

American National Standard
for Information Technology—
Metamodel for the Management of Shareable Data

Secretariat

Information Technology Industry Council (ITI)

Approved (to come)

American National Standards Institute, Inc.

Abstract

This standard specifies the structure of a data registry. The structure is stated in the form of a conceptual data model. A data registry is a place to keep facts about characteristics of data that are necessary to clearly describe, inventory, analyze, and classify data. A data registry supports data sharing with cross-system and cross-organization descriptions of common units of data. A data registry assists users of shared data to have a common understanding of a unit of data's meaning, representation, and identification.

This standard will be of interest to information managers, data administrators, and others who are responsible for making data shareable. It will also be of particular interest to manufacturers of repository and CASE tool products.

Notice:

This is a draft proposed American National Standard. As such, this is not a completed standard. The Technical Committee may modify this document as a result of comments received during public review and its approval as a standard.

Permission is granted to members of NCITS, its technical committees, and their associated task groups to reproduce this document for the purposes of NCITS standardization activities without further permission, provided this notice is included. All other rights are reserved. Any commercial or for-profit reproduction is strictly prohibited.

Contents

	Page
Foreword	v
1 Scope, purpose, and application	1
1.1 Scope	1
1.2 Purpose	1
1.3 Application	1
2 Normative references	2
3 Definitions	3
4 Introduction	4
5 Macro-view of the metamodel	6
6 Requirements for a conformant implementation	7
6.1 Conformance	7
6.2 Extensibility	7
7 Description of metamodel	8
7.1 Stewardship	9
7.1.1 Stewardship metamodel region components	9
7.1.2 Stewardship metamodel region attributes	10
7.2 Naming and Identification	11
7.2.1 Naming and identification metamodel region components	11
7.2.2 Naming and identification metamodel region attributes	11
7.3 Classification	12
7.3.1 Classification metamodel region components	12
7.3.2 Classification metamodel region attributes	13
7.4 Data element concept administration	14
7.4.1 Object class	14
7.4.2 Property	15
7.4.3 Data element concept	15
7.4.4 Data element concept metamodel administration region attributes	15
7.5 Conceptual and value domain administration	16
7.5.1 Conceptual domain	16

7.5.2.	Value domain.....	17
7.5.3.	Conceptual and value domain administration metamodel region attributes	17
7.6	Data element administration.....	19
7.6.1.	Data element administration metamodel region components	19
7.6.2.	Data element administration metamodel region attributes	20
8	Metamodel specification	21
8.1	Metamodel diagram	22
8.2	Modeling object description specifications	23
8.2.1	Metadata entity and relationship field description specification.....	23
8.2.2	Metadata attribute field description specification	23
8.3	Alphabetical list of metadata objects	24
Annex A	Modeling symbols.....	61
Annex B	Alternative modeling representations	66
B.1	IDEF1X representation model	66
B.2.	Object Role Modeling (ORM) – Natural language Information Analysis Method (NIAM) representation model.....	73
B.3.	Unified Modeling Language (UML) representation model	87
Annex C	Principles of managing shared data	95
Annex D	Data registry uses and users.....	101
Annex E	Conceptual and logical data models.....	103
Annex F	Bibliographic references	105

Figures

	Page
Figure 1 – Data registry basic structure.....	6
Figure 2 – Metamodel regions	8
Figure 3 – Stewardship metamodel region.....	9
Figure 4 – Naming and identification metamodel region	11
Figure 5 – Classification metamodel region	12
Figure 6 – Data element concept administration metamodel region.....	14
Figure 7 – Conceptual and value domain administration metamodel region.....	16
Figure 8 – Data element administration metamodel region	19
Figure 9 – Conceptual metamodel entity-relationship diagram.....	22
Figure A1 – Sample modeling diagram.....	61
Figure A2 – Entity modeling representation	61
Figure A3 – Relationship modeling representation	62
Figure A4 – Entity-relationship modeling representation	62
Figure A5 – Entity-relationship cardinality modeling representation	62
Figure A6 – Associative entity modeling representation	63
Figure A7 – Subtype modeling representation	63
Figure A8 – Overlapping subtype modeling representation	63
Figure A9 – Exhaustive subtype modeling representation.....	64
Figure A10 – Dependency modeling representation.....	64
Figure A11 – Composition modeling representation.....	65
Figure A12 – Entity-attribute modeling representation.....	65
Figure B1 – IDEF1X stewardship, naming and identification metamodel regions	68
Figure B2 – IDEF1X data element concept administration and classification administration metamodel regions.....	69
Figure B3 – IDEF1X conceptual and value domain administration metamodel region.....	70
Figure B4 – IDEF1X data element administration region.....	71
Figure B5 – IDEF1X type hierarchy	72
Figure B6 – ORM stewardship region	74
Figure B7 – ORM naming and identification region.....	75
Figure B8 – ORM classification region.....	76

Figure B9 – ORM data element concept administration region	77
Figure B10 – ORM conceptual domain and value domain administration region.....	78
Figure B11 – ORM data element administration region.....	79
Figure B12 – ORM type hierarchy.....	80
Figure B13 – UML stewardship region	88
Figure B14 – UML naming and identification region.....	89
Figure B15 – UML classification region.....	89
Figure B16 – UML data element concept administration region	90
Figure B17 – UML conceptual and value domain administration region	91
Figure B18 – UML data element administration region.....	92
Figure B19 – UML full model	93

Foreword (This foreword is not part of American National Standard X3.285-1998).

ANSI Accredited Standards Committee X3 Technical Committee L8, Data Representation, develops standards and guidelines for the semantic content and the syntax of units of data shared among disparate users and information processing systems.

This standard specifies the structure of a data registry. The structure is stated in the form of a conceptual data model.

In a data model, an entity is any concrete or abstract thing of interest to the business. An attribute is a characteristic of an entity that the business chooses to record as data. For each type of entity, there are often many attributes of interest to the business.

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. It is one layer more abstract than a logical data model that describes a particular computer-based system. The conceptual data model provides an excellent place to start modeling data within the sphere of interest. It is also the most viable level at which to integrate different data models because object representational differences are excluded.

A data registry is a place to keep facts about characteristics of data that are necessary to clearly describe, inventory, analyze, and classify data. A data registry supports data sharing with cross-system and cross-organization descriptions of common units of data. A data registry assists users of shared data to have a common understanding of a unit of data's meaning, representation, and identification.

This data model describes the schema of a registry where descriptions of shareable data are stored. The descriptions of units of data are considered "metadata". A data model of metadata is called a "metamodel". Thus, this is a "Conceptual Metamodel".

The main text of this standard begins with clause 4. However, to understand more fully the principles underlying this metamodel, it would be beneficial to first examine annex C – *Principles of managing shared data*. It may also be useful to read annex D – *Data registry uses and users* describing many of the uses and users of a data registry. Clause 5 provides a simplified overview of the structure of a data registry. The specifications of the structure begin in clause 6 — what is required to be considered a conformant implementation. Clause 7 describes the metamodel in narrative form and includes entity-relationship diagrams with attributes. The symbols used in the entity-relationship diagrams are explained in annex A. Clause 8 contains details of the data model in the form of a data dictionary. Each entity, relationship, and attribute is fully documented. The combination of the entity-relationship diagram along with the data dictionary is the metamodel. Annexes A to F of this American National Standard are for information only.

[suggestions referral statement]

[approval statement]

NCITS MEMBERSHIP

Ms. Karen Higginbottom, NCITS Chairman (Acting)
Ms. Karen Higginbottom, NCITS Vice Chairman
Ms. Kate McMillan, NCITS Secretary

<i>Organization Represented</i>	<i>Name of Representative</i>
AMP Incorporated	Mr. John Hill
	Mr. Charles Brill (Alt.)
Apple Computer Inc.	Mr. David K. Michael
	Mr. Jerry Kellenbenz (Alt.)

AT&T	Mr. Thomas F. Frost Mr. Paul Bartoli (Alt.)
Bull HN Information Systems Inc	Mr. Patrick L. Harris.
Compaq Computers	Mr. Stephen Heil Mr. Steve Park (Alt.)
Digital Equipment Corporation	Mr. Scott K. Jameson Mr. Richard Hovey (Alt.)
Eastman Kodak Company	Mr. Michael Nier
Hewlett-Packard	Ms. Karen Higginbottom Mr. Donald C. Loughry (Alt.)
Hitachi America Ltd.	Mr. John Neumann Mr. Kei Yamashita (Alt.)
Hughes Aircraft Company	Mr. Harold Zebrack
IBM Corporation	Mr. Ronald Sellitti Mr. Joel Urman (Alt.)
Institute for Certification of Computer Professionals (ICCP)	Mr. Kenneth Zemrowski
Imation	Mr. Philip E. Friedlund
Lucent Technologies	Mr. Herbert Bertine Mr. Tom Rutt (A)
National Communications Systems	Mr. Dennis Bodson Mr. William Olden (A) Mr. Frank McClelland (A)
National Institute of Standards & Technology	Mr. Michael D. Hogan (P)
Panasonic Technologies	Mr. Y. Machida
Share Inc.	Mr. Gary Ainsworth Mr. David Thewlis (Alt.)
Sony	Mr. Masataka Ogawa Mr. Michael Deese (Alt.)
Storage Technology Corporation	Mr. Joseph S. Zajackowski
Sun Microsystems Inc.	Mr. Gary S. Robinson
Sybase Inc.	Mr. Donald R. Deutsch
Texas Instruments Inc.	Mr. Clyde Camp Mr. Fritz Whittington (Alt.)
Unisys Corporation	Mr. Arnold Winkler Mr. Stephen P. Oksala (Alt.)
U.S. Department of Defense	Mr. Jerry L. Smith Mr. C.J. Pasquariello (Alt.)
U.S. Department of Energy	Ms. Carol S. Blackston Mr. Bruce R. White (Alt.)
Xerox Corporation	Mr. John Flannery Mr. Roy Pierce (Alt.)

The L8 Technical Committee on Data Representation, which developed this standard, had the following members:

Phong Ngo, Chair, Science Applications International Corporation
 Daniel W. Gillman, Special Interest Group Chair, U.S. Bureau of the Census
 Judith Newton, Special Interest Group Vice Chair, National Institute of Standards and Technology

Battelle Memorial Institute	Douglas D. Mann
Bell Communications Research	M. Paul Hawes
	Paul R. Levine, (Alt.)
Lawrence Berkeley National Laboratory	Frank Olken
	John McCarthy, (Alt.)
Los Alamos National Laboratory	Douglas Evalenko
Meta4 Incorporated.....	Revere Perkins
	William P. Doyle (Alt.)
Mitretek Systems	Andrew Schoka
National Institute of Standards and Technology	Judith J. Newton
	Bruce Rosen, (Alt.)
	William P. LaPlant Jr. (Liasion)
Paladin Integration Engineering	Burton G. Parker
Science Applications International Corporation	Phong Xuan Ngo
	Joseph Leahy (Alt.)
self	Henry Heffernan
self	William H. Kenworthy, Jr.
San Diego Supercomputer Center	Arcot Rajasekar
University of South Carolina	Michael N. Huns
	Larry M. Stephens (Alt.)
U.S. Bureau of the Census	Daniel W. Gillman
	Martin Appel (Alt.)
U.S. Environmental Protection Agency	Bruce E. Bargmeyer, Vice Chair
	Larry Fitzwater (Alt.)
U.S. Geological Survey	Roger L. Payne,
	Louis A. Yost IV (Alt.)
U.S. Naval Observatory	Brent A. Archinal,
U.S. Veterans Administration	Philip H. Diamond

American National Standard
for Information Technology—

Metamodel for the Management of Shareable Data

1 Scope, purpose, and application

1.1 Scope

The goal of data registration is the maintenance of the meaning, representation, and identification of units of data over time and distance. Coherence and persistence of data descriptions and usage depend on accurate documentation. The process for documenting data descriptions is data registration and the place for documenting these descriptions is a data registry.

A data registry is a place to keep facts about characteristics of data that are necessary to clearly describe, inventory, analyze, and classify data. A data registry supports data sharing with cross-system and cross-organization descriptions of common data. Units of shareable data have precise identifiers, meanings, structures, and values. They are consistently deployed among users and systems and are centrally administered within an organization. Data registries may be organized into federations for interchange among many enterprises. A data registry assists users of shared data to have a common understanding of a unit of data's meaning, representation, and identification. Just as a data registry may assist users in understanding like units of data, registries also assist in the understanding of differences of similar but different units of data.

This standard specifies the structure of a data registry. The structure is stated in the form of a conceptual data model.

1.2 Purpose

International Standard ISO/IEC 11179, parts one to six, *Information technology — Specification and standardization of data elements*, developed by ISO/IEC JTC 1/SC 14 Data Element Principles, describes the attribution, classification, definition, naming, identification, and registration of data elements. This standard expands upon ISO/IEC 11179 by providing a comprehensive structure for those aspects that demonstrate their relationships to each other. That is the purpose of this American National Standard.

1.3 Application

A metamodel is necessary for coordination of data representation between tools that store, manipulate and exchange data. The metamodel enables Computer Aided Software Engineering (CASE) 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 CASE 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.

2 Normative references

The following standards contain provisions that, through reference in this text, constitute provisions of this American National Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this American National Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

ISO/IEC DIS 11179-1, *Information technology — Specification and standardization of data elements — Framework for the specification and standardization of data elements*

ISO/IEC DIS 11179-2, *Information technology — Specification and standardization of data elements — Classification for data elements*

ISO/IEC 11179-3:1994, *Information technology — Specification and standardization of data elements — Basic attributes of data elements*

ISO/IEC 11179-4:1995, *Information technology — Specification and standardization of data elements — Rules and guidelines for the formulation of data definitions*

ISO/IEC 11179-5:1995, *Information technology — Specification and standardization of data elements — Naming and identification principles for data elements*

ISO/IEC 11179-6:1997, *Information technology — Specification and standardization of data elements — Registration of data elements*

3 Definitions

For the purposes of this American National Standard, the following definitions apply.

- 3.1 attribute:** A characteristic of an object or entity.
- 3.2 concept:** A unit of thought constituted through abstraction on the basis of characteristics common to a set of objects.
- 3.3 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.
- 3.4 data:** A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means.
- 3.5 data element concept:** A concept that can be represented in the form of a data element, described independently of any particular representation. The human perception of a property class of an object class, identified and described independently of any particular representation.
- 3.6 data element:** A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes. A unit of data that in a certain context is considered indivisible.
- 3.7 data model:** A description of the organization of data in a manner that reflects an information structure.
- 3.8 data registry:** A set of files (paper, electronic, or a combination) containing the administered components and the associated information.
- 3.9 entity:** Any concrete or abstract thing of interest, including associations among things. Also see *object class*.
- 3.10 identifier:** An unambiguous name for an object within a given context.
- 3.11 metadata:** Data that defines and describes other data.
- 3.12 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 behavior follow the same rules.
- 3.13 permissible value:** An expression of a value meaning in a specific value domain. A value.
- 3.14 property:** A peculiarity common to all members of an object class.
- 3.15 relationship:** Business-based association between or among entities.
- 3.16 schema:** A description or representation of the structure of data.
- 3.17 shareable data:** Data that has precise identifiers, meaning, structures, and values.
- 3.18 value domain:** A set of permissible values. It provides representation, but has no implication as to what data element concept the values are associated nor what the values mean.

4 Introduction

This standard outlines the information elements associated with a data element concept that need to be available for determining the meaning of a data element that is to be shared between systems. The standard is a complement to the six-part ISO/IEC 11179 standard that describes the organization of a data registry for managing the semantics of data elements in data systems.

This standard focuses on the individual data element concept from the perspective of the semantic entailment of the concept. This is apart from the multiple names or terms that may be used to refer to the concept in different data systems, and independent of the classification schemes in which this concept may be included in different usage contexts. This standard does not specify a coding scheme or a representation for the metadata elements needed to maintain this information about the data element concept in a data system.

Organizations and communities of discourse that share like interests and professional disciplines undertake to manage their data and information assets by developing approved terminologies for use within the organization or community. For data elements that are represented electronically for storage, communication, and presentation, a data dictionary or catalog provides an organizing tool for improving the manageability of the data assets. For managing access to data elements, organizations and communities of discourse have developed thesauri that describe the relationships between the information elements referred to in the terminologies.

The terms selected for terminologies and related to one another in thesauri are used to identify and refer to facts, ideas, rules, or instructions. When these facts, ideas, rules, or instructions are represented in digital formats for storage and communications, therefore in a formalized manner suitable for storage, communication, interpretation, or processing by machines prior to the subsequent presentation of the data in the approved terminology or a form that is understandable to humans, the representations are "data". (This definition refers to a group of facts taken as a unit; thus it is used with a singular verb.) A data dictionary, therefore, is a specialized type of database containing metadata, which is managed by a "data dictionary system". A data dictionary is considered to be a repository of information describing the characteristics of data used to design, monitor, document, protect, and control data in information systems and databases.

The idea of a data dictionary, as a common catalog of shareable data, has been around for quite a while. The dictionary or catalog contains, for each data element, descriptions of the type of data (i.e., metadata), including the allowed values or domain of the element, but does not contain any information about instances of data. Originally, in its most rudimentary form, this catalog was called a "data dictionary". More recently in data engineering standards, the scope of the catalog was extended and renamed a "data encyclopedia". The further evolution of this catalog's contents to include all of the information necessary for clearly describing, inventorying, analyzing, and classifying the data elements in accord with their semantic relationships and domains, renamed this kind of extended catalog as a "data repository".

A data registry is a place to keep characteristics of data that are necessary to clearly describe, inventory, analyze, and classify data. A data registry supports data sharing with cross-system and cross-organization descriptions of common units of data. A data registry assists users of shared data to have a common understanding of a unit of data's meaning, representation, and identification.

Descriptive terminology escalation has caused "data" to become understood as partially equivalent to "information", resulting in the usage of the term "information repository" to describe the data registry that includes definitions and specifications of the semantics and context of the data elements. For proponents of this terminology, an information repository is a database application that contains information about systems, organizations, networks, business processes, information flows, and other facts about the enterprise's information resources. To many however, these terms "data" and

"information" have different meanings. In this standard, such differences in nuance are not debated. The fuller capability is called simply a "data registry". The registration of a data element is simply the assignment of an unambiguous identifier to the data element in a way that makes the assignment available to interested parties (ISO/IEC 11179-6).

A data element is a unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes (ISO/IEC 11179-3). A data element dictionary is an information resource that specifies, defines, and lists all relevant data elements (ISO/IEC 11179-3). A data element is a single unit of data that in a certain context is considered indivisible. It cannot be decomposed into more fundamental segments of data that have useful meanings within the scope of the enterprise. A data element concept is a human perception of a property of an object set, identified and described independently of any particular representation. And a data model is a description of the organization of data in a manner that reflects an information structure. In summary, a data registry is definitely more than just a data dictionary. A data dictionary is not adequate for the purposes stated above since it has no structure.

A data registry contains the metadata that is necessary to clearly describe, inventory, analyze, and classify data. Data registries can support data sharing with cross-system and cross-organization descriptions of common units of data if these descriptions are understood in the same way by users and organizations in the community of discourse, along with information on the differing contexts and assumptions of the different sharing organizations. A data registry gives users of shared data a common understanding of a unit of data's meaning, representation, and identification. For this purpose, a data registry need not be a database containing all the metadata found in a repository. A data registry is only a sub-set of the complete metadata that can be included in a data (or information) repository. However, that metadata sub-set needs to be structured in a way that supports the administration and retrieval of registered data.

Confirming that the data to be shared between organizations is interpreted correctly in the different user contexts and sharing organizations is not readily feasible without verification of the comparability of the metamodel instances of the different data registries.

5 Macro-view of the metamodel

Data modeling is founded on the theory that all data describes properties (attributes) of objects in the natural world. Data represents the properties of these things. The basic units of data are data elements. This metamodel uses many of the same conceptual data structures used in data modeling.

To specify data elements, relevant metadata of each data element shall be documented to ensure its representation of the natural world item is consistent, accurate, and understandable. The documentation should support the following:

“Any meaningful exchange of utterances depends upon the prior existence of an agreed upon set of semantic and syntactic rules.” ISO TR 9007:1987, *Information processing systems - Concepts and terminology for the conceptual schema and the information base.*”

A data registry shall be well organized so that the metadata can be easily analyzed and retrieved. In the registry, the description of each data element and its components shall be. Each data element shall have a value domain that shall be documented. In addition, each data element shall be documented as a property of an object class.

Metadata stored in this way is easy to locate. This structure also minimizes the manual analysis needed to identify redundant data elements. In addition, this structure helps to prevent creating new data elements that are redundant with ones that are already registered.

The data registry should contain descriptions of *shareable data* — unassociated with any specific application. The registry should support the notion of ISO TR 9007 exchange of utterances such that recipients of utterances must use only these registry documented rules to interpret the received utterances, if it is to mean the same as that which was meant by the utterer.

Figure 1 provides a simplified overview of the more important components of the data registry.

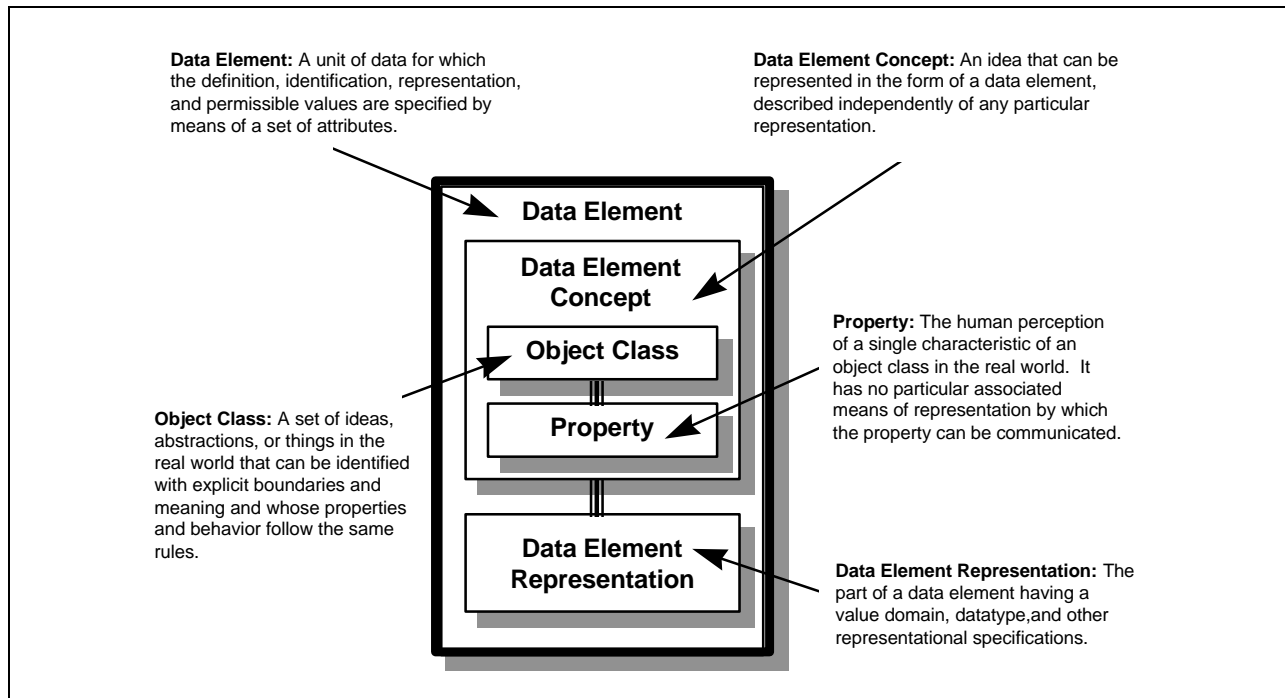


Figure 1 – Data registry basic structure

6 Requirements for a conformant implementation

6.1 Conformance

This document specifies the minimum features of a metamodel for the content of a data registry. A conformant implementation shall fully support the structure and content depicted in figure 9 – Conceptual metamodel entity-relationship diagram and further detailed in clause 8. To be conformant with this standard, the database will support the complete structure and content specified by this data model. The metamodel may be extended. However, an installation is not allowed to modify the minimal schema provided by this conceptual metamodel.

