

# ISO/IEC JTC 1/SC 32/WG 2 N 00201

Date: 1999-12-13

REPLACES: --

## ISO/IEC JTC 1/SC 32/WG 2

### Metadata

Secretariat: United States of America (ANSI)

Administered by Pacific Northwest National Laboratory on behalf of ANSI

<b>DOCUMENT TYPE</b>	<i>Working Draft</i>
<b>TITLE</b>	<i>INTERNATIONAL STANDARD IS 11179-3</i> <i>Information technology -- Specification and standardization of data elements</i> <i>Part 3:</i> <i>Basic attributes of data elements components <u>or</u></i> <i>Metadata registry structure, administration and content</i>
<b>SOURCE</b>	<i>Joe Christensen, Ray Gates</i>
<b>PROJECT NUMBER</b>	
<b>STATUS</b>	<i>For information and action as appropriate</i>
<b>REFERENCES</b>	
<b>ACTION ID.</b>	<i>FYI</i>
<b>REQUESTED ACTION</b>	<i>Comment</i>
<b>DUE DATE</b>	<i>January 26, 1999, editing meeting</i>
<b>Number of Pages</b>	<i>62</i>
<b>LANGUAGE USED</b>	<i>English</i>
<b>DISTRIBUTION</b>	<i>P &amp; L Members</i> <i>SC Chair</i> <i>WG Conveners and Secretaries</i>

*Douglas Mann, Secretariat, ISO/IEC JTC 1/SC 32*

*Pacific Northwest National Laboratory \*, 901 D Street, SW., Suite 900, Washington, DC, 20024-2115, United States of America*

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

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

**ISO/IEC JTC 1/SC 32/WG 2 N 00201**

Date: 1999/12/13

**/WD**

ISO/TC 1/SC 32/WG 2

Secretariat:

## **INTERNATIONAL STANDARD IS 11179-3**

**Information technology -- Specification and standardization of data elements**

**Part 3:**

**Basic attributes of data elements components or**

**Metadata registry structure, administration and content**

Document type: International Standard

Document subtype: Not applicable

Document stage: (20) Preparatory

Document language: E

H:\Standards\Iso-Iec\Jtc1\JTC1-SC32\SC32-WG2-MetaData\ISO-IEC 11179\11179p3r\WD  
11179-3\Revision of 11179 Pt3 - 13 Dec 1999.doc ISOSTD ISO Template Version 3.0  
1997-02-

## Contents

<i>Foreword</i> .....	v
Introduction .....	vi
1 Scope.....	1
1.1 Basic attributes of administered components.....	1
1.2 Structure of a metadata registry.....	1
2 Conformance .....	2
2.1 Basic attributes of data elements.....	2
2.2 Metamodel for the content of a metadata registry .....	2
3 Normative reference(s).....	3
4 Definition(s) .....	5
5 Attributes of administered components.....	7
5.1 Data elements.....	7
5.2 Data element concepts .....	8
5.3 Other data element components .....	8
5.4 Attributes of administered components.....	8
6 Structure of a Metadata Registry.....	9
6.1 Conceptual data model.....	9
6.2 Purpose .....	10
6.3 Application.....	10
6.4 Key concepts and terminology.....	10
6.5 Macro-view of the metamodel.....	12
6.6 Extensibility .....	14
6.7 Description of metamodel.....	14
6.8 Administration .....	15
6.9 Administration metamodel region components .....	17
6.10 Administration metamodel region attributes .....	17
6.11 Naming and Identification .....	19
6.12 Naming and identification metamodel region components .....	19
6.13 Naming and identification metamodel region attributes .....	20
6.14 Classification .....	20
6.15 Classification metamodel region components .....	23
6.16 Classification metamodel region attributes .....	23
6.17 Data element concept administration .....	23
6.18 Object Class .....	25
6.19 Property .....	25
6.20 Data element concept .....	25
6.21 Data element concept metamodel administration region attributes .....	26
6.22 Conceptual and value domain administration .....	26

6.23	Conceptual domain .....	28
6.24	Value domain .....	28
6.25	Conceptual and value domain administration metamodel region attributes.....	29
6.26	Data element administration .....	29
6.27	Data element administration metamodel region components .....	32
6.28	Data element administration metamodel region attributes .....	32
7	Metamodel specification .....	33
7.1	Metamodel diagram.....	33
7.2	Modeling object description specifications .....	34
7.2.1	Metadata entity and relationship field description specification .....	35
7.2.2	Metadata attribute field description specification .....	35
7.3	Alphabetical list of metadata objects.....	35
Annex A	(Was Annex E) (Informative) Modelling Notation.....	37
A.1	Modelling symbols .....	37
A.1.1	Classes.....	37
A.1.2	Associations .....	38
A.1.3	Associative entity.....	39
A.1.4	Subtypes .....	39
A.1.5	Aggregation .....	40
A.1.6	Composite Aggregation .....	41
A.1.7	Attribute .....	41
Annex B	(Was Annex F) (informative) Alternative modeling representations.....	43
B.1	IDEF1X representation model .....	43
B.2	Object Role Modeling (ORM) – Natural language Information Analysis Method (NIAM) representation model.....	44
<del>Annex C</del>	<del>(Was Annex A) (Informative) Additional attributes (for deletion).....</del>	<del>46</del>
Annex D	(Was Annex B) (Informative) (Removed to Content Technical report) .....	49
Annex E	(Was Annex C) (Informative) (Removed to Content Technical report).....	50
Annex F	(Was Annex D) (Informative) Recommendations for the use of additional registration descriptors of attributes (for deletion) .....	51
Annex G	(Informative) Principles of managing shared data (Removed to Content Technical Report) .....	54
Annex H	(Informative) Metadata registry uses and users (Removed to Content Technical Report) .....	55
Annex I	(Informative) Conceptual and logical data models (Removed to Content Technical Report) .....	56
Annex J	(Informative) Bibliographic references .....	57

## Table of Figures

Figure 1: Metadata registry basic structure.....	13
Figure 2: Metamodel regions .....	15
Figure 3: Administration metamodel region .....	17
Figure 4: Naming and identification metamodel region .....	19
Figure 5: Classification metamodel region.....	22
Figure 6: Data element concept administration metamodel region .....	24
Figure 7: Conceptual and value domain administration metamodel region.....	27
Figure 8: Data element administration metamodel region .....	31
Figure 9: Conceptual metamodel entity-relationship diagram .....	34
Figure A-1: Sample modelling diagram.....	37
Figure A-2: Class modelling representation.....	38
Figure A- 3: Association modelling representation .....	38
Figure A- 4: Class relationship modelling representation .....	38
Figure A- 5: Class relationship cardinality modelling representation.....	39
Figure A- 6: Associative class modelling representation .....	39
Figure A- 7: Subtype modelling representation.....	40
Figure A- 8: Aggregation modelling representation .....	40
Figure A- 9: Composite Aggregation modelling representation .....	41
Figure A- 10: Class-attribute modelling representation .....	42
Figure B- 1: ORM Administration region .....	44
Figure B- 2: ORM naming and identification region.....	44
Figure B- 3: ORM classification region .....	44

Figure B- 4: ORM data element concept administration region .....	44
Figure B- 5: ORM conceptual domain and value domain administration region.....	45
Figure B- 6: ORM data element administration region.....	45
Figure B- 7: ORM type hierarchy .....	45

## **Foreword**

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

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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

## Introduction

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

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

A prerequisite for correct and proper use and interpretation of data is that both users and owners of data have a common understanding of the meaning and representation of the source data elements. To facilitate this shared view, a number of characteristics, or attributes, have to be defined. A set of attributes required for effective interchange of metadata is described in Clauses 5 to 7.

