The relationship denoting the result of an administered data component derivation.	
Derivation Rule	Derivation Rule	The logical, mathematical, and/or other operations specifying derivation.	
Derivation Rule - derivation rule administered component	derivation rule administered component	The administered component information for a derivation rule.	
derivation rule application	derivation rule application	The relationship specifying the derivation rule for an administered data component derivation.	
Derivation Rule - derivation rule description	derivation rule description	The text of a specification of derivation.	
Designation (of administered component)	Designation (of administered component)	The designation of an administered component within a context.	
Designation (of administered component) - designation language	designation language	The identifier of the language used in the designation name.	Note: Includes natural language and special languages, not computer languages.
Conceptual Domain - dimensionality	dimensionality	The dimensionality for a conceptuality <sup>1</sup> .	For example, length, mass, velocity; Needs further discussion.
Registration Authority - documentation language	documentation language	The identifier of the language used for documentation by the registration authority.	
Administered Component - effective date	effective date	The date an administered component became/becomes available to registry users.	
Contact - electronic mail address	electronic mail address	An e-mail address for correspondence with the contact.	
Enumerated Domain	Enumerated Domain	A value domain that is specified by a list of all permissible values.	
exemplification	exemplification	A relationship between an administered data component example and its administered data component.	
Administered Component - explanatory comment	explanatory comment	Descriptive comments about the administered component.	

---

<sup>1</sup> EDITOR'S NOTE: Changed from "conceptionality", which is not a valid word in the English Language.

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Contact - fax number	fax number	A facsimile number for correspondence with the contact.	
Administered Component - identifier	identifier	A unique identifier for an administered component.	Note: If the data is to be shared with an outside registry, the identifier includes a registration authority identifier to make it globally unique. If the data is not to be shared externally, the registration authority portion of the identifier is not required.
Registration Authority Identifier - international code designator	international code designator	The identifier of an organization identification scheme.	See ISO 11179, Part 6.
Language	Language	The collection of identifiers required to identify a language or language variation for a particular purpose.	
Language - language identifier	language identifier	Information in a terminological entry which indicates the name of a language.	Note: Use the codes from ISO 639, Part 2.1998/T, with extensions if required.
Administered Component - last change date	last change date	The date the administered component was last changed.	
Datatype - maximum character quantity	maximum character quantity	The maximum number of storage units (of the corresponding datatype) to represent the administered data component value.	Needs further discussion; See ISO 11404.
Designation (of administered component) - name	name	A name by which an administered component is known within a specific context.  EDITOR'S NOTE: Should we replace the word "name" in this definition by either "designation" or "linguistic expression"? (If so, see also sub-clause D.2.56 in Annex D.) Compare definition of "organization name".	Note: The general term "name" is defined in sub-clause 3.1.15 as "The designation of an object by a linguistic expression".
Non-enumerated Domain	Non-enumerated Domain	A value domain that is not specified by a list of all permissible values.	
Non-enumerated domain - non-enumerated domain description	non-enumerated domain description	A description of a rule, reference, or range for a set of all permissible values for the value domain.	

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Object Class	Object Class	A set of ideas, abstractions, or things in the real world that can be identified with explicit boundaries and meaning and whose properties and behaviour follow the same rules.	
Object Class - object class administered component	object class administered component	The administered component information for an object class.	
Administered data component Concept - object class qualifier	object class qualifier	A qualifier of the administered data component concept object-class.	
Registration Authority Identifier - opi source	opi source	The source for the organization part identifier.	See ISO 11179, Part 6. Opi is the organization part identifier.
Organization	Organization	A unique framework of authority within which a person or persons act, or are designated to act, towards some purpose.	
Registration Authority Identifier - organization identifier	organization identifier	The identifier assigned to an organization within an organization identification scheme, and unique within that scheme.	.
Organization - organization mail address	organization mail address	The mailing address of the organization.	EDITOR'S NOTE: Compare this definition with that of "contact mail address". See also note in sub-clause D.2.54 in Annex D.
Organization - organization name	organization name	A designation for the organization.	
Registration Authority Identifier - organization part identifier	organization part identifier	An identifier allocated to a particular organization part.	See ISO 11179, Part 6. Also referred to as opi.
Administered Component - origin	origin	The source (document, project, discipline or model) that was the origin for the administered component.	
Contact - pager number	pager number	A pager number for the contact.	
Permissible Value	Permissible Value	An expression of a value meaning in a specific value domain.	

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Permissible Value - permissible value begin date	permissible value begin date	The date this value became/becomes permissible in the value domain.	A Registration Authority may determine whether this date is the date the value becomes valid in a registry or the date the value becomes part of the source domain.
Permissible Value - permissible value end date	permissible value end date	The date this value became/becomes no longer permissible in the value domain.	A Registration Authority may determine whether this date is the date the value is no longer valid in a registry or the date the value is no longer part of the source domain.
permissible value meaning	permissible value meaning	The relationship of a value meaning and a permissible value enumerated in a domain.	
permissible value set	permissible value set	The set of permissible values for an enumerated domain.	
permitted value	permitted value	The use of a value as a permissible value in an enumerated domain.	
Contact - phone number	phone number	A telephone number for verbal correspondence with the contact.	
Unit of Quantity - precision	precision	The degree of specificity for a unit of measure.	
Property	Property	A characteristic common to all members of an object class.	
Property - property administered component	property administered component	The administered component information for a property.	
Administered data component Concept - property qualifier	property qualifier	A qualifier of the administered data component concept property.	
reference	reference	The relationship between a reference document and an administered component.	
Reference Document	Reference Document	A document that provides pertinent details for consultation about a subject.	
Reference Document - reference document identifier	reference document identifier	An identifier for the reference document.	

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Reference Document - reference document language	reference document language	The natural language identifier used in the reference document.  EDITOR'S NOTE: Suggest changing this definition to read: " The identifier of the natural or special language used in the reference document."	
Reference Document - reference document title	reference document title	The title of the reference document.	
Reference Document - reference document type description	reference document type description	A description of the type of reference document.	
reference organization	reference organization	The relationship between a reference document and an organization.	
Registrar	Registrar	The representative of the Registration Authority.	
Registrar - registrar identifier	registrar identifier	A designator for the registrar.  EDITOR'S NOTE. Should we use "identifier" instead of "designator"? Compare definition of "registration authority identifier" below. (See also sub-clause D.2.78 in Annex D.)	
registration	registration	The relationship between an administered component and the Registration Authority.	
Registration Authority	Registration Authority	A body responsible for maintaining a registry.	
Component Identifier - registration authority identifier	registration authority identifier	An identifier for a registration authority. <sup>2</sup>	See ISO 11179, Part 6.
Registration Authority - registration authority identifier	registration authority identifier	An identifier assigned to a Registration Authority.	See ISO 11179, Part 6.
Registration Authority identifier	Registration Authority identifier	An identifier assigned to a Registration Authority.	See ISO 11179, Part 6.
registration authority registrar	registration authority registrar	The relationship between a registration authority and a registrar.	

---

<sup>2</sup> EDITOR'S NOTE: We should make this definition and the following two consistent. Which text is preferred? Part 6 uses the "assigned to" version.

<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Administered Component - registration status	registration status	A designation of the position in the registration life-cycle of an administered component.	
Representation Class	Representation Class	The classification of types of representations.	
Representation Class - representation class administered component	representation class administered component	The administered component information for a representation class.	
Administered data component - representation class qualifier	representation class qualifier	A representation class qualifier to the designation - name to accurately label the administered data component.	
Stewardship	Stewardship	The relationship of an administered component, a contact, and an organization involved in the stewardship of the metadata.	
Stewardship - stewardship contact	stewardship contact	The contact information for a steward.	
Submission	Submission	The relationship of an administered component, a contact, and an organization involved in a submission of metadata.	
Submission - submission contact	submission contact	The contact information for a submitter.	
Contact - telex number	telex number	A telex number for correspondence with the contact.	
Unit of Measure	Unit of Measure	A system of measurement.	
Unit of Quantity - unit of measure name	unit of measure name	The name of a unit of measure.	
Administered Component - unresolved issue	unresolved issue	Any problem that remains unresolved regarding proper documentation of the administered component.	
Administered Component - until date	until date	The date an administered component is no longer effective in the registry.	
Value	Value	A data value.	
Value Domain	Value Domain	A set of permissible values. It provides representation, but has no implication as to what administered data component concept the values may be associated with nor what the values mean.	



<u>Term (with Class)</u>	<u>Term (without Class)</u>	<u>Definition</u>	<u>Comments</u>
Value Domain - value domain administered component	value domain administered component	The administered component information for a value domain.	
Value Domain - value domain identifier	value domain identifier	The identifier of a value domain.	
Value Domain - value domain name	value domain name	The name of a value domain.	
value domain relationship	value domain relationship	A relationship between value domains.	
Value Domain - value domain unit of measure	value domain unit of measure	The unit of measure used in a value domain.	
Value - value item	value item	A representation of a value meaning in a specific value domain. The actual value.	
Value Meaning	Value Meaning	The meaning or semantic content of a value.	
Value Meaning - value meaning begin date	value meaning begin date	The effective date of this value meaning in the conceptual domain.	
Value Meaning - value meaning description	value meaning description	A description of a value meaning.	
Value Meaning - value meaning end date	value meaning end date	The date this value meaning became invalid.	
Value Meaning - value meaning identifier	value meaning identifier	The unique identifier for a value meaning.	
value meaning set	value meaning set	The relationship between a conceptual domain and a value meaning.	
Value Domain Relationship - vd relationship type description	vd relationship type description	The description of a value domain relationship.	
Component Identifier – version	version	The unique version identifier of the administered component.	

## 5 Conformance

### EDITOR'S NOTES

Although this clause looks long, it is fairly explicit about the types of conformance and the requirements for each type of conformance. There are 8 types of conformance. Many applications will claim more than one type of conformance. Metadata instances will want to claim conformance. The XML bindings are expected to become a

"conditionally normative" annex. It might be possible to reduce the size of the conformance wording, but we have tried to avoid "clever" wording that might be misinterpreted. The phrase "MDR3 administered data component attributes" is used. There might be a shorter phrase that we can all agree upon. This wording provides a firm conformance statement, while addressing the business and institutional realities of the marketplace. ]

Text has been added to reflect that conformance is affected by the in the context of roles and responsibilities of registration authorities, addressed in Part 6 of this standard.

Numbers of references to "data", other than "administered data component" have been replaced by "metadata", so as to reduce confusion between the administration of data, and that of metadata, the latter being the focus of this part of this standard.

The definition of "MDR3 writer" has been loosened to encompass applications for manual entry of metadata.

At first glance, the first sentence of this clause may seem unnecessary ... however, once the remaining sentences are added, the first sentence becomes necessary. Data sets, and hence metadata sets, don't exhibit "behavior", but implementations that store, retrieve, interpret, and allow the production of data sets all exhibit "behavior" ... thus, the notions of "undefined", "implementation-defined", and "unspecified" behaviours are necessary.]

In this Standard, "shall" is to be interpreted as a requirement on an implementation; conversely, "shall not" is to be interpreted as a prohibition. If a "shall" or "shall not" requirement is violated, the behavior is undefined. Undefined behavior is otherwise indicated in this Standard by the words "undefined behavior" or by the omission of any explicit definition of behavior. There is no difference in emphasis among these three; they all describe "behavior that is undefined".

EDITOR'S NOTE: The following statement has been inserted based on discussion of Issue 62.

This part of this standard prescribes a conceptual model, not a physical implementation. Therefore, the metamodel need not be physically implemented exactly as specified. However, it must be possible to unambiguously map between the implementation and the metamodel in both directions.

## 5.1 Conformance level

The following sub-clauses define strictly conforming implementations and conforming implementations. In the context of conformance, the terms "support", "use", "test", "access", and "probe" are defined in sub-clauses 5.5 [Application conformance] and 5.4 [MDR3 metadata instance conformance].

### 5.1.1 Caveat

#### EDITOR'S NOTE

This caveat has been included to reference the section dealing with registration authorities.

Conformance needs to be considered in the context of the roles and responsibilities of registration authorities, as covered by Part 6: Registration of data elements.

Extended conformance of systems requires formalisation of procedures, agreement of roles and responsibilities between parties, and guidelines addressing use of software products and conversions from other systems. The formalisation of these aspects must be consistent with the conformance requirements in the following clauses, and roles of registration authorities as set out in Part 6.

### 5.1.2 Rationale

EDITOR'S NOTE

This rationale has been included to help build consensus. We will need to decide whether to remove this wording towards the final stages of completion.

The distinction between "strictly conforming" and "conforming" implementations is necessary for addressing the simultaneous need for interoperability and the need for extensions. Extensions are motivated by needs of users, vendors, institutions, and industries.

### 5.1.3 Strictly conforming implementations

A strictly conforming implementation shall be at least one of: a strictly conforming application, a strictly conforming metadata set, a strictly conforming coding, a strictly conforming API, a strictly conforming protocol.

EDITOR'S NOTE

We don't have APIs and protocols yet, but we might later on. We are likely to have encodings : e.g. the XML encoding in Annex E.

A strictly conforming implementation

- 1) shall support all mandatory and optional MDR3 administered data component attributes;
- 2) shall not use, test, access, or probe for any extension features nor extensions to MDR3 administered data component attributes;
- 3) shall not exceed limits nor minimum-maxima values specified by this Standard; and
- 4) shall not interpret nor allow the production of administered data component attributes that are dependent on any unspecified, undefined, or implementation-defined behavior.

Note: The use extensions to MDR3 administered data component attributes may cause undefined behavior.

### 5.1.4 Conforming implementations

A conforming implementation shall be at least one of: a conforming application, a conforming metadata set, a conforming coding, a conforming API, a conforming protocol.

EDITOR'S NOTE

We don't have APIs and protocols yet, but we might later on. We are likely to have encodings : e.g. the XML encoding in Annex E.

A conforming implementation

- 1) shall support all mandatory and optional MDR3 administered data component attributes;
- 2) as permitted by the implementation, may use, test, access, or probe for extension features or extensions to MD3 administered data component attributes;

- 3) as permitted by the implementation, may exceed limits or minimum-maxima values specified by this Standard; and
- 4) may interpret or allow the production of administered data component attributes that are dependent on implementation-defined behavior.

Note 1: All strictly conforming implementations are also conforming implementations.

Note 2: The use extensions to MDR3 administered data component attributes may cause undefined behavior.

## 5.2 Summary of conformance labels

The following is an informative summary of the possible implementation conformance statements:

### 5.2.1 Strictly conforming MDR3 metadata instance

All mandatory MDR3 administered data component attributes shall exist; some optional MDR3 administered data component attributes may exist; extended MDR3 administered data component attributes shall not exist.

### 5.2.2 Conforming MDR3 metadata instance

All mandatory MDR3 administered data component attributes shall exist; some optional MDR3 administered data component attributes may exist; some extended MDR3 administered data component attributes may exist.

### 5.2.3 Strictly conforming MDR3 registry

Shall support storing/retrieving all mandatory MDR3 administered data component attributes, shall support storing/retrieving all optional MDR3 administered data component attributes, and data interchange applications shall not attempt to store/retrieve extended MDR3 administered data component attributes.

### 5.2.4 Conforming MDR3 registry

Shall support storing/retrieving all mandatory MDR3 administered data component attributes, shall support storing/retrieving all optional MDR3 administered data component attributes, and may support storing/retrieving some extended MDR3 administered data component attributes.

### 5.2.5 Strictly conforming MDR3 writer

Only mandatory and optional MDR3 administered data component attributes are produced, but no extended MDR3 administered data component attributes are produced.

### 5.2.6 Conforming MDR3 writer

Mandatory and optional MDR3 administered data component attributes are produced, and some extended MDR3 administered data component attributes may be produced.

### 5.2.7 Strictly conforming MDR3 reader

Only mandatory and optional MDR3 administered data component attributes are interpreted, but no extended MDR3 administered data component attributes are interpreted. Note: Supplying extended MDR3 administered data component attributes to a strictly conforming interpreter produces undefined behavior.

### 5.2.8 Conforming MDR3 reader

Mandatory and optional MDR3 administered data component attributes are interpreted and some extended MDR3 administered data component attributes may be interpreted. Interpreting extended MDR3 administered data component attributes is undefined behavior (note: not just "unspecified" behavior, but "undefined" behavior). This behavior ("undefined" behavior) may be relaxed (e.g., becomes "implementation-defined" behavior), depending upon the binding.

Note: An implementation may have more than one implementation conformance statement.

## 5.3 MDR3 metadata set conformance

Metadata set conformance is independent of binding.

### 5.3.1 Strictly conforming metadata set

A strictly conforming metadata set shall be a set of MDR3 administered data component attributes that:

- 1) is structured independent of binding;
- 2) strictly conforms to the functionality, conceptual model, and semantics of this Standard;
- 3) shall include all mandatory MDR3 administered data component attributes;
- 4) may include optional MDR3 administered data component attributes;
- 5) shall not include extended MDR3 administered data component attributes.

### 5.3.2 Conforming metadata set

A conforming metadata set shall be a set of MDR3 administered data component attributes that:

- 1) is structured independent of binding;
- 2) conforms to the functionality, conceptual model, and semantics of this Standard;
- 3) shall include all mandatory MDR3 administered data component attributes;
- 4) may include optional MDR3 administered data component attributes;
- 5) may include extended MDR3 administered data component attributes.

## 5.4 MDR3 metadata instance conformance

### 5.4.1 Strictly conforming metadata instance

A strictly conforming metadata instance shall:

- 1) be a strictly conforming metadata set, and
- 2) strictly conform to at least one MDR3 coding binding.

### 5.4.2 conforming metadata instance

A conforming metadata instance shall:

- 1) be a conforming metadata set, and
- 2) conform to at least one MDR3 coding binding.

Note: The difference between a strictly conforming/conforming metadata set, a coding, and a strictly conforming/conforming metadata instance is:

- 1) a metadata set is an instance of metadata independent of binding,
- 2) a coding can refer to an instance of metadata, a set of instances of data, or a syntax of instances of data,
- 3) a strictly conforming/conforming metadata instance is associated with a specific binding.

### 5.4.3 Rationale

In addition to the three application conformance perspectives (MDR3 registry, MDR3 reader, MDR3 writer), there is a fourth perspective on conformance: the metadata instance. Users will want to claim conformance for particular metadata instances ("My MDR3 information conforms to the Standard").

However, there is no fifth conformance perspective resulting from metadata sets. It is not possible to perform conformity assessment on metadata that is independent of binding, e.g., conformance testing requires that the metadata set to be rendered in some binding.

## 5.5 Application conformance

Application conformance is measured by how well the application behaves according to this Standard. There are two types of application conformance: strictly conforming and conforming.

### 5.5.1 Strictly conforming application

For all strictly conforming applications,

- Mandatory (required) features shall exist (or shall be available) and they shall conform to this Standard.
- Optional features may exist (or may be available) and, if they exist (or are available), they shall conform to this Standard.
- Extended features shall not be directly used and they shall not be tested for existence nor availability.

Note: A strictly conforming application may be minimally conforming but is maximally interoperable with respect to this Standard. Strict conformance concerns (1) the assessment, measurement; and/or availability of a minimal set of features; (2) the application's non-use of feature-probing; (3) the application's non-use of extended feature sets.

### 5.5.2 Conforming application

For all conforming applications,

- Mandatory (required) features shall exist (or shall be available) and they shall conform to this Standard.
- Optional features may exist (or may be available) and, if they exist (or are available), they shall conform to this Standard.

- Extended features may exist (or may be available), they may be tested for existence (or availability), and their use and behavior is implementation-defined.

Note 1: A conforming application may be more useful, but may be less interoperable with respect to this Standard. Conformance concerns (1) the assessment, measurement, and/or availability of a minimal set of features; (2) feature-probing for and/or prior agreement to the existence (or availability) of extended features, as permitted by the implementation; and (3) extended features.

Note 2: Although all strictly conforming applications are also conforming applications, the conformity assessment of strictly conforming applications may differ from the conformity assessment of conforming applications. Thus, the requirement that application must be both "a strictly conforming application" and "a conforming application", is a stronger requirement than the individual requirements of "a strictly conforming application" and "a conforming application", i.e., from the perspective of conformity assessment, an application may be "strictly conforming", "conforming", both, or neither.

## 5.6 Application varieties

There are three types of strictly conforming/conforming applications: MDR3 registry, MDR3 writer, MDR3 reader.

### 5.6.1 Rationale

There are four separate application conformance perspectives: MDR3 reader, MDR3 writer, MDR3 registry, MDR3 instance. Three of them (reader, writer, registry) are applications:

- Vendors or institutions that create MDR3 repositories will want to claim conformance: "My metadata registry conforms to the Standard".
- Vendors will want to claim conformance for their tools. For example, applications that read/interpret MDR3 information, the vendor will want to claim "My application is a conforming MDR3 reader". Similarly, for applications that write/allow the production of MDR3 information, the vendor will want to claim "My application is a conforming MDR3 writer".

The fourth conformance perspective, a MDR3 instance, is a coding but not an application. See 5.4 MDR3 Metadata Instance Conformance, above.

### 5.6.2 MDR3 registry

An MDR3 registry is an application that stores and retrieves metadata.

A strictly conforming MDR3 registry shall

- 1) receive MDR3 administered data component attributes for subsequent retrieval;
- 2) use strictly conforming interpretation for receiving MDR3 administered data component attributes;
- 3) store MDR3 administered data component attributes in persistent storage such that extensions to MDR3 administered data component attributes may not persist;
- 4) send, on request, previously stored MDR3 administered data component attributes;
- 5) use conforming metadata generation for sending MDR3 administered data component attributes;
- 6) strictly conform to at least one MDR3 coding binding, and at least one MDR3 API binding or MDR3 protocol binding.

Note 1: A strictly conforming MDR3 registry does not require "preservation" of extensions to MDR3 administered data component attributes, i.e., metadata interchange should not be dependent upon the expectation that extensions to MDR3 administered data component attributes would persist in a strictly conforming data registry.

Note 2: The storage or retrieval of extensions to MDR3 administered data component attributes may cause undefined behavior.

A conforming MDR3 registry shall

- 1) receive MDR3 administered data component attributes for subsequent retrieval;
- 2) use conforming interpretation for receiving MDR3 administered data component attributes;
- 3) store MDR3 administered data component attributes in persistent storage such that extensions to MDR3 administered data component attributes may persist;
- 4) send, on request, previously stored MDR3 administered data component attributes;
- 5) use conforming metadata generation for sending metadata;
- 6) conform to at least one MDR3 coding binding, and at least MDR3 API binding or MDR3 protocol binding.

Note 2: Upon storage, a conforming data registry may add, delete, or change extensions to MDR3 administered data component attributes for subsequent retrieval.

Note 3: A conforming MDR3 registry may store some extensions to MDR3 administered data component attributes, but it is not required to store all extensions to MDR3 administered data component attributes.

Note 4: The storage or retrieval of extensions to MDR3 administered data component attributes may cause undefined behavior.

### **5.6.3 MDR3 writer**

An MDR3 writer is an application that allows the production of MDR3 administered data component attributes. It may allow manual entry of metadata, or may automatically generate and produce it, such as by analysis of a database or a data based application.

A strictly conforming MDR3 writer application shall allow the production of MDR3 administered data component attributes that strictly conform to (1) this Standard, and (2) at least one binding of this Standard.

Note 1: A strictly conforming MDR3 writer does not allow the production of metadata extensions.

A conforming MDR3 writer application shall allow the production of MDR3 administered data component attributes that conform to (1) this Standard, and (2) at least one binding of this Standard.

Note 2: A conforming MDR3 writer may allow the production of extensions to MDR3 administered data component attributes.

### **5.6.4 MDR3 reader**

An MDR3 reader is an application that consumes and interprets MDR3 administered data component attributes.

A strictly conforming MDR3 reader application shall interpret MDR3 administered data component attributes that strictly conform to (1) this Standard, and (2) at least one binding of this Standard.

Note 1: A strictly conforming MDR3 reader does not interpret extensions to MDR3 administered data component attributes.



Note 2: Depending upon the binding of this Standard, a strictly conforming metadata reader may "ignore" extensions to MDR3 administered data component attributes, e.g.,

- a strictly conforming metadata reader might "consume" extensions to MDR3 administered data component attributes but the MDR3 reader is able ignore (not "interpret") these extensions. See definitions of "consume" and "interpret".
- a conforming MDR3 reader application shall interpret MDR3 administered data component attributes that conform to (1) this Standard, and (2) at least one binding of this Standard.

Note 3: A conforming MDR3 reader may, to the extent permitted by the implementation, interpret extensions to MDR3 administered data component attributes.