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 data 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. Annex E (informative) describes some of the normal translations, additions, and decisions when converting a conceptual data model to a logical data model.

A conformant data registry may be a stand-alone product, or may be implemented as part of an Information Resource Dictionary System (IRDS), any other information repository, or database.

6.2 Extensibility

It is not expected that this schema will completely satisfy 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.

7 Description of metamodel

The metamodel can be divided into six functional regions for descriptive purposes, as shown in figure 2. The primary region is the Stewardship 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, Data Element Administration, Data element concept administration, and Conceptual and Value Domain Administration. 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. Data elements, data element concepts, conceptual domain, and value domains are administered in their respective regions under the stewardship, naming, and classification support regions.

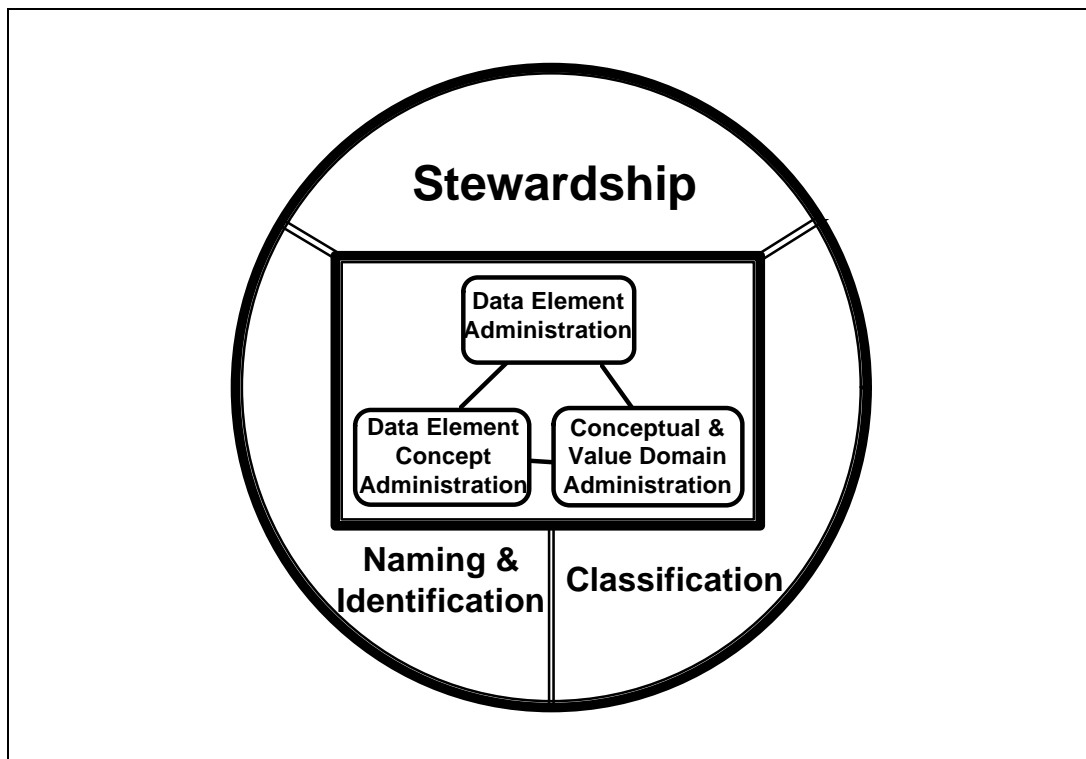


Figure 2 – Metamodel regions

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 subsections detail the six regions. The complete metamodel diagram is represented graphically in figure 9 in clause 8. Annex A contains an explanation of the symbols used in the following model diagrams. Clause 8 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 2. In the text of the section, 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 gray background.

7.1 Stewardship

The primary region is the Stewardship 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 3 represents the Stewardship region.

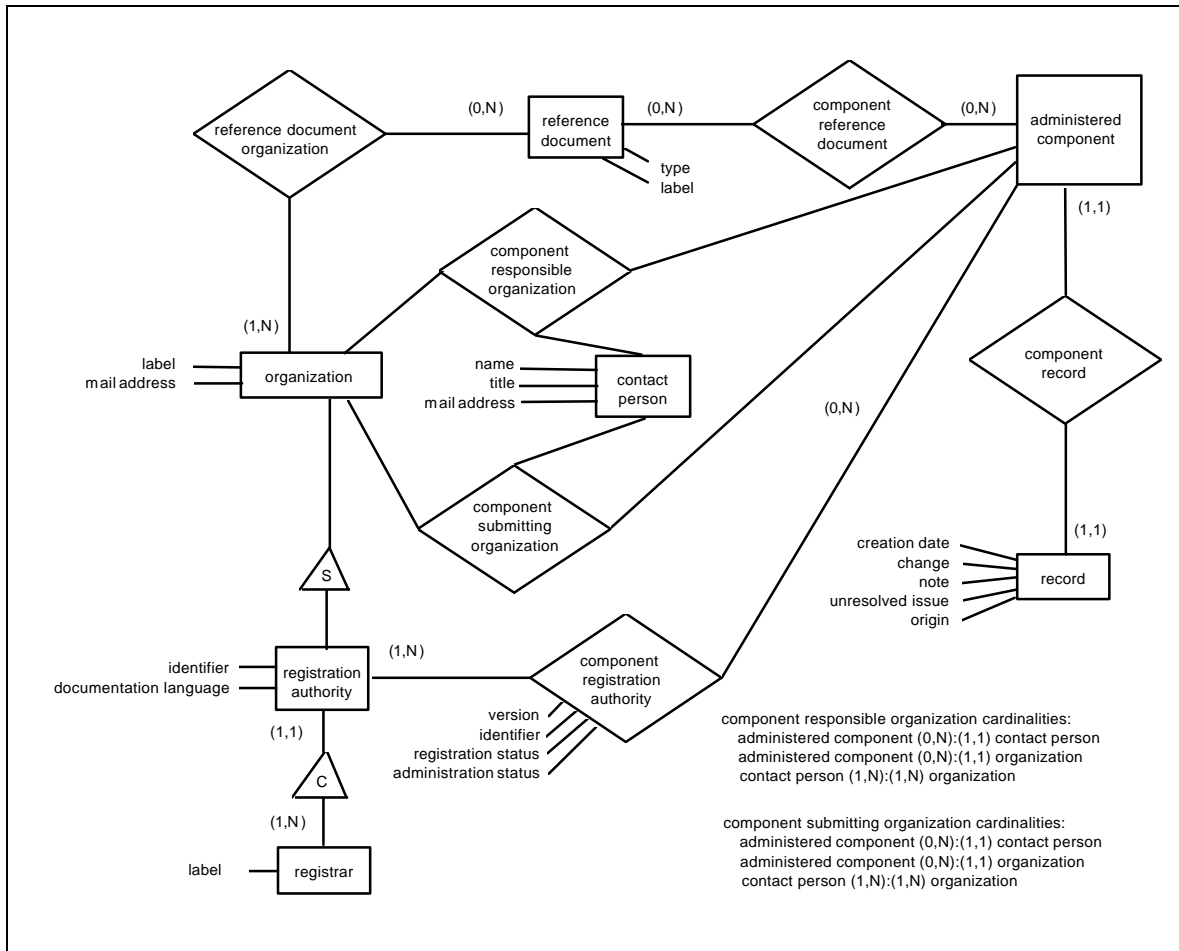


Figure 3 – Stewardship metamodel region

7.1.1. Stewardship metamodel region components

An *administered component* is a data element or its components (classification scheme, value domain, data element 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 a *data element*, *classification scheme*, *value domain*, *rule*, or *classified component*. Each of these subtypes has the relationships of the *administered component*.

A *registration authority* is an 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*, an *administered component* shall have a (*component registration authority*-) *identifier* 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*, *data element*, *data element concept*, *conceptual domain*, etc.). Each *administered component* in the owner's data registry shall have a (*component registration authority* -) *registration status* indicating the point in a registration life cycle applying to it. Each *administered component* in the owner's data registry shall also have a (*component registration authority* -) *administrative status* indicating the point in the registration processing process. A *registration authority* may register (*component registration authority*) many *administered components*. And in turn many *registration authorities* may register an *administered component*.

When an *administered component* is submitted to the data 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 data registry.

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

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.

7.1.2. Stewardship metamodel region attributes

The attributes in the Stewardship 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>
<i>component registration authority - administrative status</i>	One per <i>administered component</i>
<i>component registration authority - registration status</i>	One per <i>administered component</i>
<i>component registration authority - identifier</i>	One per <i>administered component</i>
<i>component registration authority - 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>record - change</i>	One conditional per <i>administered component</i>
<i>record - creation date</i>	One per <i>administered component</i>

<i>record - note</i>	One optional per <i>administered component</i>
<i>record - origin</i>	One optional per <i>administered component</i>
<i>record - 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>

7.2. Naming and Identification

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 4 represents the Naming and Identification region.

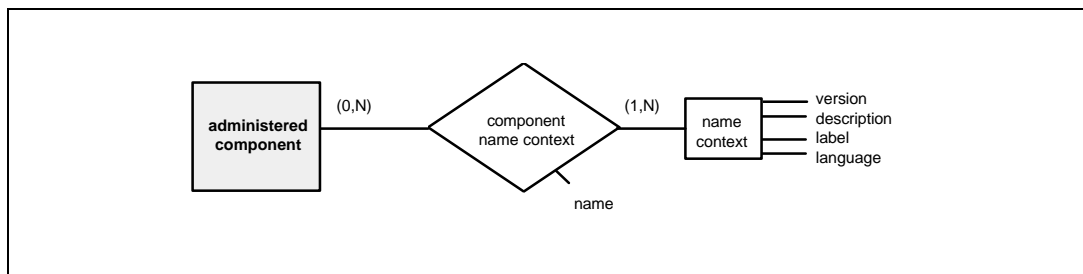


Figure 4 – Naming and identification metamodel region

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

7.2.1. Naming and identification metamodel region components

Each *administered component* has one or more names. Each name has a *name 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 a *data element* and a *value domain* cannot be the same within the same *name context*). A *name context* may be used by many *administered components*.

7.2.2. 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>component name context - name</i>	One per <i>name context</i>
<i>name context - description</i>	One or more per <i>administered component</i>
<i>name context - label</i>	One or more optional per <i>administered component</i>
<i>name context - language</i>	One or more per <i>administered component</i>

name context - version

One or more optional per *administered component*

7.3. Classification

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 5 represents the Classification region.

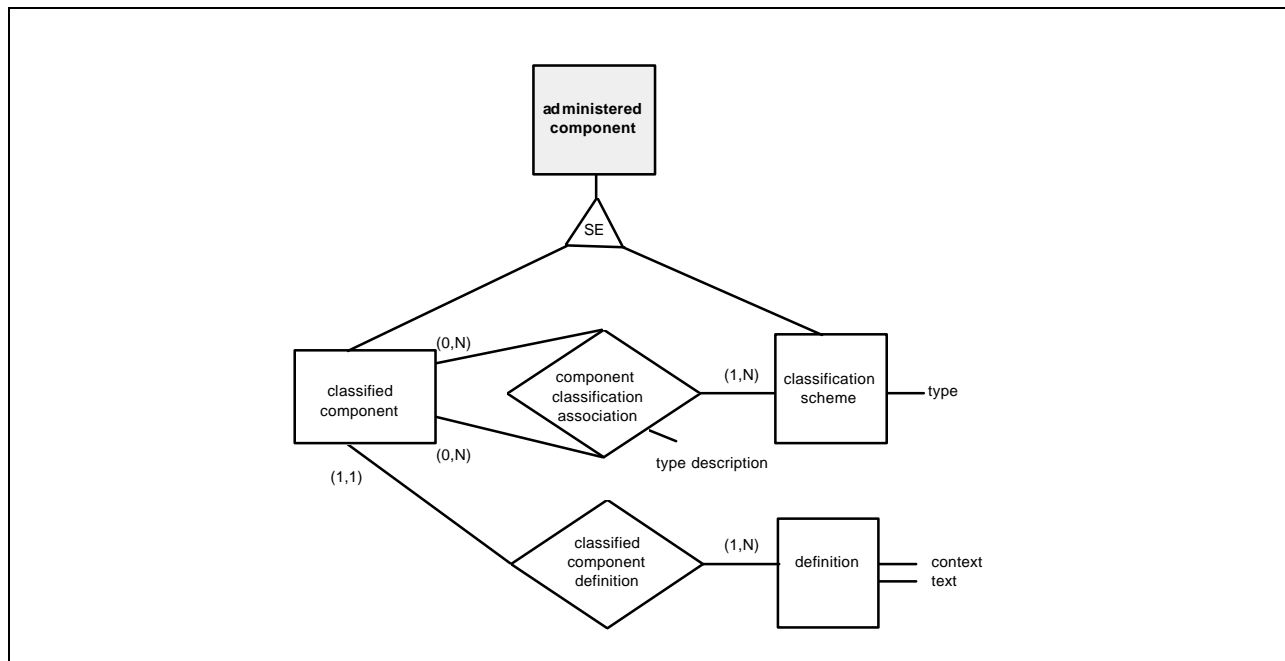


Figure 5 – Classification metamodel region

7.3.1. Classification metamodel region components

A *classified component* is an *administered component* that may be classified in one or more classification schemes. A classified component must have one or more definitions. The definitions may vary to a particular context such that a particular audience may understand the meaning of the component.

A *classified component* is a generalization for an *object class (concept or concept application association)*, *property class*, *representation class*, *data element 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 the are each from a different *name 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.

7.3.2. 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*.

7.4. Data element concept administration

The purpose of the data element concept administration region is to maintain the information on the concepts upon which the data elements 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 data element concepts.

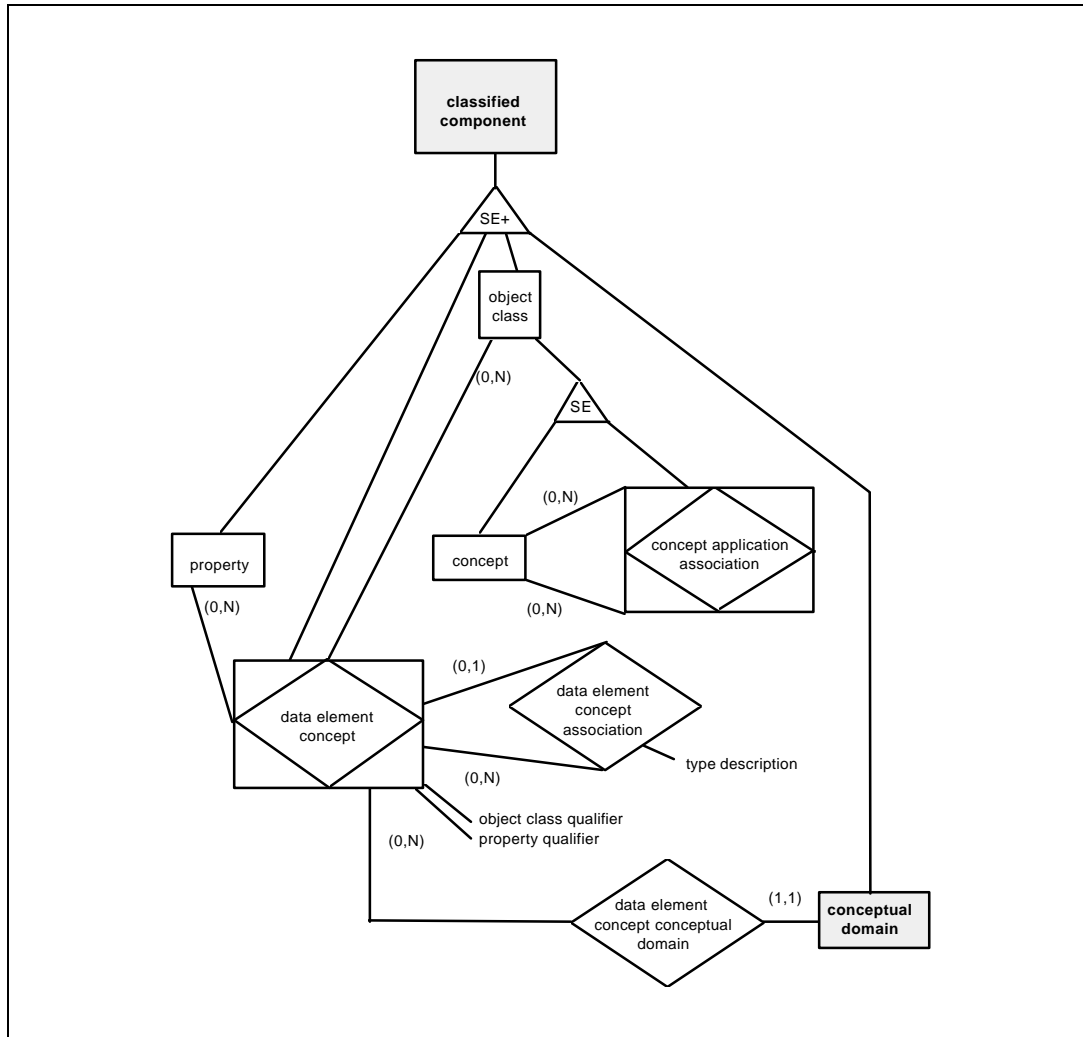


Figure 6 – Data element concept administration metamodel region

7.4.1 Object class

An *object class* is either a single or a group of associated concepts, abstractions, or things in the natural world that can be identified with explicit boundaries and meaning and whose properties and behavior all follow the same rules. 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*.

7.4.2 Property

A *property* is 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.

7.4.3 Data element concept

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

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

A *data element 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*, a *data element concept* may exist within a *classification scheme*. A *data element concept* may reside in this structure but may not necessarily have a *data element* representation.

A *data element concept* shall have one *conceptual domain*. The *conceptual domain* is all permissible value meanings of a data element concept without a specified representation. The *conceptual domain* is detailed in the Conceptual and Value Domain Administration subclause (7.5).

7.4.4 Data element concept metamodel administration region attributes

The attributes in the Data element concept metamodel region shall be as follows:

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

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

7.5. Conceptual and value domain administration

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 data element concepts and value domains support data elements.

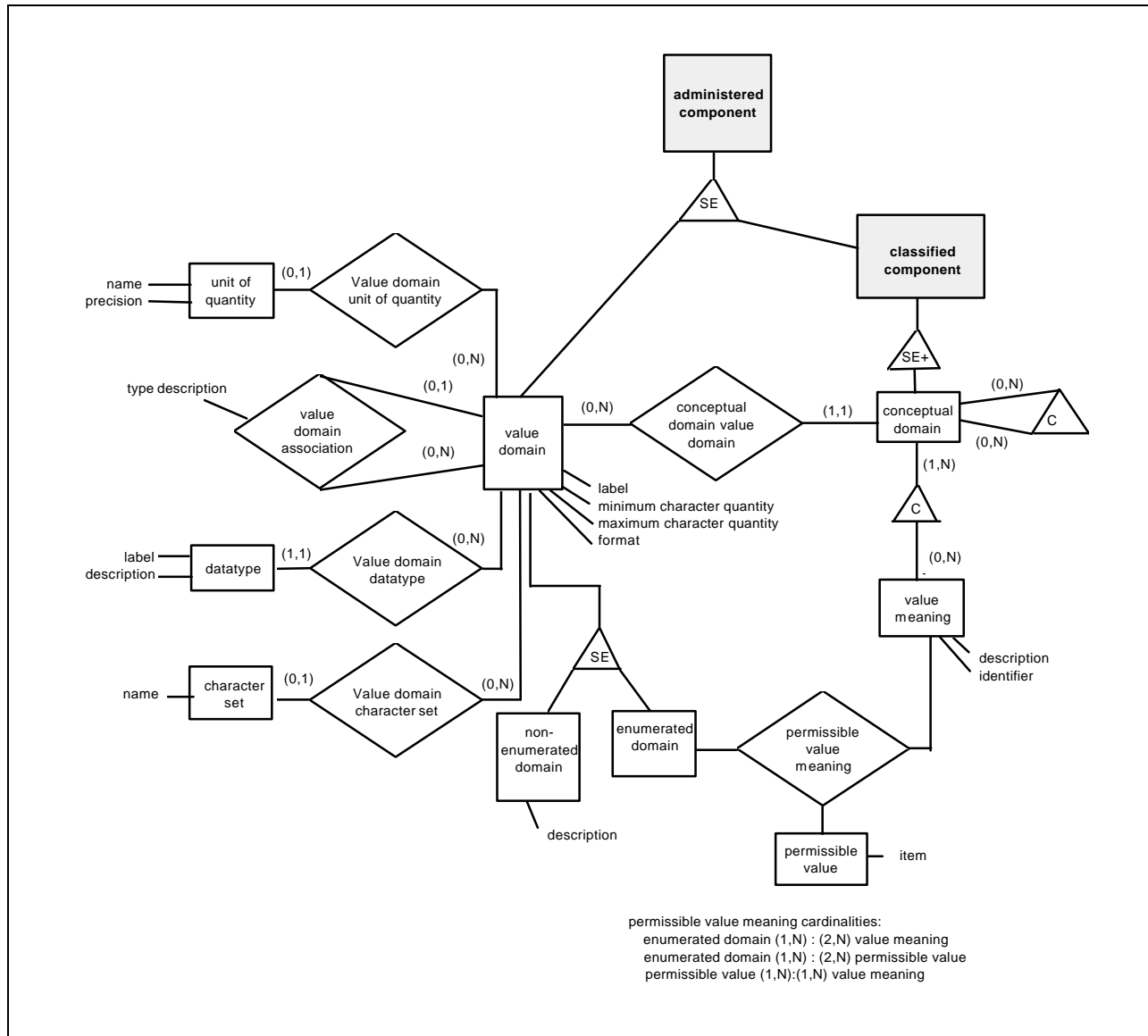


Figure 7 – Conceptual and value domain administration metamodel region

7.5.1. Conceptual domain

The *conceptual domain* is all permissible value meanings of a data element concept without a specified representation. A *data element 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*.

7.5.2. Value domain

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 *data element 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 3166 (parts 1-3), *Codes for the representation of names of countries*. For instance, ISO 3166 describes the set of 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 be dependent upon the particular value instance of another *data element*.

A *value domain* is subtype of an *administered component*. It inherits the stewardship 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* — the category used for the collection of letters, digits, and/or symbols to depict values of a *data element* based upon the operations that may be performed on the *data element*. 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. Also 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 one of the representational forms of quantity.

7.5.3. 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>

exactly one *representation class*. A *representation class* may reside in this structure without being associated with either a *data element concept* or a *value domain*.

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

A *data element* has one or more *examples* that are used to provide representative samples of the *data element*. A *data element* may have a *rule* that is a specification of derivation for the *data element*. The *rule* may range from a simple operation such as subtraction to a very complex set of derivations. *Rules* are not limited to arithmetic and logical operations.

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

7.6.2. Data element administration metamodel region attributes

The attributes in the Data Element Representation metamodel region shall be as follows:

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

8. Metamodel specification

The complete metamodel for the management of shareable data is shown in figure 9. Subclause 8.1 is the complete diagram for the metamodel. Subclause 8.2 contains the specifications that for describing the entities, relationships, and attributes that comprise the metamodel. Subclause 8.3 contains an alphabetical list of specifications of the metamodel objects.