This Part of the standard also addresses the broader administration and retrieval of data by specifying the structure of a metadata registry (Clauses 8 and 9). A metadata registry is a place to keep facts about characteristics of data that are necessary to clearly describe, record, analyse, and classify data. Clauses 8 and 9 provide a structure for the recording of metadata that supports the administration and retrieval of registered data including the attribution, classification, definition, naming, identification and registration of data elements described elsewhere in this standard. This assists users of shared data to have a common understanding of a unit of data's meaning, representation, and identification. A metadata registry supports data sharing with cross-system and cross-organization descriptions of common units of data.

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

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

## 1 Scope

### 1.1 Basic attributes of administered components

This section of the International Standard applies to activities including:

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

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

This standard does not assume nor endorse any specific system environment, database management system, database design paradigm, system development methodology, data definition language, command language, system interface, user interface, computing platform, or any technology required for implementation. The standard does not directly apply to the actual use of data in communications and information processing systems.

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

### 1.2 Structure of a metadata registry

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

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

- a) the description of attributes in Clauses 5 to 7 of Part 3 of the standard
- b) rules and guidelines for classifying data elements in Part 2
- c) rules and guidelines for the formulation of definitions in Part 4

- d) naming and identifying principles for data elements in Part 5 and
- e) rules and guidelines for registering data elements in Part 6 of the International Standard.

## 2 Conformance

### 2.1 Basic attributes of data elements

Clause 5 of this standard specifies a set of basic attributes required for a metadata registry to be considered conformant to ISO/IEC 11179. These basic attributes are specified for (a) registration of data elements and (b) registration of data element concepts.

These attributes constitute what is required for effective metadata interchange. This is only a statement of what is “required” for ISO/IEC 11179 Part 3 conformance and does not preclude individual Registration Authorities from adopting additional attributes as “required” for their implementation.

For a metadata registry to be considered to be conformant to ISO/IEC 11179 it must be able to provide content for the “required” attributes.

Nothing in the conformance statements is intended to require the redundant storage of metadata. This is stated at the “conceptual” level and it is expected to be implemented efficiently (I don't understand this sentence, and I doubt that many readers will. The phrase 'it is expected to be implemented efficiently' could be read as patronising. My suggestion is to delete it, unless it can be put in plainer, more positive language).

### 2.2 Metamodel for the content of a metadata registry

This document specifies the minimum features of a metamodel for the content of a metadata registry. A conformant implementation shall fully support the structure and content depicted in figure 9 – Conceptual metamodel entity-relationship diagram and further detailed in section 6. 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.

Conformant applications may use a specific logical data model to describe the data specified according to this Part of the Standard, as structured in an information system such as a metadata registry system, Information Resource Dictionary System (IRDS), any other information repository, or database.

This standard does not assume nor endorse any specific system environment, database management system, database design paradigm, system development methodology, data definition language, command language, system interface, user interface, computing platform, or any technology required for implementation.

A metamodel is like any other data model except it describes the structure of data that is about data. Since this standard is specified as a conceptual data model, there can be no exact mapping to an implementation of a metadata registry. A conceptual data model describes how relevant information is structured in the natural world. In other words, it is how the human mind is accustomed to thinking of the information. As a conceptual data model, there is no one-to-one match between attributes and fields,

columns, objects, et cetera in a database. There may be more than one field per attribute and some entities and relationships may be implemented as fields. There is no intent that an implementation should have a table for each relationship or entity.

It is assumed that an implementor will use this conceptual data model to develop a more specific logical data model of the identical sphere of interest. A logical data model describes the same data, but as structured in an information system. It is often referred to as a Model of the Information System. A logical data model can be directly used for database design. Annex E (informative) **(Note: This has been moved to the Content Technical Report)** describes some of the normal translations, additions, and decision when converting a conceptual data model to a logical data model.

A conformant metadata 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.

### 3 Normative reference(s)

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

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

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

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

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

ISO 704:1987, Principles and methods of terminology

ISO 2382-4:1987, Information processing systems - Vocabulary

ISO Standards Handbook 10, Data Processing - Vocabulary, 1982

ISO 1087:1990, Terminology - Vocabulary

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

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

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

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

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

ISO/IEC 11179-6 :1997, Information technology – Specification and Standardization of data elements  
Part 6 : Registration of data elements  
ISO 11404: 1995, Information technology – Language-Independent Datatypes

ISO/IEC FDTR 15452, Information technology – Specification of data value domains

## 4 Definition(s)

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

- 4.1 administered component:** A component for which administrative information is recorded. It may be 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.
- 4.1 attribute:** A characteristic of an object or entity.
- 4.2 attribute value:** A representation of an instance of an attribute.
- 4.3 basic attribute:** An attribute of a data element frequently needed to specify a data element
- 4.4 concept:** A unit of thought constituted through abstraction on the basis of characteristics common to a set of objects.
- 4.5 conceptual data model:** A data model that describes how relevant information is structured in the natural world. It is how the human mind is accustomed to thinking of the information.
- 4.6 data:** A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means.
- 4.7 data element concept:** A concept that can be represented in the form of a data element, described independently of any particular representation.
- 4.8 data element:** A unit of data that in a certain context is considered indivisible.
- 4.9 data element dictionary:** An information resource that specifies, defines and lists a set of data elements relevant in a specific field of interest.
- 4.10 data model:** A description of the organization of data in a manner that reflects an information structure.
- 4.11 metadata registry.** A system for managing structured metadata describing the semantic content of sharable data and metadata.
- 4.12 entity:** Any concrete or abstract thing of interest, including associations among things.
- 4.13 identifier.** A sequence of characters arbitrary devised to refer to a particular item.
- 4.14 metadata:** Data that defines and describes other data.
- 4.15 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.
- 4.16 permissible value:** An expression of a value meaning in a specific value domain. A value.

- 4.17 **property:** A characteristic common to all members of an object class.
- 4.18 **registration authority:** A body responsible for maintaining a register of data elements and related components.
- 4.18 **relationship:** A mental link between two or more concepts.
- 4.19 **schema:** A formalism for representing knowledge about a simple concept, an entity, or a class of objects by means of its possible values.
- 4.20 **shareable data:** Data that has precise identifiers, meaning, structures, and values.
- 4.21 **value domain:** A set of permissible values. It provides representation, but has no implication as to what data element concept the values are associated with nor what the values mean.

Definitions of all components of the metamodel are set out in Clause 7.

## 5 Attributes of administered components

Administered components need to be registered and controlled in a standard way in order to ensure understandability and to achieve consistency in the exchange of metadata.

This clause outlines a structure for describing those properties of data element components that are required to meet the above objectives.

There are many properties to choose from, yet some attributes seem obvious. Clearly, it is difficult to imagine a data element component without an identifier, a name or a definition. But these alone seem insufficient to effectively convey the semantic meaning of data element components nor are they adequate for effective data interchange. However, the formulation of an authoritative set of descriptors can be usefully guided by a structure that encapsulates a set of general properties that apply consistently to all data element components. These properties consist of:

### ***Identifying***

Attributes that provide the descriptive information for an administered component required for effective access, registration and control.