## Annex A (Normative) Modelling Notation

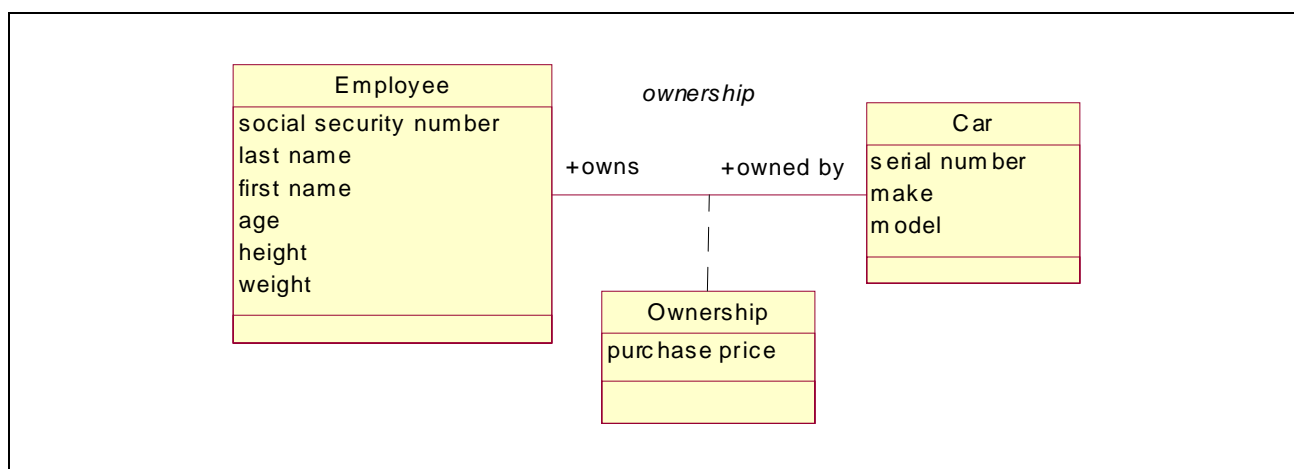
**Note** –In this normative clause of this standard, the metamodel for the management of shareable data was expressed in the Unified Modeling Language (UML).. The informative annexes immediately following the normative ones address alternative modelling notations, and are provided so that readers may see the metamodel in representations that they may use. The alternative metamodel representations are in:

<b>IDEF1X</b>	a entity relational Modeling paradigm,
<b>ORM or NIAM</b>	a natural language paradigm, and
<b>XML</b>	a data description and interchange language paradigm.

### A.1 Modelling symbols

The Unified Modeling Language (UML) is used to describe this metamodel. This notation is particularly suited to documenting a conceptual data schema. The structure used in the description (the metamodel) is compatible and is partially described by the metamodel described. Since this is a *conceptual data model*, only the *administered data component concept* information is used. A more complete description of UML may be found in ISO CD 15046-3, Geographic information - Part 3: Conceptual schema language, 1999-07-23.

The object model provides for four basic types of Modeling objects: *classes* (shown as rectangles in a diagram), *associations* among these classes (shown as lines), *operations*, and *attributes* that are associated to a class -. For example, the attributes describing employees and the cars they own could be modelled as follows:

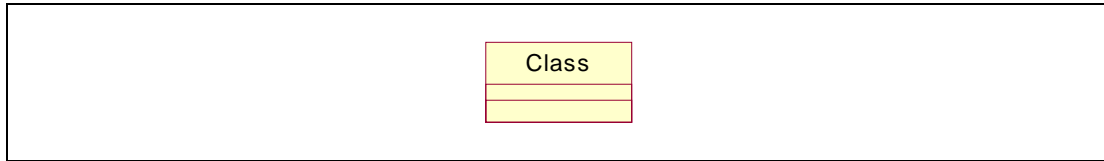


**Figure A-1: Sample modelling diagram**

At the current time operations are not shown.

#### A.1.1 Classes

Classes (Entities) are represented by rectangles and are the things about which the business processes information. A class is something that has distinctiveness and typically, properties of its own. Classes may be persons, places, concepts, events, or other fundamental things. For example, employee and car are both classes. The name of the class appears in the top area of the class rectangle.

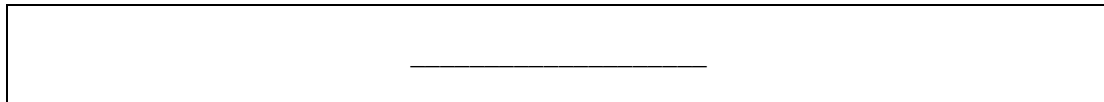


**Figure A-2: Class modelling representation**

As described below, entities may have attributes.

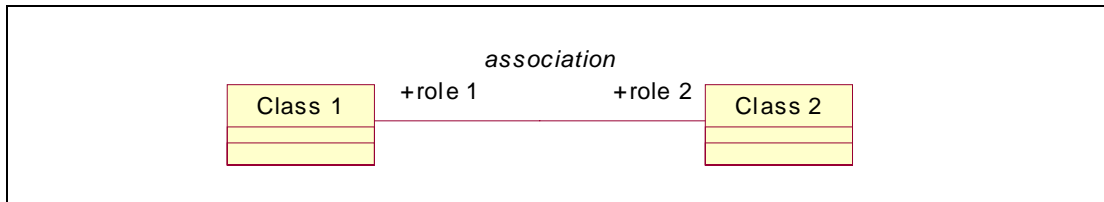
**A.1.2 Associations**

Classes have business-based *associations (relationships)* with one or more other classes. These associations are bi-directional. For example, an employee owns a car. Conversely the car is owned by the employee. This association between car and employee may be called ownership. In the UML diagram these links between classes are represented by lines.



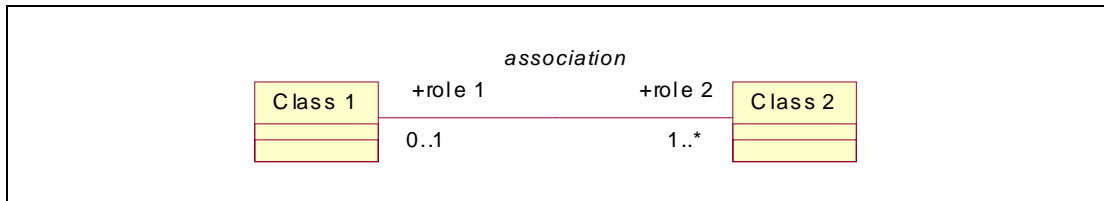
**Figure A-3: Association modelling representation**

Lines are drawn between the rectangles to identify the classes participating in an association. The role of each class in the association is specified along the association line.



**Figure A-4: Class relationship modelling representation**

The statement of the number of class instances that may participate at each end of an association is its *cardinality*. It is expressed as a minimum and a maximum, separated by a "..".



**Figure A-5: Class relationship cardinality modelling representation**

This example states that class 1 may be associated with a minimum of one and a maximum of many occurrences of class 2. Class 2 may be associated with a minimum of zero (i.e., the association is optional) and a maximum of one occurrence of class 1.

As described below, like classes, association classes may also have attributes.

### A.1.3 Associative entity

There are situations where a particular business object fits the criteria for both an class and an association. This usually occurs when an association itself is found to have a business association with an entity. Thus, an *associative class* is defined as a association that acts like an class. It is represented as a class that is connected to an association with a dashed line.

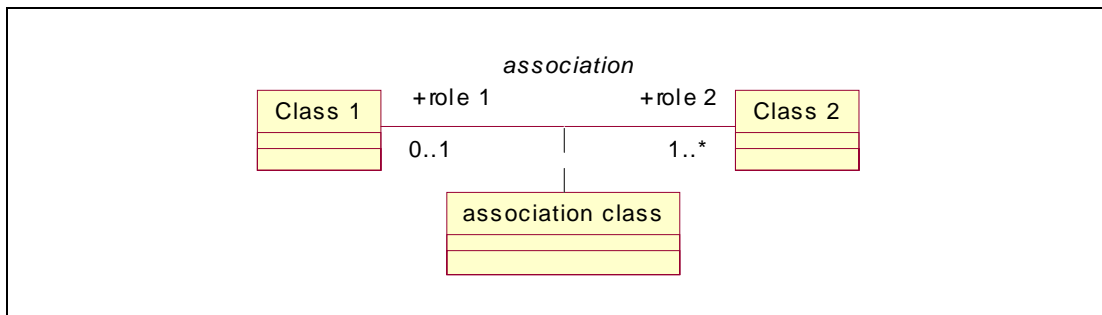


Figure A-6: Associative class modelling representation

### A.1.4 Subtypes

Classes may be decomposed into a hierarchy with increasing level of details. Each *supertype* entity may have *subtypes*. This is a generalization/specialization association.

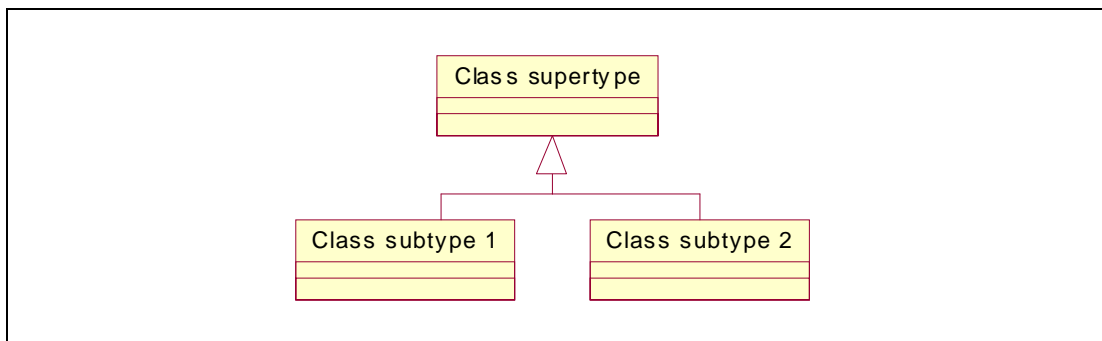


Figure A-7: Subtype modelling representation

Subtypes inherit all the attributes of their supertype. In addition to the attributes that are inherited, each subtype may have unique attributes of its own. In addition, subtypes participate in all the associations in which their supertype participates.

### A.1.5 Aggregation

A class sometimes consists of component classes. When a class consists of assemblies of “parts”, there is a special relationship between the component parts and the class representing the assembly of these parts. An aggregation is represented by a hollow diamond at the end of the association line. The tip of the diamond points toward the assembly entity.

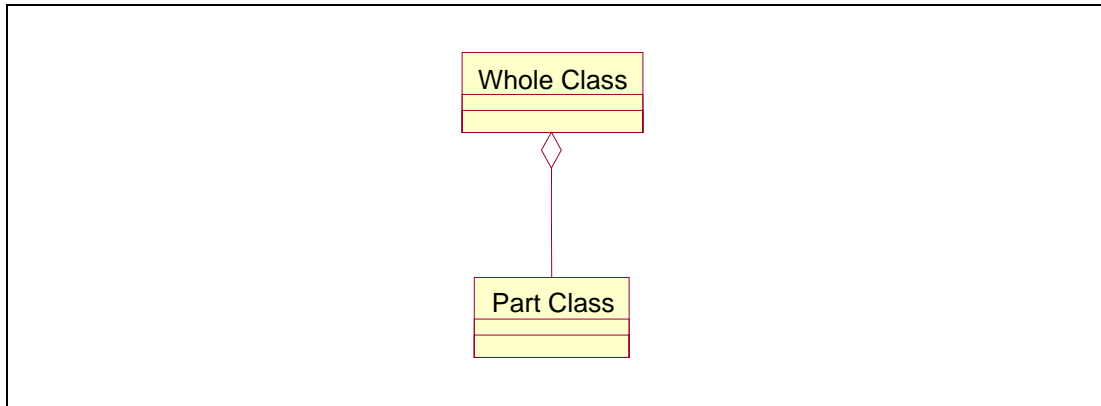


Figure A-8: Aggregation modelling representation

### A.1.6 Composite Aggregation

A class sometimes consists of component classes where the part classes cannot exist without the whole class. When a class consists of assemblies of “parts”, there is a special relationship between the component parts and the class representing the assembly of these parts. A composite aggregation is represented by a solid diamond at the end of the association line. The tip of the diamond points toward the assembly entity.

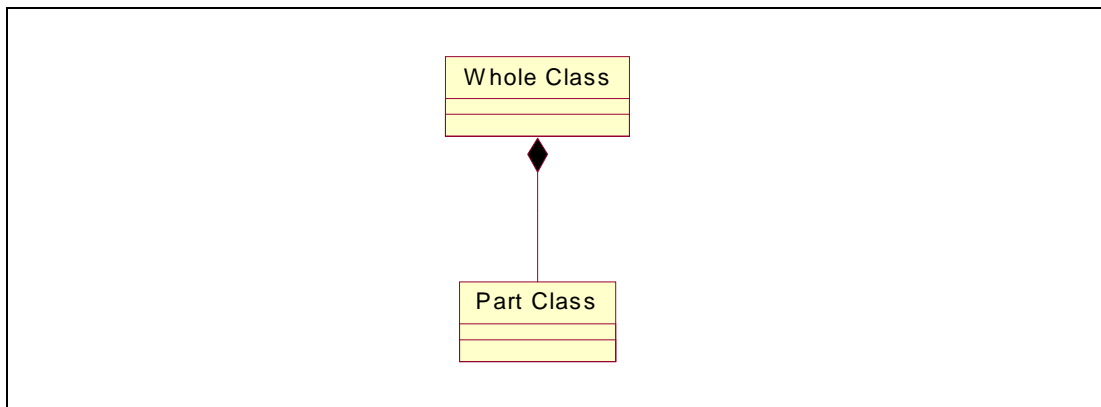
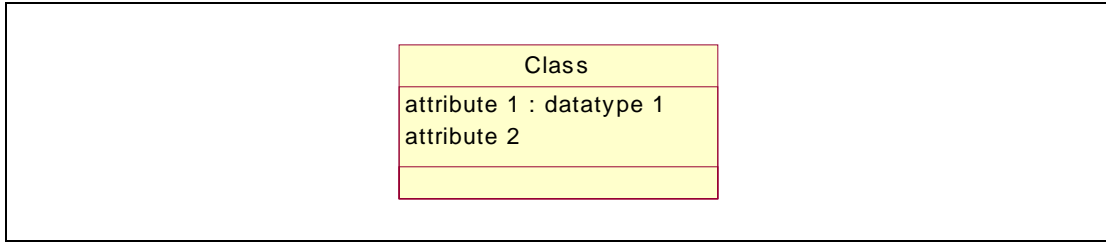


Figure A-9: Composite Aggregation modelling representation

### A.1.7 Attribute

The *properties* (or characteristics) of a class or an association class are described as *attributes*. Each attribute represents one *fact*. Alone among the other information Modeling objects, attributes have actual *permissible values*. In this document, we display attribute names in the middle area of the class. An attribute may have a datatype that is shown following the colon after the attribute name. A datatype may be a class in itself.



**Figure A-10: Class-attribute modelling representation**





## **Annex B (Normative) IDL representation of the metamodel**

[Editor's Note: This is to be developed after CD ballot as it is not ready. It will be generated by Rational Rose from the UML model.]



## Annex C (Informative) IDEF1X alternative representation model

**(Editor's Note –The figures in the following *informative IDEF1X alternative representation model Annexes will be re-generated when the normative UML model is approved.*)**

In the normative clauses of this standard, the metamodel for the management of shareable data was expressed in the Unified Modeling Language (UML).. The informative clauses in this and the immediately following annexes are provided so that readers may see the metamodel in representations that they may use. The metamodel has been represented in:

**IDEF1X** an entity-relational Modeling paradigm,  
**ORM or NIAM** a natural language paradigm (Annex D), and  
**XML** a data description and interchange language paradigm (Annex E).

The following diagrams represent the metamodel in IDEF1X notation. The definitions of the model components are the same as the components in Clause 8. The metamodel is represented in five diagrams. These diagrams are:

1. IDEF1X high-level metamodel
2. IDEF1X administration metamodel region,
3. IDEF1X naming metamodel region,
4. IDEF1X classification metamodel region,
5. IDEF1X conceptual domain and value domain administration region,
6. IDEF1X administered data component administration region, and
7. IDEF1X administered data component concept administration region.