8.1 Metamodel diagram

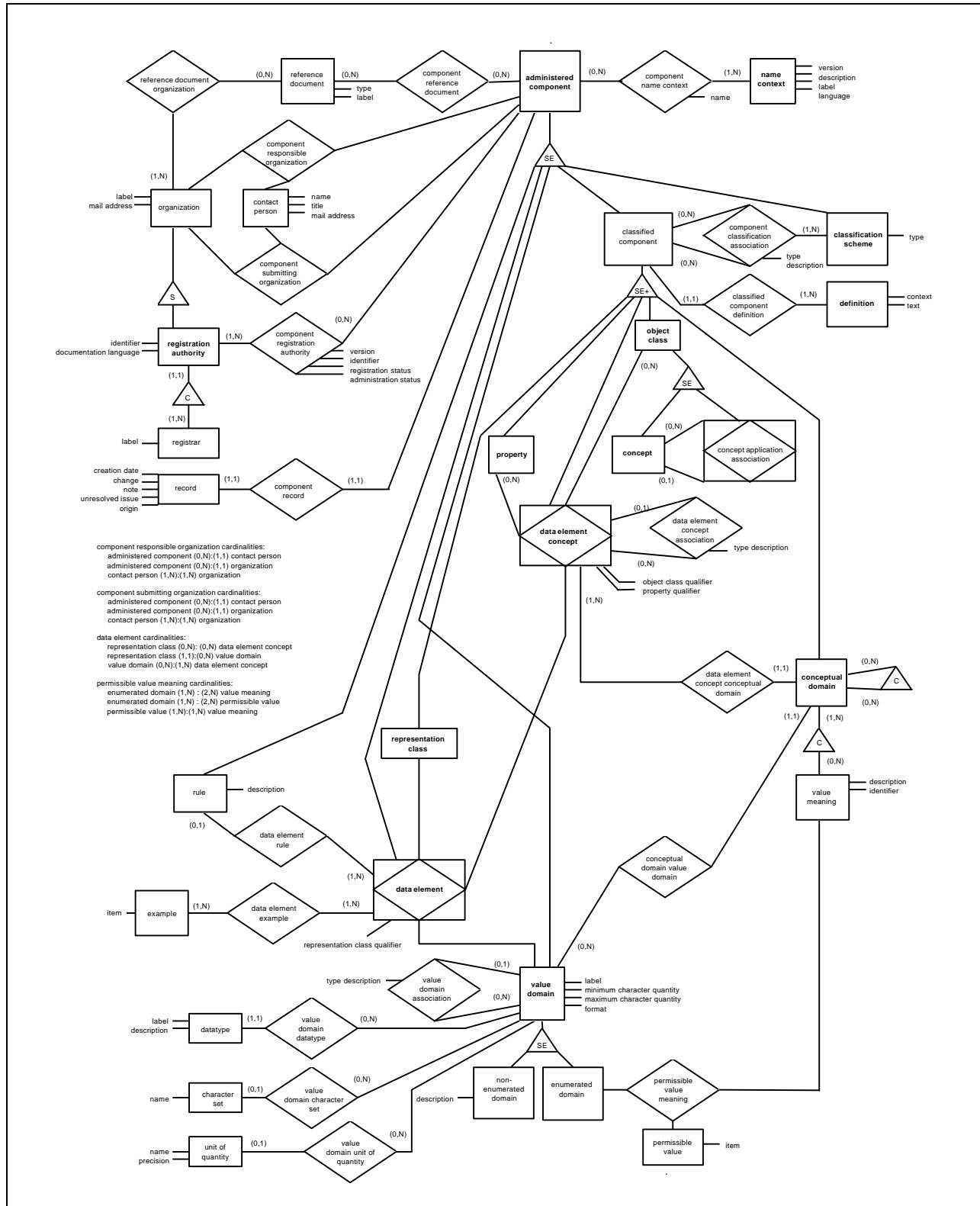


Figure 9 – Conceptual metamodel entity-relationship diagram

8.2 Modeling object description specifications

The following are the entities, relationships, and attributes in alphabetical order. These tables provide detailed descriptions of each modeling object in the entity-relationship diagram in figure 9.

8.2.1 Metadata entity and relationship field description specification

This table structure is used to describe the entities and relationships of the metamodel.

Modeling Name: The single name that identifies this entity or relationship in the data dictionary.

Name(s): The context and the name by which the entity or relationship is known.

Context

A given context identified by the name of the system, database, logical data model, standard document, discipline, organization, company, or other environment in which the subject data has meaning and is known by a name.

Name

The name by which the entity or relationship is known within a specific context.

Type: An entity or relationship may be an:

- Entity — persons, places, concepts, events, or other fundamental things.
- Relationship — associations among two or more entities.
- Associative entity — a relationship that acts like an entity.

Definition: Concise definition or description of the intelligence conveyed by the entity or relationship.

Notes/Comments: General notes or comments not recorded in other fields.

8.2.2 Metadata attribute field description specification

This table structure is used to describe the attributes of the metamodel.

Modeling Name: The single structured name that identifies this attribute in the data dictionary.

Name(s): The context and the name by which the attribute is known.

Context

A given context identified by the name of the system, database, logical data model, standard document, discipline, organization, company, or other environment in which the subject data has meaning and is known by a name.

Name

The name by which the attribute is known within a specific context.

Type: "attribute" identifies this as an attribute.

Definition: Concise definition or description of the intelligence conveyed by the attribute.

Property Obligation: States whether an attribute must be present if the entity or relationship is active. If conditional, the condition for which it is required is given.

Notes/Comments: General notes or comments not recorded in other fields.

8.3 Alphabetical list of metadata objects

Modeling Name: *administered component*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: entity

Definition: An item requiring registration.

Notes/Comments: All *administered components* must be
classification schemes,
or classified components,
or data elements,
or rules,
or value domains.

Modeling Name: *character set*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: entity

Definition: The collective symbols of a formalized writing system for a language used to intelligibly communicate data.

Notes/Comments:

Modeling Name: *character set - name*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: attribute

Definition: A label naming the character set.

Property Obligation: required

Notes/Comments:

Modeling Name: *classification scheme*

Name(s):

Context

International Standard ISO 5127-6:1983

International Standard ISO 1087:1990

Name

classification system

concept system

Type: entity

Definition: A structured set of classified components established according to the relations among them, each component being determined by its position in the set.

Notes/Comments: The classification scheme may be a list, a taxonomy, a network, an ontology, or any other schema for systematizing where the categories are mutually exclusive. The list of "class words" traditionally used by data modelers for attribute-naming standards are candidates for a controlled vocabulary of representation classes.

A classification scheme is always a kind of administered component.

*A classification scheme cannot also be a
classified component,
or a data element,
or a rule,
or a value domain.*

Modeling Name: *classification scheme - type*

Name(s):

Context

Name

Type: attribute

Definition: A description of the type of classification scheme.

Property Obligation: required

Notes/Comments: The classification scheme may be a list, a taxonomy, a network, an ontology, or any other schema for systematizing where the categories are mutually exclusive.

Modeling Name: *classified component*

Name(s):

Context

Name

Type:

entity

Definition:

An administered component that may be classified in one or more classification schemes.

Notes/Comments:

A classified component is always a kind of administered component.

*A classified component cannot also be a
classification scheme,
or a data element,
or a rule,
or a value domain.*

*All classified components must be
conceptual domains,
or data element concepts,
or object classes,
or representation classes.*

Modeling Name: *classified component definition*

Name(s):

Context

Name

Type:

relationship

Definition:

The association between a classified component and a definition where it is defined.

Notes/Comments:

A classified component must be defined by one or more definitions.

A definition must define one classified component.

Modeling Name: *component classification association*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

classification scheme

Type: relationship

Definition: An association between a classified component and one or more other classified component in a classification scheme.

Notes/Comments: *A classified component may be grouped into any number of component classification associations.*

A classified component may group any number of component classification associations

A component classification association must be grouped by one or more classified components.

A component classification association must group one or more classified components.

A component classification association must be included in one or more classification schemes.

A classification scheme may include any number of component classification associations.

Modeling Name: *component classification association - type description*

Name(s):

Context

Name

Type: attribute

Definition: A description of the type of association between a classified component and one or more other classified components in a classification scheme.

Property Obligation: required

Notes/Comments:

Modeling Name: *component name context*

Name(s):

Context

Name

Type: relationship

Definition: The association between an administered component and a name.

Notes/Comments: *An administered component must be named in one or more name contexts.
A name context may provide names to any number of administered components*

Modeling Name: *component name context - name*

Name(s):

Context

Name

International Standard ISO/IEC 11179-5:1995

name

International Standard ISO/IEC 11179-3:1994

name (data element only)

International Standard ISO/IEC 11179-3:1994

synonymous name
(data element concept only)

International Standard ISO/IEC 11179-3:1994

short name
(data element only)

International Standard ISO/IEC 11179-3:1994

programming name or
symbolic name
(data element only)

International Standard ISO/IEC 11179-3:1994

representation category
(data element only)

International Standard ISO 5127-6:1983

thesaurus descriptor

Type: attribute

Definition: A name by which an administered component is known within a specific context.

Property Obligation: required

Notes/Comments: A name from components of an administered component may be combined to produce a structured data element name. This metamodel allows the use of rules from a data element naming convention, but it does not require this structured name.

Modeling Name: *component record*

Name(s):

Context

Name

Type: relationship

Definition: A relationship between a record and the administered component.

Notes/Comments: *An administered component must be described by exactly one record.*
A record must describe exactly one administered component.

Modeling Name: *component reference document*

Name(s):

Context

Name

Type: relationship

Definition: An association between an administered component and a reference document.

Notes/Comments: *An administered component may be described by any number of reference documents.*
A reference document may provide reference to any number of administered components.

Modeling Name: *component registration authority*

Name(s):

Context

Name

Type: relationship

Definition: The association between an administered component and the registration authority.

Notes/Comments: *An administered component must be registered by one or more registration authorities.*
A registration authority may register any number of administered components.

Modeling Name: *component registration authority - administrative status*

Name(s):

Context

International Standard ISO/IEC 11179-6:1997

Name

administrative status
(data element only)

Type: attribute

Definition: A designation of the position in the processing life-cycle of a registration authority for handling registration requests.

Property Obligation: required

Notes/Comments:

Modeling Name: *component registration authority - registration status*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

registration status
(data element only)

International Standard ISO/IEC 11179-6:1997

registration status
(data element only)

Type: attribute

Definition: A designation of the position in the registration life-cycle of a data element.

Property Obligation: required

Notes/Comments:

Modeling Name: *component registration authority - identifier*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

identifier (data element only)

International Standard ISO/IEC 11179-6:1997

data identifier (data element
only)

Type: attribute

Definition: The label that uniquely identifies the administered component for a registration authority.

Property Obligation: required

Notes/Comments:

Modeling Name: *component registration authority - version*

Name(s):

<u>Context</u>	<u>Name</u>
International Standard ISO/IEC 11179-3:1994	version (data element only)
International Standard ISO/IEC 11179-3:1994	revision (data element only)
International Standard ISO/IEC 11179-6:1997	version identifier

Type: attribute

Definition: The label that uniquely identifies a version of the administered component for a registration authority.

Property Obligation: required

Notes/Comments:

Modeling Name: *component responsible organization*

Name(s):

<u>Context</u>	<u>Name</u>
International Standard ISO/IEC 11179-6:1997	responsible organization

Type: Relationship

Definition: The association between the administered component, a contact person, and an organization. The association identifies stewardship of the metadata.

Notes/Comments:

- A component responsible organization (stewardship) may be responsible for at most one administered component.*
- A component responsible organization (stewardship) must be administered by exactly one organization.*
- A component responsible organization (stewardship) must be contacted through exactly one contact person.*
- An administered component must be administered by exactly one (stewardship) component responsible organizations.*
- A contact person may be contacted for any number of (stewardships) component responsible organizations*
- An organization may be responsible for any number of (stewardships) component responsible organizations.*

Modeling Name: *component submitting organization*

Name(s):

Context

International Standard ISO/IEC 11179-6:1996

Name

submitting organization

Type: relationship

Definition: The association between the administered component, a contact person, and an organization. This association identifies the submission of metadata.

Notes/Comments: *A component submitting organization (submission) must be contacted through exactly one contact person.*

A component submitting organization (submission) must submit exactly one administered component.

A component submitting organization (submission) must be submitted by exactly one organization.

An administered component must be submitted by one (submission) component submitting organization

A contact person may be contact for any number of (submissions) component submitting organizations.

An organization may submit any number of (submissions) component submitting organizations.

Modeling Name: *concept*

Name(s):

Context

relational data modeling discipline

Name

entity type

CDIF data model

DataObject

Object Oriented paradigm

class

Fundamental Concepts of Information Modeling
(book)

object type

Type: entity

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

Notes/Comments: An object class, property class, and representation class may be from the same classification scheme or from different classification schemes.

A concept is always a kind of object class.

A concept cannot also be a concept application association.

Modeling Name: *concept application association*

Name(s):

<u>Context</u>	<u>Name</u>
logical data model	relationship
Fundamental Concepts of Information Modeling (book)	relationship

Type: associative entity

Definition: An association between a concept and one or more other concepts.

Notes/Comments: *A concept application association is always a kind of object class.*
A concept application association cannot also be a concept.
A concept application association must be used by one or more concepts.
A concept application association must use one or more concepts.
A concept may be used in any number concept application associations.
A concept may use any number of concept application associations.

Modeling Name: *conceptual domain*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: entity

Definition: A set of possible valid value meanings of a data element expressed without representation.

Notes/Comments: *A conceptual domain is always a kind of classified component.*

Modeling Name: *conceptual domain conceptual domain*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: relationship

Definition: The recursive relationship between two conceptual domains.

Notes/Comments: *A conceptual domain may be contained in any number conceptual domains.*
A conceptual domain may contain any number of conceptual domains.

Modeling Name: *conceptual domain value domain*

Name(s):

Context

Name

Type: relationship

Definition: A relationship between a conceptual domain and a value domain.

Notes/Comments: *A conceptual domain may be represented by any number of conceptual domains.
A conceptual domain must represent exactly one conceptual domain.*

Modeling Name: *conceptual domain value meaning*

Name(s):

Context

Name

Type: relationship

Definition: An association between a conceptual domain and a value meaning.

Notes/Comments: *A conceptual domain may contain any number of value meanings.
A value meaning must be contained in one or more conceptual domains.*

Modeling Name: *contact person*

Name(s):

Context

Name

Type: entity

Definition: Person providing expertise for an administered component.

Notes/Comments:

Modeling Name: *contact person - mail address*

Name(s):

Context

International Standard ISO/IEC 11179-6:1997

Name

postal address of SO's contact person

International Standard ISO/IEC 11179-6:1997

postal address of CO's contact person

Type: attribute

Definition: A description of where to send written correspondence to the contact person.

Property Obligation: required

Notes/Comments:

Modeling Name: *contact person - name*

Name(s):

Context

International Standard ISO/IEC 11179-6:1997

Name

name of SO's contact person

International Standard ISO/IEC 11179-6:1997

name of CO's contact person

Type: attribute

Definition: The name of the contact person.

Property Obligation: required

Notes/Comments:

Modeling Name: *contact person - title*

Name(s):

Context

International Standard ISO/IEC 11179-6:1997

Name

title of SO's contact person

International Standard ISO/IEC 11179-6:1997

title of CO's contact person

Type: attribute

Definition: The name of the position held by the contact person.

Property Obligation: optional

Notes/Comments:

Modeling Name: *data element concept*

Name(s):

Context

Basic Semantic Repository

Name

basic semantic unit (BSU)

Type: associative entity

Definition: A concept that can be represented in the form of a data element, described independently of any particular representation.

Notes/Comments: *A data element concept must convey the perception of one property.*
A data element concept must be the perception of one object class.
An object class may be perceived in any number of data element concepts.
A property may be perceived in any number of data element concepts.
A data element concept is always a kind of classified component.

Modeling Name: *data element concept - object class qualifier*

Name(s):

Context

Name

Type: attribute

Definition: An object-class qualifier to the component name context - name to accurately label the data element concept.

Property Obligation: optional

Notes/Comments: This is restricted to any adjectival word or phrase that is never used with the same meaning as a name of an object class.

Modeling Name: *data element concept - property qualifier*

Name(s):

Context

Name

Type: attribute

Definition: A property qualifier to the component name context - name to accurately label the data element concept.

Property Obligation: optional

Notes/Comments:

Modeling Name: *data element concept association*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

related data reference (shown as a metadata attribute)

Type: relationship

Definition: A relationship with another data element concept that this data element concept.

Notes/Comments: *A data element concept may be the source in one data element concept.*
A data element concept may link into in any number of data element concepts.

Modeling Name: *data element concept association - type description*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

procedural information

Type: attribute

Definition: The description of the type of association with another data element concept that this data element concept modifies, is modified by, or is otherwise linked.

Property Obligation: required

Notes/Comments:

Modeling Name: *data element concept conceptual domain*

Name(s):

Context

Name

Type: relationship

Definition: The association between a data element concept and the conceptual domain.

Notes/Comments: *A data element concept may have at most one conceptual domain.*
A conceptual domain may specify any number of data element concepts.

Modeling Name: *data element*

Name(s):

<u>Context</u>	<u>Name</u>
Fourth Generation Data (book)	data item
CDIF data model	DataElementType
International Standard ISO/IEC 11179	data element
Content Standards for Digital Geospatial Metadata	data element
Object Oriented paradigm	attribute
logical data model	attribute

Type: associative entity

Definition: A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes. A unit of data that in a certain context is considered indivisible.

Notes/Comments: A data element is the union of a data element concept, a value domain, and a representation class. Data is a representation of facts, concepts, or instructions in a form that allows them to be collected, organized, recorded, processed, and stored in a retrievable form for communication, interpretation, or processing by human or automated means.

A data element is always a kind of administered component.

*A data element cannot also be a
classification scheme,
or a classified component,
or a rule,
or a value domain.*

A data element must derive meaning from exactly one data element concept.

A data element must have exactly one representation class.

A data element must be represented with exactly one value domain.

A data element concept may provide meaning to any number of data elements.

A representation class may describe any number of data elements.

A value domain may provide representation to any number data elements.

Modeling Name: *data element - representation class qualifier*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: attribute

Definition: A representation class qualifier to the component name context - name to accurately label the data element.

Property Obligation: optional

Notes/Comments:

Modeling Name: *data element example*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: relationship

Definition: A relationship between an example and a data element.

Notes/Comments: *A data element may be represented by any number of examples. An example must represent one or more data elements.*

Modeling Name: *data element rule*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: relationship

Definition: A relationship between a rule and a data element.

Notes/Comments: *A data element may be derived by one rule. A rule must specify derivation of one or more data elements.*

Modeling Name: *datatype*

Name(s):

Context

Name

Type: entity

Definition: The format used for the collection of letters, digits, and/or symbols, to depict values of a data element, determined by the operations that may be performed on the data element.

Notes/Comments:

Modeling Name: *datatype - description*

Name(s):

Context

Name

Type: attribute

Definition: Describes the parameters of a datatype.

Property Obligation: required

Notes/Comments:

Modeling Name: *datatype - label*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

datatype of data element values

Type: attribute

Definition: A label identifying the category of datatype.

Property Obligation: required

Notes/Comments: May be represented by either a name or an abbreviation, or both. The set of datatype categories used in a Registry shall be determined by the registration authority. Possible sets are described by CDIF and ISO/IEC 11404:1995. They need to include non-character representations such as bar code, image, and sound.

Modeling Name: *definition*

Name(s):

Context

Name

Type: entity

Definition: An expression of the essential nature of a person or thing or class of persons or things: an answer to the question "what is x?" or "what is an x?". [This is the definition of the entity.]

Notes/Comments:

Modeling Name: *definition - context*

Name(s):

Context

Name

Type: attribute

Definition: A viewpoint in which the definition is prepared for sharing.

Property Obligation: required

Notes/Comments:

Modeling Name: *definition - text*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

definition (data element only)

Type: attribute

Definition: A word or phrase expressing the essential nature of a person or thing or class of persons or things: an answer to the question "what is x?" or "what is an x?"; a statement of the meaning of a word or word group [Webster's Third New International Dictionary of the English Language Unabridged, 1986].

Property Obligation: required

Notes/Comments:

Modeling Name: *enumerated domain*

Name(s):

Context

Name

Type: entity

Definition: A value domain that is specified by a list of all permissible values.

Notes/Comments: *An enumerated domain is always a kind of value domain.
An enumerated domain cannot also be a non-enumerated domain.*

Modeling Name: *example*

Name(s):

Context

Name

Type: entity

Definition: Representative illustration of the information described.

Notes/Comments:

Modeling Name: *example - item*

Name(s):

Context

Name

Type: attribute

Definition: Actual illustrative case of the information described.

Property Obligation: required

Notes/Comments:

Modeling Name: *name context*

Name(s):

<u>Context</u>	<u>Name</u>
ANSI X3H4 Open Repository Systems Committee	Namespace
International Standard ISO/IEC 11179-5:1995	context

Type: entity

Definition: The system, database, standard document, or other environment in which data has meaning and is known by a name.

Notes/Comments:

Modeling Name: *name context - description*

Name(s):

<u>Context</u>	<u>Name</u>
International Standard ISO/IEC 11179-3:1994	context (data element concept only)

Type: attribute

Definition: The description of the system, database, standard document, or other environment that is the name context.

Property Obligation: required

Notes/Comments: Shall convey sufficient information to describe the nature of the context.

Modeling Name: *name context - label*

Name(s):

<u>Context</u>	<u>Name</u>
International Standard ISO/IEC 11179-3:1994	context (data element concept only)
International Standard ISO/IEC 11179-5:1995	context

Type: attribute

Definition: A label identifying the system, database, standard document, or other environment that is the name context.

Property Obligation: optional

Notes/Comments:

Modeling Name: *name context - language*

Name(s):

Context

International Standard ISO 1087:1990

Name

language code

Type: attribute

Definition: The label identifying the human language used by the system, database, standard document, or other environment that is the name context.

Property Obligation: required

Notes/Comments: May be represented by either a name or an abbreviation, or both.

Modeling Name: *name context - version*

Name(s):

Context

International Standard ISO/IEC 11179-5:1995

Name

version identifier

Type: attribute

Definition: The label that uniquely identifies the version of a name context.

Property Obligation: optional

Notes/Comments:

Modeling Name: *non-enumerated domain*

Name(s):

Context

CDIF data model

Name

ValueDomainProdedure

CDIF data model

ValueDomainRule

Type: entity

Definition: A value domain where the allowed values are not a finite list.

Notes/Comments: Examples are a procedure or scheme for specifying permissible values, a format structure rule specifying permissible values, and a range used for specifying the lower limit and the upper limit of permissible values.

A non-enumerated domain is always a kind of value domain.

A non-enumerated domain cannot also be an enumerated domain.

Modeling Name: *non-enumerated domain - description*

Name(s):

Context

Name

Type: attribute

Definition: A description of procedure or rule for a set of all permissible values for the value domain or the upper and lower limit to a value domain range.

Property Obligation: required

Notes/Comments: This may be expressed as a rule or the identification of a reference document.

Modeling Name: *object class*

Name(s):

Context

Name

Fundamental Concepts of Information Modeling
(book) associative object type

Type: entity

Definition: 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.

Notes/Comments: An *object class* is always a kind of *classified component*.
All *object classes* must be
concepts,
or *concept application associations*.

Modeling Name: *organization*

Name(s):

Context

Name

Type: entity

Definition: One or more persons or groups formed into a functioning whole for a purpose.

Notes/Comments:

Modeling Name: *organization - label*

Name(s):

Context

Name

Type: attribute

Definition: The label used to identify the organization.

Property Obligation: required

Notes/Comments: May be represented by the name or the role of the person or group.

Modeling Name: *organization - mail address*

Name(s):

Context

International Standard ISO/IEC 11179-6:1997

Name

address of submitting organization

Type: attribute

Definition: The mailing address of an organization.

Property Obligation: optional

Notes/Comments:

Modeling Name: *permissible value*

Name(s):

Context

CDIF data model

Name

ValueDomainEnumeration

Type: entity

Definition: An expression of a value meaning in a specific value domain. A value.

Notes/Comments:

Modeling Name: *permissible value - item*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

permissible data element values

Type: attribute

Definition: An expression of a value meaning in a specific value domain. The actual value.

Property Obligation: required

Notes/Comments:

Modeling Name: *permissible value meaning*

Name(s):

Context

Name

Type: relationship

Definition: The association of a value meaning and a permissible value enumerated in a domain.