### ***Definitional***

Attributes required to describe the semantic aspects of a data element to ensure understandability.

These attributes may be derived by inheritance from characteristics of data element concepts, objects or entities.

### ***Relational***

Attributes required to describe associations among administered components and/or associations between administered components

### ***Representational***

Attributes required to describe the representational aspects of an administered component.

### ***Administrative***

Attributes required to describe management and control aspects of a data element.

A set of attributes that constitute what is required for effective metadata registration and interchange follows in Tables 5.1 to 5.3. Obviously, what is "required" will vary from one metadata registry implementation to another. The following tables are only a statement of what is "required" for ISO/IEC 11179 Part 3 conformance and does not preclude individual Registration Authorities from adopting additional attributes as "required" for their implementation.

### **5.1 Data elements**

(Insert table - names only of the attributes/descriptors).

### **5.2 Data element concepts**

(Insert table- names only of the attributes/descriptors).

### **5.3 Other data element components**

Other data element components such as classification schemes, objects, and entities are not specified in this standard. However, each of these types of component require a minimum set of identifying, definitional, relational, representational, and administrative attributes.

For a metadata registry to be considered to be conformant to ISO/IEC 11179 it must be able to provide content for the “required” attributes.

Definitions of each of these attributes are set out in Table 5.4

### **5.4 Attributes of administered components**

(Insert table).

## 6 Structure of a Metadata Registry

### 6.1 Conceptual data model

This section of the standard specifies the structure of a metadata registry. The structure is stated in the form of a conceptual data model referred to in the text as the 'metamodel for the management of shareable data'.

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. 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.

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 metadata registry.

A metadata registry is a place to keep facts about characteristics of data that are necessary to clearly describe, inventory, analyze, and classify data. A metadata registry supports data sharing with cross-system and cross-organization descriptions of common data. A metadata registry assists users of shared data to have a common understanding of a unit of data's meaning, representation, and identification. 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 metadata registry assists users of shared data to have a common understanding of a unit of data's meaning, representation, and identification. Just as a metadata 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 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 8.4. However, to understand more fully the principles underlying this metamodel, it would be beneficial to first examine annex G – *Principles of managing shared data*. It may also be useful to read annex H – *Metadata registry uses and users* describing many of the uses and users of a metadata registry. Clause 8.5 provides a simplified overview of the structure of a metadata registry. Clause 8.7 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 E. Clause 9 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 E to J of this International Standard are for information only.

## 6.2 Purpose

International Standard ISO/IEC 11179, parts one to six, *Information technology — Specification and standardization of data elements*, developed by ISO/IEC JTC1/SC 14 Data Element Principles, describes the attribution, classification, definition, naming, identification, and registration of data elements. This Section of the standard provides a comprehensive structure for the components of data elements, including basic data element attributes, and in addition, shows the relationships of the components to each other.

## 6.3 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.

## 6.4 Key concepts and terminology

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 Part 3 of the six-part ISO/IEC 11179 standard that describes the specification and standardization of data elements and the organization of a metadata 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 metadata registry is a structured system, set of files or repository (paper, electronic, or a combination) containing the administered components of data elements and any associated information. It is a place to keep characteristics of data that are necessary to clearly describe, inventory, analyze, and classify data. A metadata registry supports data sharing with cross-system and cross-organization descriptions of common units of data. A metadata 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 metadata 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 "metadata 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. It is a unit of data that in a certain context is considered indivisible. A data element dictionary is an information resource that specifies, defines, and lists all relevant data elements. 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 metadata 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 metadata registry contains the metadata that is necessary to clearly describe, inventory, analyze, and classify data. Metadata 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 metadata registry gives users of shared data a common understanding of a unit of data's meaning, representation, and identification. For this purpose, a metadata registry need not be a database containing all the metadata found in a repository. A metadata 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 metadata registries.

### 6.5 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 TR9007:1987 Information processing systems - Concepts and terminology for the conceptual schema and the information base

A metadata 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 metadata 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 metadata registry.

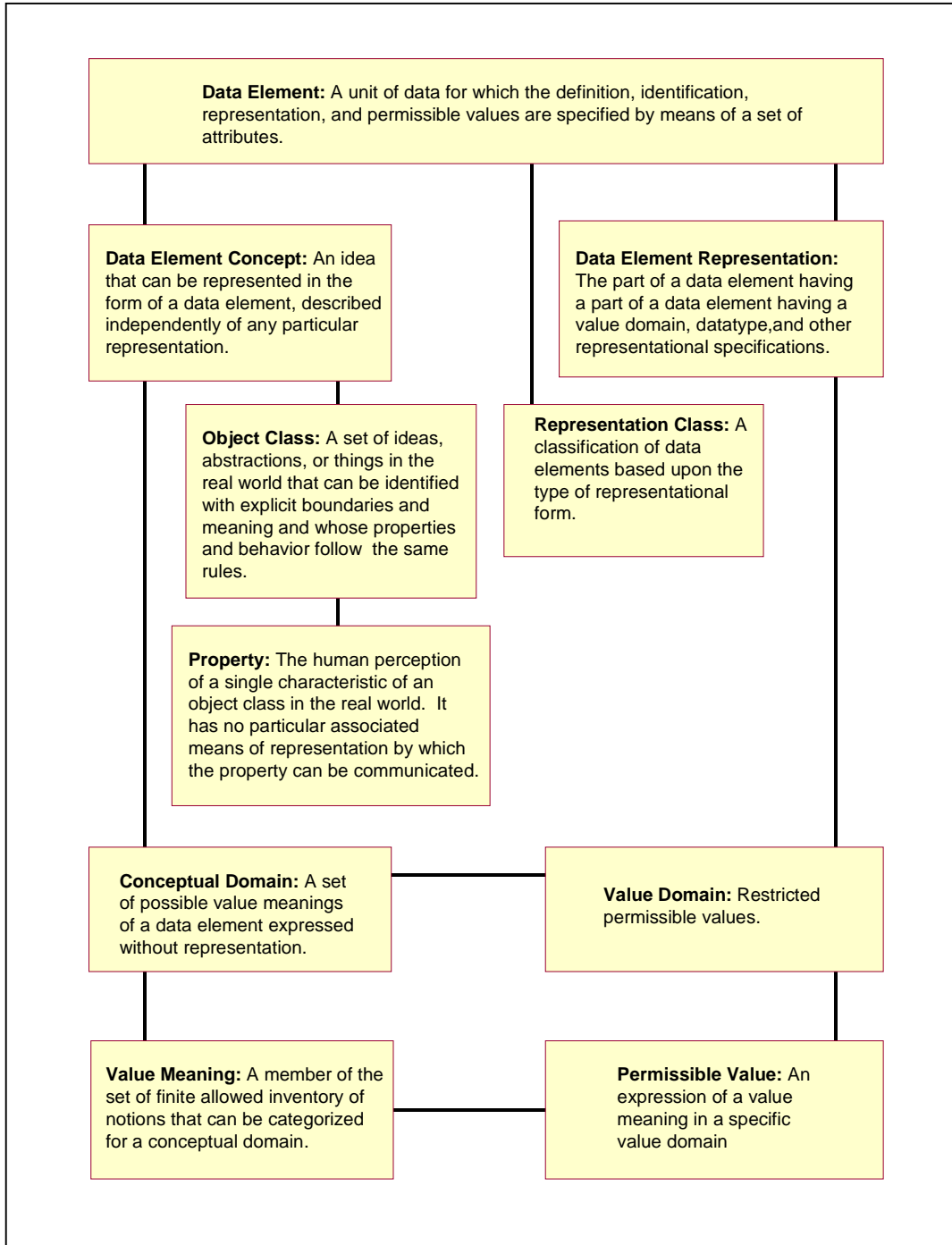


Figure 1: Metadata registry basic structure

## **6.6 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.

## **6.7 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 Administration region that supports the administrative components in a registry. It is from this region that the other regions extend. The other five regions are Naming and Identification, Classification, 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 Administration, naming, and classification support regions.