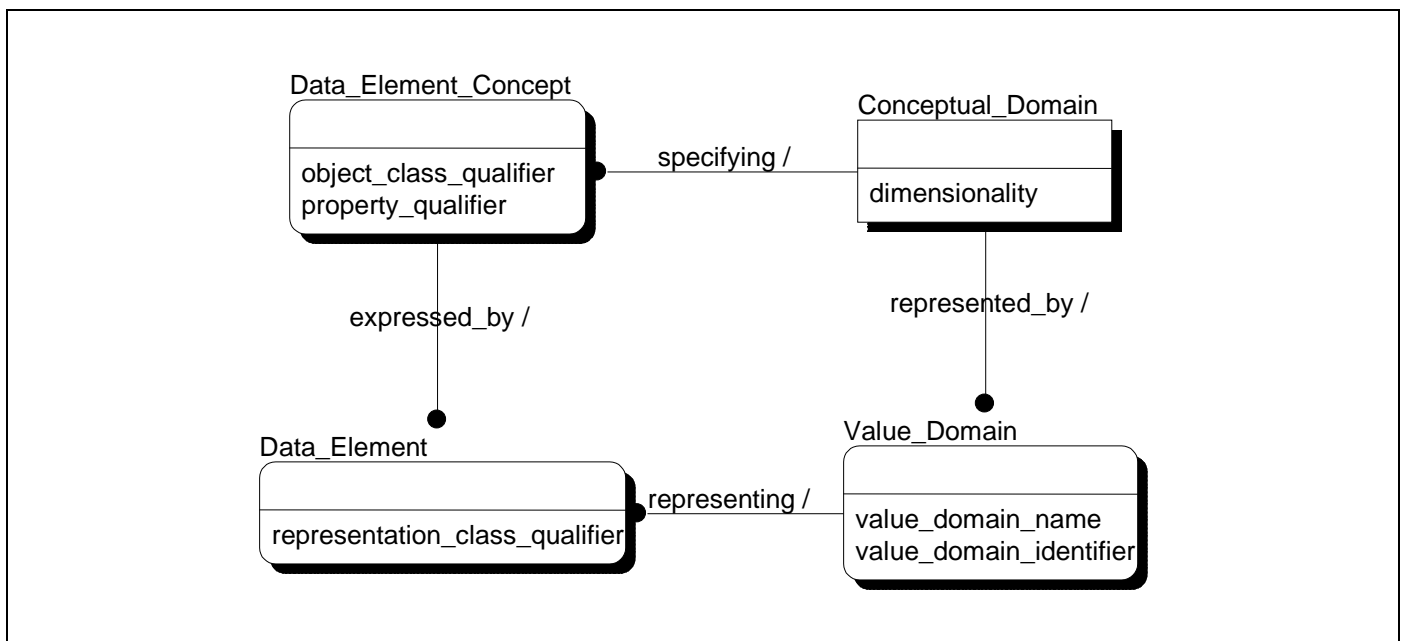


Figure C-1 – IDEF1X high-level metamodel

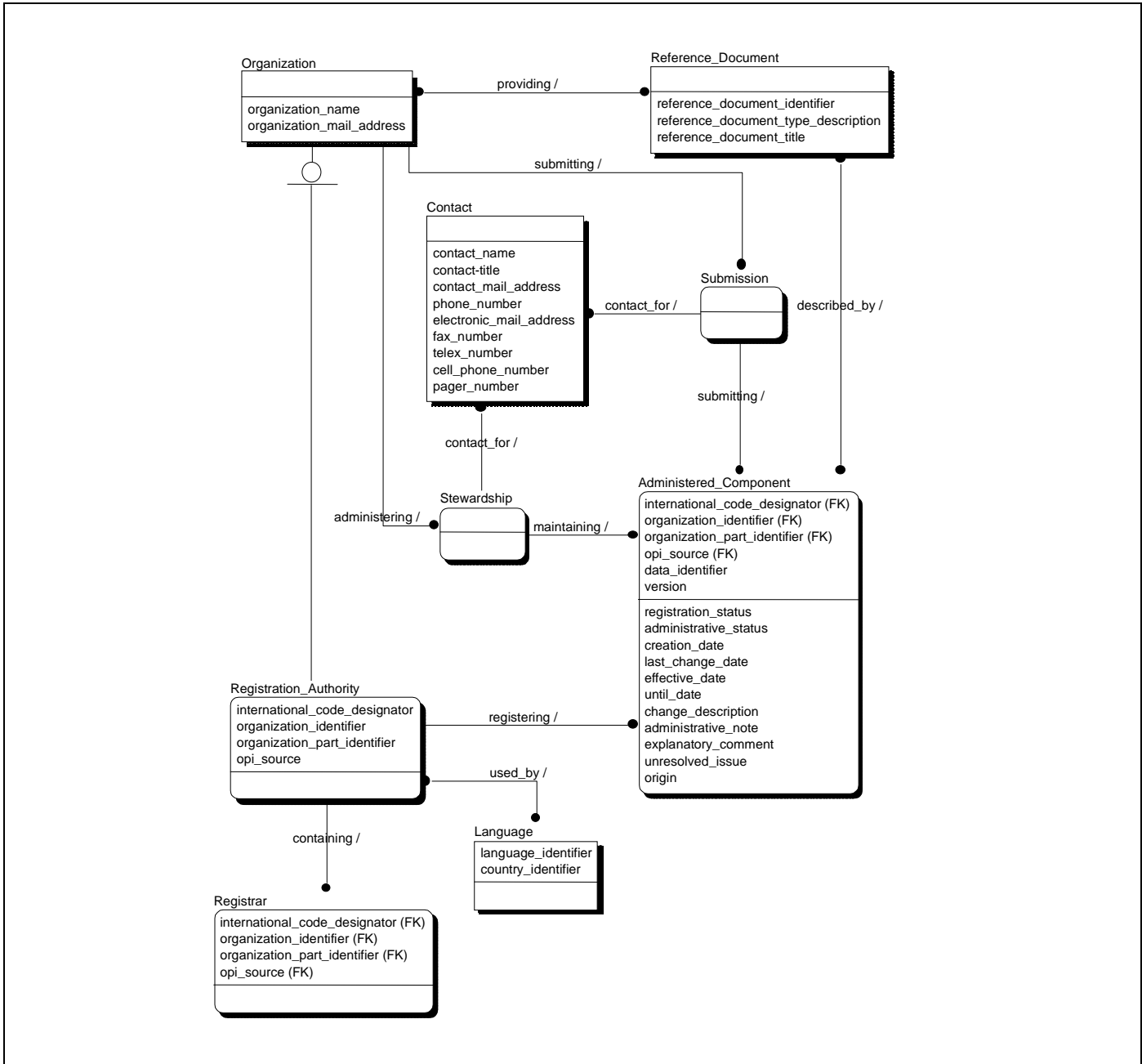


Figure C-2 – IDEF1X Administration metamodel region

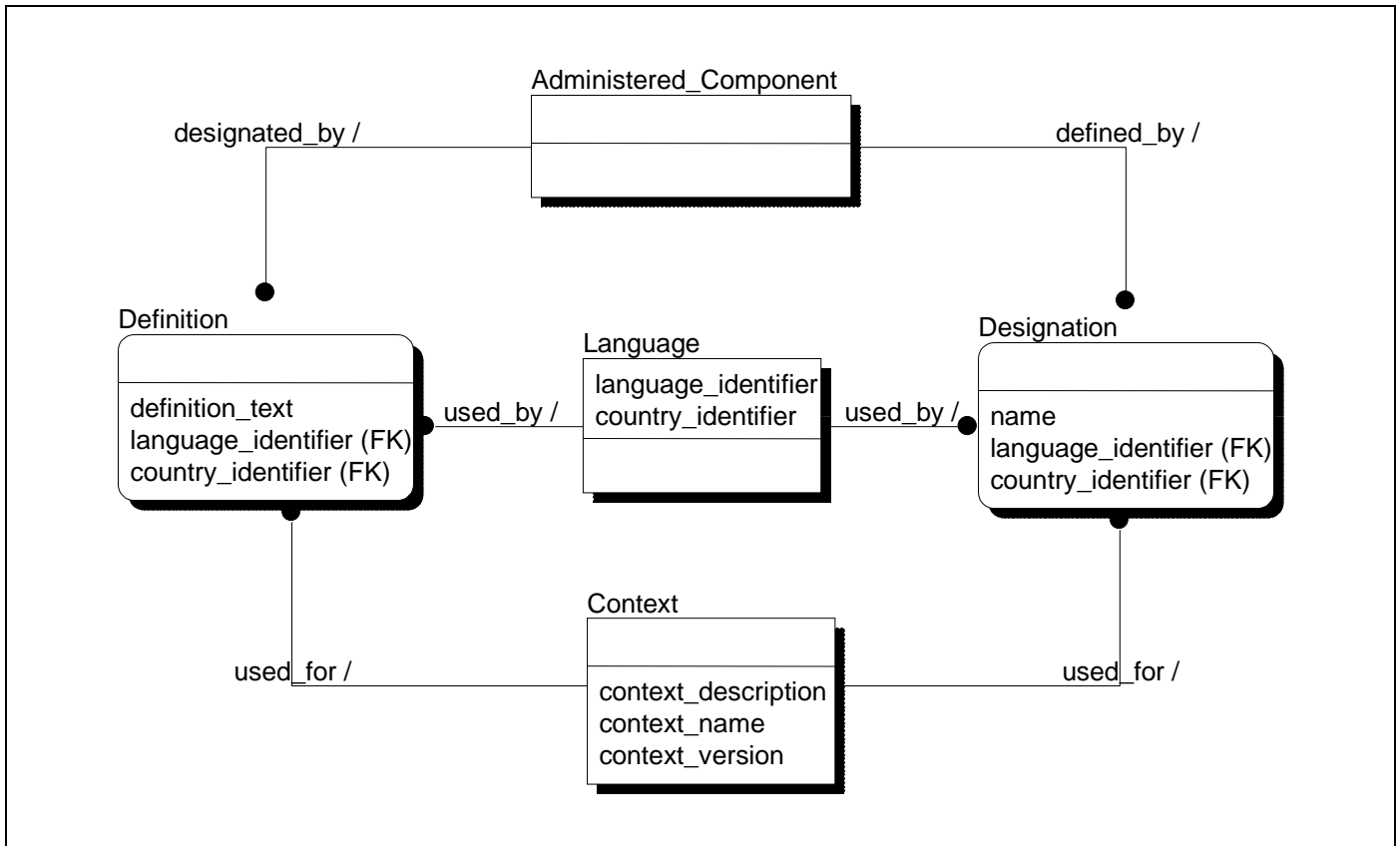


Figure C-3 – IDEF1X naming metamodel region

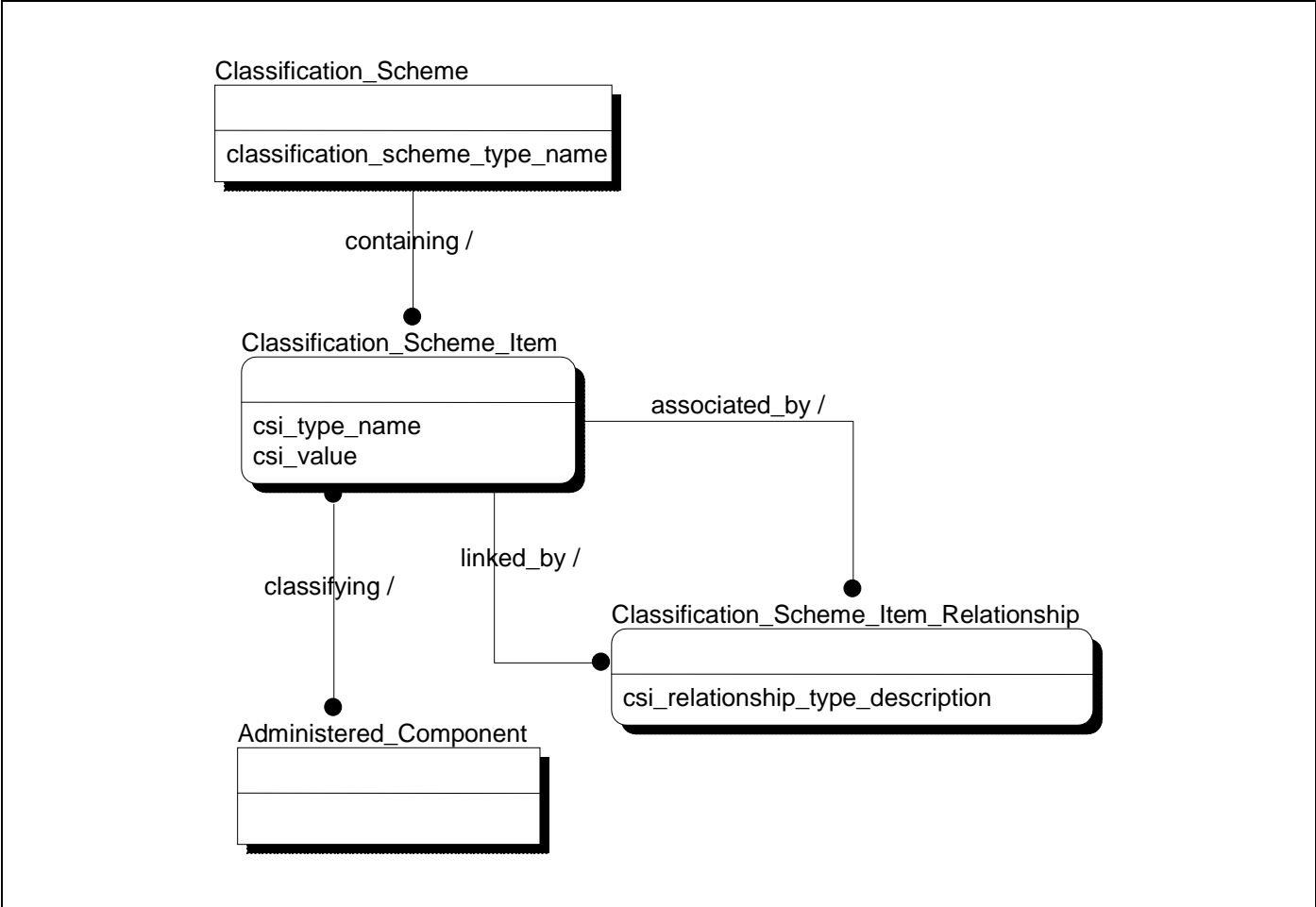


Figure C-4 – IDEF1X classification metamodel region

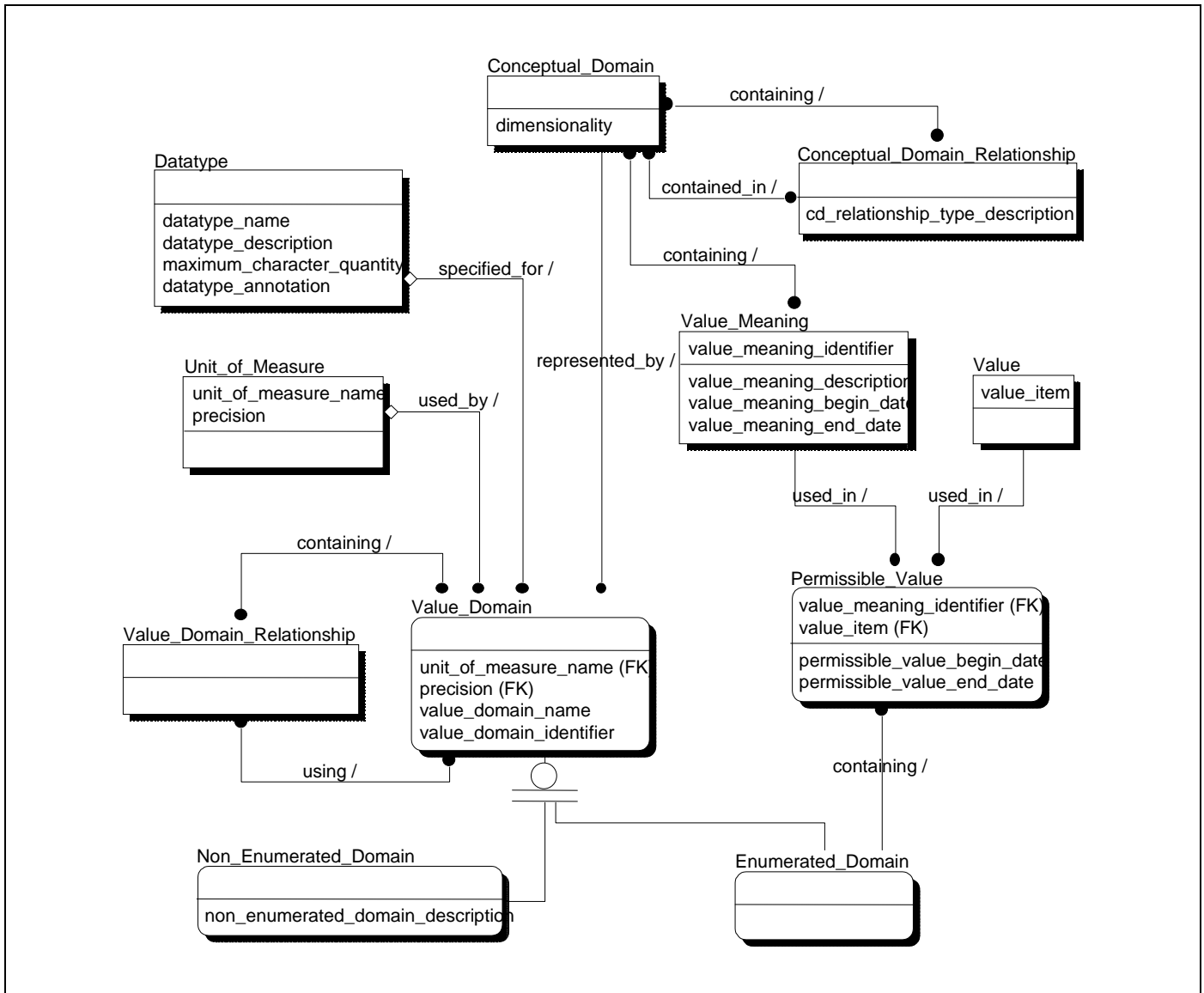


Figure C-5 – IDEF1X conceptual and value domain administration metamodel region

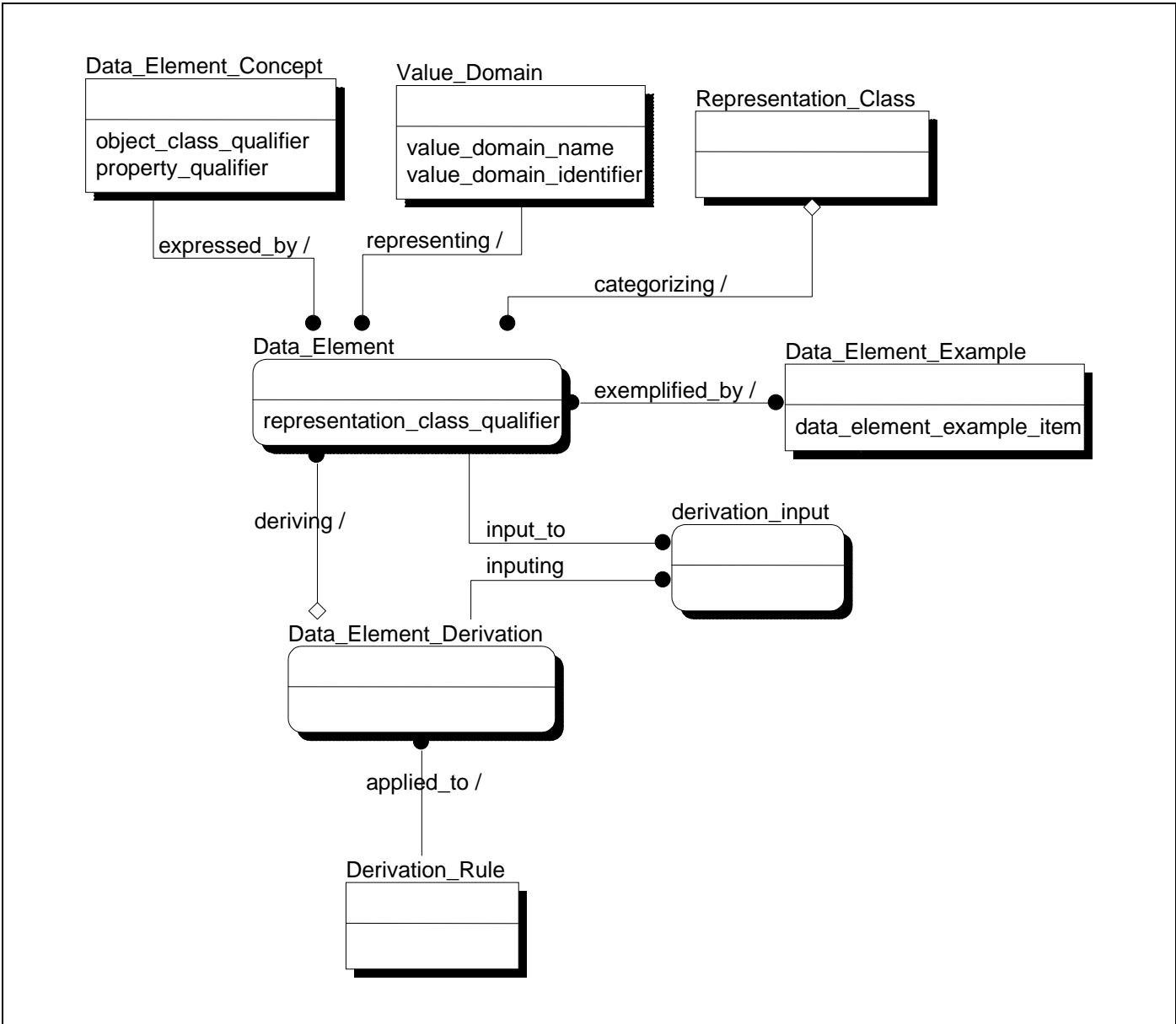


Figure C-6 – IDEF1X administered data component administration metamodel region



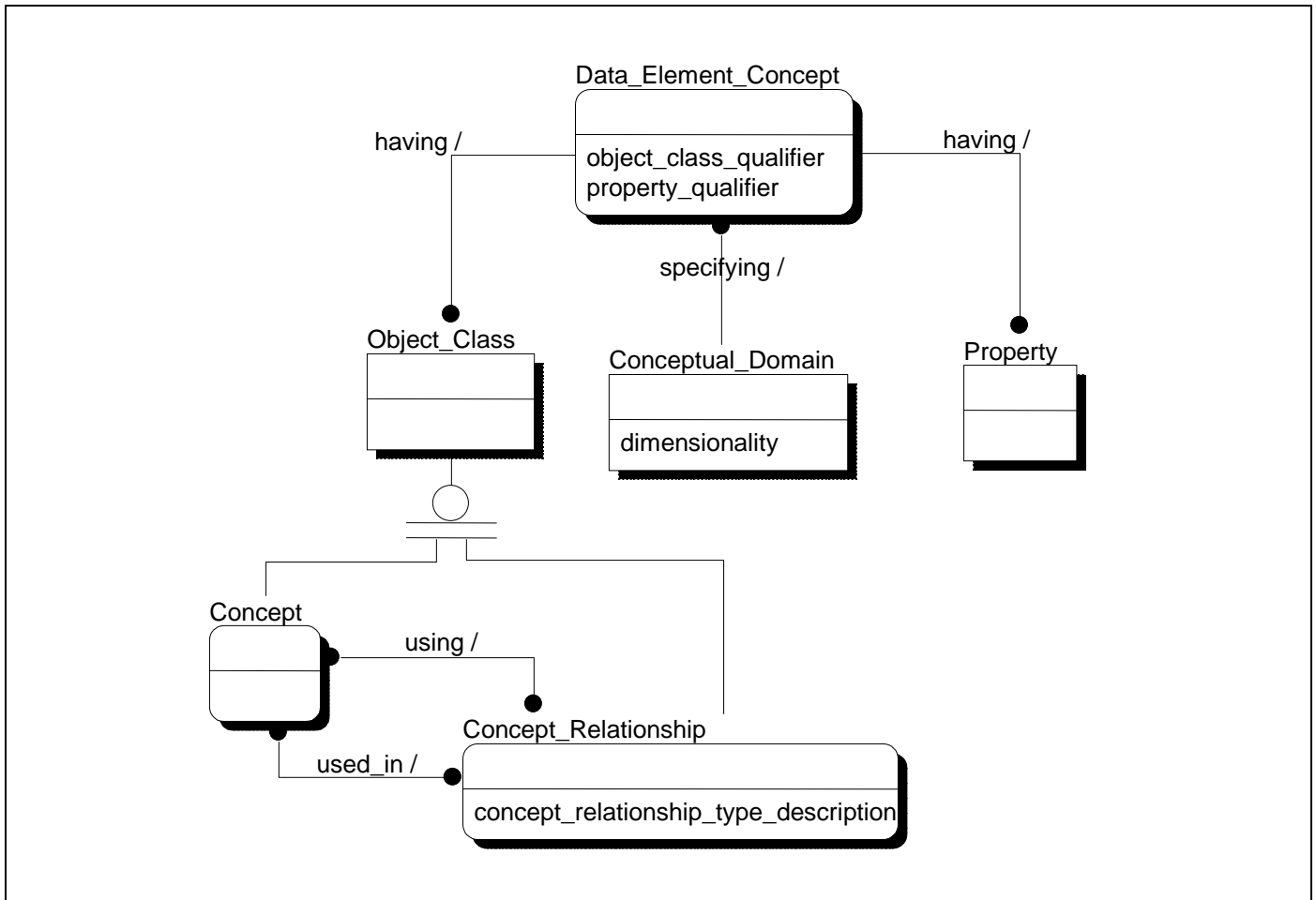


Figure C-7 – IDEF1X administered data component concept administration metamodel region



## Annex D (Informative) Object Role Modeling (ORM) - Natural language Information Analysis Method (NIAM) alternative representation model

The Object Role Modeling (ORM) – Natural language Information Analysis Method (NIAM) representation model of the Metamodel for the management of shareable data is presented in two forms. The first form is graphical. The second form is as a set of facts in sentence format.

### D.1 ISO/IEC 11179-3 Metamodel expressed using ORM graphical form

There are fifteen diagrams that compose the model. These are:

1. ORM high-level metamodel,
2. ORM administration region,
3. ORM-naming and identification region,
4. ORM-classification region,
5. ORM administered data component concept administration region,
6. ORM conceptual domain and value domain administration region,
7. ORM administered data component administration region,
8. ORM administered component,
9. ORM administered components,
10. ORM registration authority,
11. ORM value domain,
12. ORM contact,
13. ORM administered data component concept relationship,
14. ORM value domain relationship,
15. ORM conceptual domain relationship.