Notes/Comments:

- A *permissible value meaning* (legal value) must be contained in one or more enumerated domains.
- A *permissible value meaning* (legal value) must be represented by one or more *permissible values*.
- A *permissible value meaning* (legal value) must represent exactly one *value meaning*.
- An *enumerated domain* must contain two or more (legal values) *permissible value meanings*.
- A *permissible value* must represent one or more (legal values) *permissible value meanings*.
- A *value meaning* must be represented by one or more (legal values) *permissible value meanings*.

Modeling Name: *property*

Name(s):

Context

Fourth Generation Data (book)

International Standard ISO 1087:1990

International Standard ISO CD 14481

Name

data item category

characteristic

proposition

Type: entity

Definition: A peculiarity common to all members of an object class.

Notes/Comments: An object class, property, and representation class may be from the same classification scheme or from different classification schemes. A property may also be an object class within the same classification scheme.

A property is always a kind of classified component.

Modeling Name: *record*

Name(s):

Context

Name

Type: entity

Definition: A collection of known facts treated as a unit.

Notes/Comments: May exist on paper, in a database, or any other medium.

Modeling Name: *record - change*

Name(s):

Context

Name

Type: attribute

Definition: A description of what has changed in this record since the last version.

Property Obligation: Conditional - Required after first version.

Notes/Comments:

Modeling Name: *record - creation date*

Name(s):

<u>Context</u>	<u>Name</u>
International Standard ISO/IEC 11179-3:1994	date of approval
International Standard ISO/IEC 11179-6:1997	date of original assignment
International Standard ISO/IEC 11179-3:1994	date of latest modification
International Standard ISO/IEC 11179-6:1997	date of last update

Type: attribute

Definition: The date the record was first created.

Property Obligation: required

Notes/Comments:

Modeling Name: *record - note*

Name(s):

<u>Context</u>	<u>Name</u>
International Standard ISO/IEC 11179-3:1994	comments (data element only)
International Standard ISO/IEC 11179-3:1994	additional information (data element only)
International Standard ISO 1087:1990	note

Type: attribute

Definition: General notes or comments about the record.

Property Obligation: optional

Notes/Comments:

Modeling Name: *record - origin*

Name(s):

<u>Context</u>	<u>Name</u>
----------------	-------------

Type: attribute

Definition: Any document, project, discipline, model, et cetera that was the basis for this record.

Property Obligation: optional

Notes/Comments:

Modeling Name: *record - unresolved issue*

Name(s):

Context

Name

Type: attribute

Definition: Any problems that remain unresolved regarding proper documentation of this record.

Property Obligation: optional

Notes/Comments:

Modeling Name: *reference document*

Name(s):

Context

Name

Type: entity

Definition: A source document that provides pertinent details for consultation about a subject.

Notes/Comments: May be any practice, standard, or other source document that further describes and/or provides additional pertinent details.

Modeling Name: *reference document - label*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

source document

Type: attribute

Definition: A label identifying the reference document.

Property Obligation: required

Notes/Comments: May be a document name, identifier, et cetera.

Modeling Name: *reference document - type*

Name(s):

Context

Name

Type: attribute

Definition: A description of the type of reference document.

Property Obligation: optional

Notes/Comments:

Modeling Name: *reference document organization*

Name(s):

Context

Name

Type: relationship

Definition: The relationship between a reference document and an organization.

Notes/Comments: *A reference document may be provided by at most one organization.
An organization may provide any number of reference documents.*

Modeling Name: *registrar*

Name(s):

Context

CDIF data model

Name

ToolUser

Type: entity

Definition: A person who performs the administrative steps to register metadata in a data registry.

Notes/Comments:

Modeling Name: *registrar - label*

Name(s):

Context

Name

Type: attribute

Definition: A label identifying the registrar.

Property Obligation: required

Notes/Comments:

Modeling Name: *registration authority*

Name(s):

Context

Name

International Standard ISO/IEC 11179-7:1996

registration authority

Type: entity

Definition: An organization authorized to register objects.

Notes/Comments: This may be an international organization such as the International Organization for Standardization (ISO), a national organization such as the American National Standards Institute (ANSI), an industry group such as the National Information Standards Organization (NISO), a government body such as the National Institute of Standards and Technology (NIST), or a private business.

A registration authority is always a kind of organization.

Modeling Name: *registration authority - documentation language*

Name(s):

Context

Name

International Standard ISO 5127-6:1983

documentation language

International Standard ISO 1087:1990

language code

Type: attribute

Definition: The label identifying the natural language used by the registration authority in the data registry.

Property Obligation: required

Notes/Comments: May be represented by either a name or an abbreviation, or both.

Modeling Name: *registration authority - identifier*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

International Standard ISO/IEC 11179-6:1997

Name

registration authority
(data element only)

registration authority identifier
(data element only)

Type: attribute

Definition: An identifier assigned to a Registration Authority.

Property Obligation: required

Notes/Comments:

Modeling Name: *registration authority registrar*

Name(s):

Context

Name

Type: relationship

Definition: A relationship between a registration authority and a registrar.

Notes/Comments: *A registration authority must contain one or more registrars.
A registrar must be included in exactly one registration authority.*

Modeling Name: *representation class*

Name(s):

Context

Name

Type: entity

Definition: A classification of types of representations.

Notes/Comments: *An object class, property, and representation class may be from the same classification scheme or from different classification schemes.
A representation class is always a kind of classified component.*

Modeling Name: *rule*

Name(s):

Context

Name

Type: entity

Definition: The logical, mathematical, and/or other operations specifying the derivation for a data element.

Notes/Comments: A *rule* is always a kind of *administered component*.

A *rule* cannot also be a
classification scheme,
 or a *classified component*,
 or a *data element*,
 or a *value domain*.

Modeling Name: *rule - description*

Name(s):

Context

Name

Type: attribute

Definition: The text of a specification of derivation for a data element.

Property Obligation: required

Notes/Comments:

Modeling Name: *unit of quantity*

Name(s):

Context

Name

Type: entity

Definition: A system of measurement.

Notes/Comments: Quantities may include count and measurement. May be specified as another data element in a data composite.

Modeling Name: *unit of quantity - name*

Name(s):

Context

Name

Type: attribute

Definition: The name of a unit of quantity.

Property Obligation: required

Notes/Comments: Units of measure are meters, inches, foot-pounds, et cetera. Units of count are dozens, gross, et cetera. Units of monetary amount are US dollar, French franc, Japanese yen, et cetera.

Modeling Name: *unit of quantity - precision*

Name(s):

Context

Name

Type: attribute

Definition: The degree of specificity for a unit of quantity.

Property Obligation: required

Notes/Comments:

Modeling Name: *value domain*

Name(s):

Context

CDIF data model

Name

ValueDomain

Type: entity

Definition: A set of permissible values.

Notes/Comments: Used for syntactical integrity checking. It provides representation, but has no implication as to what data element concept the values are associated nor what the values mean.

A value domain is always a kind of administered component.

*A value domain cannot also be a
classification scheme,
or a classified component,
or a data element,
or a rule.*

*All value domains must be
enumerated domains,
or non-enumerated domains.*

Modeling Name: *value domain - format*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

layout of representation

Type: attribute

Definition: A template for the structure of the elements of a value domain.

Property Obligation: optional

Notes/Comments:

Modeling Name: *value domain - label*

Name(s):

Context

Name

Type: attribute

Definition: The label identifying the value domain.

Property Obligation: optional

Notes/Comments:

Modeling Name: *value domain - maximum character quantity*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

maximum size of data element values

Type: attribute

Definition: The maximum number of storage units (of the corresponding datatype) to represent the data element value.

Property Obligation: required

Notes/Comments: Any embedded punctuation is included in this count.

Modeling Name: *value domain - minimum character quantity*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

minimum size of data element values

Type: attribute

Definition: The minimum number of storage units (of the corresponding datatype) to represent the data element value.

Property Obligation: optional

Notes/Comments: Any embedded punctuation is included in this count.

Modeling Name: *value domain association*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

Name

related data reference (shown as a metadata attribute)

Type: relationship

Definition: A relationship between value domains.

Notes/Comments: The permissible values of one value domain may dependent upon the particular value instance of another data element.

A value domain may be dependent in one value domain.

A value domain may start any number of value domains.

Modeling Name: *value domain association - type description*

Name(s):

Context

International Standard ISO/IEC 11179-3:1994

International Standard ISO/IEC 11179-3:1994

Name

type of relationship

formula

Type: attribute

Definition: The description of a value domain association.

Property Obligation: required

Notes/Comments:

Modeling Name: *value domain character set*

Name(s):

Context

Name

Type: relationship

Definition: A relationship between a value domain and a character set.

Notes/Comments: *A value domain may use at most one character set.*

A character set may be used by any number of value domains.

Modeling Name: *value domain datatype*

Name(s):

Context

Name

Type: relationship

Definition: A relationship between a value domain and a datatype.

Notes/Comments: *A value domain must use exactly one datatype.*

A datatype may be used by any number of value domains.

Modeling Name: *value domain unit of quantity*

Name(s):

Context

Name

Type: relationship

Definition: A relationship between a value domain and a unit of quantity.

Notes/Comments: *A value domain may use exactly one unit of quantity.*
A unit of quantity may be used by any number of value domains.

Modeling Name: *value meaning*

Name(s):

Context

Name

Type: entity

Definition: A valid value in a conceptual domain.

Notes/Comments:

Modeling Name: *value meaning - description*

Name(s):

Context

Name

Type: attribute

Definition: A description of a value meaning.

Property Obligation: optional

Notes/Comments:

Modeling Name: value meaning - identifier

Name(s):

Context

Name

Type:

attribute

Definition:

The label that uniquely identifies a value meaning.

Property Obligation:

required

Notes/Comments:

Annex A
(informative)

Modeling symbols

The Chen notation, as extended by Bell Communications Research, Inc., is used to describe this metamodel. This notation is particularly suited to documenting a conceptual data schema. The structure used in the description (the meta-metamodel) is compatible and is partially described by the metamodel described. Since this is a *conceptual data model*, only the *data element concept* information is used.

The entity-relationship model provides for three basic types of modeling objects: *entities* (shown as rectangles in a diagram), relationships among these entities (shown as diamonds), and data elements that are associated to an entity or relationship as attributes. For example, the data elements or attributes describing employees and the cars they own could be modeled as follows:

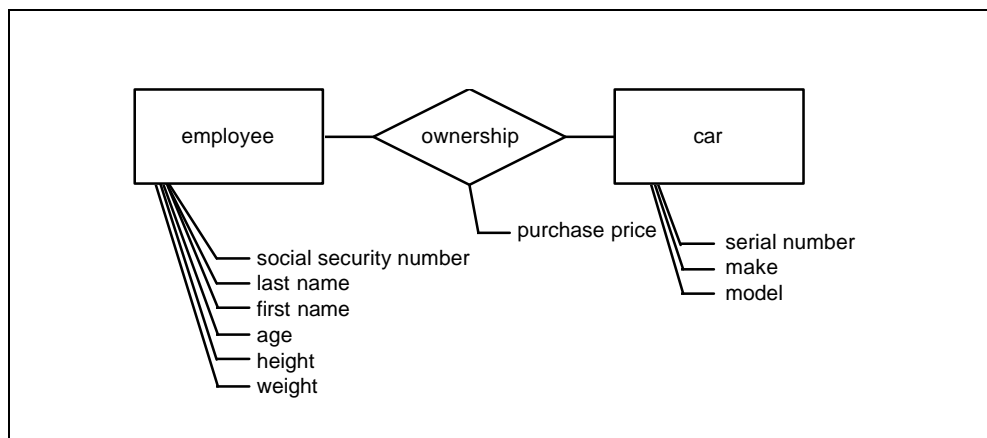


Figure A1 – Sample modeling diagram

A.1 Entities

Entities are represented by rectangles and are the things about which the business processes information. An entity is something that has distinctiveness and typically, properties of its own. Entities may be persons, places, concepts, events, or other fundamental things. For example, employee and car are both entities.

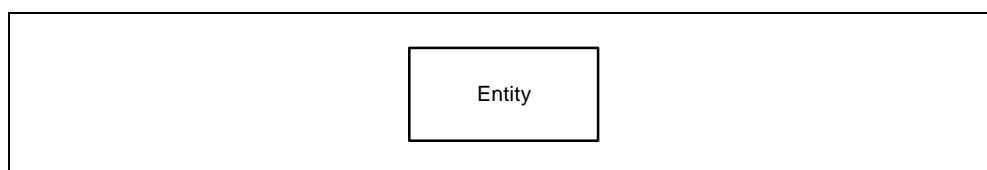


Figure A2 – Entity modeling representation

As described below, entities may have attributes.

A.2 Relationships

Entities have business-based *relationships* with one or more other entities. These relationships are bi-directional. For example, an employee owns a car. Conversely the car is owned by the employee. This relationship between car and employee may be called ownership. In an E-R diagram these links between two or more entities are represented by diamonds.

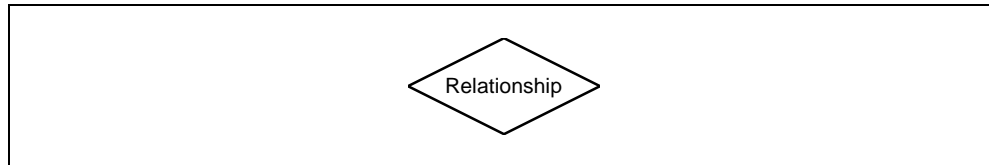


Figure A3 – Relationship modeling representation

Lines are drawn between diamonds and rectangles to identify the entities participating in a relationship.

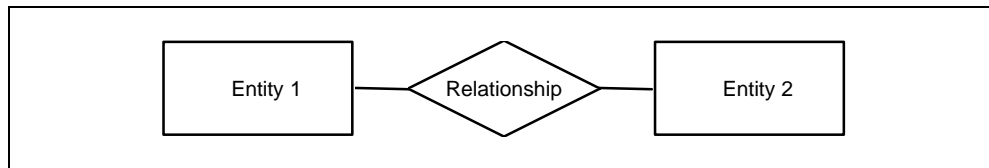


Figure A4 – Entity-relationship modeling representation

The statement of the number of entity instances that may participate at each end of a relationship is its *cardinality*. It is expressed in parentheses as a minimum and a maximum, separated by a comma.

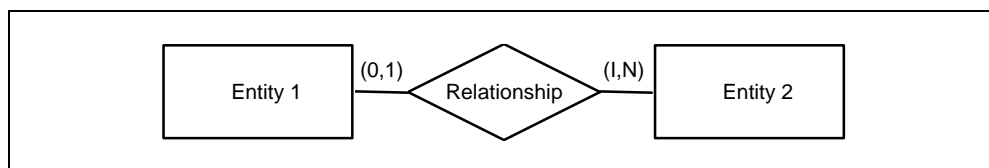


Figure A5 – Entity-relationship cardinality modeling representation

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

As described below, like entities, relationships may also have attributes.

A.3 Associative entity

There are situations where a particular business object fits the criteria for both an entity and a relationship. This usually occurs when a relationship itself is found to have a business association with an entity. Thus, an *associative entity* is defined as a relationship that acts like an entity. Its icon is a combination of the entity's rectangle and the relationship's diamond.

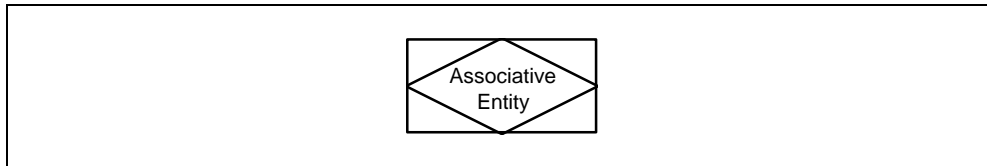


Figure A6 – Associative entity modeling representation

A.4 Subtypes

Entities may be decomposed into a hierarchy with increasing level of details. Each *supertype* entity may have two or more *subtypes*.

Disjointed (mutually exclusive) subtypes are connected with their supertype using a triangle enclosing the letter "S". The tip of the triangle points to the supertype.

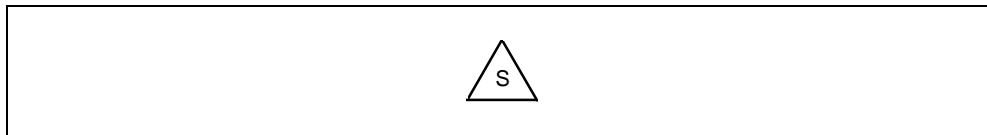


Figure A7 – Subtype modeling representation

There are cases when subtypes are not disjointed, but rather, they are overlapping. Overlapping subtypes are modeled using a triangle enclosing the characters "S+".

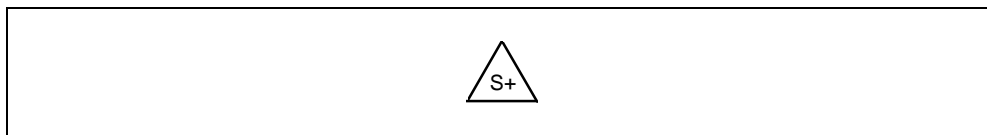


Figure A8 – Overlapping subtype modeling representation

If the list of subtypes is exhaustive, an “E” notation is appended to the “S” or “S+” symbol.



Figure A9 – Exhaustive subtype modeling 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. Also, subtypes participate in all the relationships in which their supertype participates.

Usually, only one subtype at a time may exist. This is the OR construct. The AND construct applies when all subtypes may exist simultaneously. In that case, the symbols are replicated. In any case, the subtype may or may not exist. Thus, the cardinality is always (0,1) and is not shown.

A.5 Dependencies

In some cases, the existence as well as the identification of an entity depend upon the existence of a *superior entity*. What causes the dependency is the existence of a business rule that states that the *dependent entity* may exist if and only if it participates in a relationship with the superior entity. Thus, a dependency is a special type of relationship. In an E–R diagram, a dependency is represented by a triangle containing the letter “D”. The tip of the triangle points toward the superior entity.

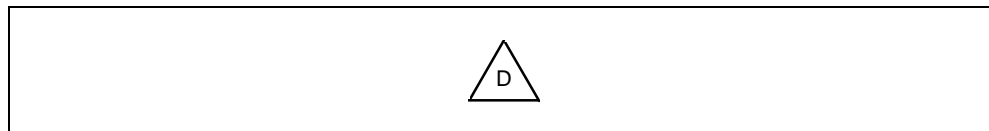


Figure A10 – Dependency modeling representation

A.6 Composition

An entity sometime consists of component entities. When an entity consists of assemblies of “parts”, there is a special relationship between the component parts and the entity representing the assembly of these parts. A composition is represented by a triangle containing the letter “C”. The tip of the triangle points toward the assembly entity.

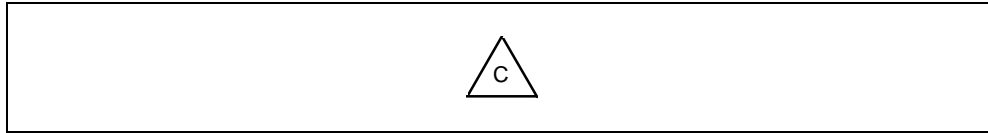


Figure A11 – Composition modeling representation

A.7 Attribute

The *properties* of an entity or a relationship 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 with lines linking them to their entity or relationship.

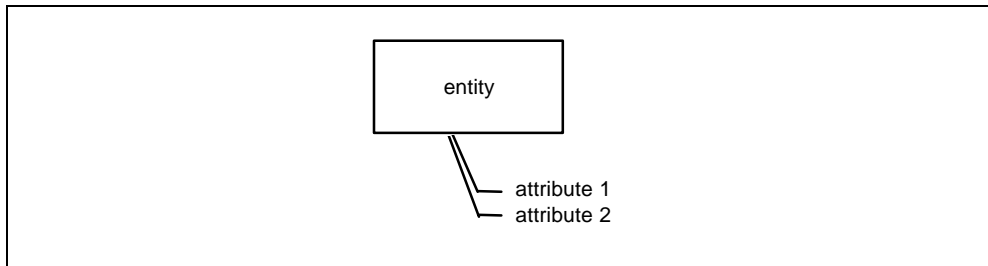


Figure A12 – Entity-attribute modeling representation

Annex B
(informative)

Alternative modeling representations

In normative section of this standard, the metamodel for the management of shareable data was expressed in the Chen notation as extended by Bell Communications Research, Inc. This informative annex is provided so that readers may see the metamodel in representations that they use. The metamodel has been represented in:

IDEF1X	a relational modeling paradigm,
ORM or NIAM	a natural language paradigm, and
UML	an object-oriented paradigm.

B.1 IDEF1X representation model

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. Stewardship, naming and identification regions,
2. Data element concept administration and classification regions,
3. Conceptual domain and value domain administration region,
4. Data element administration region, and
5. Type hierarchy.