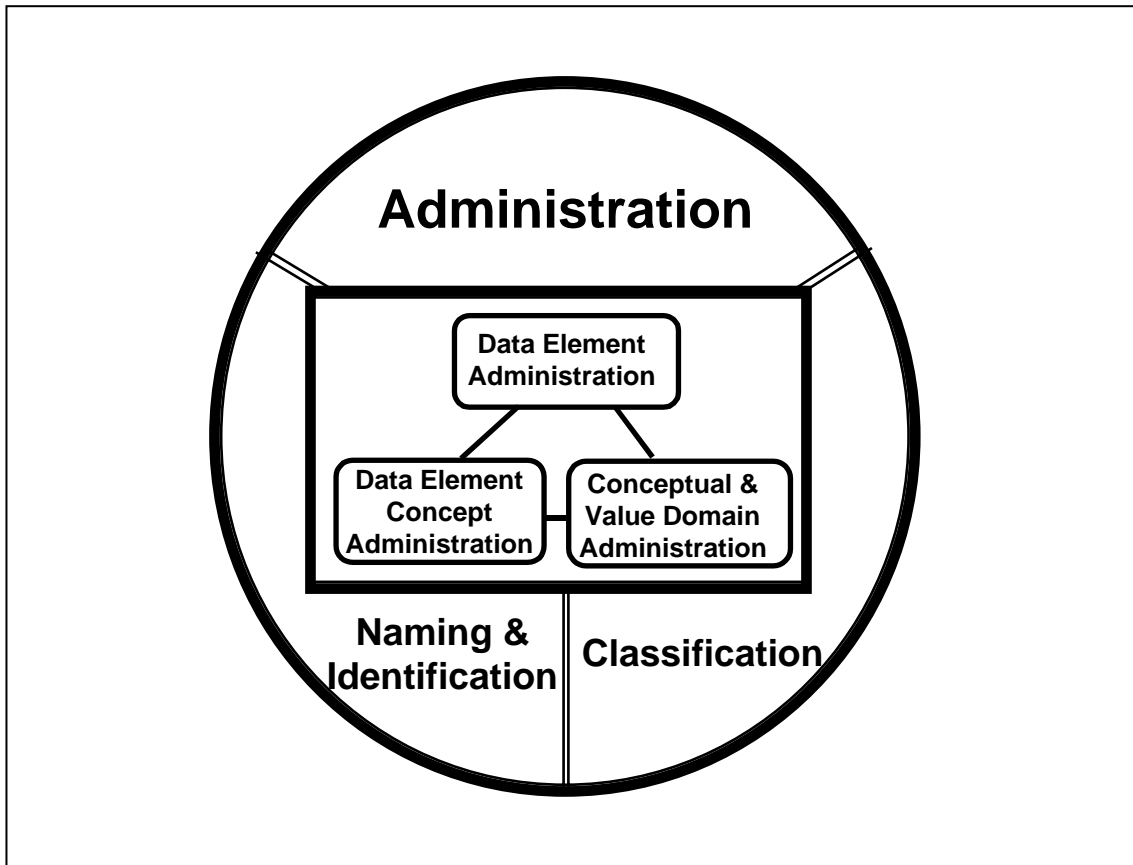


Figure 2: Metamodel regions

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 section 8. Annex ? contains an explanation of the symbols used in the following model diagrams. Section 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.

### 6.8 Administration

The primary region is the Administration region that supports the administrative aspects of components in a registry. It is from this region that the other regions extend. This region addresses the *organizations* that are responsible for

and/or have submitted components to the registry. The supporting documentation is also addressed in this region. Figure 3 represents the Administration region.

Note: This figure is being revised to make consistent with the revised model in document WG2 N00204.