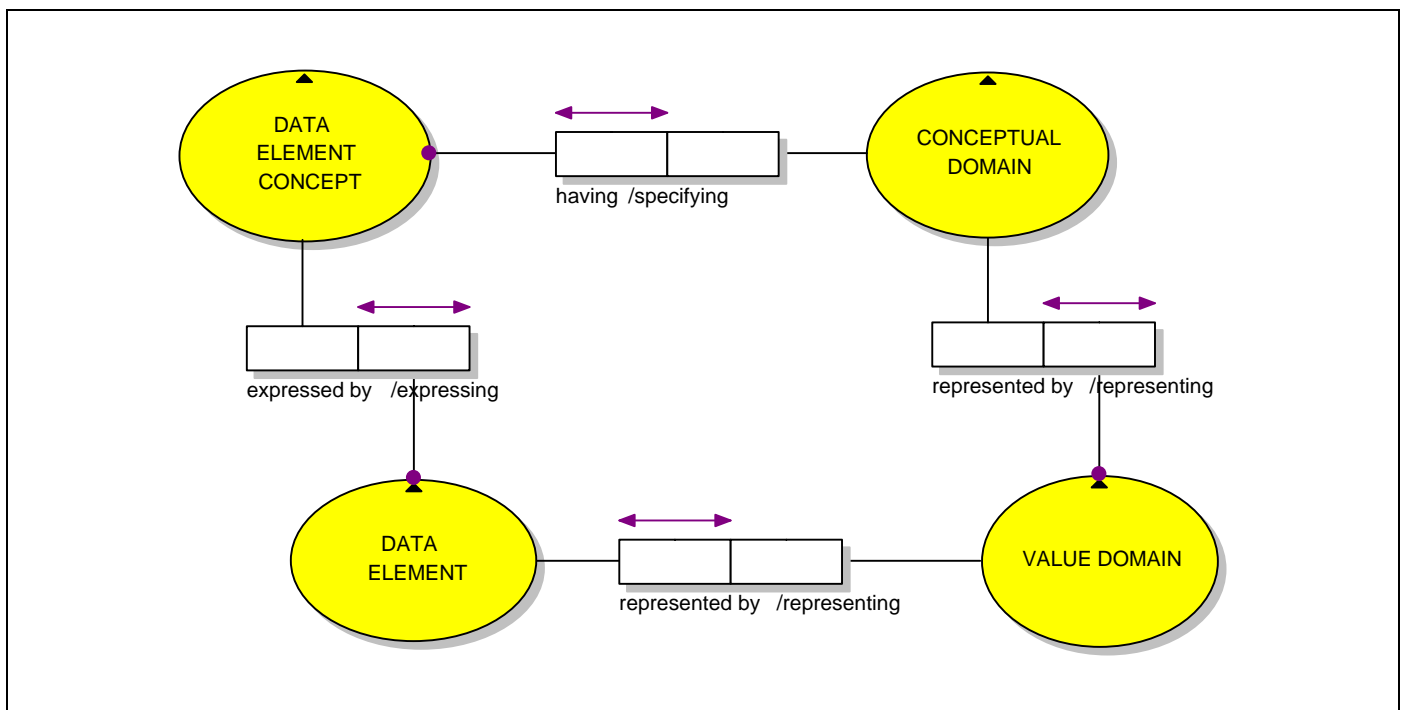


Figure D-1: ORM high-level metamodel

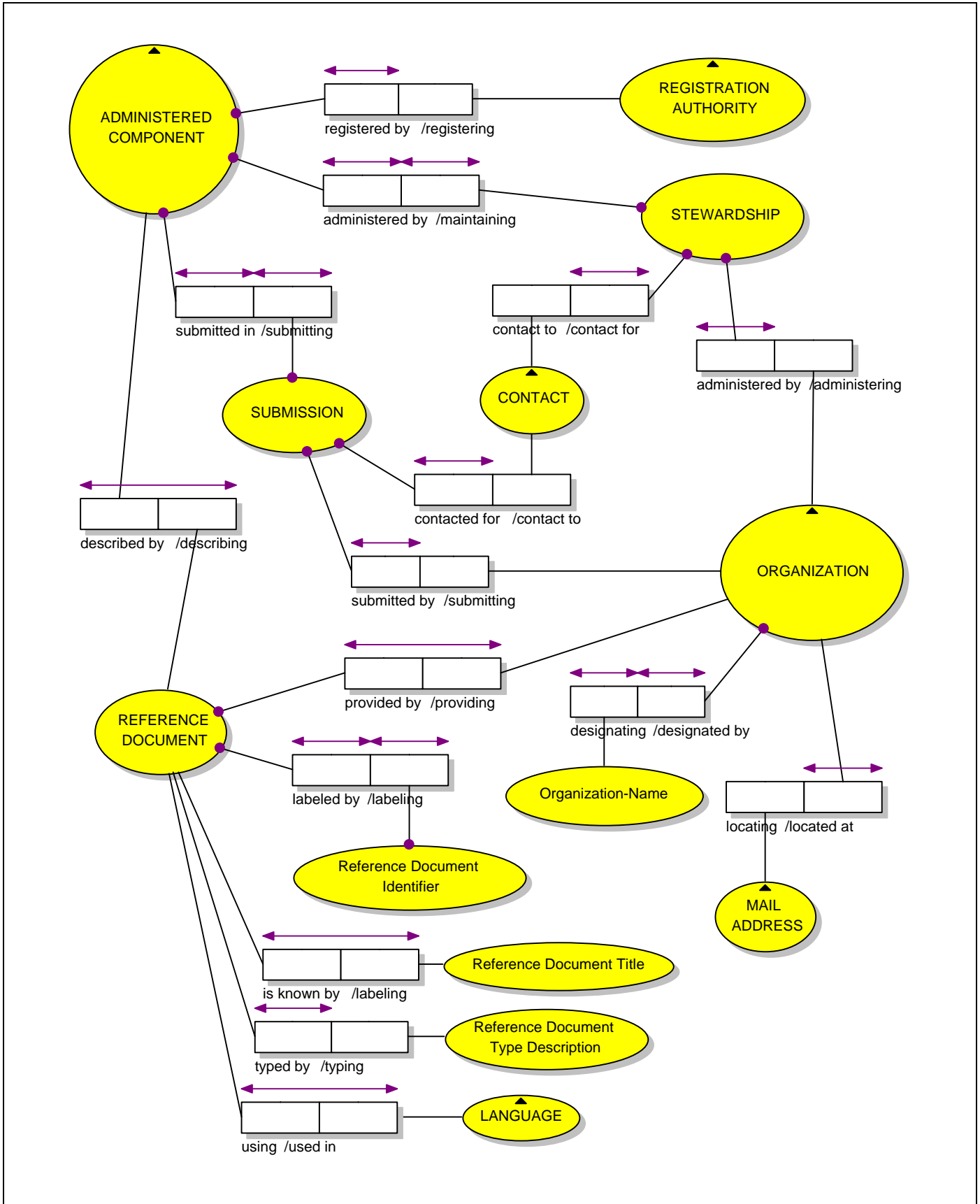


Figure D-2: ORM administration metamodel region

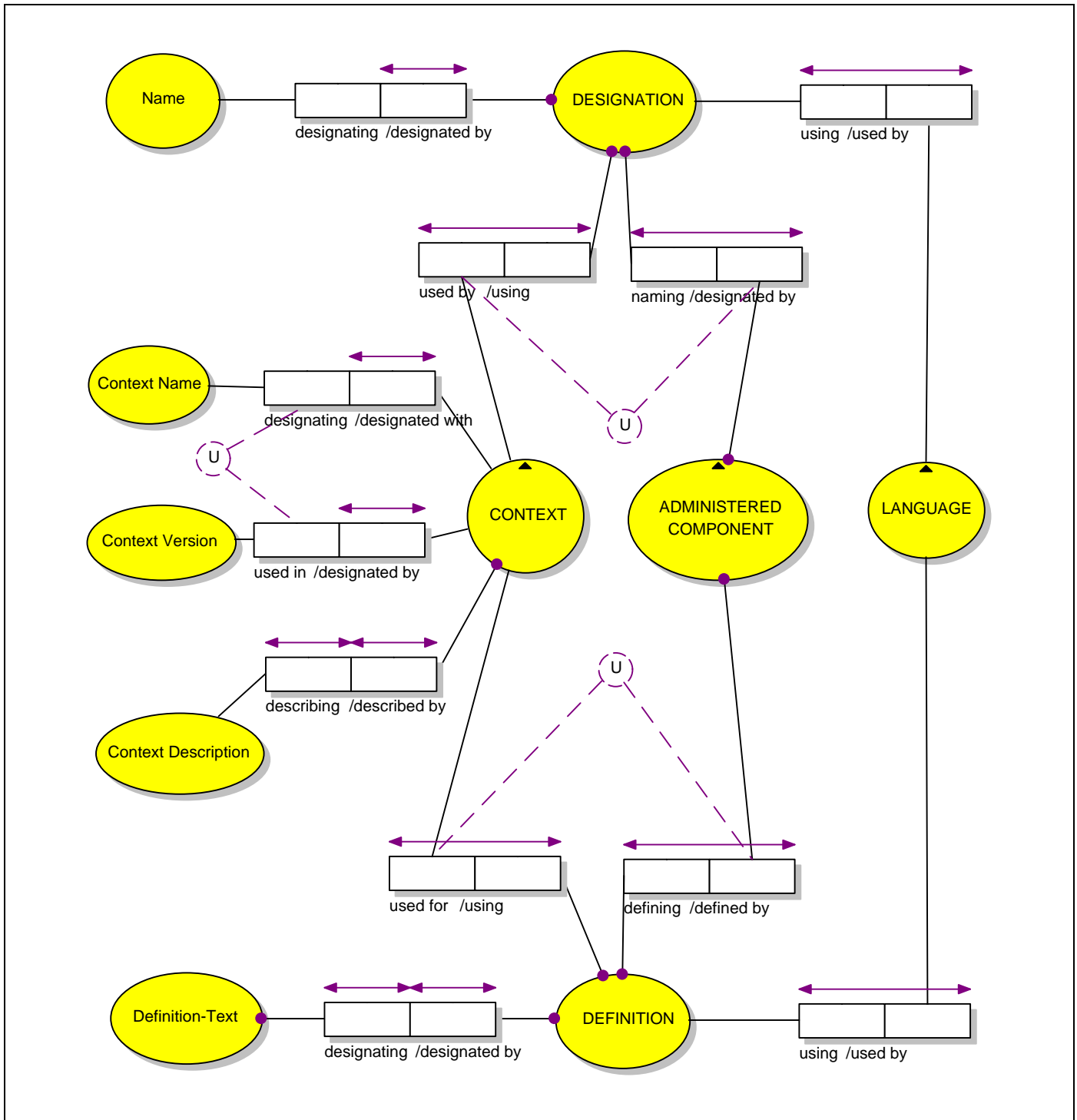


Figure D-3: ORM naming and identification metamodel region

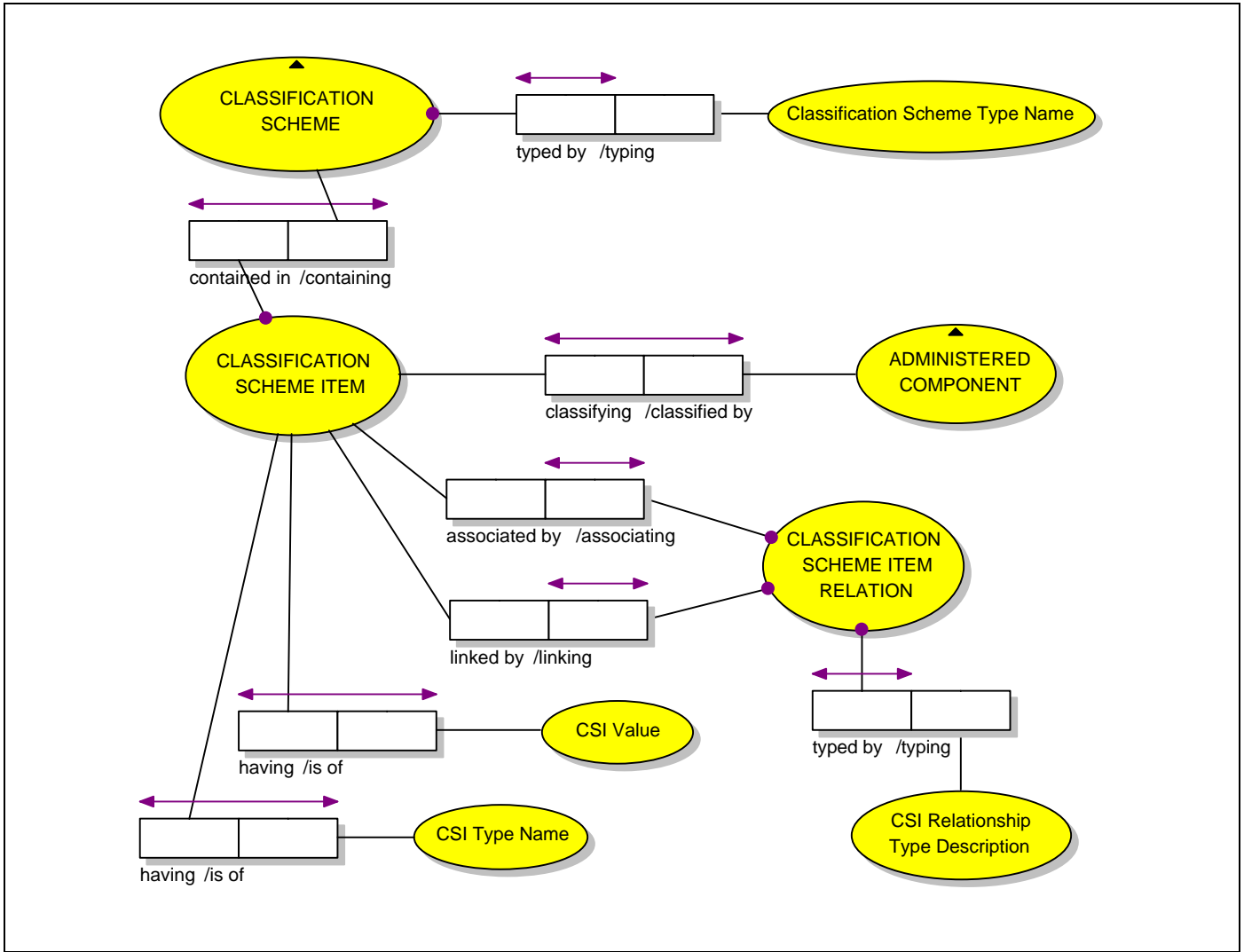


Figure D-4: ORM classification metamodel region

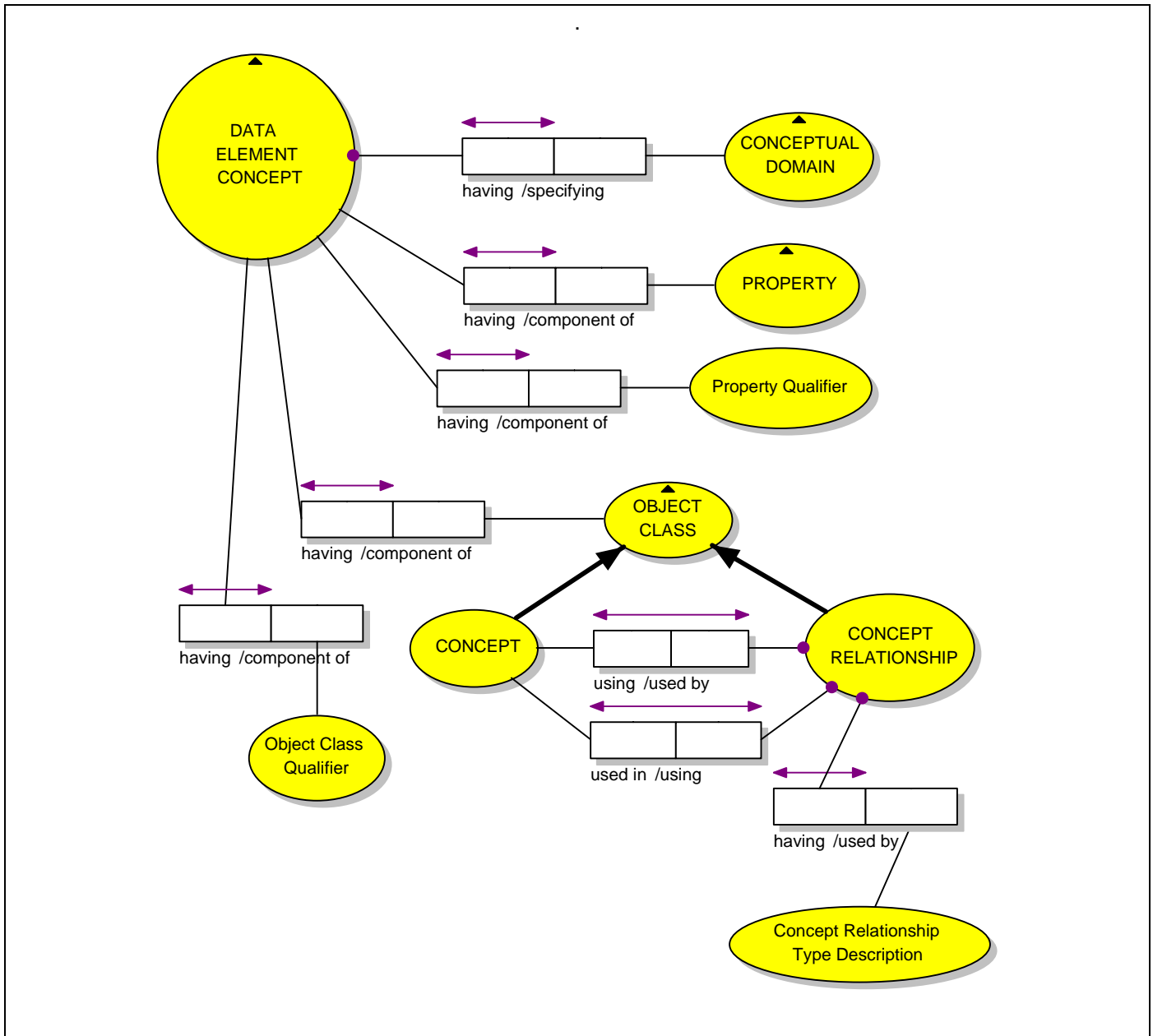


Figure D-5: ORM administered data component concept administration region

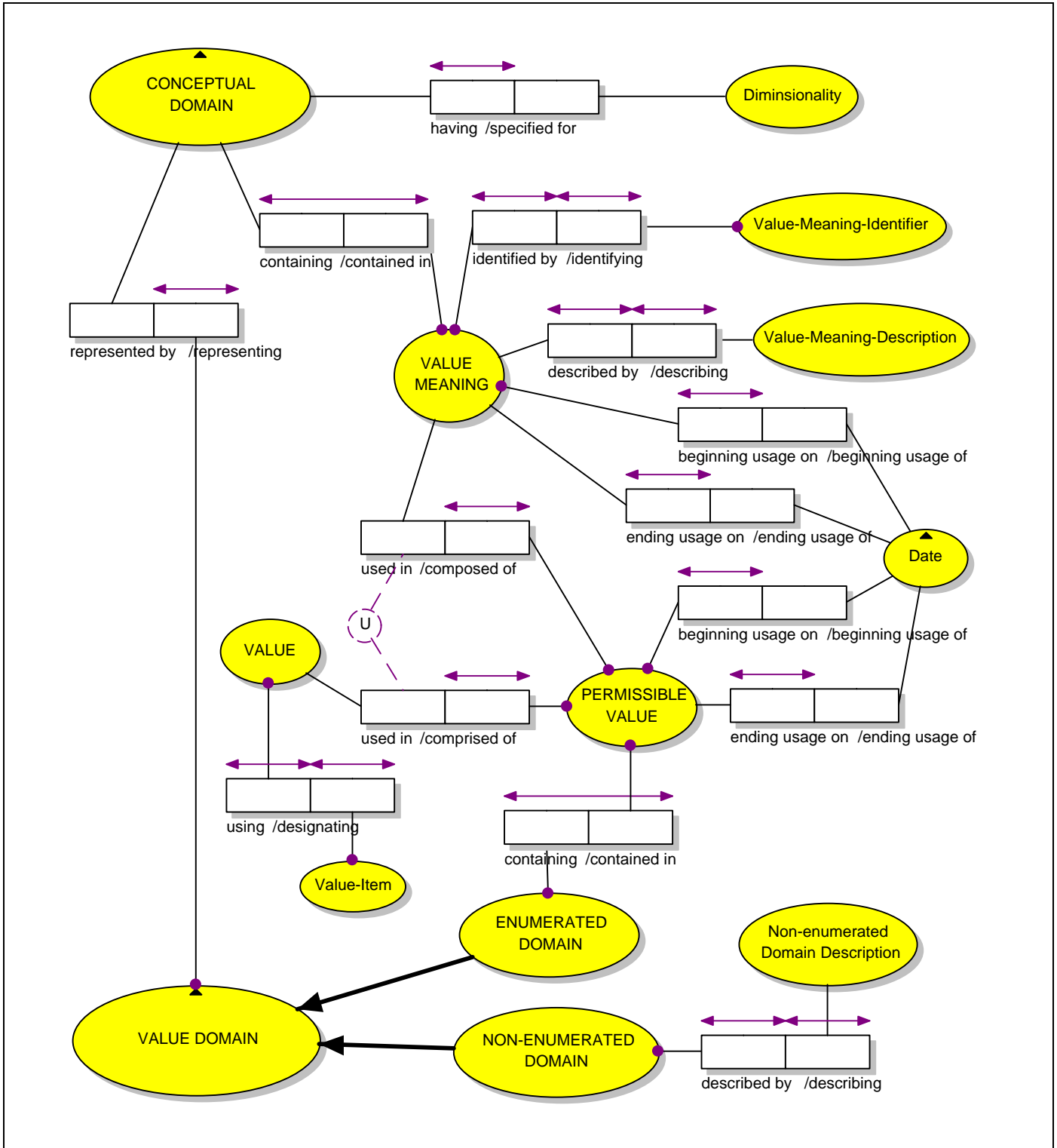


Figure D-6: ORM conceptual domain and value domain administration region



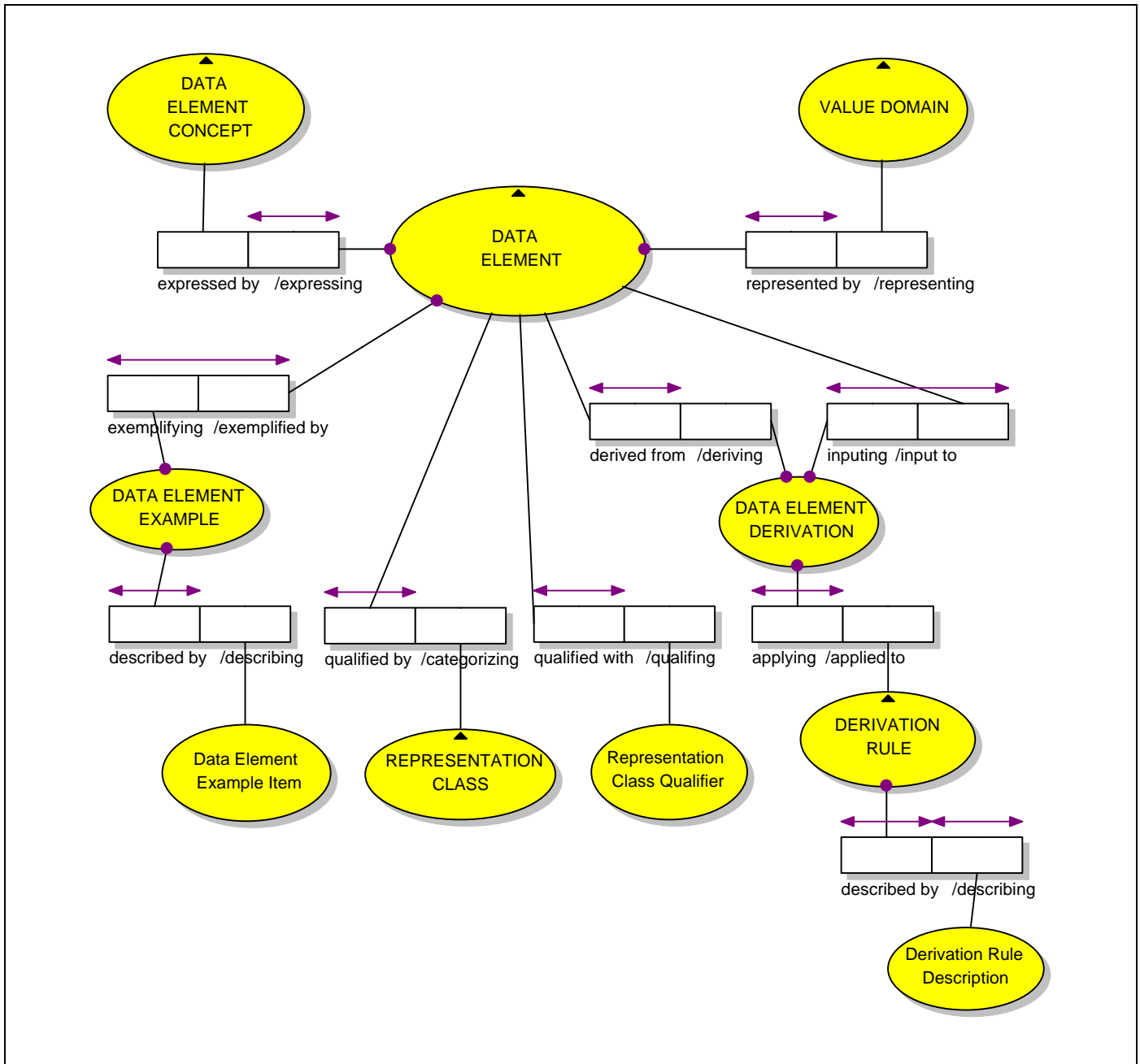


Figure D-7: ORM administered data component administration region

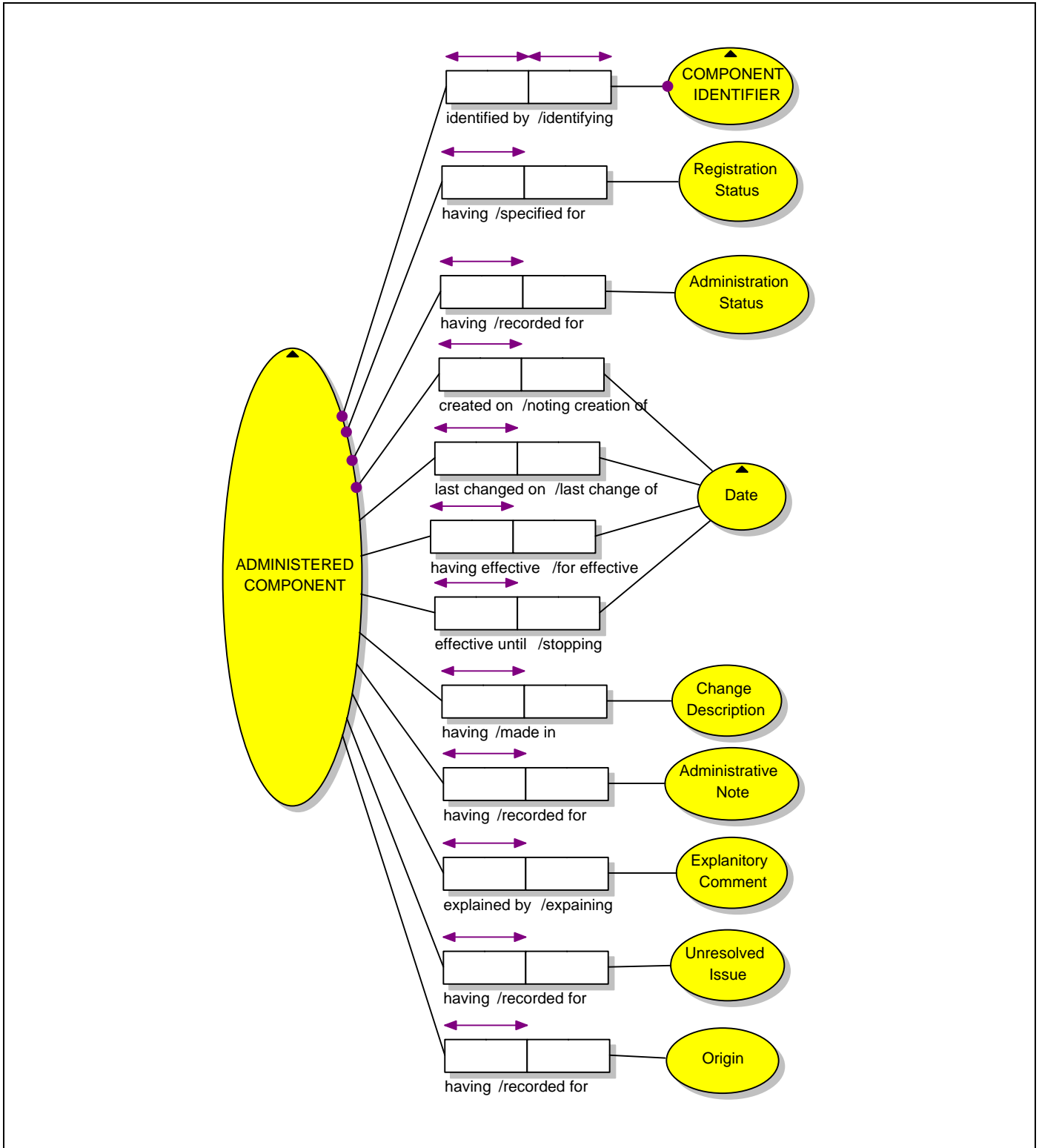


Figure D-8: ORM administered component

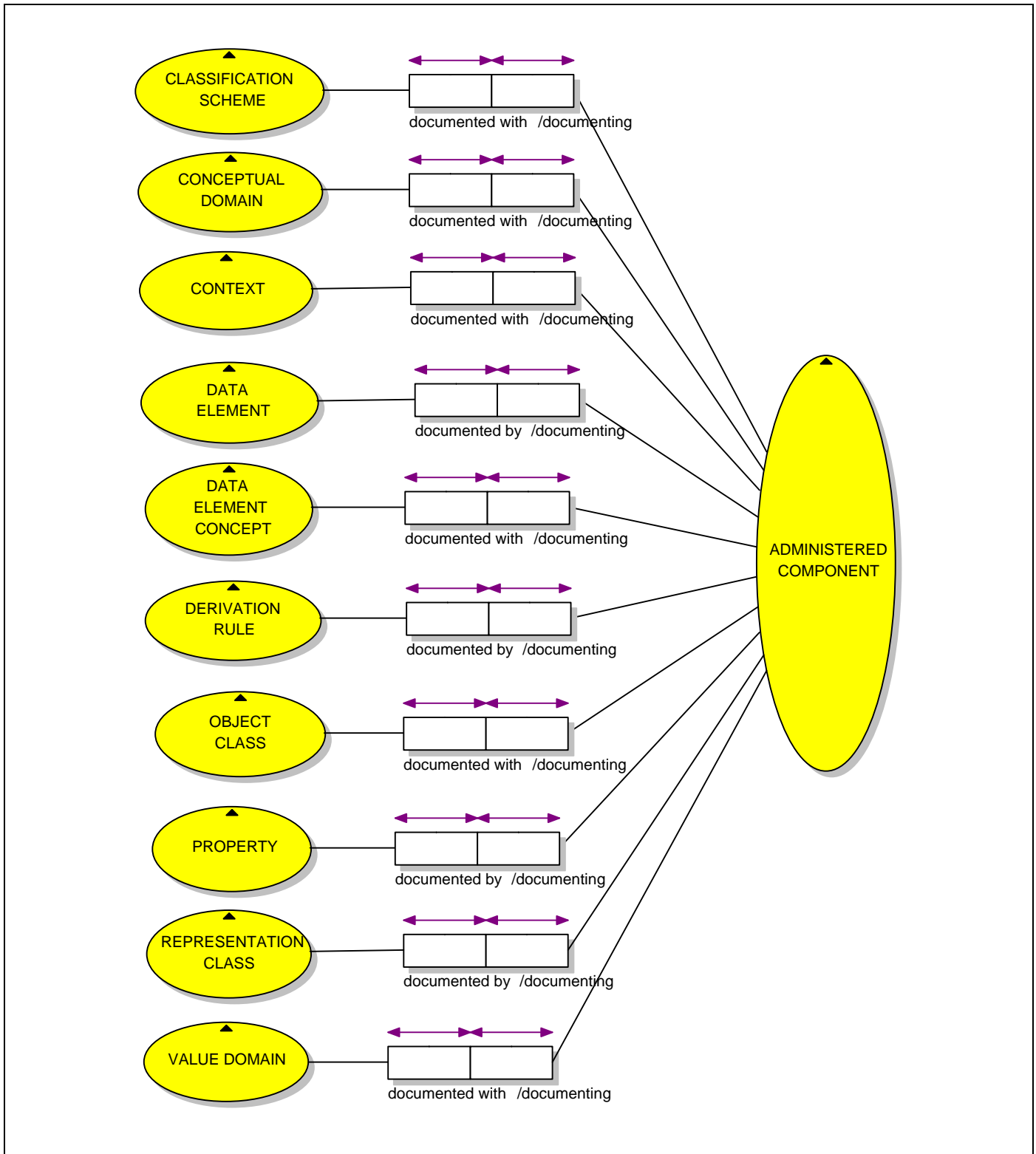


Figure D-9: ORM administered components

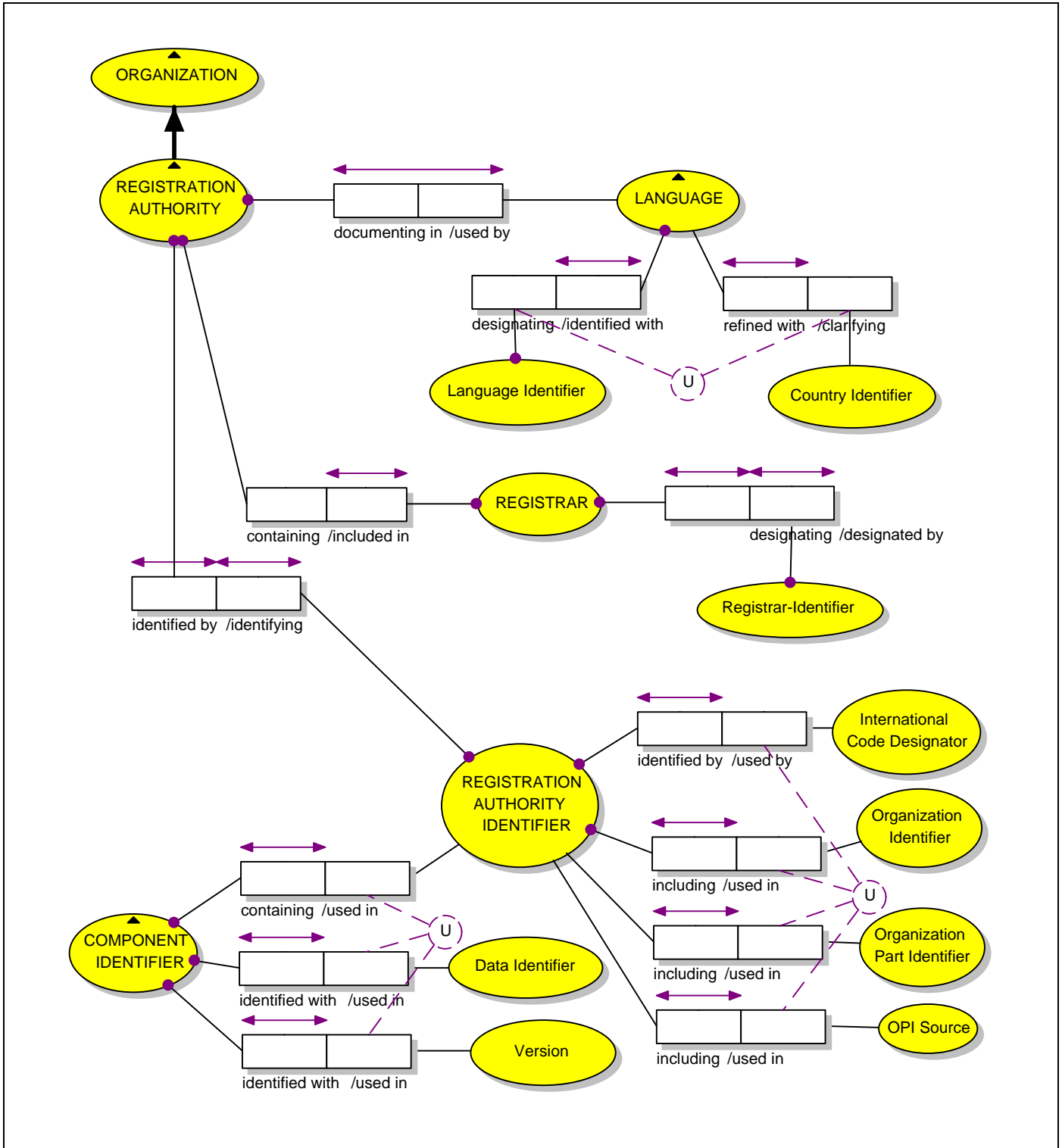


Figure D-10: ORM registration authority

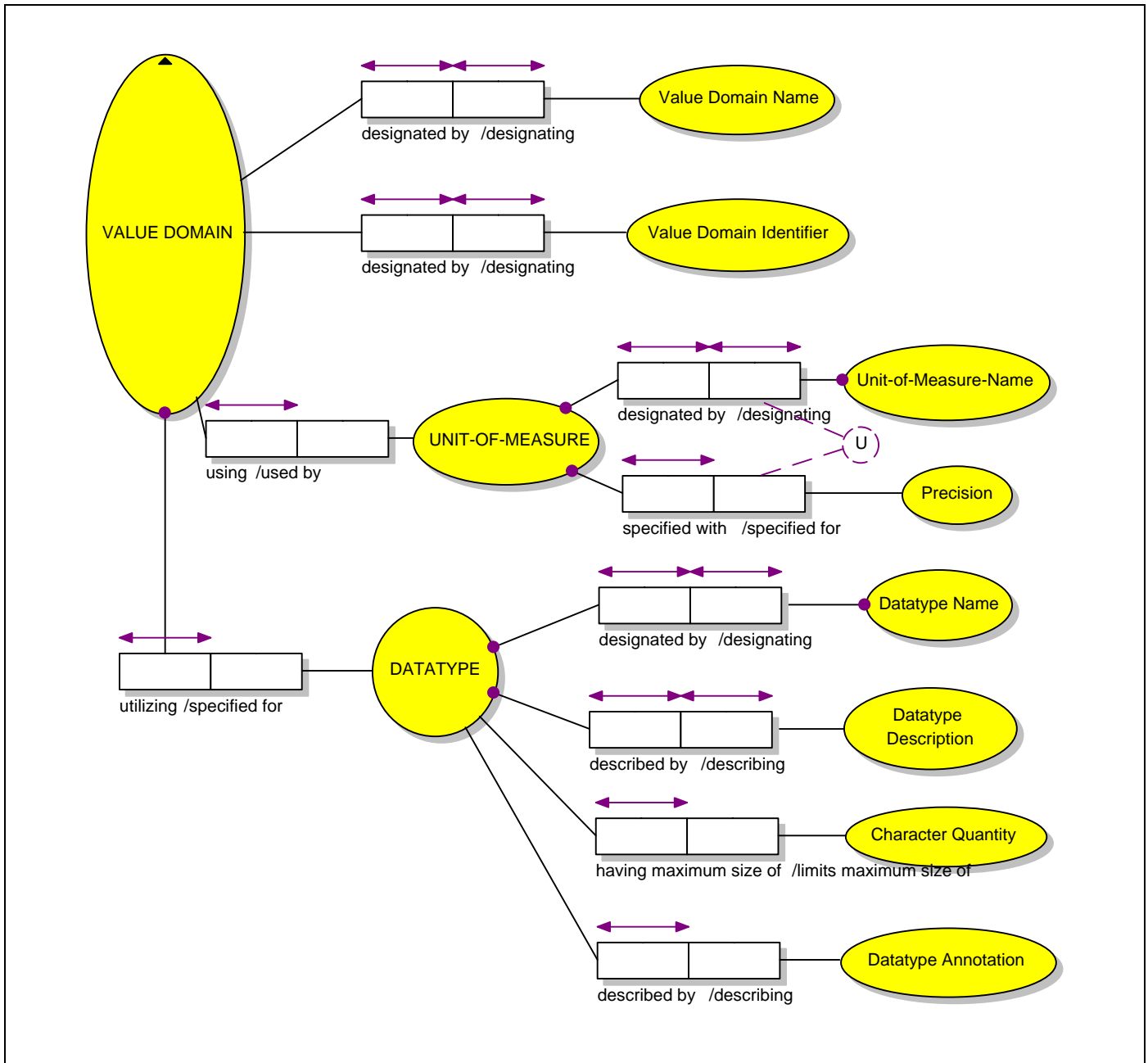


Figure D-11: ORM value domain

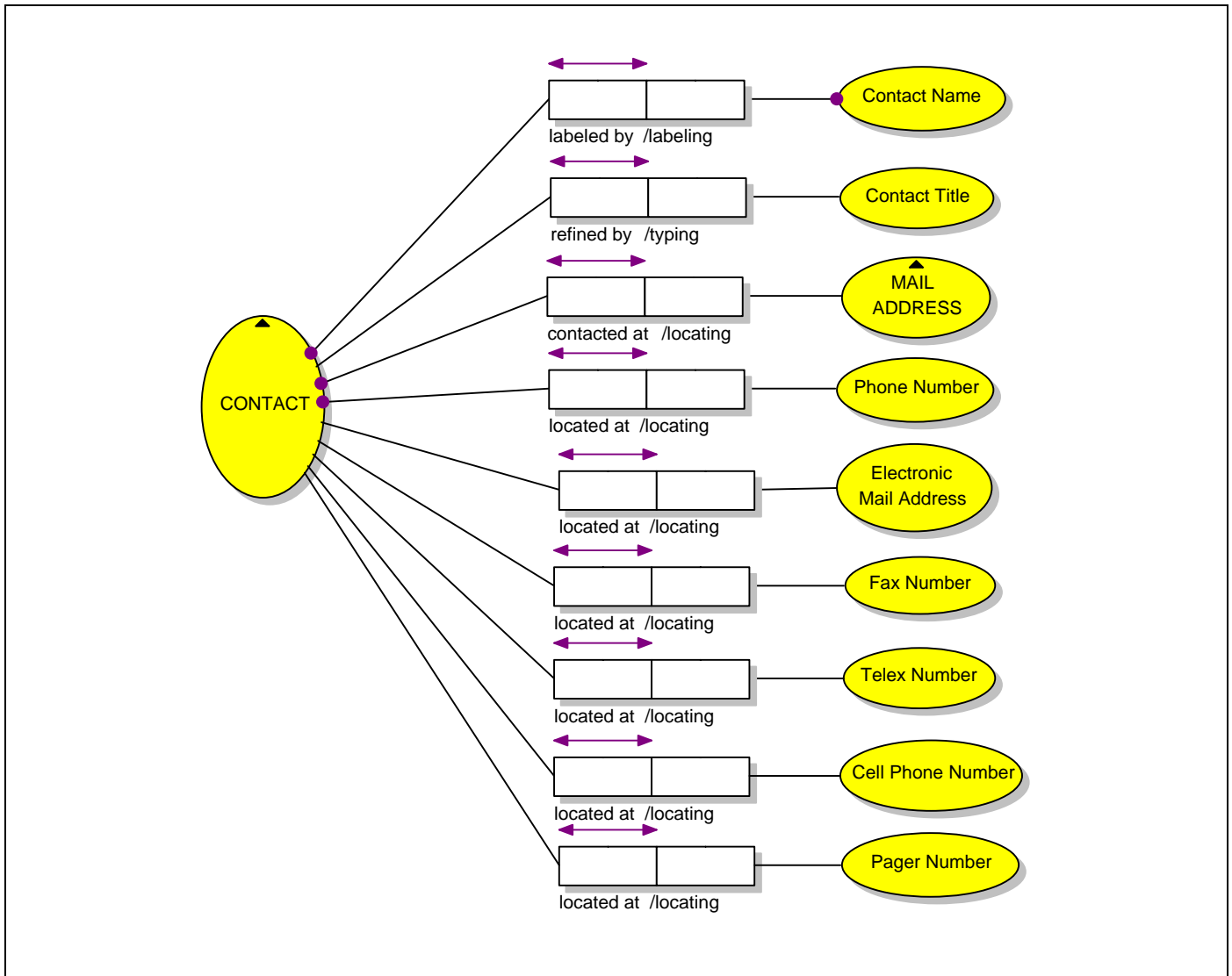


Figure D-12: ORM contact

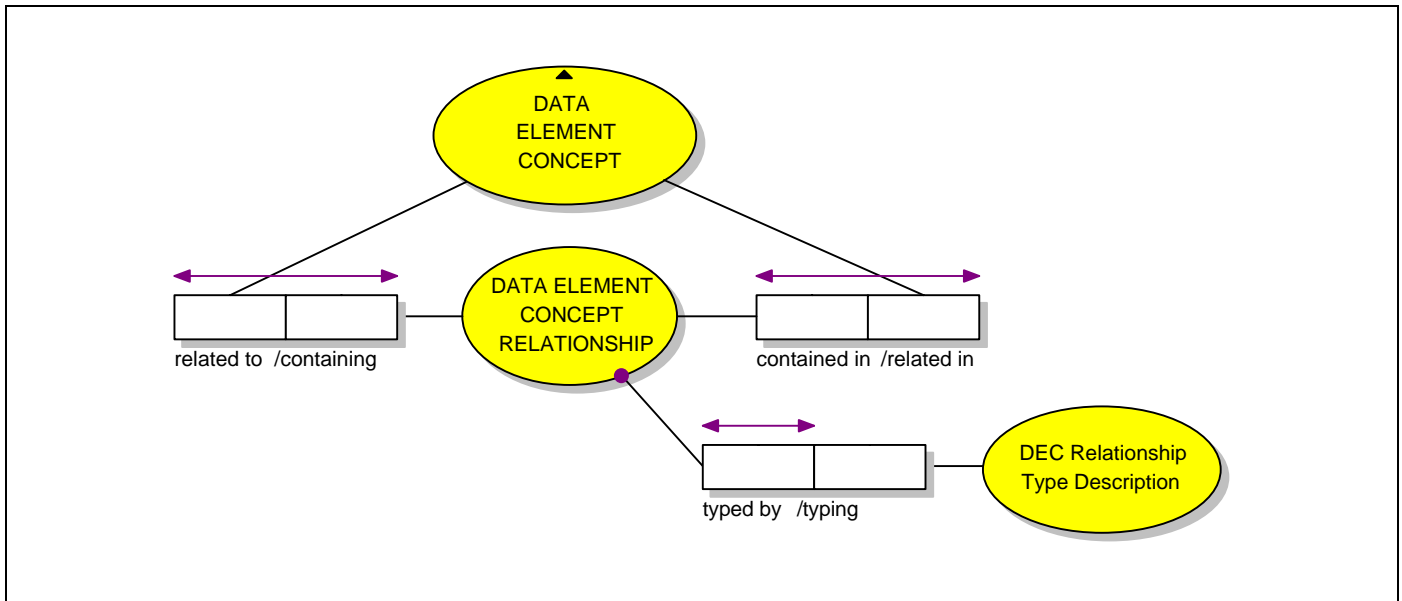


Figure D-13: ORM administered data component concept relationship

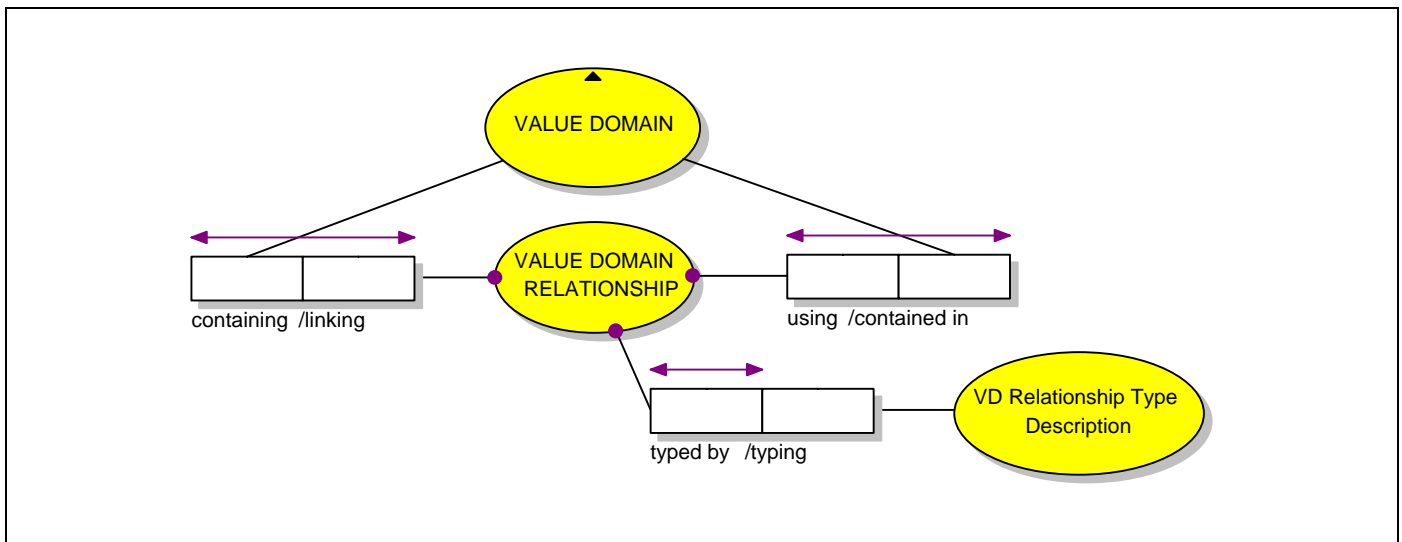


Figure D-14: ORM value domain relationship

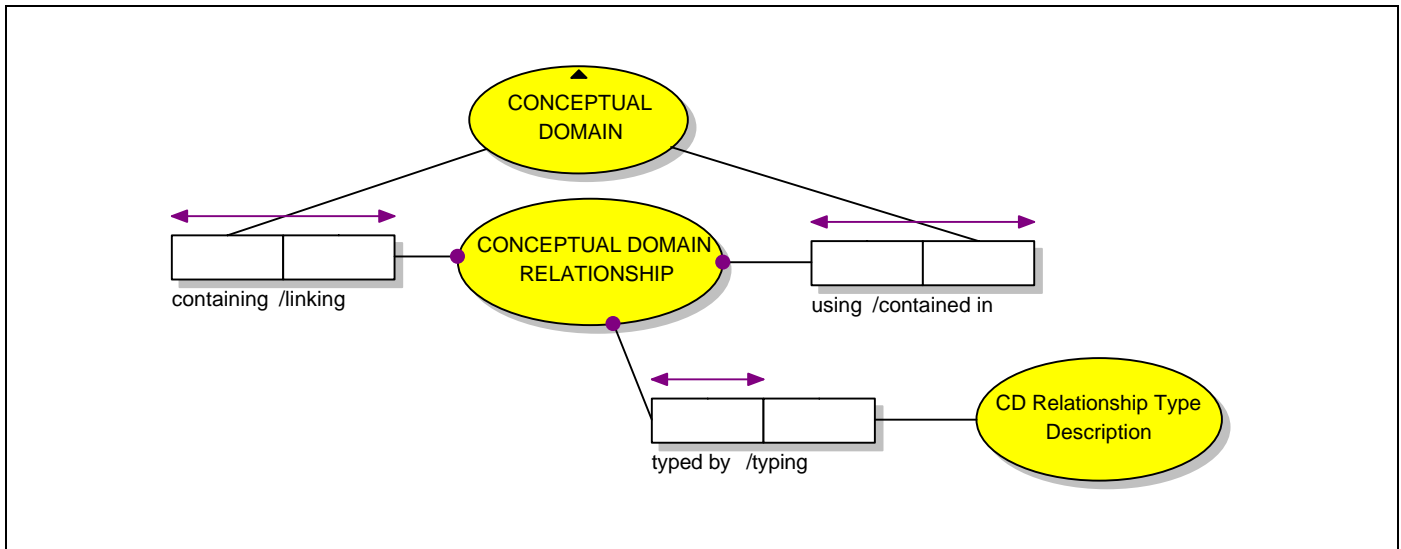


Figure D-15: ORM conceptual domain relationship

## D.2 ISO/IEC 11179-3 Metamodel Elementary Sentences from the Object Role Model.

### EDITOR'S NOTE

Differences between the ORM model and the UML model are identified in this clause by editor's notes. National Bodies are requested to determine whether the differences are material, and if so, what needs to be done about them.

Note that UML allows simple relationships to be named and defined. In the ORM model, only attributed relationships get named and defined.

### D.2.1 "Administered Component"

### EDITOR'S NOTE

In the UML model, separately named attribute capsules exist for each of the "DOCUMENTING" relationships below.

- An "Administered Component" must be ADMINISTERED-BY exactly one "Stewardship".
- An "Administered Component" may be CLASSIFIED-BY any number of "Classification Scheme Items".
- An "Administered Component" must be CREATED-ON exactly one "date".
- An "Administered Component" must be DEFINED-BY one or more "Definition"s.
- An "Administered Component" may be DESCRIBED-BY any number of "Reference Document"s.
- An "Administered Component" may be DESCRIBING at most one "Context".
- An "Administered Component" must be DESIGNATED-BY one or more "Designation"s.
- An "Administered Component" may be DOCUMENTING at most one "Classification Scheme".
- An "Administered Component" may be DOCUMENTING at most one "Conceptual Domain".
- An "Administered Component" may be DOCUMENTING at most one "Administered data component Concept".



An "Administered Component" may be DOCUMENTING at most one "Administered data component".

An "Administered Component" may be DOCUMENTING at most one "Derivation Rule".

An "Administered Component" may be DOCUMENTING at most one "Object Class".

An "Administered Component" may be DOCUMENTING at most one "Property".