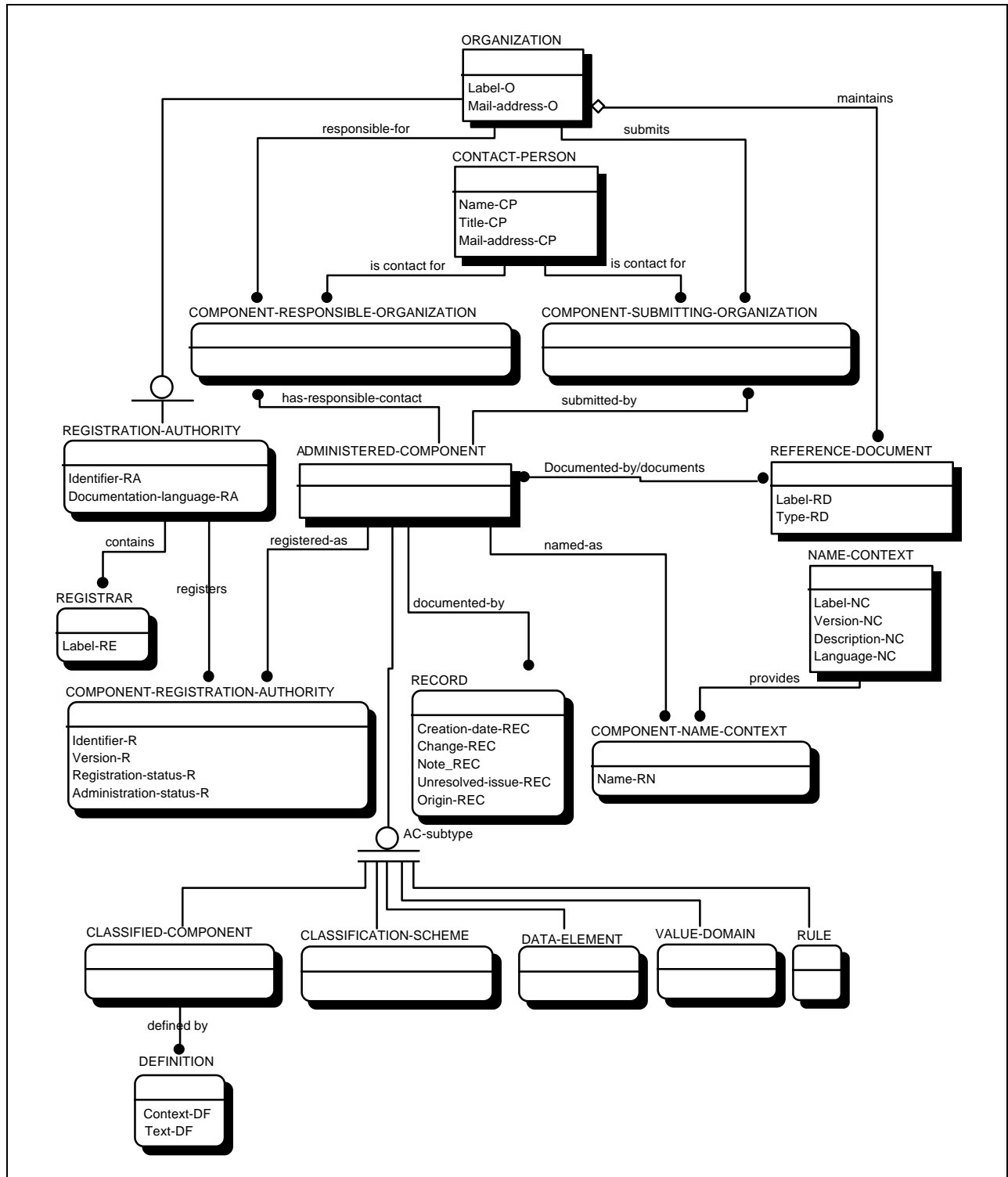


Figure B1 – IDEF1X stewardship, naming and identification metamodel regions

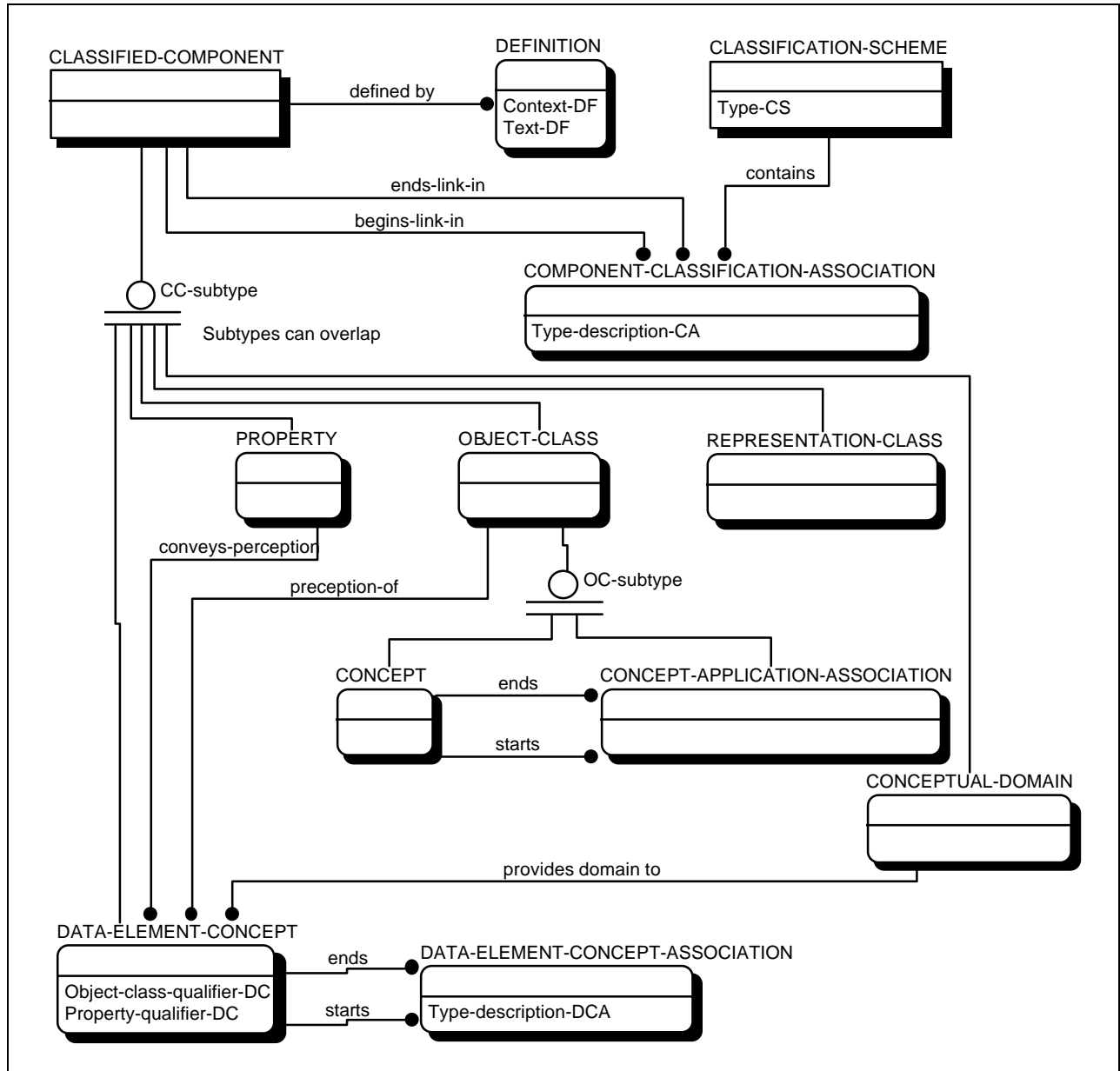


Figure B2 – IDEF1X data element concept administration and classification metamodel regions

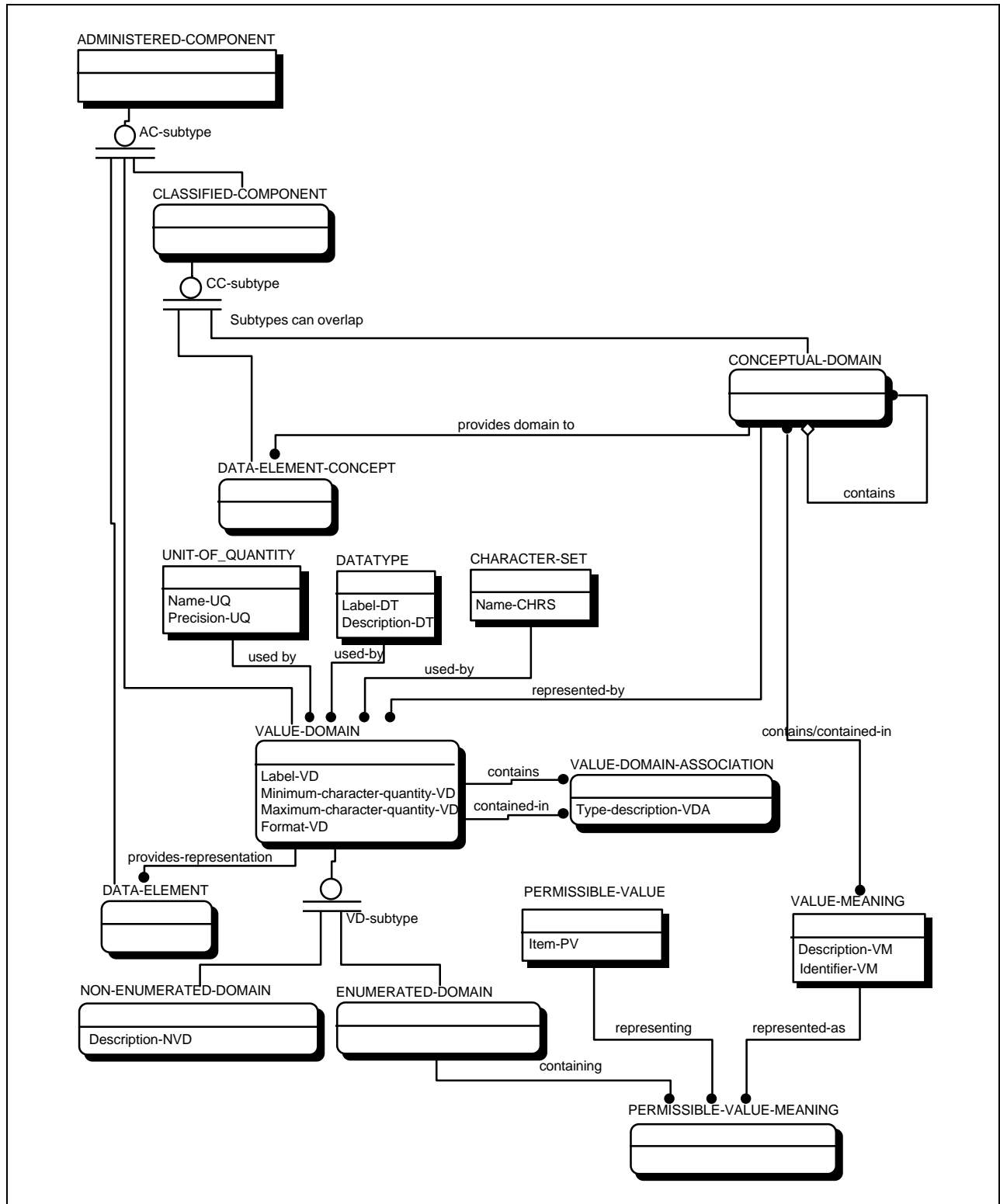


Figure B3 – IDEF1X conceptual and value domain administration metamodel region

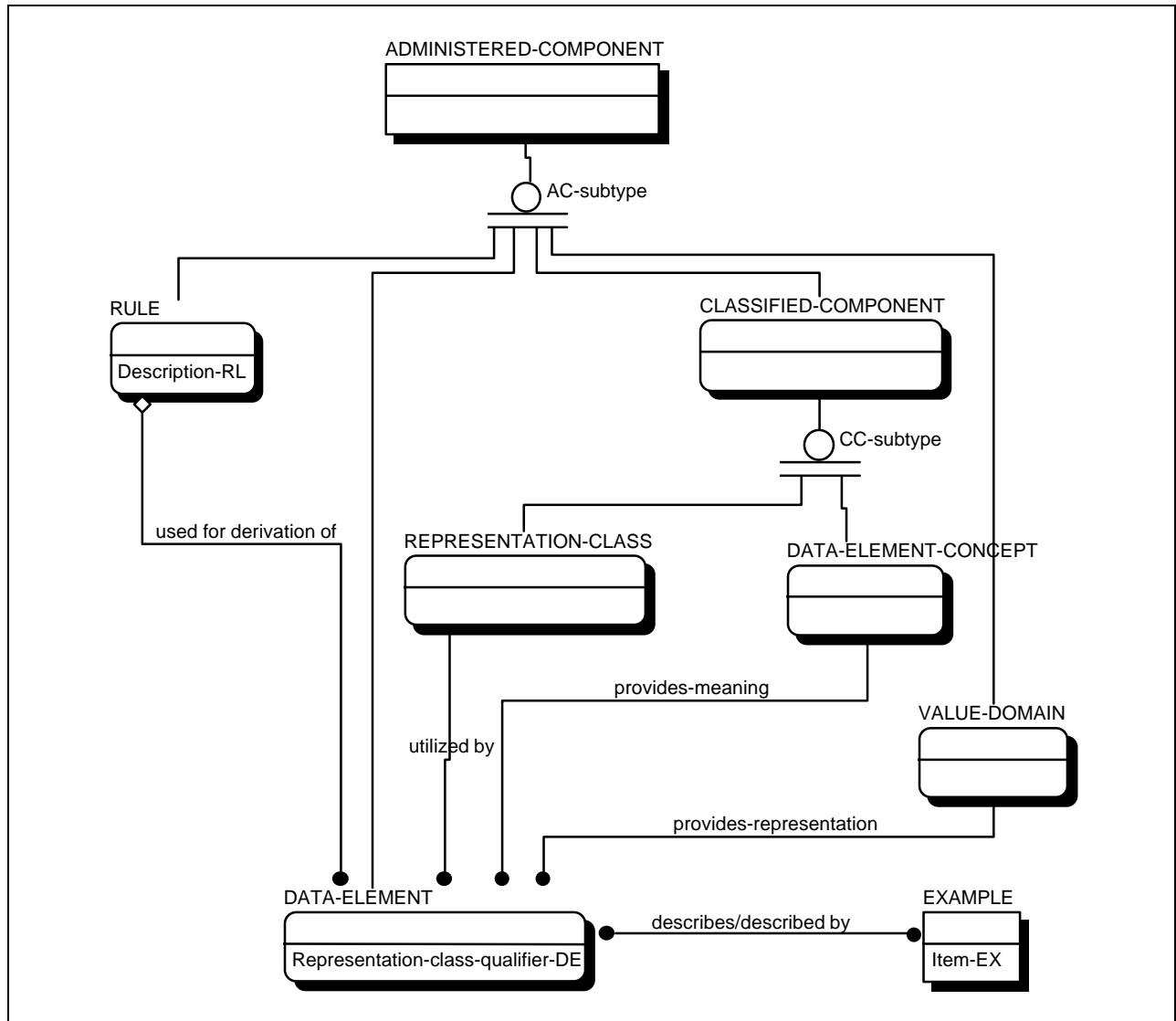


Figure B4 – IDEF1X data Element administration metamodel region

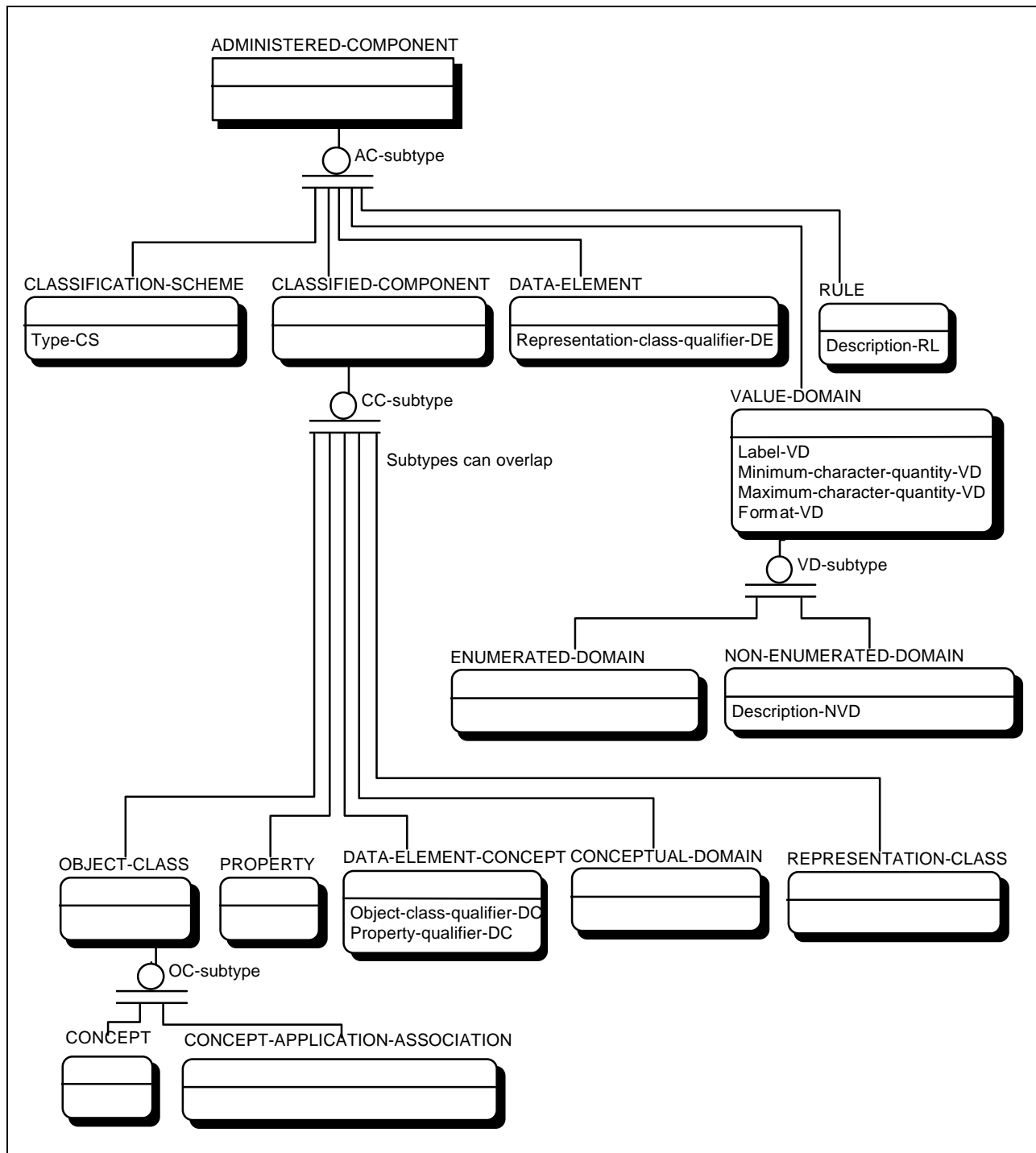


Figure B5 – IDEF1X type hierarchy

B.2. Object Role Modeling (ORM) – Natural language Information Analysis Method (NIAM) 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. There are seven diagrams that compose the model. These are:

1. Stewardship region,
2. Naming and identification region,
3. Classification region,
4. Data element concept administration region,
5. Conceptual domain and value domain administration region,
6. Data element administration region, and
7. Type hierarchy.