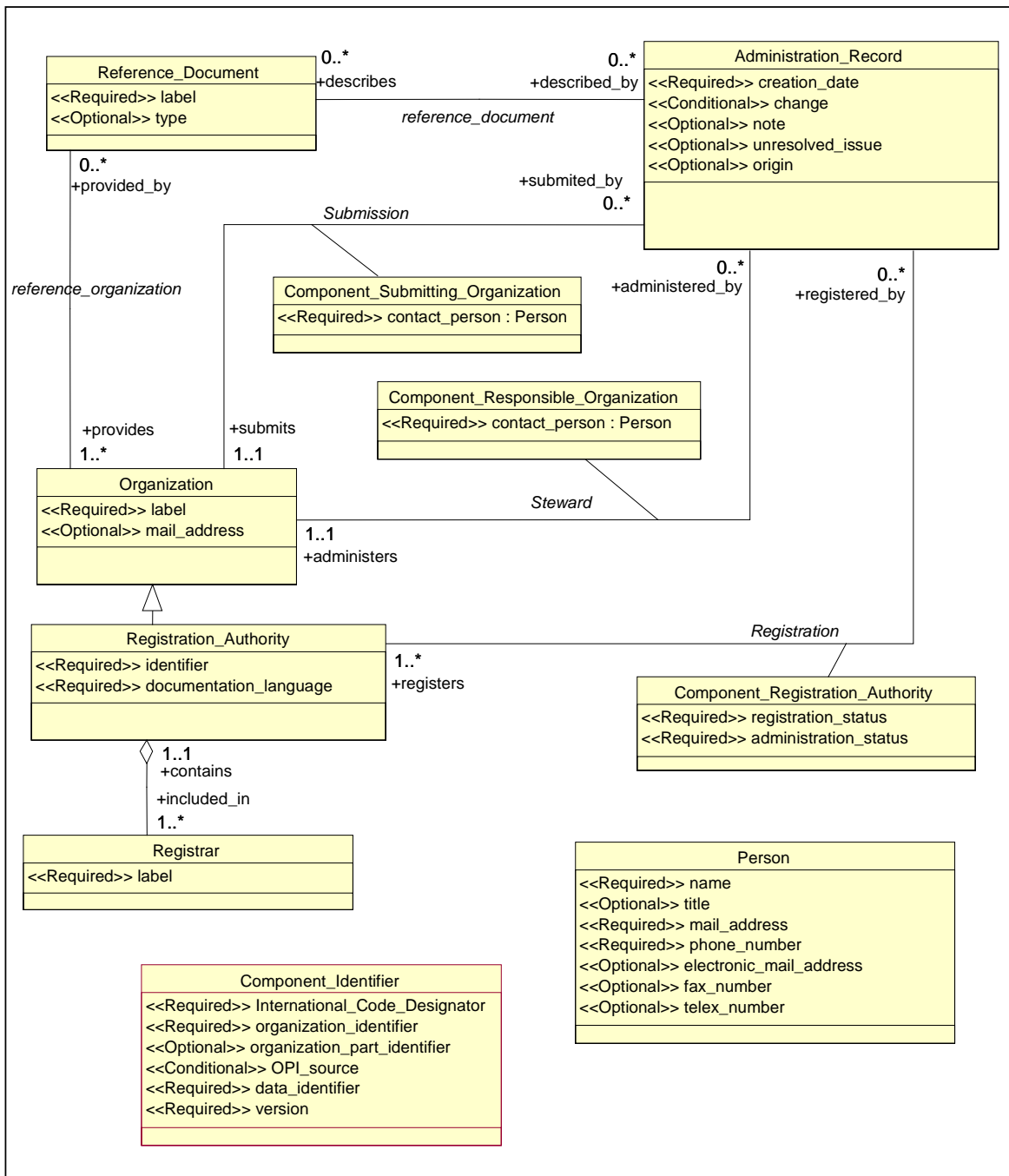


Figure 3: Administration metamodel region

## 6.9 Administration metamodel region components

An *administered component* is a component for which administrative information is recorded. It may be 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 any organization authorized to register metadata. A *registration authority* is a subtype of organization and inherits all of its attributes and relationships. An *administered component* has a *registration authority* that is its owner.

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

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

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

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

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

## 6.10 Administration metamodel region attributes

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

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

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

### 6.11 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.

Note: This figure is being revised to make consistent with the revised model in document WG2 N00204.

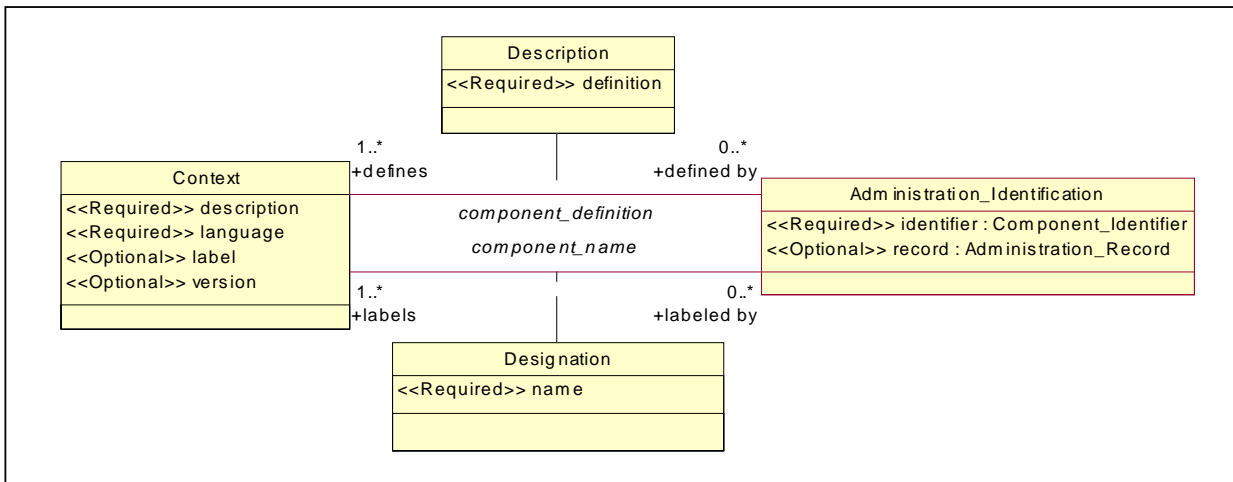


Figure 4: Naming and identification metamodel region

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

### 6.12 Naming and identification metamodel region components

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

### 6.13 Naming and identification metamodel region attributes

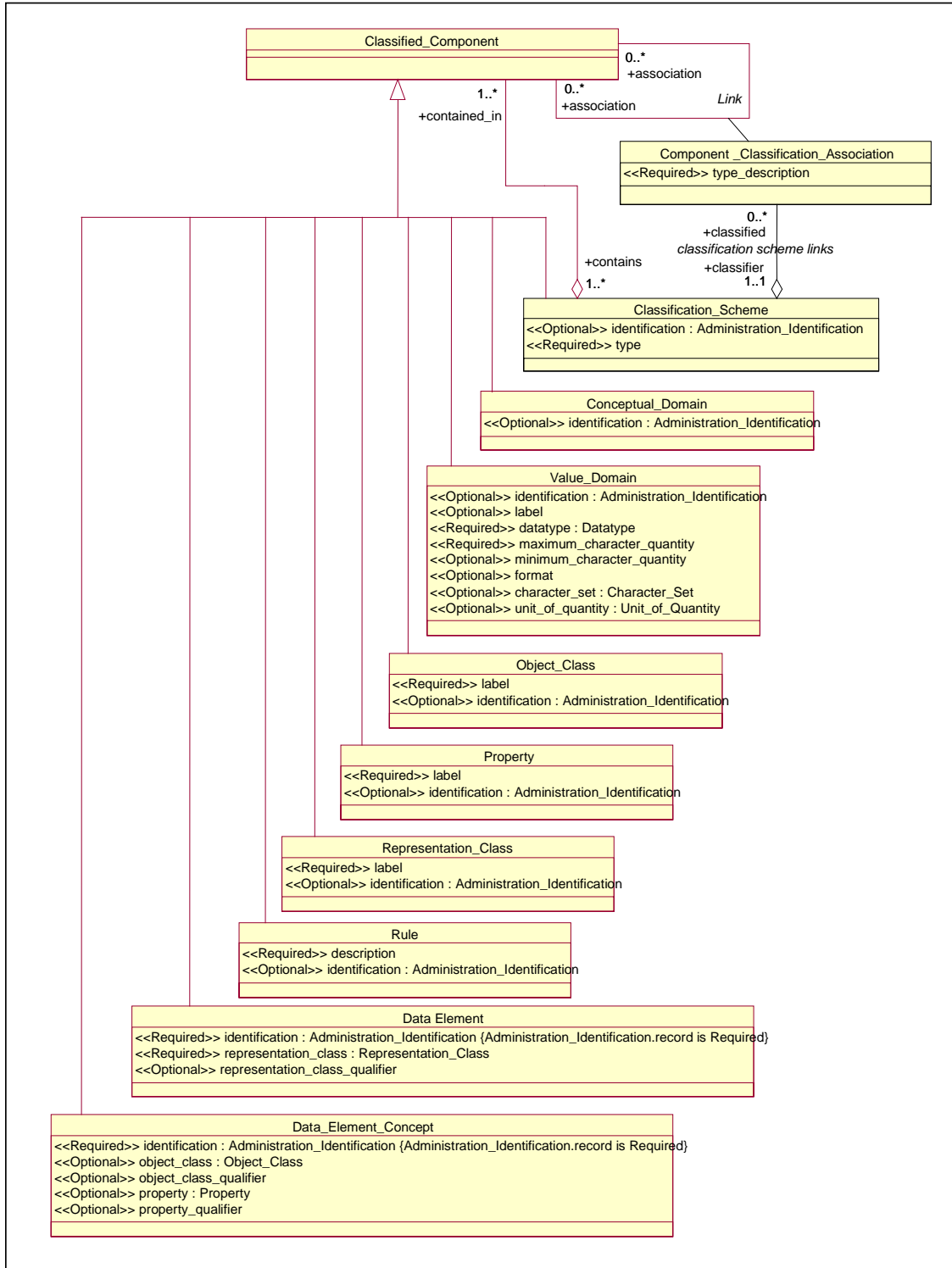
The attributes in the Naming and Identification metamodel region shall be as follows:

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

### 6.14 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.

Note: This figure is being revised to make consistent with the revised model in document WG2 N00204.