An "Administered Component" may be DOCUMENTING at most one "Representation Class".

An "Administered Component" may be DOCUMENTING at most one "Value Domain".

An "Administered Component" may be EFFECTIVE-UNTIL at most one "date".

An "Administered Component" may be EXPLAINED-BY at most one "explanatory comment".

An "Administered Component" may be HAVING at most one "administrative note".

An "Administered Component" must be HAVING exactly one "administrative status".

An "Administered Component" may be HAVING at most one "change description".

An "Administered Component" may be HAVING at most one "origin".

An "Administered Component" must be HAVING exactly one "registration status".

An "Administered Component" may be HAVING at most one "unresolved issue".

An "Administered Component" may be HAVING-EFFECTIVE at most one "date".

An "Administered Component" must be IDENTIFIED-BY exactly one "Component Identifier".

An "Administered Component" may be LAST-CHANGED-ON at most one "date".

An "Administered Component" must be REGISTERED-BY exactly one "Registration Authority".

An "Administered Component" must be SUBMITTED-BY exactly one "Submission".

No "Administered Component" may be simultaneously

DOCUMENTING an "Administered data component Concept"

and

DOCUMENTING an "Administered data component".

No "Administered Component" may be simultaneously

DOCUMENTING an "Administered data component Concept"

and

DOCUMENTING an "Value Domain".

No "Administered Component" may be simultaneously

DOCUMENTING an "Administered data component Concept"

and

DOCUMENTING an "Conceptual Domain".

No "Administered Component" may be simultaneously

DOCUMENTING an "Administered data component Concept"

and

DOCUMENTING an "Object Class".

No "Administered Component" may be simultaneously

DOCUMENTING an "Administered data component"

and

DOCUMENTING an "Conceptual Domain".

No "Administered Component" may be simultaneously

DOCUMENTING an "Administered data component Concept"

and

DOCUMENTING an "Property".

EDITOR'S NOTE: The following constraint is a duplicate.

No "Administered Component" may be simultaneously  
— DOCUMENTING an "Administered data component Concept"  
— and  
— DOCUMENTING an "Value Domain".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Administered data component Concept"  
and  
DOCUMENTING an "Derivation Rule".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Derivation Rule"  
and  
DOCUMENTING an "Property".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Object Class"  
and  
DOCUMENTING an "Property".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Administered data component"  
and  
DOCUMENTING an "Derivation Rule".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Administered data component"  
and  
DOCUMENTING an "Object Class".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Administered data component"  
and  
DOCUMENTING an "Property".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Value Domain"  
and  
DOCUMENTING an "Conceptual Domain".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Value Domain"  
and  
DOCUMENTING an "Derivation Rule".