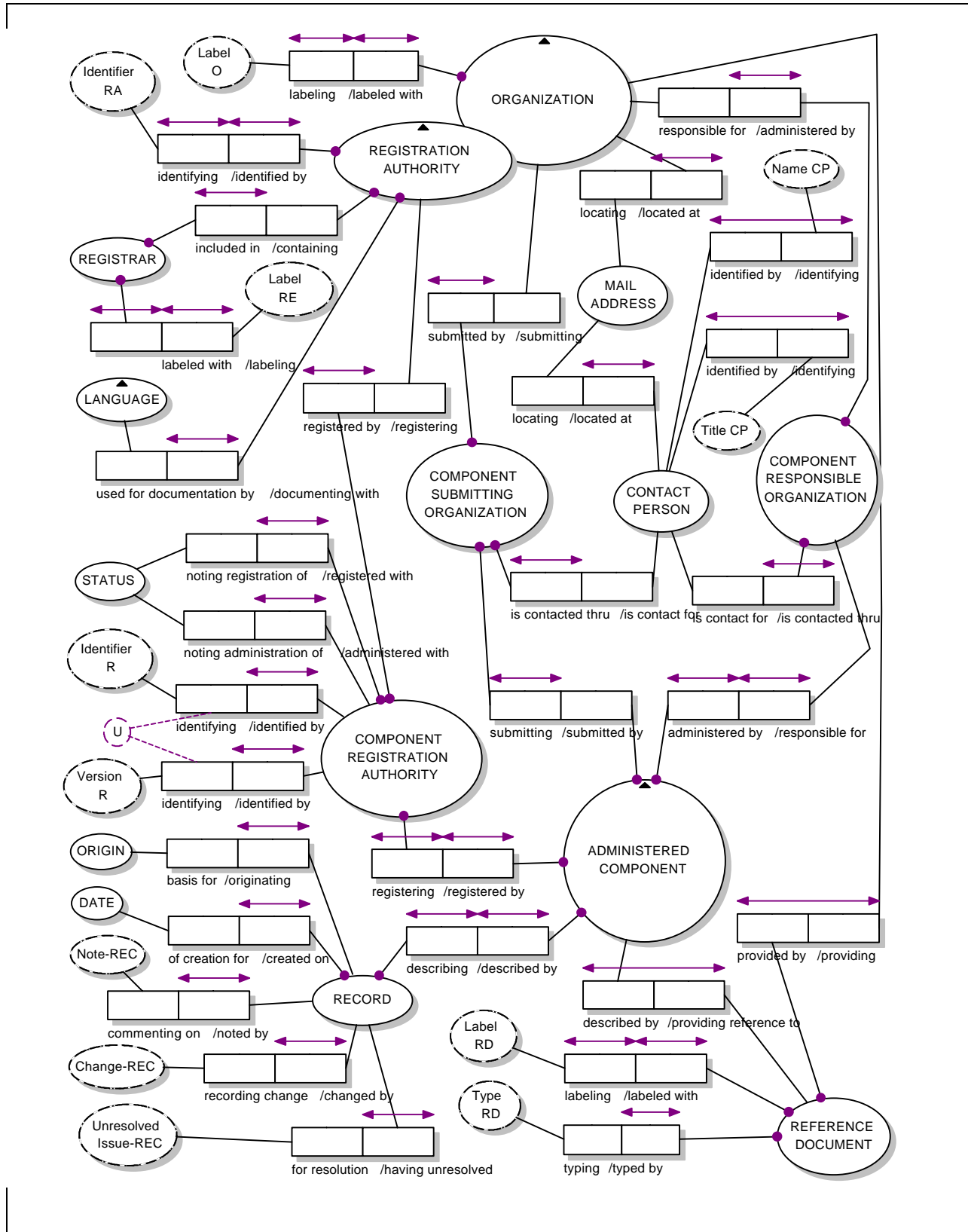


Figure B6 – ORM stewardship region

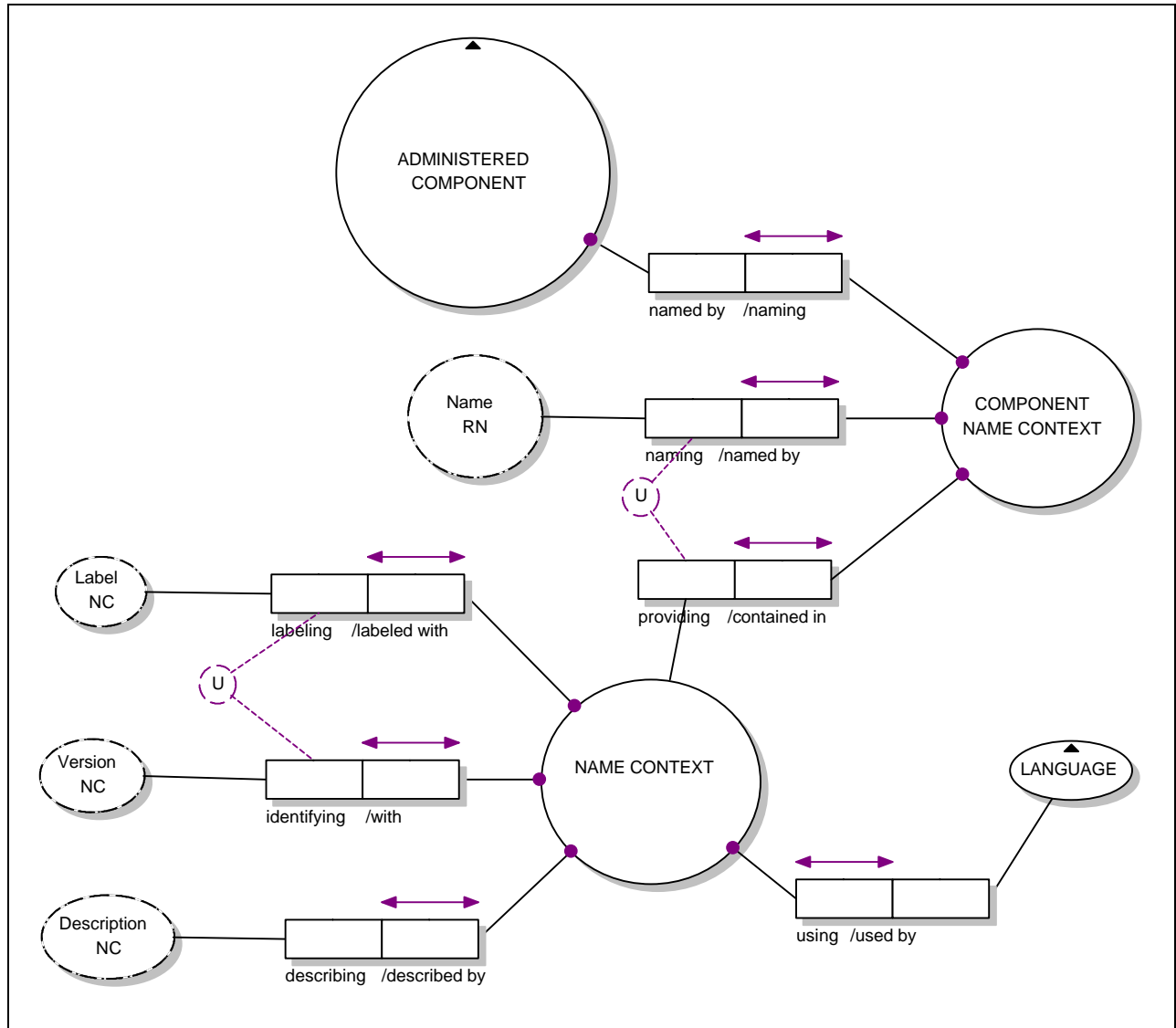


Figure B7 – ORM naming and identification region

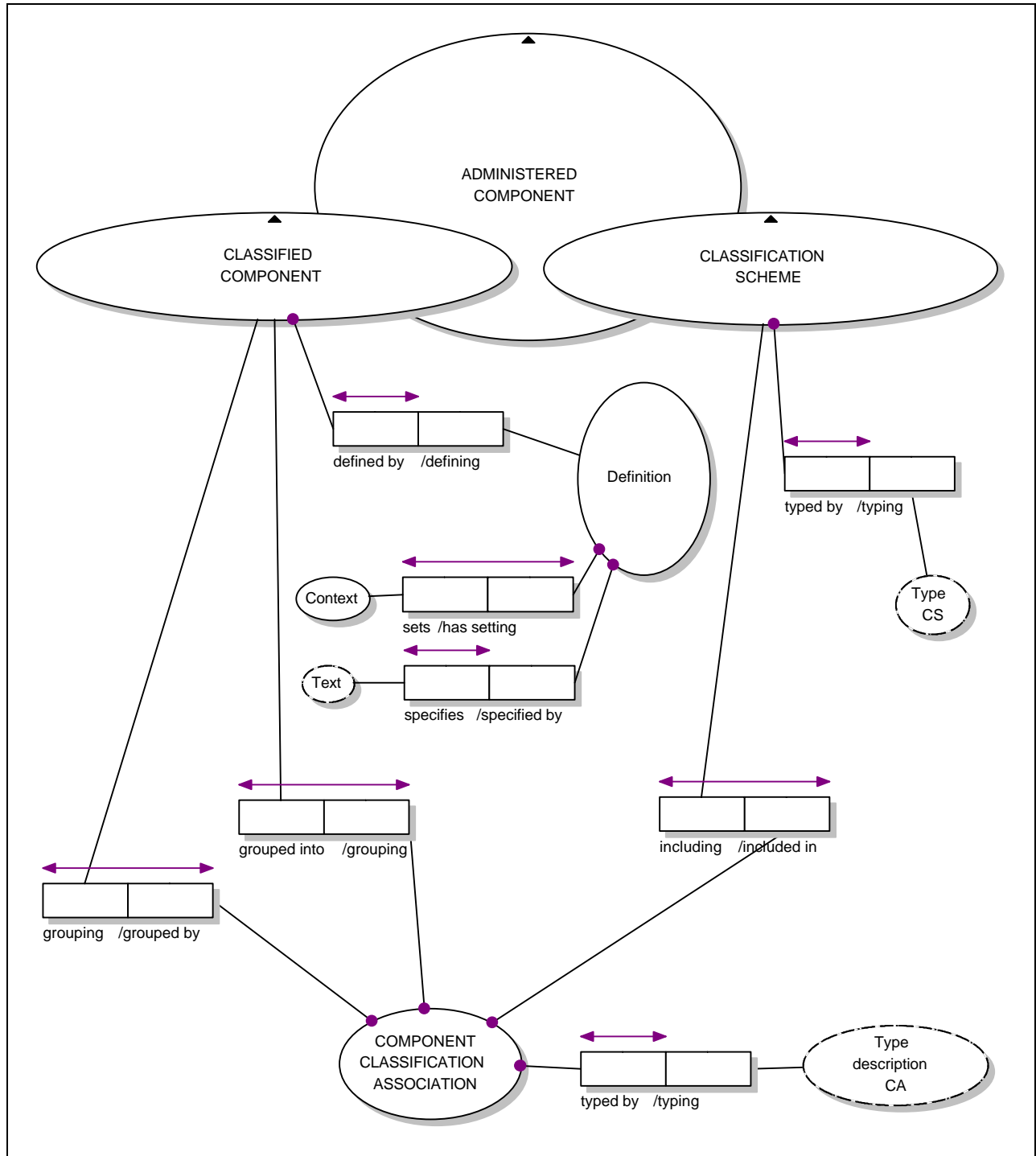


Figure B8 – ORM classification region

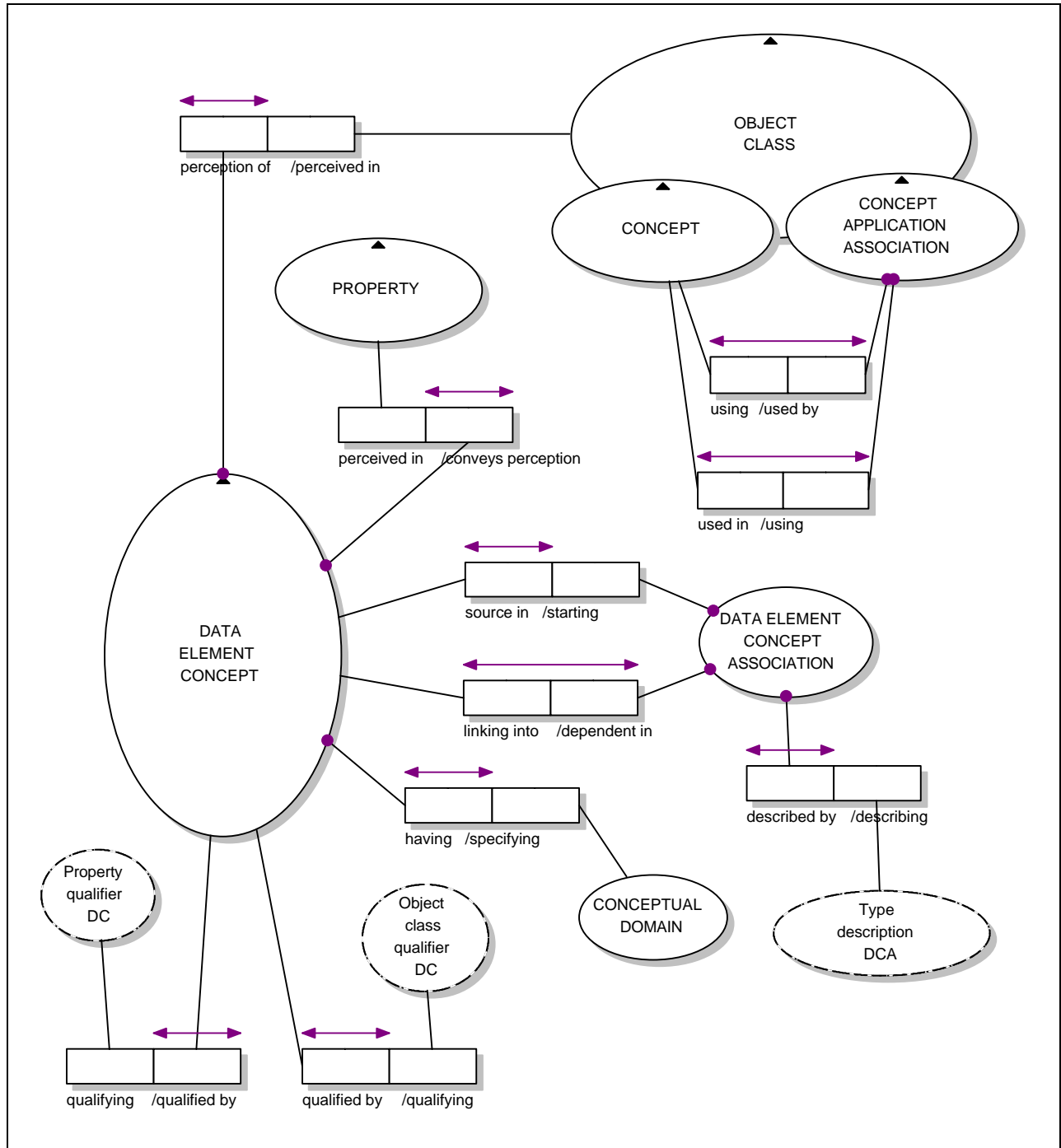


Figure B9 – ORM data element concept administration region

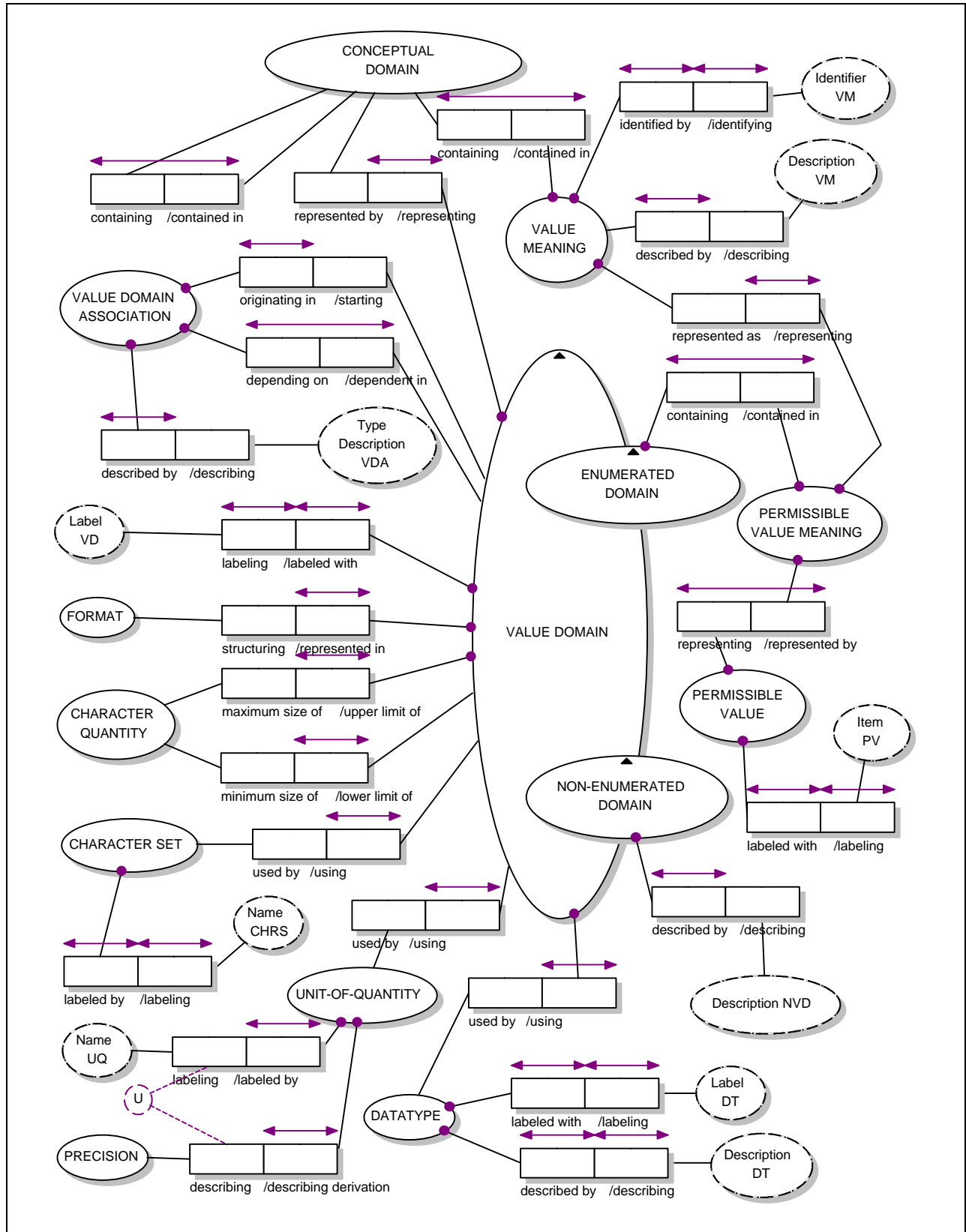


Figure B10 – ORM conceptual domain and value domain administration region

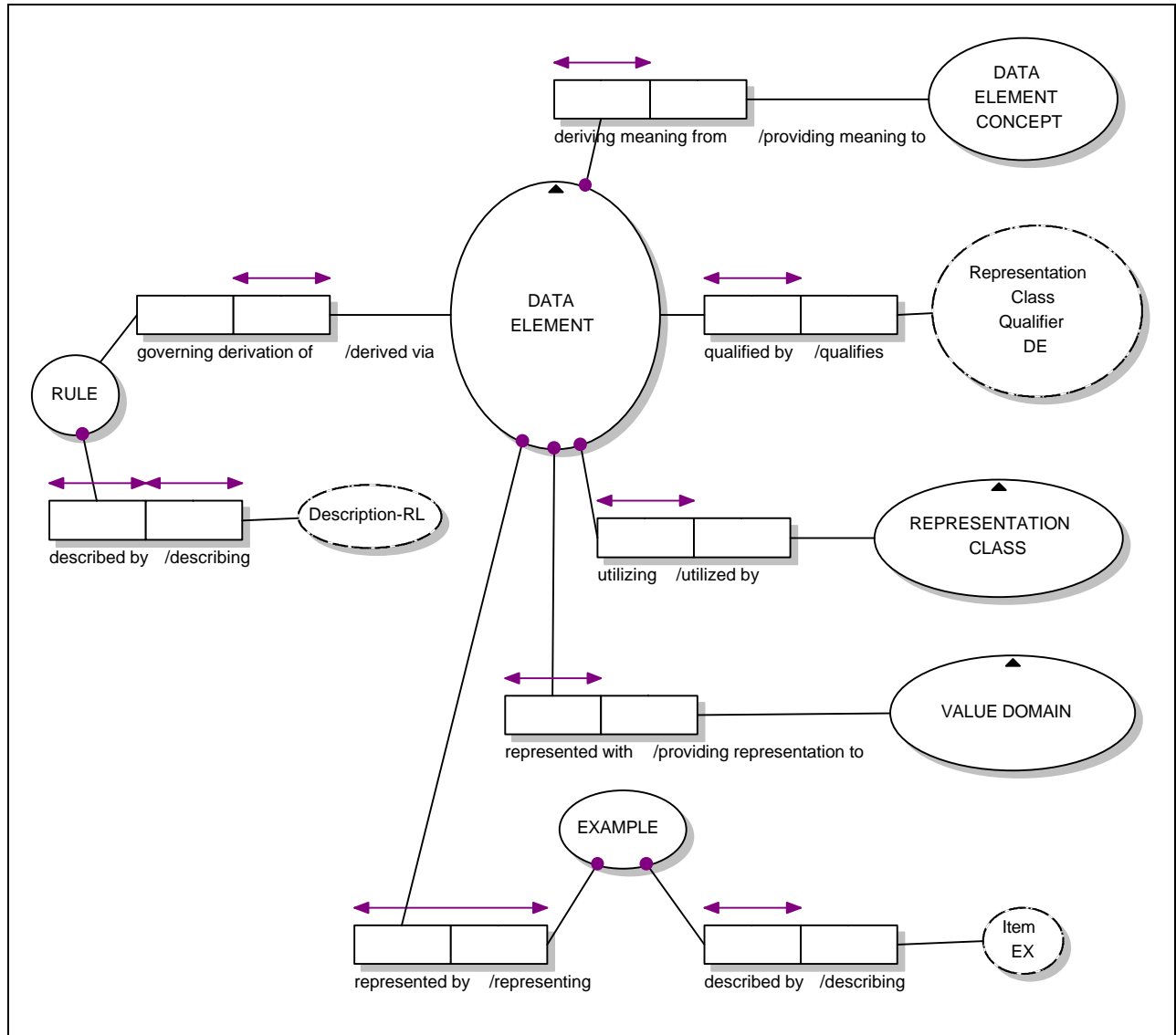


Figure B11 – ORM data element administration region

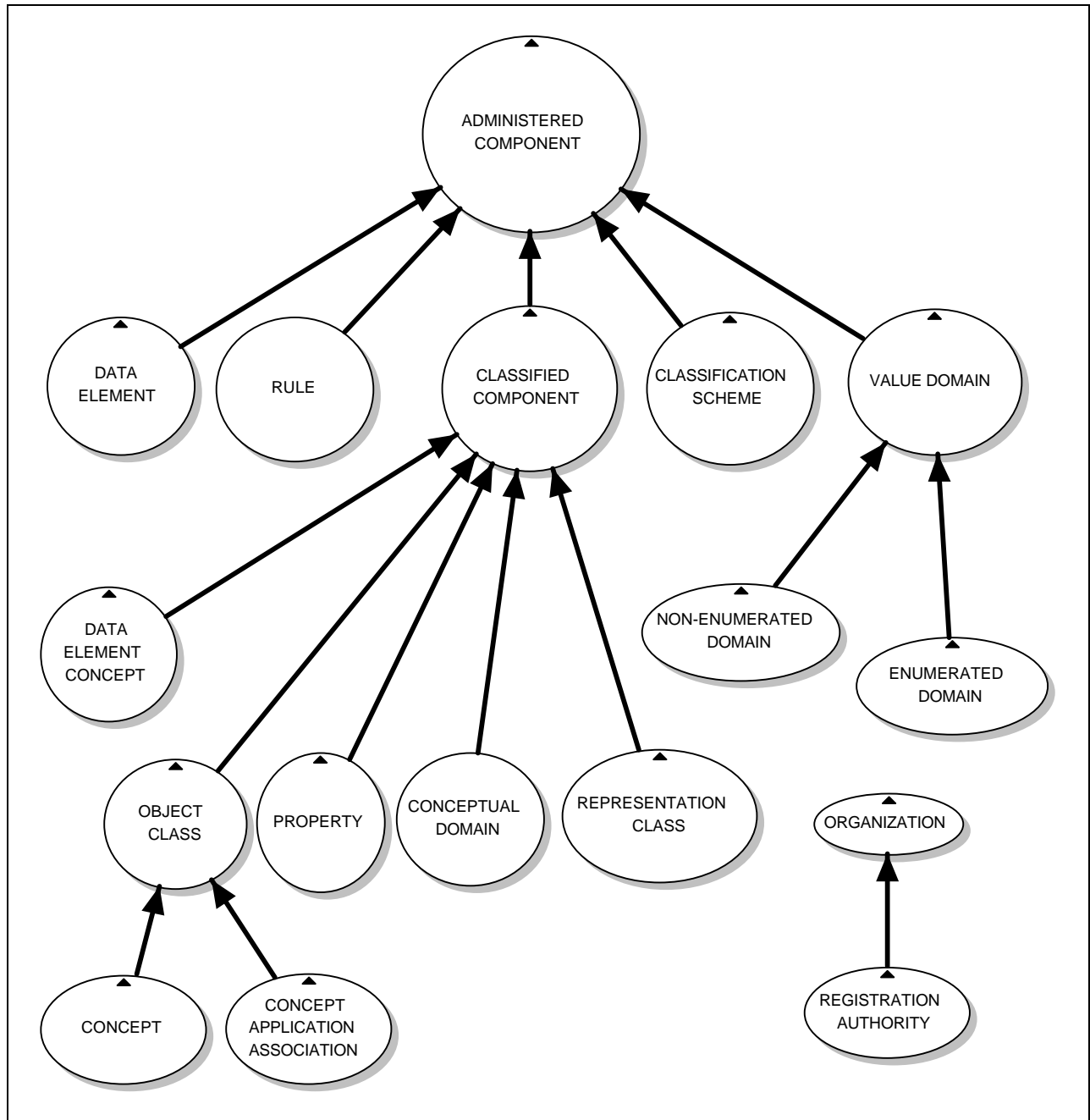


Figure B12 – ORM type hierarchy

The metamodel as a set of object role model sentences follows:

All administered components must be
classification schemes,
or classified components,
or data elements,
or rules,
or value domains.

An administered component must be administered by exactly one component responsible organization.

An administered component must be described by exactly one record.

An administered component may be described by any number of reference documents.

An administered component must be named by one or more component name contexts.

An administered component must be registered by one or more component registration authorities.

An administered component must be submitted by exactly one component submitting organization.

A change record may be recording change in any number of records.

A character quantity may be the maximum size of any number of value domains.

A character quantity may be the minimum size of any number of value domains.

A character set must be labeled by exactly one name [character set].

A character set may be used by any number of value domains.

A component classification association must be grouped by one or more classified components.

A component classification association must be grouping one or more classified components.

A component classification association must be included in one or more classification schemes.

A component classification association must be typed by exactly one type description [component classification association].

A classification scheme is always a kind of administered component.

A classification scheme cannot also be
a classified component,
or a data element,
or a rule,
or a value domain.

A classification scheme may include any number of component classification associations.

A classification scheme must be typed by exactly one type [classification scheme].

A classified component is always a kind of administered component.

A classified component cannot also be a
classification scheme,
or a data element,
or a rule,
or a value domain.

All classified components must be
conceptual domains,
or data element concept's,
or object class's,
or representation class's.

A classified component must be defined by one or more definitions.

A classified component may be grouped into any number of component classification associations.

A classified component may be grouping any number of component classification associations.

A component name context must be contained in one or more name contexts.

A component name context must be named by one or more name [component name context]s.

A component name context must be naming exactly one administered component.

A component responsible organization must be administered by exactly one organization.

A component responsible organization must be contacted through exactly one contact person.

A component responsible organization may be responsible for at most one administered component .

A component registration authority may be administered with at most one status.

A component registration authority must be identified by exactly one identifier [component registration authority].

A component registration authority must be identified by exactly one version [component registration authority].

A component registration authority must be registered by exactly one registration authority.

A component registration authority must be registered with exactly one status.

A component registration authority must be registering exactly one administered component.

Every component registration authority is associated uniquely with one combination of a identifier [component registration authority] identifying the component registration authority and a version [component registration authority] identifying the component registration authority.

A component submitting organization must be contacted through exactly one contact person.

A component submitting organization must be submitting exactly one administered component .

A component submitting organization must be submitted by exactly one organization.

A concept is always a kind of object class.

A concept cannot also be a concept application association.

A concept may be used in any number of concept application associations.

A concept may be using any number of concept application associations.

A concept application association is always a kind of object class.

A concept application association cannot also be a concept.

A concept application association must be used by one or more concepts.

A concept application association must be using one or more concepts.

A conceptual domain is always a kind of classified component.

A conceptual domain may be contained in any number of conceptual domains.

A conceptual domain may be containing any number of conceptual domains.

A conceptual domain may be containing any number of value meanings.

A conceptual domain may be represented by any number of value domains.

A conceptual domain may be specifying any number of data element concepts.

A contact person must be identified by exactly one name [contact person].

A contact person may be identified by at most one title [contact person].

A contact person may be the contact for any number of component submitting organizations.

A contact person may be the contact for any number of component responsible organizations.

A contact person may be located at any number of mail addresses.

A context may be utilized by any number of definitions.

A data element concept is always a kind of classified component.

A data element concept must be conveying the perception of exactly one property.

A data element concept must be having exactly one conceptual domain.

A data element concept may be linked into any number of data element concept associations.

A data element concept must be the perception of exactly one object class.

A data element concept may be providing meaning to any number of data elements.

A data element concept may be qualified by at most one object class qualifier.

A data element concept may be qualified by at most one property qualifier.

A data element concept may be the source in at most one data element concept association.

-
- A data element concept association must be dependent in one or more data element concepts.
- A data element concept association must be described by exactly one type description [data element concept association].
- A data element concept association must be starting one or more data element concepts.
- A data element is always a kind of administered component.
- A data element cannot also be a
classification scheme,
or a classified component,
or a rule,
or a value domain.
- A data element may be derived via at most one rule.
- A data element must be deriving meaning from exactly one data element concept.
- A data element must have exactly one representation class.
- A data element may be qualified by at most one representation class qualifier.
- A data element must be represented by one or more examples.
- A data element must be represented with exactly one value domain.
- A datatype must be described by exactly one description [datatype].
- A datatype must be labeled with exactly one label [datatype].
- A datatype may be used by any number of value domains.
- A date may be the creation for any number of records.
- A definition must be defining exactly one classified component.
- A definition must be specified by one or more text [definition]s.
- A definition must be utilizing one or more contexts.
- A description [datatype] may be describing at most one datatype.
- A description [name context] may be describing any number of name contexts.
- A description [non-enumerated domain] may be describing any number of non-enumerated domains.
- A description [rule] may be describing at most one rule.
- A description [value meaning] may be describing any number of value meanings.
- 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 meanings.
- An example must be described by exactly one item [example].
- An example must be representing one or more data elements.
- A format may be structuring any number of value domains.
- An identifier [component registration authority] may be identifying any number of component registration authorities.
- An identifier [registration authority] may be identifying at most one registration authority.
- An identifier [value meaning] may be identifying at most one value meaning.
- An item [example] may be describing any number of examples.
- An item [permissible value] may be labeling at most one permissible value.
- A label [datatype] may be labeling at most one datatype.
- A label [name context] may be labeling any number of name contexts.
- A label [organization] may be labeling at most one organization.
- A label [reference document] may be labeling at most one reference document.
- A label [registrar] may be labeling at most one registrar.
- A label [value domain] may be labeling at most one value domain.