*Figure 5: Classification metamodel region*

### 6.15 Classification metamodel region components

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

A *classified component* is a generalization for an *object class* (*concept* or *concept application association*), *property class*, *representation class*, *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 they are each from a different *context*. Conversely, a *classified component* may exist in this structure without being associated with any other *classified component*. In essence, having a *classification scheme* is optional.

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

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

### 6.16 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*.

### 6.17 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.

Note: This figure is being revised to make consistent with the revised model in document WG2 N00204.

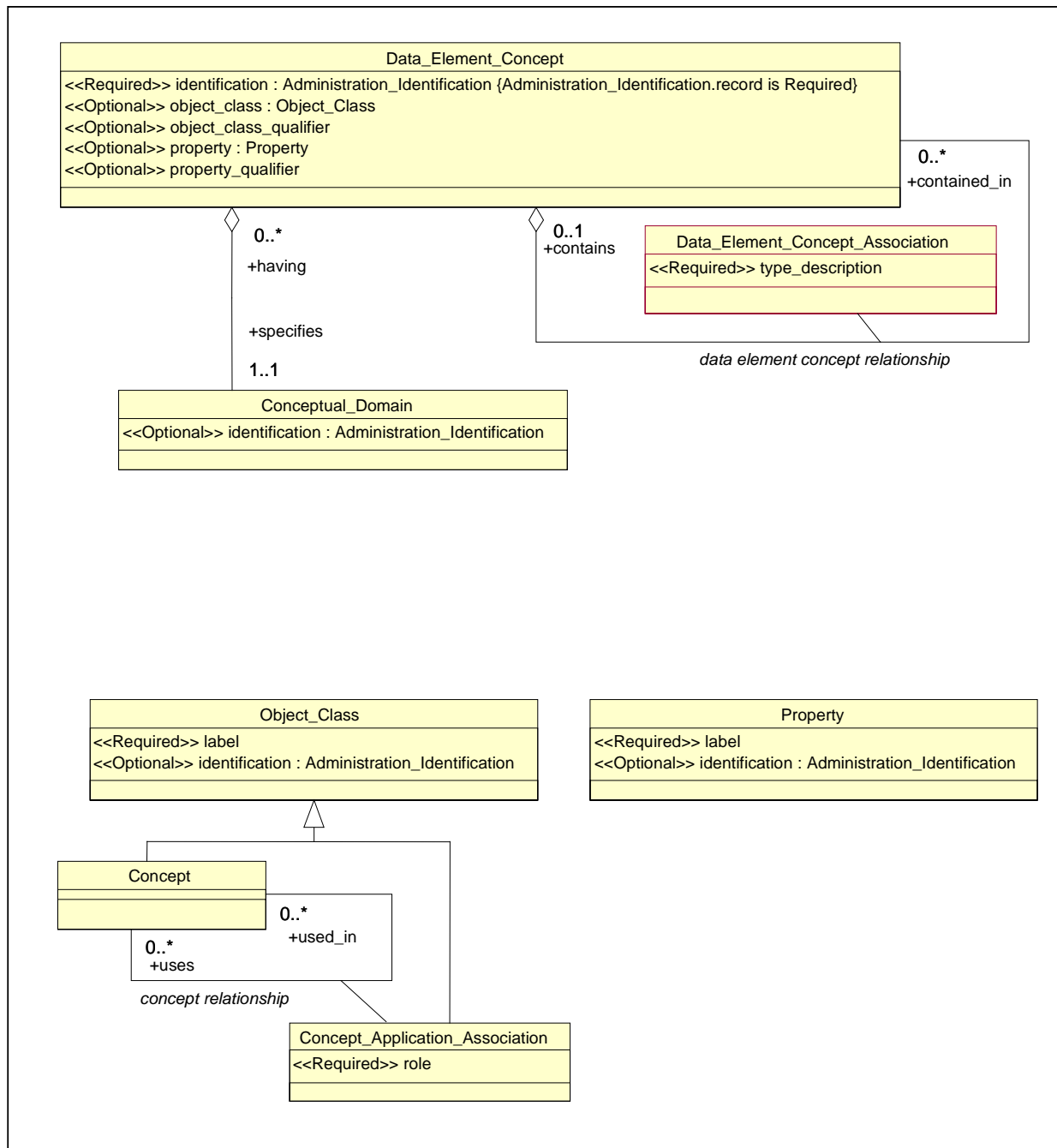


Figure 6: Data element concept administration metamodel region

### 6.18 Object Class

An *object class* is a set of ideas, abstractions, or things in the real world that can be identified with explicit boundaries and meaning and whose properties and behavior follow the same rules. It may be either a single or a group of associated concepts, abstractions, or things. An *object class* may be a single unit of thought (i.e., *concept*) or a set of *concepts* in a relationship with each other to form a more complex concept (i.e., *concept application association*).

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

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

### 6.19 Property

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

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

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

### 6.20 Data element concept

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

### 6.21 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*.

### 6.22 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.

Note: This figure is being revised to make consistent with the revised model in document WG2 N00204.