No "Administered Component" may be simultaneously  
DOCUMENTING an "Value Domain"  
and  
DOCUMENTING an "Object Class".

No "Administered Component" may be simultaneously

DOCUMENTING an "Value Domain"

and

DOCUMENTING an "Property".

No "Administered Component" may be simultaneously

DOCUMENTING an "Derivation Rule"

and

DOCUMENTING an "Object Class".

An "Administered Component" is described as "A component for which administrative information is recorded".

#### EDITOR'S NOTE

The following lines were added manually.

An "Administered Component" may be DOCUMENTING at most one "Context".

No "Administered Component" may be simultaneously

DOCUMENTING an "Context"

and

DOCUMENTING an "Classification Scheme".

No "Administered Component" may be simultaneously

DOCUMENTING an "Context"

and

DOCUMENTING an "Conceptual Domain".

No "Administered Component" may be simultaneously

DOCUMENTING an "Context"

and

DOCUMENTING an "Administered data component".

No "Administered Component" may be simultaneously

DOCUMENTING an "Context"

and

DOCUMENTING an "Administered data component Concept".

No "Administered Component" may be simultaneously

DOCUMENTING an "Context"

and

DOCUMENTING an "Derivation Rule".

No "Administered Component" may be simultaneously

DOCUMENTING an "Context"

and

DOCUMENTING an "Object Class".

No "Administered Component" may be simultaneously

DOCUMENTING an "Context"

and

DOCUMENTING an "Property".

No "Administered Component" may be simultaneously

DOCUMENTING an "Context"

and  
DOCUMENTING an "Representation Class".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Context"  
and  
DOCUMENTING an "Value Domain".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Conceptual Domain".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Administered data component".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Administered data component Concept".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Derivation Rule".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Administered data component".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Object Class".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Property".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Representation Class".  
No "Administered Component" may be simultaneously  
DOCUMENTING an "Classification Scheme"  
and  
DOCUMENTING an "Value Domain".  
No "Administered Component" may be simultaneously

DOCUMENTING an "Conceptual Domain"

and

DOCUMENTING an "Derivation Rule".

No "Administered Component" may be simultaneously

DOCUMENTING an "Conceptual Domain"

and

DOCUMENTING an "Derivation Rule".

No "Administered Component" may be simultaneously

DOCUMENTING an "Conceptual Domain"

and

DOCUMENTING an "Object Class".

No "Administered Component" may be simultaneously

DOCUMENTING an "Conceptual Domain"

and

DOCUMENTING an "Property".

No "Administered Component" may be simultaneously

DOCUMENTING an "Conceptual Domain"

and

DOCUMENTING an "Representation Class".

No "Administered Component" may be simultaneously

DOCUMENTING an "Representation Class "

and

DOCUMENTING an "Administered data component".

No "Administered Component" may be simultaneously

DOCUMENTING an "Representation Class "

and

DOCUMENTING an "Administered data component Concept".

No "Administered Component" may be simultaneously

DOCUMENTING an "Representation Class "

and

DOCUMENTING an "Derivation Rule".

No "Administered Component" may be simultaneously

DOCUMENTING an "Representation Class "

and

DOCUMENTING an "Object Class".

No "Administered Component" may be simultaneously

DOCUMENTING an "Representation Class "

and

DOCUMENTING an "Property".

No "Administered Component" may be simultaneously

DOCUMENTING an "Representation Class "

and

DOCUMENTING an "Value Domain".

No "Administered Component" may be simultaneously

DOCUMENTING an "Value Domain"

and

DOCUMENTING an "Administered data component".

### **D.2.2 "administrative note"**

An "administrative note" may be RECORDED-FOR any number of "Administered Component"s.

An "administrative note" is described as "Any general notes about the administered component".

### **D.2.3 "administrative status"**

An "administrative status" may be RECORDED-FOR any number of "Administered Component"s.

An "administrative status" is described as "A designation of the position in the processing life-cycle of a registration authority for handling registration requests".

### **D.2.4 "cd relationship type description"**

A "cd relationship type description" may be TYPING any number of "Conceptual Domain Relationship"s.

A "cd relationship type description" is described as "A description of the type of relationship between a conceptual domain and one or more other conceptual domains".

### **D.2.5 "cell phone number"**

A "cell phone number" may be LOCATING any number of "Contact"s.

A "cell phone number" is described as "A cell phone number for communication with the contact".

### **D.2.6 "change description"**

A "change description" may be MADE-IN any number of "Administered Component"s.

A change description is described as "The description of what has changed in the administered component since the prior version of the administered component".

### **D.2.7 "Classification Scheme"**

A "Classification Scheme" may be CONTAINING any number of "Classification Scheme Item"s.

A "Classification Scheme" may be DOCUMENTED-BY at most one "Administered Component".

A "Classification Scheme" must be TYPED-BY exactly one "classification scheme type name".

A "Classification Scheme" is described as "The descriptive information for an arrangement or division of objects into groups based on characteristics, which the objects have in common".

### **D.2.8 "Classification Scheme Item"**

A "Classification Scheme Item" may be ASSOCIATED-BY any number of "Classification Scheme Item Relationship"s.

A "Classification Scheme Item" may be CLASSIFYING any number of "Administered Component"s.

A "Classification Scheme Item" must be CONTAINED-IN one or more "Classification Scheme"s.

A "Classification Scheme Item" may be HAVING at most one "csi type name".

A "Classification Scheme Item" may be LINKED-BY any number of "Classification Scheme Item Relationship"s.

A "Classification Scheme Item" may be POSSESSING at most one "csi value".

A "Classification Scheme Item" is described as "An item of content in a classification scheme".

### D.2.9 "Classification Scheme Item Relationship"

A "Classification Scheme Item Relationship" must be ASSOCIATING one or more "Classification Scheme Item"s.

EDITOR'S NOTE: The following sentence is a former wording of the previous one.

~~A "Classification Scheme Item Relationship" must be LINKING one or more "Classification Scheme Item"s.~~

EDITOR'S NOTE: Prefix corrected from "cs" to "csi" in the following sentence.

A "Classification Scheme Item Relationship" must be TYPED-BY exactly one "csi relationship type description".

A "Classification Scheme Item Relationship" is described as "The relationship among items in a classification scheme".

### D.2.10 "classification scheme type name"

A "classification scheme type name" may be TYPING any number of "Classification Scheme"s.

A "classification scheme type name" is described as "The name of the type of classification scheme".

### D.2.11 "Component Identifier"

A "Component Identifier" must be CONTAINING exactly one "registration authority identifier".

A "Component Identifier" must be IDENTIFIED-WITH exactly one "data identifier".

A "Component Identifier" must be IDENTIFIED-WITH exactly one "version".

A "Component Identifier" must be IDENTIFYING exactly one "Administered Component".

Every "Component Identifier"

is associated uniquely with one combination of

a "registration authority identifier" USED IN the "Component Identifier"

and a "data identifier" USED IN the "Component Identifier"

and a "version" USED IN the "Component Identifier".

A "Component Identifier" is described as "An identifier for a component".

### D.2.12 "Concept"

A "Concept" is always a kind of "Object Class".

A "Concept" may be USED-IN any number of "Concept Relationship"s.

A "Concept" may be USING any number of "Concept Relationship"s.

A "Concept" is described as "A unit of thought constituted through abstraction on the basis of characteristics common to a set of objects".

### D.2.13 "Concept Relationship"

A "Concept Relationship" is always a kind of "Object Class".

A "Concept Relationship" must be HAVING exactly one "concept relationship type description".

A "Concept Relationship" must be USED-BY one or more "Concept"s.

A "Concept Relationship" must be USING one or more "Concept"s.

A "Concept Relationship" is described as "A semantic link between concepts".

### D.2.14 "concept relationship type description"

EDITOR'S NOTES

(1) Would "DESCRIBING" be better than "USED-BY"?

(2) In the UML model, this is an attribute on only one relationship, not any number, as described below.

A "concept relationship type description" may be USED-BY any number of "Concept Relationship"s.

A "concept relationship type description" is described as "A description of the type of relationship between the concepts".

### D.2.15 "Conceptual Domain"

A "Conceptual Domain" may be CONTAINING any number of "Value Meaning"s.

A "Conceptual Domain" may be DOCUMENTED-BY at most one "Administered Component".

A "Conceptual Domain" may be HAVING at most one "dimensionality".

A "Conceptual Domain" may be RELATED-IN any number of "Conceptual Domain Relationship"s.

A "Conceptual Domain" may be RELATED-TO any number of "Conceptual Domain Relationship"s.

A "Conceptual Domain" may be REPRESENTED-BY any number of "Value Domain"s.

A "Conceptual Domain" may be SPECIFYING any number of "Administered data component Concept"s.

A "Conceptual Domain" is described as "A set of possible valid value meanings of an administered data component concept, expressed without representation".

### D.2.16 "Conceptual Domain Relationship"

A "Conceptual Domain Relationship" must be CONTAINED-IN one or more "Conceptual Domain"s.

A "Conceptual Domain Relationship" must be CONTAINING one or more "Conceptual Domain"s.

A "Conceptual Domain Relationship" must be TYPED-BY one or more "Conceptual Domain Relationship" TYPE DESCRIPTION"s.

A "Conceptual Domain Relationship" is described as "A relationship between two conceptual domains".

### D.2.17 "Contact"

EDITOR'S NOTE: The UML model gives separate names and definitions to the attribute capsules based on Contact.

A "Contact" may be "Contact"-FOR any number of "Stewardship"s.

A "Contact" may be "Contact"-FOR any number of "Submission"s.

A "Contact" must be "Contact"ED-AT exactly one "mail address".

A "Contact" must be LABELLED-BY exactly one "contact name".

A "Contact" may be LOCATED-AT at most one "cell phone number".

A "Contact" may be LOCATED-AT at most one "electronic mail address".

A "Contact" may be LOCATED-AT at most one "fax number".

A "Contact" may be LOCATED-AT at most one "pager number".

A "Contact" must be LOCATED-AT exactly one "phone number".

A "Contact" may be LOCATED-AT at most one "telex number".

A "Contact" may be REFERENCED-BY at most one "contact title".

A "Contact" is described as "An instance of a role of an individual or an organization (or organization part or organization person) to whom an information item(s), a material object(s) and/or person(s) can be sent to or from in a specified context".

### D.2.18 "contact name"

A "contact name" must be LABELLING one or more "Contact"s.

A "contact name" is described as "The name of the contact".



**D.2.19 "contact title"**

A "contact title" may be TYPING any number of "Contact"s.

A "contact title" is described as "The name of the position held by the contact".

**D.2.20 "Context"**

A "Context" may be DESCRIBED-BY at most one "Administered Component".

A "Context" must be DESCRIBED-BY exactly one "context description".

A "Context" may be DESIGNATED-BY at most one "context name".

A "Context" may be DESIGNATED-BY at most one "context version".

A "Context" may be USED-BY any number of "Designation"s.

A "Context" may be USED-FOR any number of "Definition"s.

Every "Context"

is associated uniquely with one combination of

a "context name" DESIGNATING the "Context"

and a "context version" USED IN the "Context".

A "Context" is described as "A universe of discourse in which a name or definition is used".

**D.2.21 "context description"**

A "context description" may be DESCRIBING at most one "Context".

A "context description" is described as "The textual description of the context".

**D.2.22 "context name"**

A "context name" may be DESIGNATING any number of "Context"s.

A "context name" is described as "The context name".

**D.2.23 "context version"**

A "context version" may be USED-IN any number of "Context"s.

A "context version" is described as "The unique version identifier of a context".

**D.2.24 "country identifier"**

A "country identifier" may be CLARIFYING any number of "Language"s.

A "country identifier" is described as "A country identifier further specifying the geographic region associated with the language".

**D.2.25 "csi relationship type description"**

EDITOR'S NOTE: The generated name was prefixed "cs" instead of "csi".

A "csi relationship type description" may be TYPING any number of "Classification Scheme Item Relationship"s.

A "csi relationship type description" is described as "A description of the type of relationship between a classification scheme item and one or more other classification scheme items in a classification scheme".

**D.2.26 "csi type name "**

A "csi type name " may be TYPING any number of "Classification Scheme Item"s.

A "csi type name " is described as "The name of the type of the classification scheme item value".

### D.2.27 "csi value"

A "csi value" may be REPRESENTING any number of "Classification Scheme Item"s.

A "csi value" is described as "An instance of a classification scheme item".

### D.2.28 "Administered data component"

EDITOR'S NOTE: In the UML model, the attribute encapsulating Representation Class is given a unique name.

A "Administered data component" may be CATEGORIZED-BY at most one "Representation Class".

A "Administered data component" may be DERIVED-FROM at most one "Administered data component Derivation".

A "Administered data component" must be DOCUMENTED-BY exactly one "Administered Component".

A "Administered data component" must be EXEMPLIFIED-BY one or more "Administered data component Example"s.

A "Administered data component" must be EXPRESSING exactly one "Administered data component Concept".

A "Administered data component" may be INPUT-TO any number of "Administered data component Derivation"s.

A "Administered data component" may be QUALIFIED-WITH at most one "representation class qualifier".

A "Administered data component" must be REPRESENTED-BY exactly one "Value Domain".

A "Administered data component" is described as "A unit of data for which the definition, identification, representation and permissible values are specified by means of a set of attributes".

### D.2.29 "Administered data component Concept"

EDITOR'S NOTE: In the UML model, the attributes encapsulating Object Class and Property are given unique names.

A "Administered data component Concept" must be DOCUMENTED-BY exactly one "Administered Component".

A "Administered data component Concept" may be EXPRESSED-BY any number of "Administered data component"s.

A "Administered data component Concept" must be HAVING exactly one "Conceptual Domain".

A "Administered data component Concept" may be HAVING at most one "Object Class".

A "Administered data component Concept" may be HAVING at most one "object class qualifier".

A "Administered data component Concept" may be HAVING at most one "Property".

A "Administered data component Concept" may be HAVING at most one "property qualifier".

A "Administered data component Concept" may be RELATED-IN any number of "Administered data component Concept Relationship"s.

A "Administered data component Concept" may be RELATED-TO any number of "Administered data component Concept Relationship"s.

A "Administered data component Concept" is described as "A concept that can be represented in the form of an administered data component, described independently of any particular representation".

### D.2.30 "Administered data component Concept Relationship"

A "Administered data component Concept Relationship" must be CONTAINED-IN one or more "Administered data component Concept"s.

A "Administered data component Concept Relationship" must be CONTAINING one or more "Administered data component Concept"s.

A "Administered data component Concept Relationship" must be TYPED-BY exactly one "dec relationship type description".

A "Administered data component Concept Relationship" is described as "The attribution of a relationship of an administered data component concept with another administered data component concept".

**D.2.31 "Administered data component Derivation"**

A "Administered data component Derivation" must be APPLYING exactly one "Derivation Rule".

A "Administered data component Derivation" must be DERIVING one or more "Administered data component"s.

A "Administered data component Derivation" must be INPUTTING one or more "Administered data component"s.

A "Administered data component Derivation" is described as "The relationship among a derived administered data component, the rule controlling its derivation, and the administered data component(s) from which it is derived".

**D.2.32 "Administered data component Example"**

A "Administered data component Example" must be EXEMPLIFYING one or more "Administered data component"s.

A "Administered data component Example" must be ILLUSTRATED-BY exactly one "administered data component example item".

A "Administered data component Example" is described as "Representative illustration of the administered data component".

**D.2.33 "administered data component example item"**

A "administered data component example item" may be ILLUSTRATING at most one "Administered data component Example".

A "administered data component example item" is described as "Actual illustrative case of the administered data component".

**D.2.34 "data identifier"**

A "data identifier" may be USED-IN any number of "Component Identifier"s.

A "data identifier" is described as "The unique identifier for an administered component within a registration authority".

**D.2.35 "Datatype"**

A "Datatype" may be DESCRIBED-BY at most one "datatype annotation".

A "Datatype" must be DESCRIBED-BY exactly one "datatype description".

A "Datatype" must be DESIGNATED-BY exactly one "datatype name".

A "Datatype" may be HAVING-MAX-SIZE-OF at most one "character quantity".

A "Datatype" may be SPECIFIED-FOR any number of "Value Domain"s.

A "Datatype" is described as "A set of distinct values, characterized by properties of those values and by operations on those values".

**D.2.36 "datatype annotation"**

A "datatype annotation" may be DESCRIBING any number of "Datatype"s.

A "datatype annotation" is described as ...

"[EDITOR'S NOTE: To be added]".

**D.2.37 "datatype description"**

A "datatype description" may be DESCRIBING at most one "Datatype".

A "datatype description" is described as "A description of the parameters of a datatype".

**D.2.38 "datatype name"**

A "datatype name" must be DESIGNATING exactly one "Datatype".

A "datatype name" is described as "A designation for the category of datatype".

**D.2.39 "date"**

## EDITOR'S NOTE

"date" is defined here as a generic attribute. The UML model uses "date" as a datatype for specific attributes in various classes, each with unique names: creation date,

- A "date" may be BEGINNING-USAGE-OF any number of "Permissible Value"s.
- A "date" may be BEGINNING-USAGE-OF any number of "Value Meaning"s.
- A "date" may be ENDING-USAGE-OF any number of "Permissible Value"s.
- A "date" may be ENDING-USAGE-OF any number of "Value Meaning"s.
- A "date" may be FOR-EFFECTIVE any number of "Administered Component"s.
- A "date" may be LAST-CHANGE-OF any number of "Administered Component"s.
- A "date" may be NOTING-CREATION-OF any number of "Administered Component"s.
- A "date" may be STOPPING any number of "Administered Component"s.
- A "date" is described as  
 "[EDITOR NOTE: To be added]".

**D.2.40 "dec relationship type description"**

- A "dec relationship type description" may be TYPING any number of "Administered data component Concept Relationship"s.
- A "dec relationship type description" is described as "The description of the type of relationship with another administered data component concept that this administered data component concept modifies, is modified by, or is otherwise linked with".

**D.2.41 "Definition"**

- A "Definition" must be DEFINING one or more "Administered Component"s.
- A "Definition" must be DESIGNATED-BY exactly one "definition text".
- A "Definition" must be USING one or more "Context"s.
- A "Definition" may be USING any number of "Language"s.
- Every "Definition"  
 is associated uniquely with one combination of  
 an "Administered Component" DEFINED BY the "Definition"  
 and a "Context" USED FOR the "Definition".
- A "Definition" is described as "The definition of an administered component within a context".

**D.2.42 "definition text"**

- A "definition text" must be DESIGNATING exactly one "Definition".
- A "definition text" is described as "The text of the definition".

**D.2.43 "Derivation Rule"**

- A "Derivation Rule" may be APPLIED-TO any number of "Administered data component Derivation"s.
- A "Derivation Rule" must be DESCRIBED-BY exactly one "derivation rule description".
- A "Derivation Rule" may be DOCUMENTED-BY at most one "Administered Component".
- A "Derivation Rule" is described as "The logical, mathematical, and/or other operations specifying derivation".

**D.2.44 "derivation rule description"**

A "derivation rule description" may be DESCRIBING at most one "Derivation Rule".

A "derivation rule description" is described as "The text of a specification of derivation".

**D.2.45 "Designation"**

A "Designation" must be DESIGNATED-BY exactly one "name".

A "Designation" must be NAMING one or more "Administered Component"s.

A "Designation" must be USING one or more "Context"s.

A "Designation" may be USING any number of "Language"s.

Every "Designation"

is associated uniquely with one combination of

an "Administered Component" DESIGNATED BY the "Designation"

and a "Context" USED BY the "Designation".

A "Designation" is described as "The designation of an administered component within a context".

**D.2.46 "dimensionality"**

A "dimensionality" may be SPECIFIED-FOR any number of "Conceptual Domain"s.

A "dimensionality" is described as "The dimensionality for a conceptuality".

**D.2.47 "electronic mail address"**

An "electronic mail address" may be LOCATING any number of "Contact"s.

An "electronic mail address" is described as "An e-mail address for correspondence with the contact".

**D.2.48 "Enumerated Domain"**

An "Enumerated Domain" is always a kind of "Value Domain".

An "Enumerated Domain" cannot also be a "Non-enumerated Domain".

An "Enumerated Domain" must be CONTAINING one or more "Permissible Value"s.

An "Enumerated Domain" is described as "A value domain that is specified by a list of all permissible values".

**D.2.49 "explanatory comment"**

An "explanatory comment" must be EXPLAINING one or more "Administered Component"s.

An "explanatory comment" is described as "Descriptive comments about the administered component".

**D.2.50 "fax number"**

A "fax number" may be LOCATING any number of "Contact"s.

A "fax number" is described as "A facsimile number for correspondence with the contact".

**D.2.51 "international code designator"**

An "international code designator" may be USED-BY any number of "registration authority identifier"s.

An "international code designator" is described as "The identifier of an organization identification scheme".

### D.2.52 "Language"

EDITOR'S NOTE: The UML model assigns unique names and definitions to each attribute capsule based on Language.

A "Language" must be DESIGNATED-WITH exactly one "language identifier".

A "Language" may be REFINED-WITH at most one "country identifier".

A "Language" may be USED-BY any number of "Definition"s.

A "Language" may be USED-BY any number of "Designation"s.

A "Language" may be USED-BY any number of "Registration Authority"s.

A "Language" may be USED-IN any number of "Reference Document"s.

Every "Language"

is associated uniquely with one combination of

a "language identifier" DESIGNATING the "Language"

and a "country identifier" CLARIFYING the "Language".

A "Language" is described as "The collection of identifiers required to identify a language or language variation for a particular purpose".

### D.2.53 "language identifier"

A "language identifier" must be DESIGNATING one or more "Language"s.

A "language identifier" is described as "Information in a terminological entry that indicates the name of a language".

### D.2.54 "mail address"

EDITOR'S NOTE: The UML model does not contain this class. Instead it contains two specific attributes: "contact mail address" and "organization mail address". It may well make sense to make "mail address" an attribute capsule in the UML model.

A "mail address" may be LOCATING any number of "Contact"s.

A "mail address" may be LOCATING any number of "Organization"s.

A "mail address" is described as ???

EDITOR'S NOTE: To be completed.

In the UML model, there are two definitions:

A "contact mail address" is described as "A description of where to send written correspondence to the contact".

An "organization mail address" is described as "The mailing address of the organization".

### D.2.55 "Maximum character quantity"

EDITOR'S NOTE: The name in the generated text was "character quantity". This has been changed to match the name in the UML model. "LIMITS-MAX-SIZE-OF" would read better as "LIMITING-MAX-SIZE-OF"

A "maximum character quantity" may be LIMITS-MAX-SIZE-OF any number of "Datatype"s.

A "maximum character quantity" is described as "The maximum number of storage units (of the corresponding datatype) to represent the administered data component value".

**D.2.56 "name"**

A "name" may be DESIGNATING any number of "Designation"s.

A "name" is described as "A name by which an administered component is known within a specific context".

**D.2.57 "Non-enumerated Domain"**

A "Non-enumerated Domain" is always a kind of "Value Domain".

A "Non-enumerated Domain" cannot also be an "Enumerated Domain".

A "Non-enumerated Domain" must be DESCRIBED-BY exactly one "non-enumerated domain description".

A "Non-enumerated Domain" is described as "A value domain that is not specified by a list of all permissible values"

**D.2.58 "non-enumerated domain description"**

A "non-enumerated domain description" may be DESCRIBING at most one "Non-enumerated Domain".

A "non-enumerated domain description" is described as "A description of a rule, reference, or range for a set of all permissible values for the value domain".

**D.2.59 "Object Class"**

An "Object Class" may be COMPONENT-OF any number of "Administered data component Concept"s.

An "Object Class" may be DOCUMENTED-BY at most one "Administered Component".

An "Object Class" is described as "A set of ideas, abstractions, or things in the real world that can be identified with explicit boundaries and meaning and whose properties and behaviour follow the same rules".

**D.2.60 "object class qualifier"**

An "object class qualifier" may be COMPONENT-OF any number of "Administered data component Concept"s.

An "object class qualifier" is described as "A qualifier of the administered data component concept object-class".

**D.2.61 "opi source"**

An "opi source" may be USED-IN any number of "registration authority identifier"s.

An "opi source" is described as "The source for the organization part identifier".

**D.2.62 "Organization"**

An "Organization" may be ADMINISTERING any number of "Stewardship"s.

An "Organization" may be "Contact"ED-AT at most one "mail address".

An "Organization" must be DESIGNATED-BY exactly one "organization name".

An "Organization" may be PROVIDING any number of "Reference Document"s.

An "Organization" may be SUBMITTING any number of "Submission"s.

An "Organization" is described as "A unique framework of authority within which a person or persons act, or are designated to act, towards some purpose"

**D.2.63 "organization identifier"**

An "organization identifier" may be USED-IN any number of "registration authority identifier"s.

An "organization identifier" is described as "The identifier assigned to an organization within an organization identification scheme, and unique within that scheme".

### **D.2.64 "organization name"**

An "organization name" may be DESIGNATING at most one "Organization".

An "organization name" is described as "A designation for the organization".