A language may be used by any number of name contexts.

A language may be used for documentation by any number of registration authorities.

A mail address may be locating any number of contact persons.

A mail address may be locating any number of organizations.

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 description [non-enumerated domain].

A name [character set] may be labeling at most one character set.

A name [contact person] may be identifying any number of contact persons.

A name [component name context] may be naming any number of component name contexts.

A name [unit of quantity] may be labeling any number of unit of quantities.

A name context must be described by exactly one description [name context].

A name context must be labeled with exactly one label [name context].

A name context may be providing at most one component name context.

A name context must be using exactly one language.

A name context must be with exactly one version [name context].

Every name context is associated uniquely with one combination of
a label [name context] labeling the name context
and a version [name context] identifying the name context.

A note [record] may be commenting on any number of records.

An object class qualifier may be qualifying any number of data element concepts.

An object class is always a kind of classified component.

All object class's must be
concept's,
or concept application associations.

An object class may be perceived in any number of data element concepts.

An organization must be labeled with exactly one label [organization].

An organization may be located at exactly one mail address.

An organization may be providing any number of reference documents.

An organization may be responsible for any number of component responsible organizations.

An organization may be submitting any number of component submitting organizations.

An origin may be the basis for any number of records.

A permissible value must be labeled with exactly one item [permissible value].

A permissible value must be representing one or more permissible value meanings.

A permissible value meaning must be contained in one or more enumerated domains.

A permissible value meaning must be represented by one or more permissible values.

A permissible value meaning must be representing exactly one value meaning.

A precision may be describing any number of unit of quantities.

A property is always a kind of classified component.

A property may be perceived in any number of data element concepts.

A property qualifier may be qualifying any number of data element concepts.

A representation class qualifier may be qualifying any number of data elements.

A record may be changed by at most one change record.

A record must be created on exactly one date.

A record must be describing exactly one administered component.

A record may be having unresolved at most one unresolved issue.

- A record may be noted by at most one note [record].
- A record may be originating at most one origin.
- A reference document must be labeled with exactly one label [reference document].
- A reference document must be provided by one or more organizations.
- A reference document may be providing reference to any number of administered components.
- A reference document must be typed by exactly one type [reference document].
- A registrar must be included by exactly one registration authority.
- A registrar must be labeled by exactly one label [registrar].
- A registration authority is always a kind of organization.
- A registration authority must be containing one or more registrars.
- A registration authority must be documenting with exactly one language.
- A registration authority must be identified by exactly one identifier [registration authority].
- A registration authority may be registering any number of component registration authorities.
- A representation class is always a kind of classified component.
- A representation class may be describing any number of data elements.
- A rule is always a kind of administered component.
- A rule cannot also be a
classification scheme,
or a classified component,
or a data element,
or a value domain.
- A rule must be described by exactly one description [rule].
- A rule may be governing derivation for any number of data elements.
- A status may be noting administration of any number of component registration authorities.
- A status may be noting registration of any number of component registration authorities.
- A text [definition] may be specifying at most one definition.
- A title [contact person] may be identifying any number of contact persons.
- A type [classification scheme] may be typing any number of classification schemes.
- A type [reference document] may be typing any number of reference documents.
- A type description [data element concept association] may be describing any number of data element concept associations.
- A type description [value domain association] may be describing any number of value domain associations.
- A type description [component classification association] may be typing any number of component classification associations.
- A unit of quantity must be describing derivation of exactly one precision.
- A unit of quantity must be labeled by exactly one name [unit of quantity].
- A unit of quantity may be used by any number of value domains.
- Every unit of quantity is associated uniquely with one combination of
a name [unit of quantity] labeling the unit of quantity
and a precision describing the unit of quantity.
- An unresolved issue may be for resolution of any number of records.
- A value domain is always a kind of administered component.
- A value domain cannot also be a
classification scheme,
or a classified component,

or a data element,
or a rule.

All value domains must be enumerated domains,
or non-enumerated domains.

A value domain may be dependent in any number of value domain associations.

A value domain must be labeled with exactly one label [value domain].

A value domain may be with a lower limit of at most one character quantity.

A value domain may be providing representation to any number of data elements.

A value domain must be represented in exactly one format.

A value domain must be representing exactly one conceptual domain.

A value domain may be starting any number of value domain associations.

A value domain must be with an upper limit of exactly one character quantity.

A value domain may be using at most one character set.

A value domain must be using exactly one datatype.

A value domain may be using at most one unit of quantity.

A value domain association must be depending on one or more value domains.

A value domain association must be described by exactly one type description [value domain association].

A value domain association must be originating in exactly one value domain.

A value meaning must be contained in one or more conceptual domains.

A value meaning must be described by exactly one description [value meaning].

A value meaning must be identified by exactly one identifier [value meaning].

A value meaning must be represented as one or more permissible value meanings.

A version [name context] may be identifying any number of name contexts.

A version [component registration authority] may be identifying any number of component registration authorities.

B.3. Unified Modeling Language (UML) representation model

The Unified Modeling Language (UML) representation model of the Metamodel for the management of shareable data is presented as a data model. This model does not represent any of the methods for the objects in the model. There are seven diagrams that comprise the model. These are:

1. Stewardship region,
2. Naming and identification region,
3. Classification region,
4. Data element concept administration region,
5. Conceptual domain and value domain administration region,
6. Data element administration region, and
7. Full model.