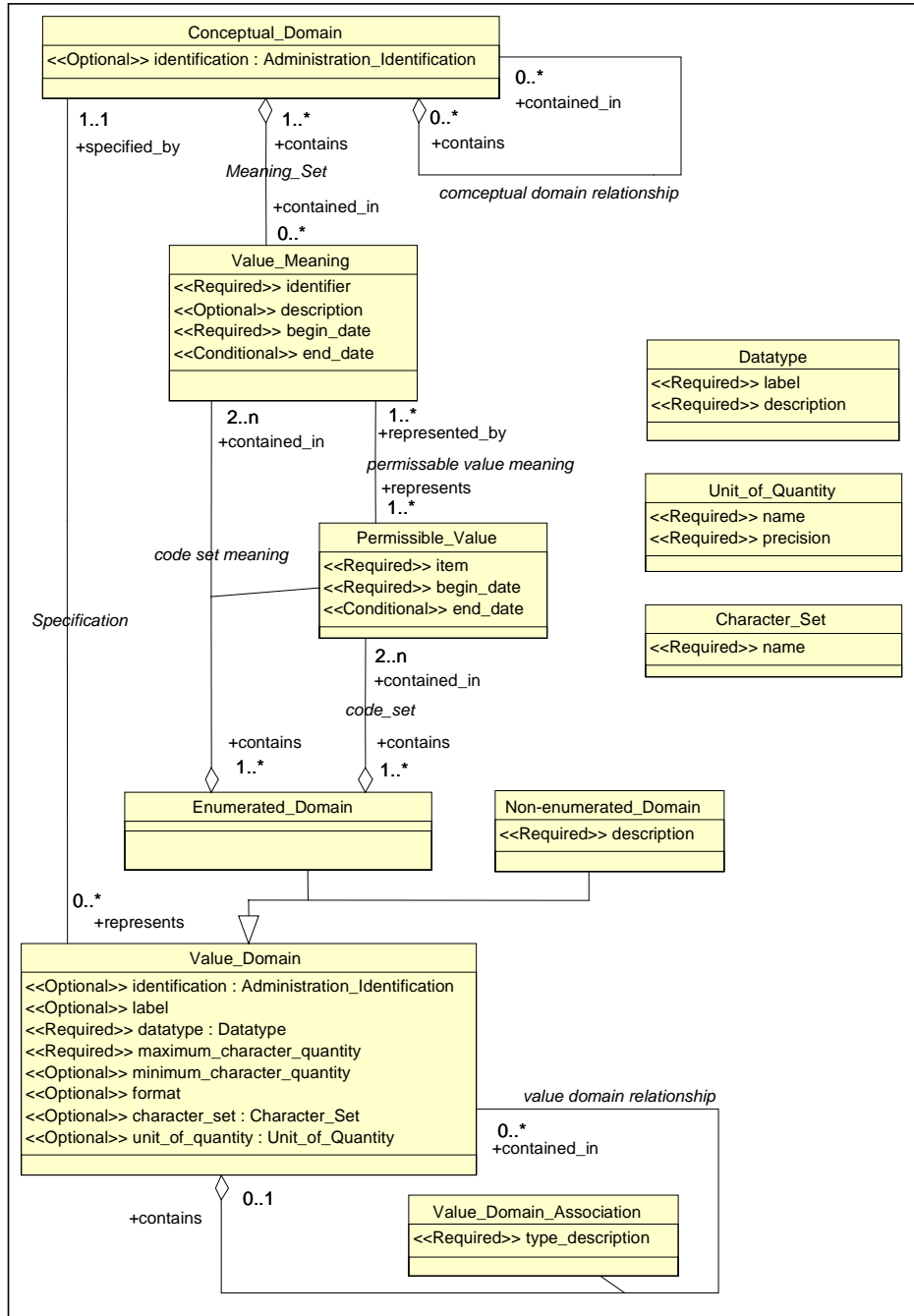


Figure 7: Conceptual and value domain administration metamodel region

### 6.23 Conceptual domain

The *conceptual domain* is a set of value meanings of a data element concept, expressed without 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*.

### 6.24 Value domain

One of the key components of a representation is the *value domain*. A *value domain* provides representation, but has no implication as to what *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 standard 3166, Codes for the representation of names of countries. For instance, ISO 3166 describes the set of by five value domains: short name in English, official name in English, alpha-2 code, alpha-3 code, and numeric code.

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

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

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

A *value domain* is associated with a *datatype* — a set of distinct values, characterized by properties of those values and by operations on those values, for example the category used for the collection of letters, digits, and/or symbols to depict values of a *data element* determined by 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 a one of the representational forms of quantity.

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

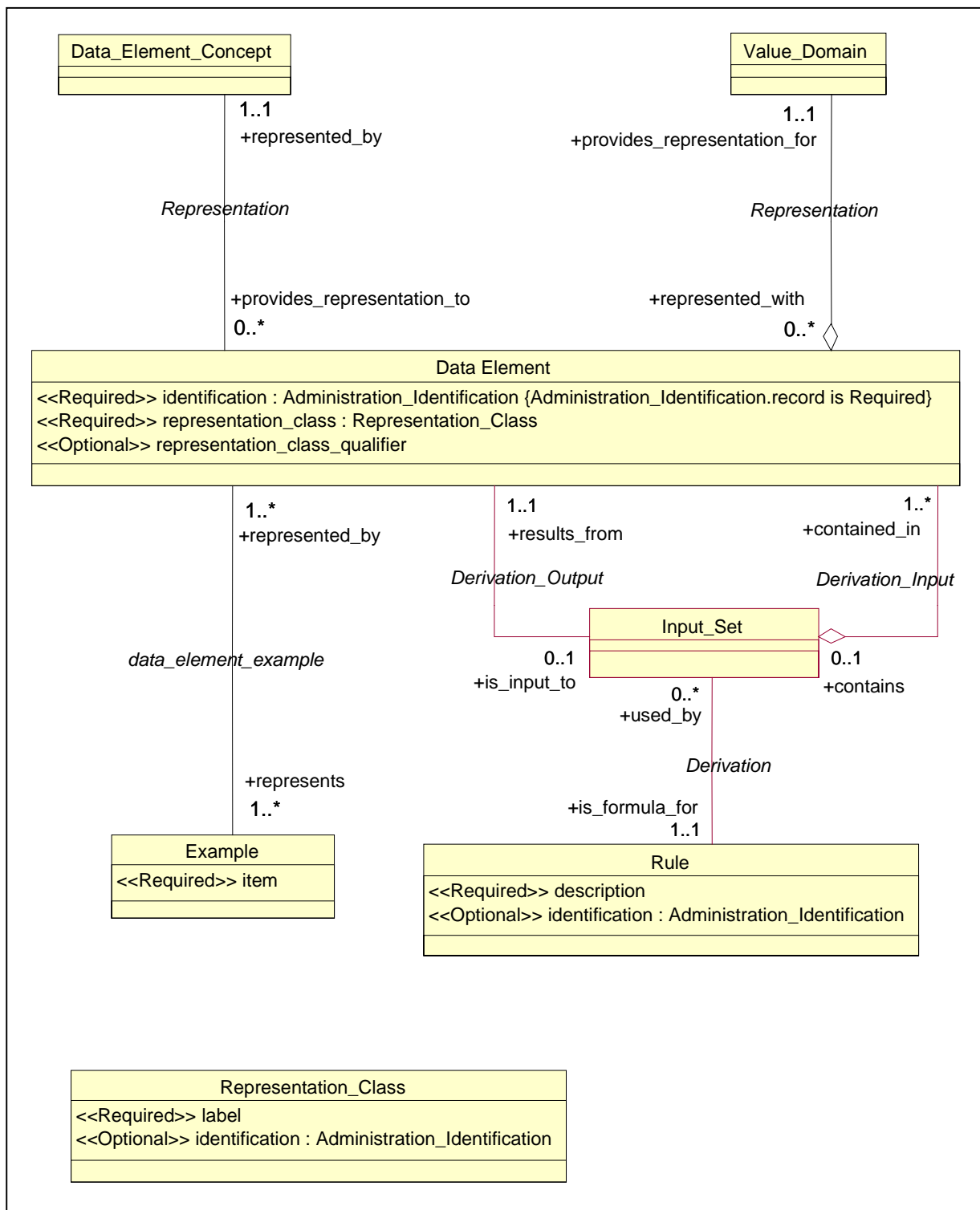
The attributes in the Conceptual and Value Domain Administration metamodel region shall be as follows:

<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>

### 6.26 Data element administration

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

Note: This figure is being revised to make consistent with the revised model in document WG2 N00204.