### **D.2.65 "organization part identifier"**

An "organization part identifier" may be USED-IN any number of "registration authority identifier"s.

An "organization part identifier" is described as "An identifier allocated to a particular organization part".

### **D.2.66 "origin"**

An "origin" may be RECORDED-FOR any number of "Administered Component"s.

An "origin" is described as "The source (document, project, discipline or model) that was the origin for the administered component".

### **D.2.67 "pager number"**

A "pager number" may be LOCATING any number of "Contact"s.

A "pager number" is described as "A pager number for the contact".

### **D.2.68 "Permissible Value"**

A "Permissible Value" must be BEGINNING-USAGE-ON exactly one "date".

A "Permissible Value" must be COMPRISED-OF exactly one "Value".

A "Permissible Value" must be COMPRISED-OF exactly one "Value Meaning".

A "Permissible Value" must be CONTAINED-IN one or more "Enumerated Domain"s.

A "Permissible Value" may be ENDING-USAGE-ON at most one "date".

A "Permissible Value" is described as "An expression of a value meaning in a specific value domain"  
"permissible value begin date"

### **D.2.69 "phone number"**

A "phone number" may be LOCATING any number of "Contact"s.

A "phone number" is described as "A telephone number for verbal correspondence with the contact".

### **D.2.70 "precision"**

A "precision" may be SPECIFIED-FOR any number of "Unit of Measure"s.

A "precision" is described as "The degree of specificity for a unit of measure".

### **D.2.71 "Property"**

A "Property" may be COMPONENT-OF any number of "Administered data component Concept"s.

A "Property" may be DOCUMENTED-BY at most one "Administered Component".

A "Property" is described as "A characteristic common to all members of an object class".

### **D.2.72 "property qualifier"**

A "property qualifier" may be COMPONENT-OF any number of "Administered data component Concept"s.

A "property qualifier" is described as "A qualifier of the administered data component concept property".



**D.2.73 "Reference Document"**

A "Reference Document" may be DESCRIBING any number of "Administered Component"s.

A "Reference Document" must be DESIGNATED-BY exactly one "reference document identifier".

A "Reference Document" must be PROVIDED-BY one or more "Organization"s.

A "Reference Document" may be TITLED any number of "reference document title"s.

A "Reference Document" may be TYPED-BY at most one "reference document type description".

A "Reference Document" may be USING any number of "Language"s.

A "Reference Document" is described as "A document that provides pertinent details for consultation about a subject".

**D.2.74 "reference document identifier"**

A "reference document identifier" must be DESIGNATING exactly one "Reference Document".

A "reference document identifier" is described as "An identifier for the reference document".

**D.2.75 "reference document title"**

A "reference document title" may be TITLING any number of "Reference Document"s.

A "reference document title" is described as "The title of the reference document".

**D.2.76 "reference document type description"**

A "reference document type description" may be TYPING any number of "Reference Document"s.

A "reference document type description" is described as "A description of the type of reference document".

**D.2.77 "Registrar"**

A "Registrar" must be DESIGNATED-BY exactly one "registrar identifier".

A "Registrar" must be INCLUDED-IN exactly one "Registration Authority".

A "Registrar" is described as "The representative of the Registration Authority".

**D.2.78 "registrar identifier"**

A "registrar identifier" must be DESIGNATING exactly one "Registrar".

A "registrar identifier" is described as "A designator for the registrar".

**D.2.79 "Registration Authority"**

A "Registration Authority" is always a kind of "Organization".

A "Registration Authority" must be CONTAINING one or more "Registrar"s.

A "Registration Authority" must be DOCUMENTING-IN one or more "Language"s.

A "Registration Authority" must be IDENTIFIED-BY exactly one "registration authority identifier".

A "Registration Authority" may be REGISTERING any number of "Administered Component"s.

A "Registration Authority" is described as "A body responsible for maintaining a registry".

**D.2.80 "registration authority identifier"**

A "registration authority identifier" must be IDENTIFYING exactly one "Registration Authority".

A "registration authority identifier" must be INCLUDING exactly one "international code designator".

A "registration authority identifier" may be INCLUDING at most one "opi source".

A "registration authority identifier" may be INCLUDING at most one "organization part identifier".

A "registration authority identifier" must be INCLUDING exactly one "organization identifier".

A "registration authority identifier" may be USED-IN any number of "Component Identifier"s.

A "registration authority identifier" is described as "An identifier assigned to a Registration Authority".

### **D.2.81 "registration status"**

A "registration status" may be SPECIFIED-FOR any number of "Administered Component"s.

A "registration status" is described as "A designation of the position in the registration life-cycle of an administered component".

### **D.2.82 "Representation Class"**

A "Representation Class" may be CATEGORIZING any number of "Administered data component"s.

A "Representation Class" may be DOCUMENTED-BY at most one "Administered Component".

A "Representation Class" is described as "The classification of types of representations".

### **D.2.83 "representation class qualifier"**

A "representation class qualifier" may be QUALIFYING any number of "Administered data component"s.

A "representation class qualifier" is described as A representation class qualifier to the designation - name to accurately label the administered data component".

### **D.2.84 "Stewardship"**

A "Stewardship" must be ADMINISTERED-BY exactly one "Organization".

A "Stewardship" must be "Contact"-TO exactly one "Contact".

A "Stewardship" must be MAINTAINING exactly one "Administered Component".

A "Stewardship" is described as "The relationship of an administered component, a contact, and an organization involved in the stewardship of the metadata".

### **D.2.85 "Submission"**

A "Submission" must be "Contact"-TO exactly one "Contact".

A "Submission" must be SUBMITTED-BY exactly one "Organization".

A "Submission" must be SUBMITTING exactly one "Administered Component".

A "Submission" is described as "The relationship of an administered component, a contact, and an organization involved in a submission of metadata".

### **D.2.86 "telex number"**

A "telex number" may be LOCATING any number of "Contact"s.

A "telex number" is described as "A telex number for correspondence with the contact".

### **D.2.87 "Unit of Measure"**

EDITOR'S NOTE: The UML model gives a separate name and definition to the attribute capsule based on Unit of Measure.

An "Unit of Measure" must be DESIGNATED-BY exactly one "unit of measure name".

An "Unit of Measure" must be SPECIFIED-WITH exactly one "precision".

An "Unit of Measure" may be USED-BY any number of "Value Domain"s.

Every "Unit of Measure"

is associated uniquely with one combination of  
 an "unit of measure name" DESIGNATING the "Unit of Measure"  
 and a "precision" SPECIFIED FOR the "Unit of Measure".

An "Unit of Measure" is described as "A system of measurement".

### **D.2.88 "unit of measure name"**

An "unit of measure name" must be DESIGNATING one or more "Unit of Measure"s.

An "unit of measure name" is described as "The name of a unit of measure".

### **D.2.89 "unresolved issue"**

EDITOR'S NOTE: In the generated text, "RECORDED" was misspelled. It has been manually corrected.

An "unresolved issue" may be RECORDED-FOR any number of "Administered Component"s.

An "unresolved issue" is described as "Any problem that remains unresolved regarding proper documentation of the administered component".

### **D.2.90 "Value"**

A "Value" must be DESIGNATED-BY exactly one "value item".

A "Value" may be USED-IN any number of "Permissible Value"s.

A "Value" is described as "A data value".

### **D.2.91 "Value Domain"**

All "Value Domain"s must be "Enumerated Domain"s,

or "Non-enumerated Domain"s.

A "Value Domain" may be CONTAINED-IN any number of "value domain relationship"s.

A "Value Domain" may be CONTAINING any number of "value domain relationship"s.

A "Value Domain" may be DESIGNATED-BY at most one "value domain identifier".

A "Value Domain" may be DESIGNATED-BY at most one "value domain name".

A "Value Domain" may be DOCUMENTED-WITH at most one "Administered Component".

A "Value Domain" must be REPRESENTING exactly one "Conceptual Domain".

A "Value Domain" may be REPRESENTING any number of "Administered data component"s.

A "Value Domain" may be USING at most one "Unit of Measure".

A "Value Domain" must be UTILIZING exactly one "Datatype".

A "Value Domain" is described as "A set of permissible values. It provides representation, but has no implication as to what administered data component concept the values may be associated with nor what the values mean".

### **D.2.92 "value domain identifier"**

A "value domain identifier" must be DESIGNATING exactly one "Value Domain".

A "value domain identifier" is described as "The identifier of a value domain".

### **D.2.93 "value domain name"**

A "value domain name" must be DESIGNATING exactly one "Value Domain".

A "value domain name" is described as "The name of a value domain".

#### **D.2.94 "value domain relationship"**

A "value domain relationship" must be RELATED-IN one or more "Value Domain"s.

A "value domain relationship" must be RELATED-TO one or more "Value Domain"s.

A "value domain relationship" must be TYPED-BY exactly one "vd relationship type description".

A "value domain relationship" is described as "A relationship between value domains".

#### **D.2.95 "value item"**

A "value item" must be DESIGNATING exactly one "Value".

A "value item" is described as "A representation of a value meaning in a specific value domain. The actual value".

#### **D.2.96 "Value Meaning"**

A "Value Meaning" must be BEGINNING-USAGE-ON exactly one "date".

A "Value Meaning" must be CONTAINED-IN one or more "Conceptual Domain"s.

A "Value Meaning" may be DESCRIBED-BY at most one "value meaning description".

A "Value Meaning" must be DESIGNATED-BY exactly one "value meaning identifier".

A "Value Meaning" may be ENDING-USAGE-ON at most one "date".

A "Value Meaning" may be USED-IN any number of "Permissible Value"s.

A "Value Meaning" is described as "The meaning or semantic content of a value".

#### **D.2.97 "value meaning description"**

A "value meaning description" may be DESCRIBING at most one "Value Meaning".

A "value meaning description" is described as "A description of a value meaning".

#### **D.2.98 "value meaning identifier"**

A "value meaning identifier" must be DESIGNATING exactly one "Value Meaning".

A "value meaning identifier" is described as "The unique identifier for a value meaning".

#### **D.2.99 "vd relationship type description"**

A "vd relationship type description" may be TYPING any number of "value domain relationship"s.

A "vd relationship type description" is described as "The description of a value domain relationship".

#### **D.2.100 "version"**

A "version" may be USED-IN any number of "Component Identifier"s.

A "version" is described as "The unique version identifier of the administered component".

## Annex E (Informative) XML Encoding for Data Registry Contents

### EDITOR'S NOTE

This Annex is maintained by Frank Olken (Lawrence Berkeley National Laboratory) [olken@lbl.gov](mailto:olken@lbl.gov)  
 Last revised: 2000-06-15 9:13 AM PDT]

This annex describes an XML (Extensible Markup Language) representation for the contents of a Data Registry of ISO/IEC 11179 Part 3. The starting point of this design is the normative UML metamodel for the data registry.

Topics:

- Encoding of the UML Metamodel for the registry
- Generation of an XML Schema to contain the registry contents
- Framing of XML queries against the registry contents
- Generation of an XML document to contain the registry contents

This document does not (at present) specify an encoding of the UML metamodel for the data registry.

This document does describe the generation of an XML Schema suitable for the specification of an XML document that would contain the contents of an ISO 11179 Data Registry.

This document does not describe how to formulate an XML Query Language query against the data registry contents - because such Query Language is not yet standardized by the W3C. However, it is envisaged that the XML Schema that we do specify could be used as the basis against which such queries could be formulated once XML QL is standardized. We acknowledge that the schema design embodied in this document has sacrificed some ease of querying in favour of simplicity of design. We note that one consideration in the design of the XML schema herein was to assure that the resulting document could be uniformly queryable.

This document does not specify in detail the process of encoding the contents of a data registry into an XML document. The XML schema we specify for such a document is of sufficient simplicity that the requisite process should be self-evident. In any case it will vary somewhat depending on implementation decisions made by the registry designer.

### E.1 Semi-automatic Encoding of 11179 Part 3 Metamodel into XML Schema

There are many different ways that the contents of an ISO 11179 Data Registry could be encoded into XML for data transfer. Interoperability requires that there be only one or at most a small number of standard ways to conduct such interchanges. There are number of reasons for specifying a semi-automatic algorithmic approach to the generation of the XML Schema:

- less tedious manual labor

- fewer opportunities for typographical errors
- easier to maintain as the metamodel is revised
- facilitates the automatic synthesis of programs to dump and restore the contents of the registry

EDITOR'S NOTE: Should the above reference to "data transfer" be talking about "metadata transfer?"

The process described here is not fully automated because we discovered that the UML model for the data registry is not sufficiently informative. Specifically, it does not explicitly specify keys for every object, we have inferred that "identifier" attributes of objects constitute keys. There are some exceptions in the registry metamodel.

### E.1.1 Use of XML Schema Language

We have adopted the most recent public specification of the XML Schema Language to specify the structure of the XML document to be used to encode the contents of a data registry, in lieu of an XML DTD (Document Type Definition). The XML Schema Language will soon supersede the DTDs. A DTD can be mechanically generated from the XML Schema, albeit with some loss of information (e.g., concerning types and keys). We anticipate that the XML Schema Language Recommendation will be adopted by the World Wide Web Consortium prior to the completion of the final International Standard for ISO 11179 Part 3.

### E.1.2 Why not XMI?

We had originally envisioned the use of XMI (XML Metadata Interchange Format) which has been standardized by the OMG (Object Management Group). This approach would have met our functional requirements, and has been the subject of much more detailed design studies. However, members of the L8<sup>3</sup> committee felt strongly that the XMI encodings were far too verbose. Furthermore, there was a reluctance to make the 11179 registry XML encoding dependent on an OMG standard.

The implication of the decision not to use XMI is that it will be necessary to write our own detailed specifications for the construction of the XML schema. Furthermore, it will be necessary to construct software (schema synthesizers, dumpers, loaders) which are specific to the ISO 11179 standard, rather than commercially available XMI tools.

### E.1.3 Design Criteria

- reasonably concise encoding of registry content
- globally (within the document) unique element tag names (required in XML 1.0)
- uniform treatment of UML object attributes, irrespective of whether they are simple or complex types
- where practicable, use of persistent keys (IDs, IDREFs) for encoding relationships

### E.1.4 Mapping UML graphs onto XML trees

The fundamental problem we confront in constructing an XML encoding of the data registry content is that the data registry schema is a graph, whereas XML (i.e., the nested element structure) is basically a

---

<sup>3</sup> EDITOR'S NOTE: This reference to L8 needs to be changed to SC32/WG2 for ISO.

tree. Thus we need to encode a collection of graphs (registry contents) into a forest of trees. Questions that need to be resolved include the following.

- Which objects should constitute the roots of the trees?
- Should we construct tall trees (Sequoias) or shallow trees (shrubs)?
- How should we decide which associations (relationships) to represent as containment relations (nested elements) and which to represent otherwise (IDREFs, etc.)?
- If we use IDREFs to link objects, what sort of IDs should we use? surrogate keys (artificial) or keys from the registry database?
- How do we construct element tag names that are globally unique to the entire document: for objects? for attributes?

We have decided to make each object in the registry metamodel into an element, with the attributes of an object becoming nested elements. This solves the problem of choosing the roots of the trees - every object is the root of its own tree. This design has the advantage of simplicity, ease of dumping from a relational DBMS, and uniform treatment of all object instances. However, it is likely to be cumbersome to query - in effect requiring multiple joins to reconstruct the deep structure of the database.

The shallow tree design (shrubs) requires that every association (relationship) be encoded using IDREFs - again a very simple design.

We propose to use IDs constructed from persistent keys from the data registry. The advantage of this approach is that it does not require a separately constructed symbol (for the surrogate keys), and the key (IDs) are persistent, and hence can be used for navigational queries.

XML 1.0 requires that tag names are globally unique within the document. However, it is common Modeling practice to reuse attribute names in multiple object classes. Hence we will prefix attribute names with the object class names to assure uniqueness. Because this has already been done in some portions of the registry data model, we will remove duplicate object class name prefixes wherever we encounter them.

## E.2 Schema Generation Rules

UML construct	XML construct
objects	elements
object class names	tag names - blanks changed to underscores
attributes	nested elements
attributes names	tag names - prefixed with object class name (separated by double hyphen)
associations	IDREFs

keys (see discussion)	IDs
specialization relations	IDREFs
containment relations	IDREFs

### E.2.1 Objects

Each object in the UML metamodel will be mapped into an element in the XML schema. The UML object class name will be used to construct the XML element tag name. Since XML tag names cannot contain blanks or colons (which are used to indicate namespace prefixes) we replace spaces with underscores and colons with hyphens. We assume that object class names are unique within the UML metamodel. Note that we have wrapped each collection of object instances in `<object_class_name_collection>` tags.

### E.2.2 Attributes of Objects

Each attribute of an object in the UML metamodel will be mapped into an element in the XML schema, nested within the object element. The UML attribute name will be prefixed with the object class name (separated by a hyphen) to construct the XML element tag name. Again we will replace spaces with underscores, and colons with double hyphens. If the object class name has already been prefixed to the attribute name in the UML, it will be removed, i.e., only one copy of the object class name will appear in the final element tag name.

Note that we do not propagate object class name into complex types (nested element structures), since we assume that complex types are uniquely named within the entire registry metamodel. Hence, we prefix the name of the complex type to the attributes of the complex type in order to assure that the corresponding element names are unique within the XML document. See examples below.

### E.2.3 Associations (relationships)

Associations will be mapped in empty (i.e., singleton) XML tags. Empty tags have no content or matching closing tags. The tag name will be generated by concatenating the object class name with the role name, separated by a double hyphen.

### E.2.4 Keys

Keys are not explicitly specified in UML. Hence we have inferred that attribute names ending in "identifier" are usually keys or part of keys. We concatenate all of the key attributes into a single string to construct a key (ID) for the object instance. We use double hyphens to separate key components and underscores to replace embedded spaces in the key components. Note that we prefix each key with the name of object class. This is to insure that these synthetic keys are globally unique across the entire document, which is required for IDs/IDREFs by XML 1.0.

## E.3 Example of Registry Content Encoded in XML

Note that this example is incomplete.



```

<ISO_11179_Data_Registry_Contents>

<Administered_Component_Collection>
  <Administered_Component
    ID="Administered_Component--DB--Lawrence_Berkeley_Lab--NERSC--unknown--data_ID--
Version_7.3" >
    <Administered_Component--identifier>
      <Component_Identifier--registration_authority_identifier>
        <Registration_Authority_Identifier--International_Code_Designator>
          DB
        </Registration_Authority_Identifier--International_Code_Designator>
        <Registration_Authority_Identifier--organization_identifier>
          Lawrence_Berkeley_Lab
        </Registration_Authority_Identifier--organization_identifier>
        <Registration_Authority_Identifier--organization_part_identifier>
          NERSC
        </Registration_Authority_Identifier--organization_part_identifier>
        <Registration_Authority_Identifier--OPI_source>
          DOE
        </Registration_Authority_Identifier--OPI_source>
      </Component_Identifier--registration_authority_identifier>
      <Component_Identifier--data_identifier>
        data_ID
      </Component_Identifier--data_identifier>
      <Component_Identifier--version>
        Version_7.3
      </Component_Identifier--version>
    </Administered_Component--identifier>
    <Administered_Component--registration_status>
      registered
    </Administered_Component--registration_status>
    <Administered_Component--administrative_status>
      unknown
    </Administered_Component--administrative_status>
    <Administered_Component--creation_date>
      1999-07-14
    </Administered_Component--creation_date>
    <Administered_Component--effective_date>
      1999-07-14
    </Administered_Component--effective_date>
    <Administered_Component--until_date>
      2000-07-14
    </Administered_Component--until_date>
    <Administered_Component--change_description>
      unchanged
    </Administered_Component--change_description>
    <Administered_Component--administrative_note>
      none
    </Administered_Component--administrative_note>
    <Administered_Component--explanatory_comment>
      no_explanation
    </Administered_Component--explanatory_comment>
    <Administered_Component--unresolved_issues>
      no_unresolved_issues
    </Administered_Component--unresolved_issues>
    <Administered_Component--origin>
      United_Nations
  </Administered_Component

```

```

    </Administered_Component--origin>
    <Administered_Component--described_by
      IDREF="Reference_Document--ISO_11179_Part_3--2000" />
    <Administered_Component--described_by
      IDREF="Reference_Document--ISO_11404" />
  </Administered_Component>
</Administered_Component_Collection>

<Reference_Document_collection>
  <Reference_Document ID="Reference_Document--ISO_111793_Part_3--2000" >
    <Reference_Document--identifier>
      ISO_111793_Part_3--2000
    </Reference_Document--identifier>
    <Reference_Document--type_description>
      ISO Standard
    </Reference_Document--type_description>
    <Reference_Document--language>
      English
    </Reference_Document--language>
    <Reference_Document--title>
      Specification and Standardization of Administered data components,
Part 3
    </Reference_Document--title>
    <Reference_Document--describing
      IDREF="Administered_Component--DB--Lawrence_Berkeley_Lab--NERSC--DOE--
data_ID--Version_7.3" />
  </Reference_Document>

  <Reference_Document ID="Reference_Document--ISO_11404" >
    <Reference_Document--identifier>
      ISO_11404
    </Reference_Document--identifier>
    <Reference_Document--type_description>
      ISO Standard
    </Reference_Document--type_description>
    <Reference_Document--language>
      English
    </Reference_Document--language>
    <Reference_Document--title>
      Language Independent Data Types
    </Reference_Document--title>
    <Reference_Document--describing
      IDREF="Administered_Component--DB--Lawrence_Berkeley_Lab--NERSC--DOE--
data_ID--Version_7.3"/>
  </Reference_Document>
</Reference_Document_collection>

</ISO_11179_Data_Registry_Contents>

```

## Annex F (Informative) Mapping the ISO/IEC 11179-3:1994 Basic Attributes to the new metamodel

### F.1 Introduction

ISO/IEC 11179-3:1994 lists 23 basic attributes of data elements, as shown in Figure 1.

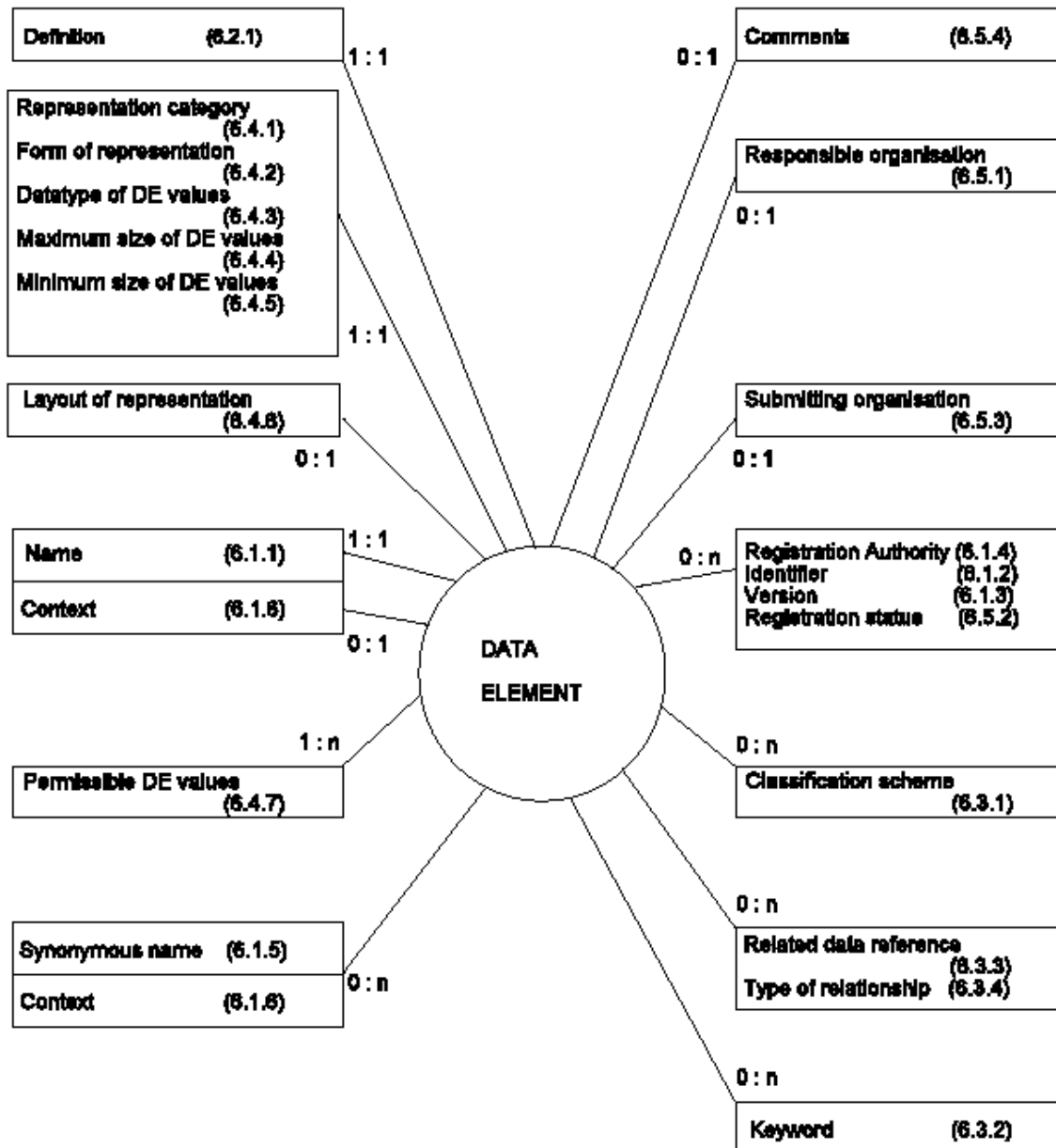


Figure F-1: Basic Attributes of Administered data components

The metamodel in the 11179-3 revision has been generalized to support not only administered data components, but associated “components of administered data components”, such as administered data component concepts, conceptual domains and value domains.