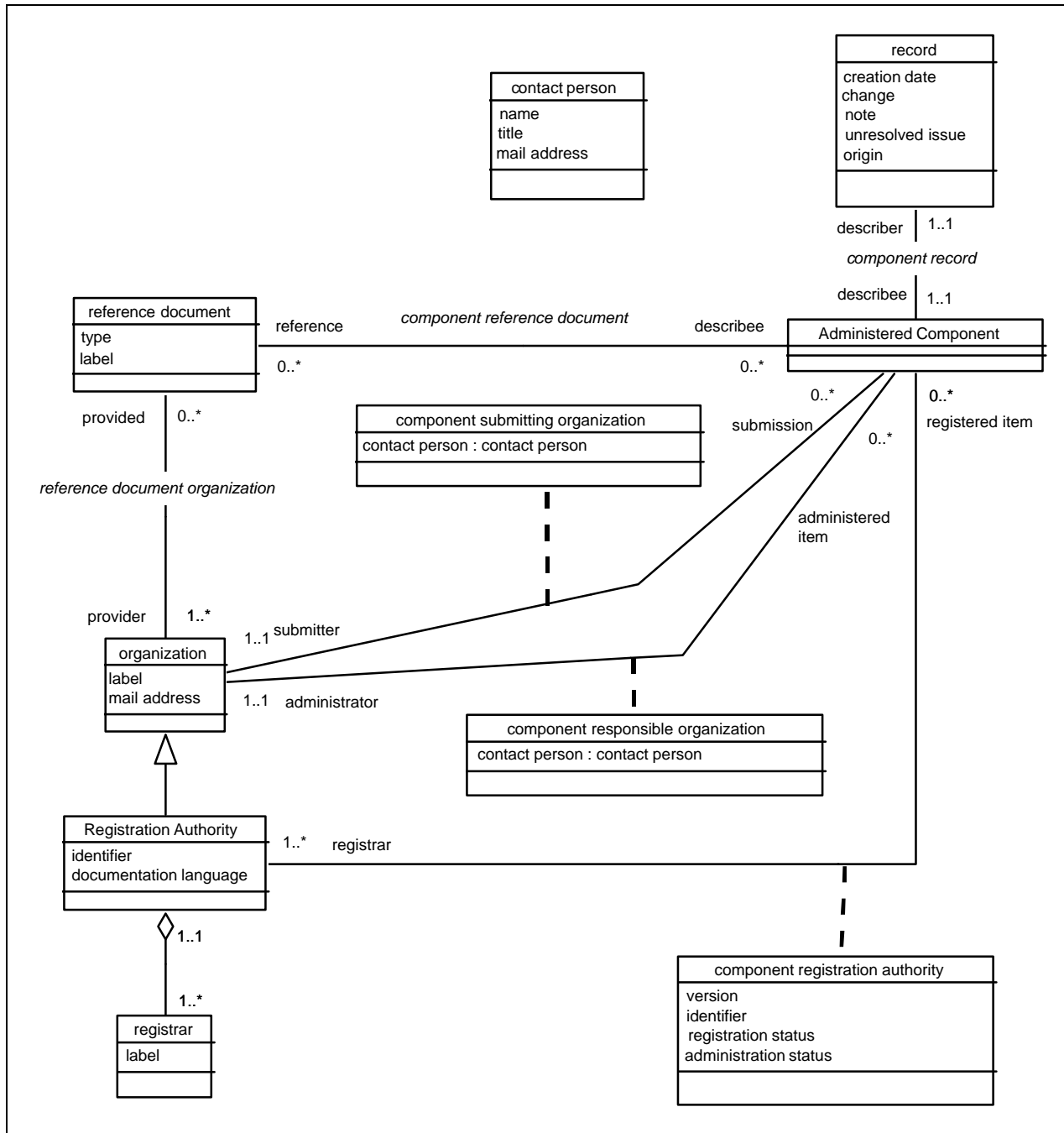


Figure B13 – UML stewardship region

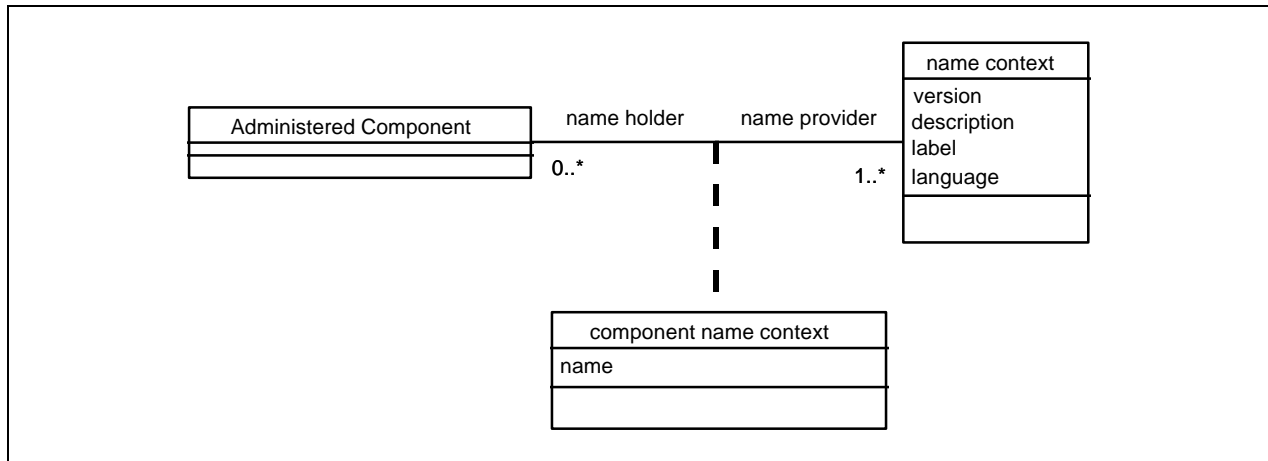


Figure B14 – UML naming and identification region

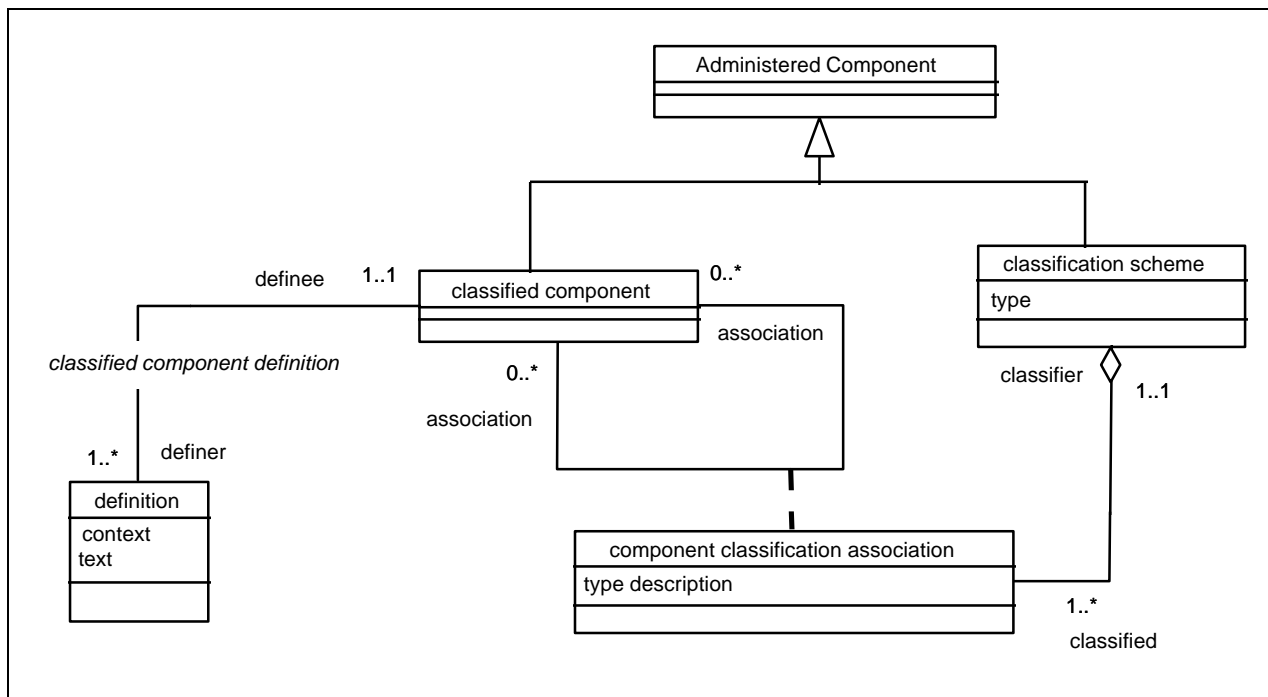


Figure B15 – UML classification region

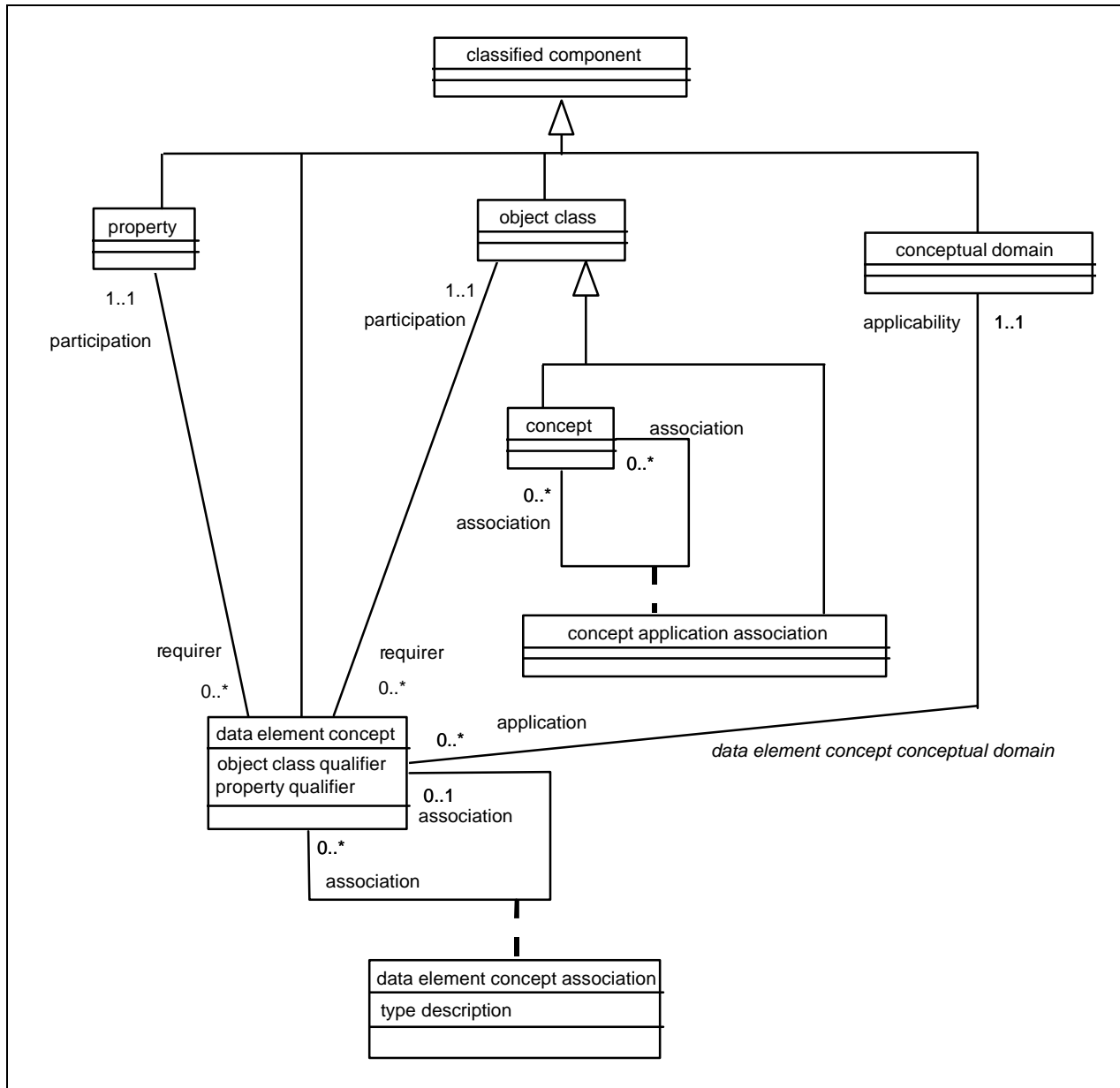


Figure B16 – UML data element concept administration region

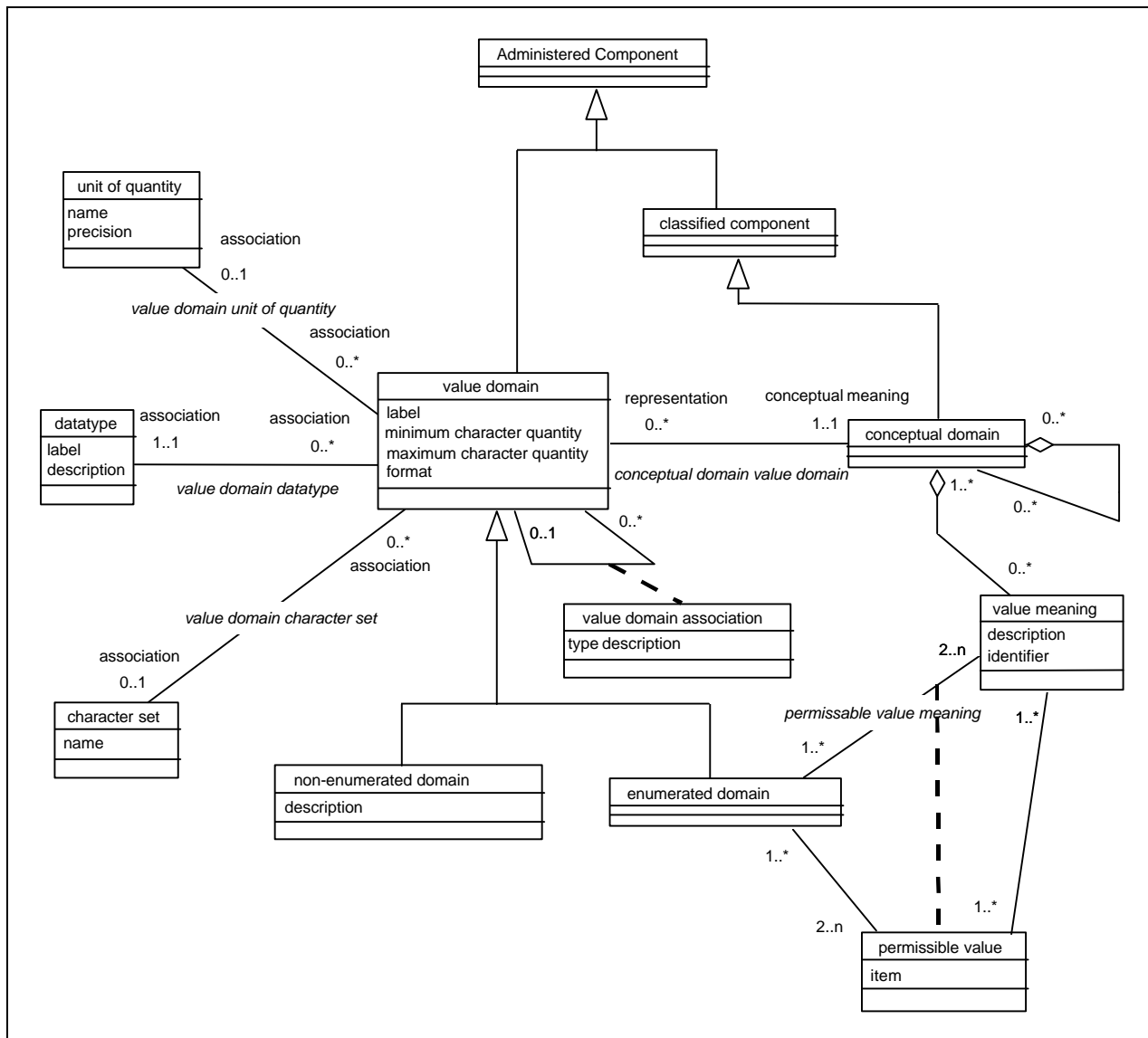


Figure B17 – UML conceptual domain and value domain administration region

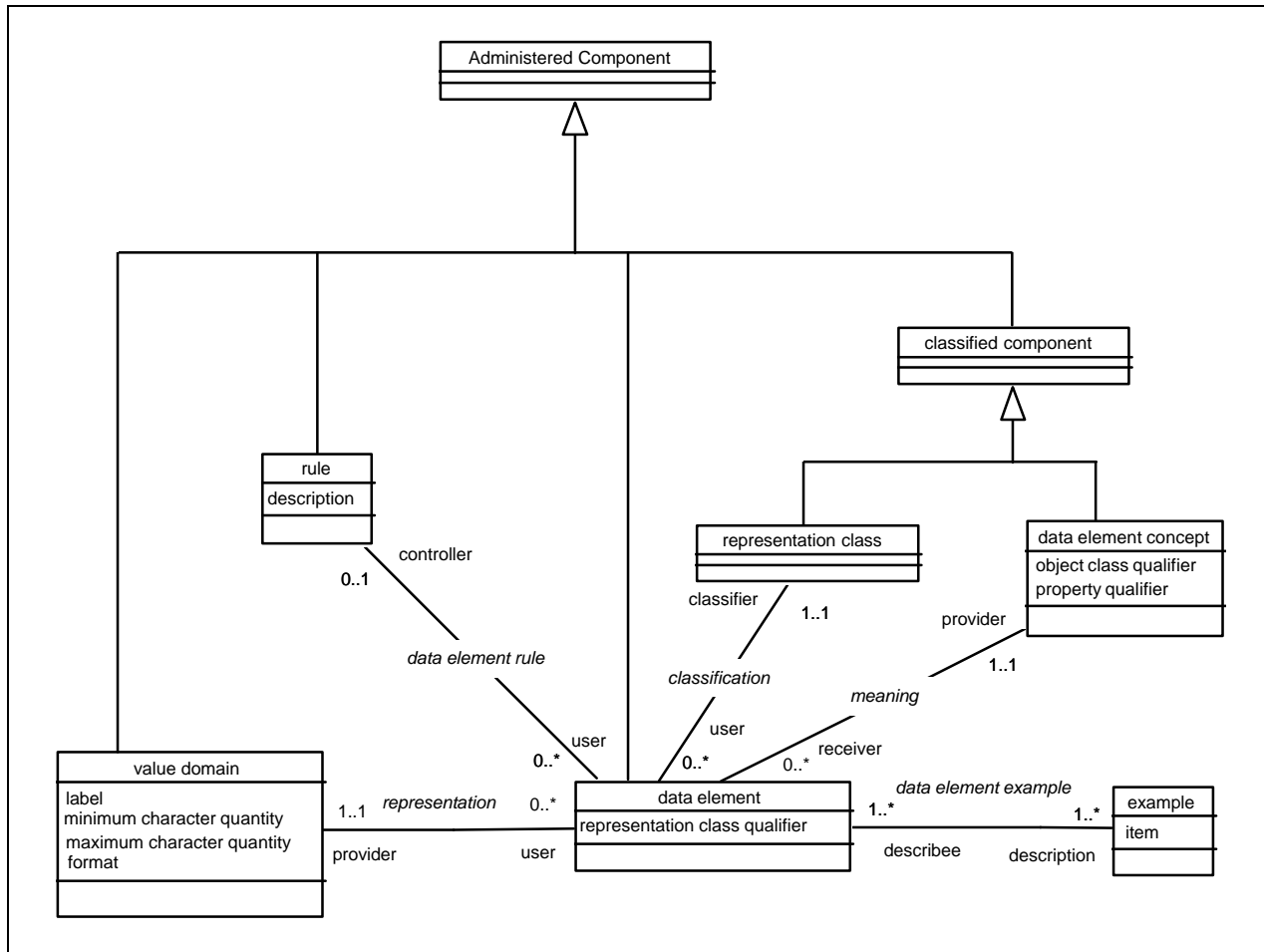


Figure B18 – UML data element administration class region

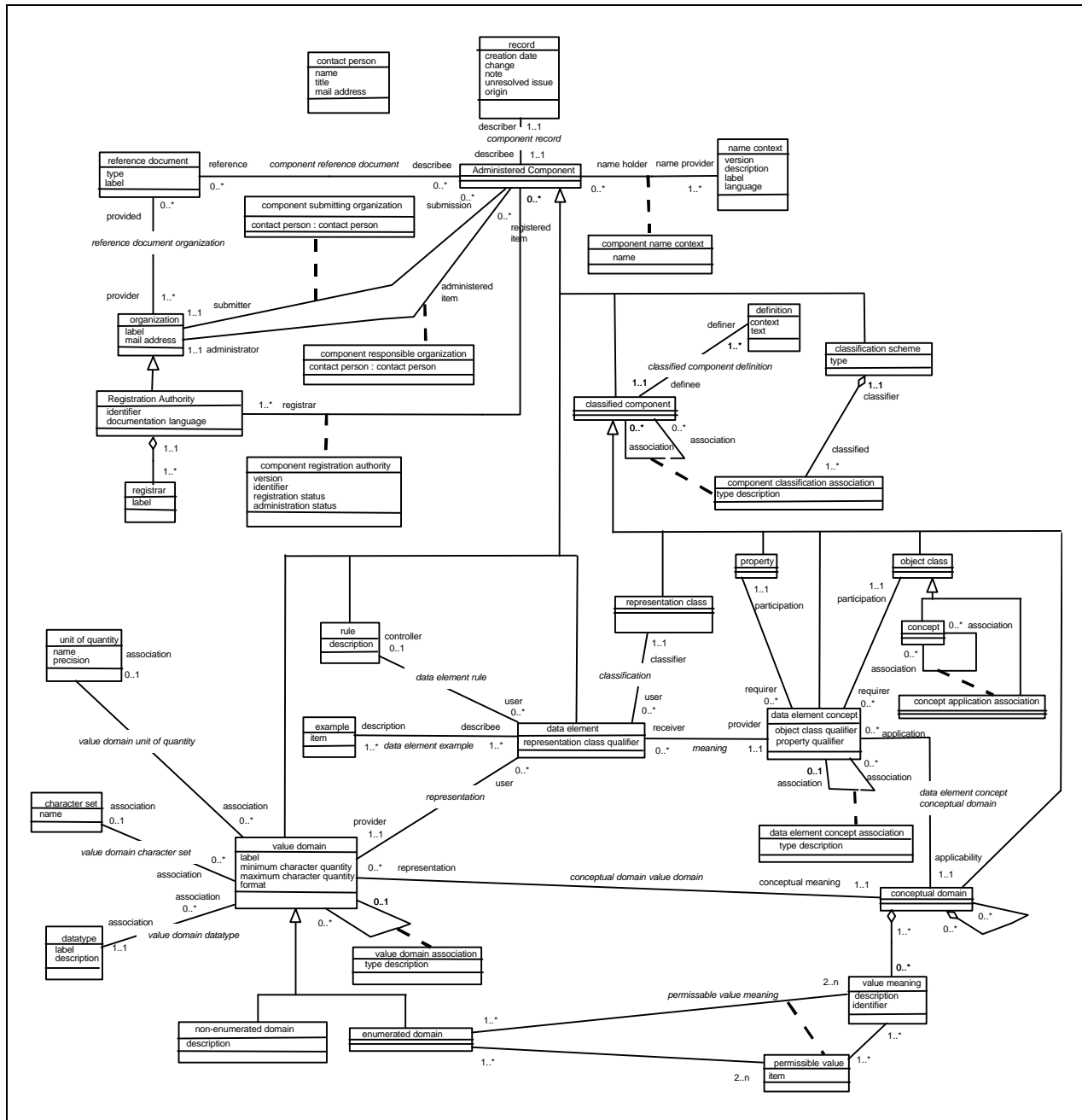


Figure B19 – UML full model

Annex C (informative)

Principles of managing shared data

These principles were used while developing the metamodel. Each principle is directly or indirectly supported by the metamodel. Conversely, this is an itemized description of much of the conceptual data structure depicted in the data model. It includes many of the more significant:

- Fundamental principles and “business rules” for data registration.
- Definitions that are applicable within the scope of this standard.
- Constraints and integrity rules for the data used for data registration.
- Structural relationships and cardinalities among data element components.
- References to terminology used elsewhere.
- Objectives for good information management.

These principles are fundamental to the use of a conformant implementation this metamodel. If the user deviates from any principle, to resulting data registry may not realize expectations.

C.1 Data

- C.1.1 Data is a representation of a fact, idea, or instruction in a formalized manner suitable for communication, interpretation, or processing by humans or by machines. (This definition refers to a group taken as a unit thus it is used with a singular verb.)
- C.1.2 Data must be able to be created, collected, organized, recorded, processed, and stored in a medium in a retrievable form.
- C.1.3 Data represents data element concepts (i.e., the properties of object classes) by using a set of symbols that are perceived. These may be words made up of characters, icons, sounds, Braille, etc.
- C.1.4 Data allows us to consider an object that exists in the real world without having the actual object present. In other words, data provides an abstraction of the real world object.
- C.1.5 Data that is derived should be registered the same as any other data if it is stored.
- C.1.6 An instantiation of a single element of data is called a “data item” (a.k.a. datum).
- C.1.7 A single type or class of structured data treated as a cohesive whole is called a “data unit”.
- C.1.8 A single unit of data that is considered indivisible within a universe of discourse is called a “data element”. It is identical to what some others call a “simple data element”.
- C.1.9 Data used to describe the meaning or characteristics of data is called “metadata”.

C.2 Concept

- C.2.1 A concept is a unit of thought (an idea) constituted by the abstraction of the common characteristics of a set of objects.

- C.2.2 An object may be any person, place, event, or other thing that has separate and distinct existence in the real world.
- C.2.3 Each concept can be shown as a more specialized type, or a component part, of one or more higher-ordered concepts.
- C.2.4 A concept inherits characteristics from one or more generalized supertype(s).

C.3 Object class

- C.3.1 Humans tend to group objects when they have similar traits. When we group a set of similar things, we refer to it as a "type" or "class". A single category of "things" or "objects" is called an "object class".
- C.3.2 An object class is a set of concepts, abstractions, or things in the natural world that can be identified with explicit boundaries and meaning and whose properties and behavior all follow the same rules.
- C.3.3 Object classes may be a single concept or a set of concepts in a relationship with each other to form a more complex concept. Concepts in relationship with other concepts are sometimes called "concept systems".
- C.3.4 Data is a representation of properties of object classes.
- C.3.5 An object class is the same as an entity (entity type) or relationship in the relational paradigm.
- C.3.6 It is desirable to describe object classes without redundancy within the universe of discourse. The same object class, but with different names and/or wording of definitions, should eventually be normalized.

C.4 Property

- C.4.1 A property is a classification of any feature that humans naturally use to distinguish one individual object from another.
- C.4.2 When we describe an object, we describe its properties. If we know nothing about the kind of properties an object has, we are not aware of the object.
- C.4.3 A property class refers to the conceptual part of an attribute, i.e., without representation.
- C.4.4 A property class has no particular associated means of representation by which it can be communicated.
- C.4.5 A property class may be associated with more than one object class where it describes a conceptual attribute (one without representation).
- C.4.6 A property class is a concept playing the role of a property class in a data element concept. Only certain concepts have the ability to behave as a property class. Whether one of these concepts is acting as a property class cannot be determined until it is associated with an object class in a data element concept.
- C.4.7 It is desirable to describe properties without redundancy within the universe of discourse. The same property class, but with different names and/or wording of definitions, should eventually be normalized (a.k.a. harmonized or rationalized).
- C.4.8 Properties are sometimes called "characteristics".

C.5 Data element concept

- C.5.1 A data element concept is the union of two or more concepts with one concept playing the role of a property.
- C.5.2 A data element concept is the human perception of a single property of an object class, identified and described independently of any particular representation.
- C.5.3 A data element concept has a definition different from its object class or property.
- C.5.4 While any specifically defined data element concept may have several representations in a universe of discourse, each such data element concept should have a preferred data element representation in a data registry.
- C.5.5 If an object class and a property are normalized across the universe of discourse, the data element concept will also be normalized.
- C.5.6 Since the object class and the property have no representation, the data element concept will have no representation.
- C.5.7 A data element concept may be represented as a data element.
- C.5.8 Data element concepts are sometimes called "Basic Semantic Units".

C.6 Attribute

- C.6.1 An attribute is a characteristic of an object class that the business chooses to record as data.
- C.6.2 An attribute is always associated with only one object class.
- C.6.3 An attribute is complex. It is composed of both a property and a representation. The concept of an attribute is separate from how it is represented.
- C.6.4 When a characteristic of a data unit is being described, the attribute is called a "meta-attribute".
- C.6.5 The metadata used to describe data units requires many meta-attributes. A set of meta-attributes of data units bundled together as a module for reusability is called a "metadata set".

C.7 Representation

- C.7.1 Before a data element concept can become a data unit it must be expressed as a term, character, symbol, et cetera that represents a meaning of the property class. Such a notation is called "representation".
- C.7.2 Representation describes how a data element concept appears in a persistent store, on a screen, on paper, et cetera. Representations are human-interpretable (sound, tactile, visual).

C.8 Data element representation

- C.8.1 A data element representation is the part of a data element having a value domain, datatype, and, if a quantity, a unit of quantity.
- C.8.2 A set of similar data element representations (i.e., a "type" or "class") are grouped as a representation class for classification purposes.
- C.8.3 A data element representation may be associated with one or more data element concepts.

- C.8.4 The permissible values of a value domain may be expressed by specifying the range from its lower to upper limit, by a rule, by a procedure or scheme, or by enumeration in a finite list.
- C.8.5 A data element representation may have a "compound datatype" that separates the representation into constituent parts. A compound datatype would only be plausible where the data element representation could be used as the representation of a single data element concept.
- C.8.6 A value domain may be an aggregation of a set of smaller value domains.
- C.8.7 It is desirable to describe data element representations without redundancy within the universe of discourse. Data element representations with the identical value domain, datatype, and, if a quantity, a unit of quantity, should eventually be normalized.

C.9 Data element

- C.9.1 A data element is a single unit of data that is considered indivisible in its shareable universe.
- C.9.2 A data element cannot be decomposed into more fundamental constituent parts of information that have useful meanings within its shareable universe.
- C.9.3 A data element is an electronic or written representation of a data element concept.
- C.9.4 Data elements are the basic building blocks of data.
- C.9.5 A data element is the association of a data element concept with a data element representation.
- C.9.6 There may be more than one alternate way a data element concept is represented as a data element by associating it with different data element representations.
- C.9.7 A data element concept associated with two or more data element representations are different data elements.
- C.9.8 The term "data element" refers to a type or class (i.e., the complete set of instances) and not any particular instantiation of a value for a data element. Where a specific data element specimen occurs, it is called a "data element instance".
- C.9.9 Each data element will represent no more than a single data element concept.
- C.9.10 A data element is identical to an attribute in many data modeling paradigms. In a logical data model, a data element is often considered an attribute.
- C.9.11 Data elements are individual, discontinuous or discrete pieces of information. They are not defined in analog or digital flows as used in electronically transmitted audio or video.
- C.9.12 Data elements can be "persistent data" or "transient data" — data that is created and consumed without ever being stored in a database.
- C.9.13 A data element is described independent of the physical space in which it is stored or transmitted. A single physical space (e.g., a field or column in a database) may be reused for more than one data element.
- C.9.14 If a data element concept and a data element representation are normalized across the universe of discourse, the data element will also be normalized.
- C.9.15 Each data element should have one identifier, one definition, one representation, one data steward, and one common set of business rules governing that element throughout the enterprise.

C.9.16 A data element is associated with a specific set of values. Any value can be expressed by a set of symbols.

C.9.17 A data element always takes on a value from a set of allowed data values. If it cannot be associated with a set of distinct values, it is not a data element. These values can include written characters, sounds, or images.

C.10 Enumerated domain

C.10.1 Each value in an enumerated domain represents an abstraction of an object in the real world.

C.10.2 The collection of the object concepts in an enumerated domain is called a "conceptual domain". It is composed of a set of all permissible value meanings without a specified representation.

C.10.3 Once a data element concept is associated with a data element representation with an enumerated domain, a value meaning must be associated with each permissible value in the set.

C.10.4 Each value meaning in a conceptual domain may be associated with a permissible value member of more than one enumerated domain representations.

C.11 Identifier

C.11.1 Each data element, object class, object class, property, data element concept, conceptual domain, value domain, and representation class will be uniquely identified by its identifier within a Registration Authority.

C.11.2 Identifiers will carry no intelligence.

C.12 Name

C.12.1 A name will not be used as an identifier.

C.12.2 Various names for various contexts where the names are used and have meaning are important metadata.

C.12.3 Classification names can be constructed from the various name meta-attributes associated with object classes and representation classes.

C.13 Quality

C.13.1 Data elements have several levels of quality.

C.13.2 All data used in the enterprise should be recognized, regardless of quality.

C.14 Registration Authority

C.14.1 A Registration Authority is self-nominated.

C.14.2 A Registration Authority obtains a registration authority identifier.

C.14.3 A Registration Authority manages a data registry.

C.14.4 Each Registration Authority establishes the datatype categories used in its data registry.

C.14.5 Each Registration Authority establishes the procedures used to register data.

C.14.6 A Registration Authority may have an organization or individual within acting as a registrar.

C.15 Data Registry

- C.15.1 A data registry is a structure to store data about data that may be shared among Information Systems and/or organizations.
- C.15.2 A data registry does not include data about Information Systems.
- C.15.3 A data registry does not include data about the (conceptual, logical, or physical) structure of databases.
- C.15.4 A data registry will be administered by a Registration Authority who acts as a resource to the registry's clients for establishing metadata about registered data and their applications.
- C.15.5 A data registry is a place to keep characteristics of classes of objects that exist in the real world that the business chooses to record as data.
- C.15.6 A data registry provides a centralized directory to describe the meaning, representation, and identification of units of data and their values.
- C.15.7 A data registry enables data to be well described so that users know exactly what facts are represented.
- C.15.8 A data registry supports data sharing with cross-system and cross-organization descriptions of common data.
- C.15.9 A data registry is a database with appropriate analysis and user interface software.
- C.15.10 A data registry may be a stand-alone system, or may be part of an Information Resource Dictionary System (IRDS) or any other information repository.
- C.15.11 A data registry assists in preventing redundancy of registering the same data (described by a metadata set) multiple times within the same registry.
- C.15.12 A data registry assists in preventing unplanned redundancy of the same business fact in different data elements.
- C.15.13 A data registry promotes reusability of data descriptions. Metadata in a data registry should be structured as modules to maximize the reusability of these metadata sets.
- C.15.14 The structure of the data registry is purposely contrived to avoid the common confusion between multiple-element units of data and single elements of data.
- C.15.15 Descriptions of shareable data must be conveniently and immediately accessible to all users.
- C.15.16 Registered data will be organized for easy accessibility.
- C.15.17 Each data element will be classified by the object class for which it represents a property.
- C.15.18 A data registry that is available to all interested parties facilitates harmonization and interchange among the parties.
- C.15.19 A data registry incorporates all of the fundamental principles itemized above.
- C.15.20 A data registry is sometimes called a "register".

Annex D (informative)

Data registry uses and users

Data users can share data if they use a common database. However, users often wish to exchange data across organizations and systems without incurring the delay and cost of creating a communal database. A more practical way of sharing data is to create a **catalog** of descriptions of shareable data. The catalog contains descriptions of the type of data we have reason to share with others. It does not contain any information about **instances** of data. It describes **types** of data including their allowed values. This data describing shareable data is what we call metadata.

With this approach, the key to sharing data is thus to share and reuse metadata. We can put this metadata in a catalog that is organized in a way that all stakeholders can use it. Users can have direct access to items in the catalog with convenient retrieval procedures.

When we catalog all the data used in an enterprise, we are confronted with several ways to represent the same "fact". Information in the catalog can be organized to assist data administrators to identify redundancy. Data administrators can use the metadata catalog to standardize preferable data descriptions. By labeling well-described and sanctioned units of data in the catalog, other users will know which form of data representation to use.

Software engineers can view descriptions of data that others have already documented in the catalog. If software engineers find it easy to copy from others, they promote shareable data. The efficient software engineer can simply use what other analysts had already done. Not only will they make data shareable, their task will be easier. Also, ultimately their clients will likely be happier since this will reduce software development time. It also increases the quality of the information system product.

Electronic data interchange (EDI) data element designers' needs are similar to those of software engineers. They know what types of information trading partners need to share, but they need to describe it as data elements. If it exists in a catalog, they can use it. If it does not exist, they describe a new data element and put it into the catalog.

End users have trouble finding the data that interests them. They often do not know its definition, what it is called, the possible values, what the values mean, et cetera. The catalog can give them the information they need. Of course, the structure of the metadata must allow them to find what they are looking for. That is also true for the other users.

Originally, in its most rudimentary form, we called this catalog a *data dictionary*. More recently it has expanded to become the *data encyclopedia*. The even more comprehensive *data repository* or *information repository* came next. In the form described in this standard, the directory is a *data registry*. The data registry is only a sub-set of the complete metadata that can be included in a data or information repository. However, that metadata sub-set is structured in a way that supports administration and retrieval of registered data. A data registry is definitely more than just another data dictionary.

A data registry facilitates sharing data without requiring that all users obtain this data from a single communal database. Data can be shared among disparate databases and users.

Annex E (informative)

Conceptual and logical data models

A conceptual data model describes how relevant information is structured in the natural world. This has (somewhat inaccurately or cryptically) been called the "model of the business" (it is not always a *business*) or "enterprise model" (the term *enterprise* has several common uses). The conceptual data model provides an excellent place to start modeling data within universe of discourse. It is also the most viable level at which to integrate different data models.

A conceptual data model can be used to develop a more specific logical data model of the identical universe of discourse.

A logical data model describes the same data as structured in an information system. It is often and accurately referred to as a "model of the information system". A logical data model can be directly used for database design. This is the level where most software engineers start. This often hinders the identification of the basic concepts to be represented by the data. It also makes correct integration of data models significantly more difficult.

A conceptual data model is converted to a logical data model with several translations, additions, and decisions. Generally these:

- Add any control and interface objects or entities.
- Eliminate or resolve many-to-many relationships.
- Combine entities with one-to-one relationships.
- Identify key attributes.
- Decide which entities from related entities can become attributes based upon the intended use and importance of the data.
- Specify which entities will inherit foreign keys.
- Specify representation class, datatypes, character count, and other value domain metadata attributes that describe the data elements used to represent the data element concepts described in the conceptual model.
- Convert all special relationships such as subtypes, components, and dependencies into conventional relationships.
- Specify whether each attribute is mandatory, conditional, or optional.

Annex F
(informative)

Bibliography

- ASC X12S/91-1000, *Electronic Data Interchange X12 Standards*, Data Interchange Standards Association. December 1991.
- ANSI X3/TR-14-95. *IRDS Conceptual Schema, Part 1: Conceptual Schema for IRDS, Part 2: Modeling Language Analysis*. 1995
- ANSI X3.138-1988, *Information Resource Dictionary System*
- Bellcore Special Report SR-OPT-001826, *Information Modeling Concepts and Guidelines*, Bell Communications Research, January 1991.
- Bellcore Special Report SR-STS-002515, *Shareable Data Registration*, Bell Communications Research, June 1994.
- Bruce, Thomas. *Designing Quality Databases with IDEF1X Information Models*. Dorset House Publishing: 1992.
- English, Larry P. Accountability to the Rescue. *Database Programming & Design* 6(5): 54-59; 1993 April.
- Flavin, Matt. *Fundamental Concepts of Information Modeling*. Yourdon Press; 1981.
- Fowler, Martin. *UML Distilled*. Addison Wesley Publishing Company: 1997.
- Giordano, Robert & Barbara Von Halle. Database Design—Remember the Standards. *Database Programming & Design* 4(6): 11-13; June 1991.
- Giordano, Robert & Barbara Von Halle. Database Design—The Heart of the Matter. *Database Programming & Design* 4(5) 15-18; May 1991.
- Giordano, Robert. The Data 'Where' House. *Database Programming & Design* 6(9): 54-61; September 1993.
- Halpin, Terry. *Conceptual Schema and Relational Database Design*. Prentice Hall: 1995.
- ISO 1087:1990, *Terminology — Vocabulary*.
- ISO 3166-1:1997, *Codes for the representation of countries and their subdivisions – Part 1: Country codes*.
- ISO DIS 3166-2, *Codes for the representation of countries and their subdivisions – Part 2: Country subdivision codes*.
- ISO DIS 3166-3, *Codes for the representation of countries and their subdivisions – Part 3: Code for formerly used names of countries*.
- ISO 5127-6:1983, *Documentation and Information – Vocabulary — Part 6: Documentary languages*
- ISO/IEC CD 14481,
- ISO TR 9007:1987 *Information processing systems - Concepts and terminology for the conceptual schema and the information base*
- ISO/IEC 10027:1990, *Information technology – Information Resource Dictionary System (IRDS) Framework*

ISO/IEC 10165-1:1993, *Information Technology — Open Systems Interconnection — Structure of management information: Management Information Model*

ISO/IEC 11404:1995, *Information technology — Language-Independent Datatypes*

ISO/IEC TR 9789:1994, *Information technology — Guidelines for the organization and representation of data elements for data interchange — Coding methods and principles*

Marks, Dana M. & Terry Moriarty. Enterprise View—Unraveling the Standards. *Database Programming & Design* 6(13): 68-69; December 1993.

Moriarty, Terry. Enterprise View—Repository and CASE: Harmony?, *Database Programming & Design* 7(4): 68-71; April 1994.

Moriarty, Terry. Enterprise View—The Next Generation of IRDS, *Database Programming & Design* 7(3): 67-69; March 1994.

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.

Tasker, Dan. *Fourth Generation Data: A Guide to Data Analysis for New and Old Systems*. Prentice-Hall; 1989.