This annex attempts to map the original basic attributes to the new metamodel.

## F.2 Mapping the Basic Attributes to the Metamodel

The clause numbers used in the following headings are those from ISO/IEC 11179-3:1994.

To accomplish the mapping, it is necessary to navigate relationships within the model from one entity/object class to another. The “dot” notation:

“Administered data component” . “Administered Component” . “Designation” . “name”

specifies to follow the path in the model from “Administered data component” to “Administered Component” to “Designation” to the attribute “name”.

EDITOR’S NOTE: The following clauses used to be clause 6 of this part of the draft standard.

### F.2.1 Identifying attributes

#### F.2.1.1 Name

	<u>Prior</u>	<u>Current</u>
Attribute name:	Name	“Administered_data_component” . “Administered_Component” . “Designation” . “name”
Definition:	Single or multi word designation assigned to a data element.	A name by which an administered component is known within a specific context.
Obligation:	Mandatory	Mandatory
Data type:	Character string	String

### F.2.2 Identifier

	<u>Prior</u>	<u>Current</u>
Attribute name:	Identifier	“Administered_data_component” . “Administered_Component” . “Component_Identifier” . “data_identifier”
Definition:	A language independent unique identifier of a data element within a	The unique identifier for an administered component within a registration authority.

	Registration Authority.	
Obligation:	Conditional	Mandatory
Condition:	If the attribute 'Name of data element' (6.1.1) is not unique within a Registration Authority this attribute is mandatory.	
Data type:	Character	String
Comment:	Assignment of a unique identifier may be made mandatory as part of the registration procedure of any Registration Authority.	

### F.2.3 Version

	<u>Prior</u>	<u>Current</u>
Attribute name:	Version	"Administered_data_component" . "Administered_Component" . "Component_Identifier" . "version"
Definition:	Identification of an issue of a data element specification in a series of evolving data element specifications within a Registration Authority.	The unique version identifier of the administered component.
Obligation:	Conditional	Mandatory
Condition:	This attribute is mandatory if updates on attributes occur which meet the maintenance rules for allocating new versions as set by the Registration	
Data type:	Character	String

### F.2.4 Registration Authority

	<u>Prior</u>	<u>Current</u>
Attribute name:	Registration Authority	"Administered_data_component" . "Administered_Component" . "registration" . "Registration_Authority" . "Organization" . "organization_name"
Definition:	Any organisation authorized to register data elements.	<b>Registration_Authority:</b> A body responsible for maintaining a registry.

		<b>Organization_name:</b> A designation for the organization.
Obligation:	Conditional	Mandatory
Condition:	One Registration Authority shall be specified for each Identifier present.	
Data type:	Character string	String
Comment:		Registration Authority in the model is a class, with a unique identifier. However, the attribute corresponding most closely to that in the 1994 standard is the associated organization name.

**F.2.5 Synonymous name**

	<u>Prior</u>	<u>Current</u>
Attribute name:	Synonymous name	"Administered_data_component" . "Administered_Component" . "Designation" . "name"
Definition:	Single word or multi word designation that differs from the given name, but represents the same data element concept.	A name by which an administered component is known within a specific context.
Obligation:	Optional	Optional
Data type:	Character string	String
Comment:	Synonymous names are often familiar names in a certain application environment. If this is the case use attribute 'Context' (6.1.6) to specify the context. If more synonymous names occur the attributes 'Synonymous name' and 'Context' shall be specified as a pair.	An administered component may have multiple names in the same or different contexts. The distinction between "name" and "synonymous name" does not exist in the new metamodel. <b>There is currently no explicit way to indicate whether one name is preferred over another, if multiple names exist for the same context, though a note could be added to the definition.</b>

**F.2.6 Context**

	<u>Prior</u>	<u>Current</u>
Attribute name:	Context	"Administered_data_component" . "Administered_Component" . "Designation" . "Context" .

		<p>"context_description"</p> <p>OR</p> <p>"Administered_data_component" .  "Administered_Component" .  "Designation" . "Context" .  "context_name"</p> <p>OR</p> <p>"Administered_data_component" .  "Administered_Component" .  "Designation" . "Context" .  "Administered_Component" .  "Designation" . "name"</p> <p>OR</p> <p>"Administered_data_component" .  "Administered_Component" .  "Designation" . "Context" .  "Administered_Component" .  "Designation" . "definition_text"</p>
Definition:	<p>A designation or description of the application environment or discipline in which a name and/or synonymous name is applied or originates from.</p> <p>Note: The new metamodel differentiates designations from descriptions.</p>	<p><b>Context:</b> A universe of discourse in which a name or definition is used.</p> <p><b>context_description:</b> The textual description of the context.</p> <p><b>context_name:</b> The context name.</p> <p><b>name:</b> A name by which an administered component (in this case the Context) is known within a specific context (where the context for a context is probably the registry).</p> <p><b>definition_text:</b> The text of the definition (of the Context).</p>
Obligation:	Conditional	<p><b>Context:</b> Mandatory</p> <p><b>context_description:</b> Mandatory</p> <p><b>context_name:</b> Optional.</p> <p><b>name:</b> Conditional</p> <p><b>definition_text:</b> Conditional</p>
Condition:	This attribute is mandatory for each occurrence of the attribute 'Synonymous name' (6.1.5). This attribute is mandatory when the	If Context is treated as an Administered Component, it must be given a name and definition_text.

	attribute 'Name' (6.1.1) occurs in an information exchange.	
Data type:	Character string	String
Comment:	Assignment of the attribute 'Context' to the attribute 'Name' (6.1.1) may be made mandatory as part of the procedures of any Registration Authority.	In the metamodel, context_description and context_name exist as two separate attributes.

### F.3 Definitional attributes

#### F.3.1 Definition

	<u>Prior</u>	<u>Current</u>
Attribute name:	Definition	"Administered_data_component" . "Administered_Component" . "Definition" "definition_text"
Definition:	Statement that expresses the essential nature of a data element and permits its differentiation from all other data elements.	<b>Definition:</b> The definition of an administered component within a context.  <b>Definition_text:</b> The text of the definition.
Obligation:	Mandatory	Mandatory
Data type:	Character string	String

### F.4 Relational attributes

#### F.4.1 Classification scheme

	<u>Prior</u>	<u>Current</u>
Attribute name:	Classification scheme	"Administered_data_component" . "Administered_Component" . "Administered_Component_Classification" . "Classification_Scheme_Item" . "Classification_Scheme_Membership" . "Classification_Scheme" . "classification_scheme_type_name"  OR "Administered_data_component" .



		<p>“Administered_Component” .  “Administered_Component_Classification” . “Classification_Scheme_Item” .  “Classification_Scheme_Membership” .  “Classification_Scheme” . “Administered_Component” . “Designation” . “name”</p> <p>OR</p> <p>“Administered_data_component” .  “Administered_Component” .  “Administered_Component_Classification” .  “Classification_Scheme_Item” .  “Classification_Scheme_Membership” .  “Classification_Scheme” . “Administered_Component” . “identifier”</p>
Definition:	A reference to (a) class(es) of a scheme for the arrangement or division of objects into groups based on characteristics that the objects have in common, e.g. origin, composition, structure, application, function etc.	<p><b>Classification Scheme:</b> The descriptive information for the arrangement or division of objects into groups based on characteristics which the objects have in common.</p> <p><b>Classification_scheme_type_name:</b> The name of the type of classification scheme.</p> <p><b>name:</b> A name by which an administered component (in this case the Classification Scheme) is known within a specific context.</p>
Obligation:	Optional	<p><b>Classification Scheme:</b> Optional</p> <p><b>classification_scheme_type_name:</b> Conditional</p> <p><b>name:</b> Conditional</p> <p><b>identifier:</b> Conditional</p>
Condition	N/A	<p>If Classification Scheme is present, classification_scheme_type_name is mandatory.</p> <p>If Classification Scheme is treated as an Administered Component, name and identifier are mandatory.</p>
Data type:	Character string	String
Comment	The definition does not specify whether the reference is by name or identifier.	

**F.4.2 Keyword**

	<u>Prior</u>	<u>Current</u>
Attribute name:	Keyword	“Administered data component” . “Administered Component” . “Administered Component Classification” . “Classification Scheme Item” . “csi_value”
Definition:	One or more significant words used for retrieval of data elements.	<b>Classification Scheme Item:</b> An item of content in a classification scheme.  <b>Csi_value:</b> An instance of a classification scheme item.
Obligation:	Optional	Optional
Data type:	Character string	String
Comment:	This attribute can be used for recording keywords (search keys) associated with the data element in question.	The metamodel treats keywords as just another classification scheme.

**F.4.3 Related data reference**

	<u>Prior</u>	<u>Current</u>
Attribute name:	Related data reference	“Administered data component” . “administered data component concept expression” . “Administered data component Concept” . “administered data component concept relationship” . “Administered data component Concept” . “administered data component concept expression” . “Administered data component” . “Administered Component” . “Component Identifier” . “data identifier”  OR “Administered data component” . “administered data component concept expression” . “Administered data component Concept” . “administered data component concept expression” . “Administered data component” . “Administered Component” . “data identifier”
Definition:	A reference between the data element and any related data.	<b>Administered data component Concept:</b> A concept that can be represented in the form of an administered data component, described

		independently of any particular representation.  Administered data component Concept Relationship: <b>The attribution of a relationship of an administered data component concept with another administered data component concept.</b> <b>Data identifier:</b> The unique identifier for an administered component within a registration authority.
Obligation:	Optional	Optional
Data type:	Character string	String
Comment:	If this attribute occurs it shall be specified in pair with the attribute 'Type of relationship' (6.3.4).	The model does not support direct relationships between administered data components. Rather, an administered data component is an expression of an administered data component concept, and administered data component concepts may be related. Also, multiple administered data components may be related to the same administered data component concept, and are therefore indirectly related to each other.

#### F.4.4 Type of relationship

	<u>Prior</u>	<u>Current</u>
Attribute name:	Type of relationship	"Administered data component" . "administered data component concept expression" . "Administered data component Concept" . "administered data component concept relationship" . "dec relationship type description"
Definition:	An expression that characterizes the relationship between the data element and related data.	The description of the type of association with another administered data component concept that this administered data component concept modifies, is modified by, or is otherwise linked with.
Obligation:	Conditional	Conditional
Condition:	This attribute is mandatory if the attribute 'Related data reference' occurs.	This attribute is mandatory if the "Administered data component Concept Relationship" exists.

Data type:	Character string	String
Comment:	Examples of type of relationships are: 'qualifier of', 'qualified by', 'subject of', 'part of', 'physical condition', 'external reference', 'higher standard', 'data element concept'.	Where the relationship is that of two administered data components to one administered data component concept, this has to be inferred. There is no way to specify this explicitly.

## F.5 Representational attributes

### F.5.1 Representation category

	<u>Prior</u>	<u>Current</u>
Attribute name:	Representation category	"Administered data component" . "administered data component representation class"
Definition:	Type of symbol, character or other designation used to represent a data element.	The class of representation of an administered data component.
Obligation:	Mandatory	Mandatory
Data type:	Character string	Representation Class
Comment:	<p>1. The representation category shall be specified by the relevant standard.</p> <p>Examples of possible representation categories:</p> <ul style="list-style-type: none"> <li>- character representation (ISO/IEC 646)</li> <li>- character/symbol representation (ISO registration no. 143)</li> <li>- bar coded representation (EIA-556)</li> <li>- graphical representation</li> </ul> <p>2. Example: The instances of data element: 'consignment number' are bit patterns in EDI-messages and bar codes on physical packages. For this data element this attribute occurs twice, one with value: 'character representation (ISO/IEC 646)' and one with 'bar code representation EIA-556)'.</p>	The exact manner in which this is to be specified is unclear.

### F.5.2 Form of representation

	<u>Prior</u>	<u>Current</u>
Attribute name:	Form of representation	There appears to be no equivalent in the new metamodel.
Definition:	Name or description of the form of representation for the data element, e.g. 'quantitative value', 'code', 'text', 'icon'.	
Obligation:	Mandatory	
Data type:	Character string	
Comment:	<p>1. See Part 2 of this International Standard for appropriate terms ('property words' or 'class words') to be used.</p> <p>2. Example 1: For the data element named: 'country of origin code' this attribute contains: 'code'.</p> <p>3. Example 2: For the data element: 'product description' this attribute contains: 'text'.</p> <p>4. Example 3: For the data element: 'weight of consignment' this attribute contains: 'quantitative value'.</p>	

### F.5.3 Datatype of administered data component values

	<u>Prior</u>	<u>Current</u>
Attribute name:	Datatype of data element values	"Administered data component" . "administered_data_component representation" . "Value Domain" . "value_domain_datatype" . "Datatype" . "datatype_name"
Definition:	A set of distinct values for representing the data element value.	<p><b>Datatype:</b> A set of distinct values characterized by properties of those values and by operations on those values.</p> <p><b>datatype_name:</b> A designation for the category of datatype.</p>
Obligation:	Mandatory	Mandatory

Data type:	Character string	String
Comment:	<p>Examples: Possible instances are: 'character', 'ordinal number', 'integer', 'real', 'scaled', 'bit', 'rational'.</p> <p>Note: The examples suggest the attribute is intended to be the <u>name</u> of the datatype, whereas the definition implies it is a <u>set of values</u>.</p>	

**F.5.4 Maximum size of administered data component values**

	<u>Prior</u>	<u>Current</u>
Attribute name:	Maximum size of data element values	“Administered data component” . “administered_data_component representation” . “Value Domain” . “value_domain_datatype” . “Datatype” . “maximum_character_quantity”
Definition:	The maximum number of storage units (of the corresponding datatype) to represent the data element value.	The maximum number of storage units (of the corresponding datatype) to represent the administered data component value.
Obligation:	Mandatory	Optional
Data type:	Integer	Integer
Comment:	<p>1. Example 1:</p> <p>For data element: 'invoice number' the attribute 'datatype' has instance 'character'</p> <p>and the attribute 'maximum size of data element value' has value: '17'. The data element value of 'invoice number' shall have a maximum of 17 characters.</p> <p>2. The two attributes 'maximum and minimum (see 6.4.5) size of data element values' indicate whether data element values are 'fixed' (maximum and minimum size are equal) or 'variable' (maximum and minimum size vary).</p>	<p>Note: The name of the attribute suggests it applies only to character datatype, while the definition suggests it is applicable to any datatype.</p>

### F.5.5 Minimum size of administered data component values

	<u>Prior</u>	<u>Current</u>
Attribute name:	Minimum size of data element values.	This attribute has been deleted from the metamodel.
Definition:	The minimum number of storage units (of the corresponding datatype) to represent the data element value.	
Obligation:	Mandatory	
Data type:	Integer	
Comment:	<p>1. Example 1:</p> <p>For data element: 'product description' the attribute 'datatype' has instance 'character' and the attribute 'minimum size of data element value' has instance: '10'.</p> <p>The data element value of 'product description' shall have a minimum of 10 characters.</p> <p>2. The two attributes 'maximum (see 6.4.4) and minimum size of data element values' indicate whether data element values are 'fixed' (maximum and minimum size are equal) or 'variable' (maximum and minimum size vary).</p>	

### 5.6.5 Layout of representation

	<u>Prior</u>	<u>Current</u>
Attribute name:	Layout of representation	This attribute has been deleted from the metamodel.
Definition:	The layout of characters in data element values expressed by a character string representation.	
Obligation:	Conditional	
Condition:	If the data element is of the class 'quantitative data' this attribute is mandatory. If the attribute 'form of	

	representation' is 'code' the use of this attribute is recommended if the code representation has to have a specific structure or layout.	
Data type:	Character string	
Comment:	<p>1. For quantitative data it is necessary to distinguish between integers, decimal mark and floating point notations.</p> <p>Example:</p> <p>Integers may be indicated with 'n', for decimal mark the number of characters before and after the decimal mark are specified as: n(5).n(3), for floating point notations the layout convention for a value with exponents shall comply with ISO 6093: n(3).n(3)E2, where 'E2' stands for max. 2 digits for the power of 10.</p> <p>2. For code representations having a specific structure or layout the type of character for each position in the code structure is important for validation purposes.</p> <p>Example:</p> <p>The data element 'flight number' has an international code representation structure consisting of two alphabetic characters of the airline company followed by a three-digit number identifying the flight (from starting-point to destination).</p> <p>The contents of the attribute: 'layout of representation' is: 'AA999'.</p>	

**F.5.6 Permissible administered data component values**

	<u>Prior</u>	<u>Current</u>
Attribute name:	Permissible data element values	<p>"Administered data component" . "administered_data_component representation" . "Value Domain"</p> <p>Note: A Value Domain will be further sub-typed as "Enumerated" or "Non-enumerated". An Enumerated Domain has an associated "Permissible Value</p>



		Set", comprising the set of "Permissible Values", each of which has an associated Value Meaning.
Definition:	The set of representations of permissible instances of the data element, according to the representation form, layout, datatype and maximum and minimum size specified in the corresponding attributes. The set can be specified by name, by reference to a source, by enumeration of the representation of the instances or by rules for generating the instances.	<p>Value Domain: <b>A set of permissible values. It provides representation, but has no implication as to what administered data component concept the values are associated with, or what the values mean.</b></p> <p><b>Non-enumerated Domain:</b> A value domain that is not specified by a list of all permissible values.</p> <p><b>Enumerated Domain:</b> A value domain that is specified by a list of all permissible values.</p> <p><b>Permissible Value Set:</b> The set of Permissible Values for an Enumerated Domain.</p> <p><b>Permissible Value:</b> An expression of a Value Meaning in a specific Value Domain.</p> <p><b>Value Meaning:</b> The meaning or semantic content of a Value.</p>
Obligation:	Mandatory	Value Domain is Mandatory.
Data type:	Character string	Value Domain is a class with multiple attributes.
Comment:	<p>When the permissible data element values are an enumeration of coded representations each data element value and instance shall be presented as a pair.</p> <p>Example 1 : Permissible data element values of data element: 'Priority indicator':</p> <p><i>Permissible value instances:</i> 1 High, 2 Low, 3 None</p> <p>Example 2: Permissible data element values of data element: 'Currency qualifier':</p> <p><i>Permissible value instances</i></p> <p>1 insurance currency</p> <p>2 home currency</p>	Note that Values and Value Meanings are separated out in the new model. Each Value and each Value Meaning will be specified as separate attributes, with appropriate relationships.

	<p>3 invoicing currency</p> <p>4 reference currency</p> <p>5 target currency</p> <p>Example 3: A data element named: 'Country code, 2-alpha' has a domain that corresponds with the set of 2-alpha codes and corresponding names of countries appearing in the text of the current ISO 3166.</p> <p>Example 4: A data element named: 'Number of employees' may contain values which are integers, equal or greater than zero.</p> <p>Example 5: A data element named: 'name of head of department' has a domain that is selected from a company's register of employees and that corresponds with the formal organizational structure.</p> <p>Example 6: A data element named: 'Product description' has a domain that corresponds with the set of products for which 'product identifications' are allocated and that is controlled by the 'product management function', and that conforms to the rules for defining products as laid down in document 'xyz'.</p> <p>Example 7: data element 'radio frequency' has a domain that ranges from 3 kHz-300 GHz and conforms to IEC50-specifications.</p>	
--	--	--

## F.6 Administrative attributes

### F.6.1 Responsible organization

	<u>Prior</u>	<u>Current</u>
Attribute name:	Responsible organization	"Organization" . "Stewardship"
Definition:	The organization or unit within an organization that is responsible for the contents of the mandatory attributes by which the data element is specified.	<b>Organization:</b> A unique framework of authority, within which a person or persons act, or are designated to act, towards some purpose.

		<b>Stewardship:</b> The relationship of an Administered Component, a Contact and an Organization involved in the stewardship of metadata.
Obligation:	Optional	Conditional
Condition:	N/A	The relationship is mandatory if Administered Component information is maintained.
Data type:	Character string	Objectified relationship: Stewardship
Comment:	The organisation shall be considered as 'owner' of the data element.	

### F.6.2 Registration status

	<b><u>Prior</u></b>	<b><u>Current</u></b>
Attribute name:	Registration status	"Administered Component" . "registration status"
Definition:	A designation of the position in the registration life-cycle of a data element.	A designation of the position in the registration life-cycle of an administered component.
Obligation:	Conditional	Conditional
Condition:	This attribute is mandatory during the data element life-cycle specified by any Registration Authority.	The attribute is mandatory if Administered Component information is maintained.
Data type:	Character	String
Comment:	The type of registration status to be distinguished and the allocation of the registration status shall follow the rules that are described in the procedures for the registration of data elements (see Part 6 of this International Standard).	

### F.6.3 Submitting organization

	<b><u>Prior</u></b>	<b><u>Current</u></b>
Attribute name:	Submitting organization	"Organization" . "Submission"

Definition:	The organization or unit within an organization that has submitted the data element for addition, change or cancellation/withdrawal in the data element dictionary.	<p><b>Organization:</b> A unique framework of authority, within which a person or persons act, or are designated to act, towards some purpose.</p> <p><b>Submission:</b> The relationship of an Administered Component, a Contact and an Organization involved in a submission of metadata.</p>
Obligation:	Optional	Conditional
Condition:	N/A	The relationship is mandatory if Administered Component information is maintained.
Data type:	Character string	Objectified relationship: Submission

**F.6.4 Comments**

	<u>Prior</u>	<u>Current</u>
Attribute name:	Comments	"Administered Component" . "explanatory comment"
Definition:	Remarks on the data element.	Descriptive comments about the administered component.
Obligation:	Optional	Optional
Data type:	Character string	String

## Bibliography

- [1] ASC X12S/91-1000, *Electronic Data Interchange X12 Standards*, Data Interchange Standards Association. December 1991.
- [2] Bruce, Thomas. *Designing Quality Databases with IDEF1X Information Models*. Dorset House Publishing: 1992.
- [3] English, Larry P. Accountability to the Rescue. *Database Programming & Design* 6(5): 54-59; 1993 April.
- [4] Flavin, Matt. *Fundamental Concepts of Information Modeling*. Yourdon Press; 1981.
- [5] Fowler, Martin. *UML Distilled*. Addison Wesley Publishing Company: 1997.
- [6] Giordano, Robert & Barbara Von Halle. Database Design—Remember the Standards. *Database Programming & Design* 4(6): 11-13; June 1991.
- [7] Giordano, Robert & Barbara Von Halle. Database Design—The Heart of the Matter. *Database Programming & Design* 4(5) 15-18; May 1991.
- [8] Giordano, Robert. The Data 'Where' House. *Database Programming & Design* 6(9): 54-61; September 1993.
- [9] Halpin, Terry. *Conceptual Schema and Relational Database Design*. Prentice Hall: 1995.
- [10] ISO 1087:1990, *Terminology — Vocabulary*
- [11] ISO 5127-6:1983, *Documentation and Information – Vocabulary — Part 6: Documentary languages*
- [12] ISO TR9007:1987 *Information processing systems - Concepts and terminology for the conceptual schema and the information base*
- [13] ISO/IEC 10027:1990, *Information technology – Information Resource Dictionary System (IRDS) Framework*
- [14] ISO/IEC 10165-1:1993, *Information Technology — Open Systems Interconnection — Structure of management information: Management Information Model*
- [15] ISO/IEC 11404:1995, *Information technology – Language-Independent Datatypes*
- [16] ISO/IEC TR 9789:1994, *Information technology – Guidelines for the organization and representation of data elements for data interchange – Coding methods and principles*
- [17] Marks, Dana M. & Terry Moriarty. Enterprise View—Unravelling the Standards. *Database Programming & Design* 6(13): 68-69; December 1993.
- [18] Moriarty, Terry. Enterprise View—Repository and CASE: Harmony?, *Database Programming & Design* 7(4): 68-71; April 1994.
- [19] Moriarty, Terry. Enterprise View—The Next Generation of IRDS, *Database Programming & Design* 7(3): 67-69; March 1994.
- [20] National Institute of Standards and Technology Special Publication 500-208, *Manual for Data Administration*, edited by Judith J. Newton and Daniel C. Wahl, National Institute of Standards and Technology, 1993.

- [21] Tasker, Dan. *Fourth Generation Data: A Guide to Data Analysis for New and Old Systems*. Prentice-Hall; 1989.