*Figure 8: Data element administration metamodel region*

## 6.27 Data element administration metamodel region components

A *data element* is a unit of data that is considered in context to be indivisible. It is a unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes. In the simplest way of thinking about it, a *data element* links a *value domain* with a meaning. From a different and more complete prospective, *data elements* are formed when a *data element concept* is assigned a representation. One of the key components of a representation is the *value domain*, i.e., restricted 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. The association between a *data element concept*, a *representation class*, and a relevant *value domain* is a *data element*. (In general usage, the term *data element* and *data element type* are used interchangeably. In this standard, the shorter term *data element* is used.) The *data element concept* may be associated with several *value domains* resulting in a different *data element* for each association. A *value domain* may not reside in this structure without being the representation of an identified *data element concept*. Each *value domain* must be assigned to exactly one *representation class*. A *representation class* may reside in this structure without being associated with either 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 metadata 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 (derivation being defined as a relationship between a rule and an input set upon which it acts). *Rules* are not limited to arithmetic and logical operations. A derivation

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

## 6.28 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>

## **7 Metamodel specification**

The complete metamodel for the management of shareable data is shown in figure 9. Paragraph 7.1 is the complete diagram for the metamodel. Paragraph 7.2 contains the specifications that for describing the entities, relationships, and attributes that comprises the metamodel. Paragraph 7.3 contains an alphabetical list of specifications of the metamodel objects.

### **7.1 Metamodel diagram**

Note: This figure is being revised to make consistent with the revised model in document WG2 N00204.

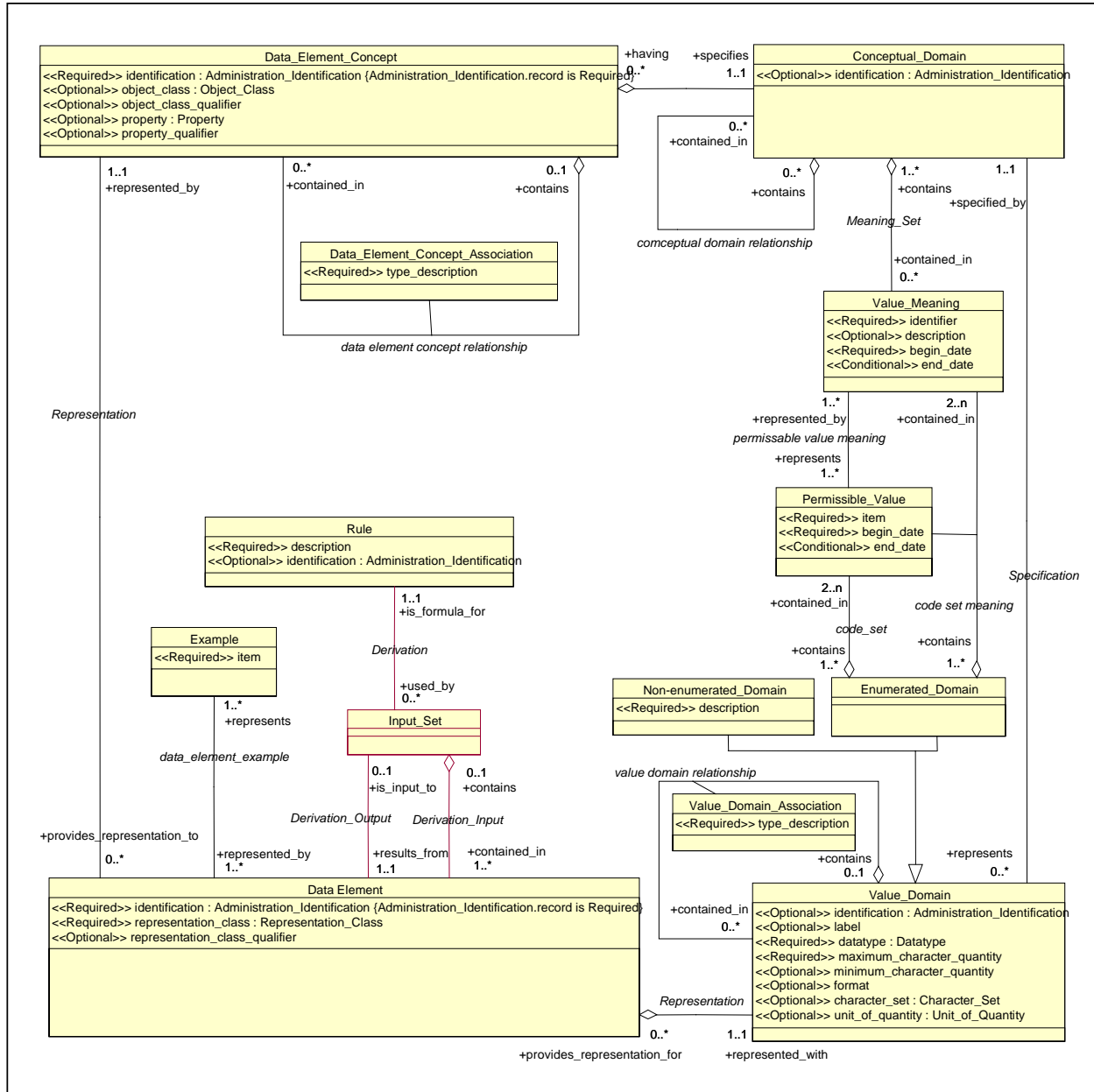


Figure 9: Conceptual metamodel entity-relationship diagram

## 7.2 Modeling object description specifications

The Table in 7.2.1 provides detailed descriptions of each modeling object in the entity-relationship diagram in figure 9.

### 7.2.1 Metadata entity and relationship field description specification

The following structure is used to describe the *entities and relationships* of the metamodel.

---

<b>Modelling Name:</b>	The single name that identifies this entity or relationship in the data dictionary.	
<b>Name(s):</b>	The context and the name by which the entity or relationship is known.	
	<b><u>Context</u></b>	<b><u>Name</u></b>
	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.	The name by which the entity or relationship is known within a specific context.
<b>Type:</b>	An entity or relationship may be an: <ul style="list-style-type: none"> <li>• Entity — persons, places, concepts, events, or other fundamental things.</li> <li>• Relationship — associations among two or more entities.</li> <li>• Associative entity — a relationship that acts like an entity.</li> </ul>	
<b>Definition:</b>	Concise definition or description of the intelligence conveyed by the entity or relationship.	
<b>Notes/Comments:</b>	General notes or comments not recorded in other fields.	

### 7.2.2 Metadata attribute field description specification

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

---

<b>Modelling Name:</b>	The single structured name that identifies this attribute in the data dictionary.	
<b>Name(s):</b>	The context and the name by which the attribute is known.	
	<b><u>Context</u></b>	<b><u>Name</u></b>
	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.	The name by which the attribute is known within a specific context.
<b>Type:</b>	"attribute" identifies this as an attribute.	
<b>Definition:</b>	Concise definition or description of the intelligence conveyed by the attribute.	
<b>Property Obligation:</b>	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.	
<b>Notes/Comments:</b>	General notes or comments not recorded in other fields.	

### 7.3 Alphabetical list of metadata objects

The content for this subclause will be imported from document WG 2 N 00203 (or documents superseding it) when the revisions suggested in N 00203 have been accepted.



## Annex A (Was Annex E) (Informative) Modelling Notation

### A.1 Modelling symbols

The Unified Modeling Language (UML) is used to describe this metamodel. This notation is particularly suited to documenting a conceptual data schema. The structure used in the description (the 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 object model provides for four basic types of modeling objects: *classes* (shown as rectangles in a diagram), *associations* among these classes (shown as lines), *operations*, and *attributes* that are associated to a class -. For example, the attributes describing employees and the cars they own could be modeled as follows:

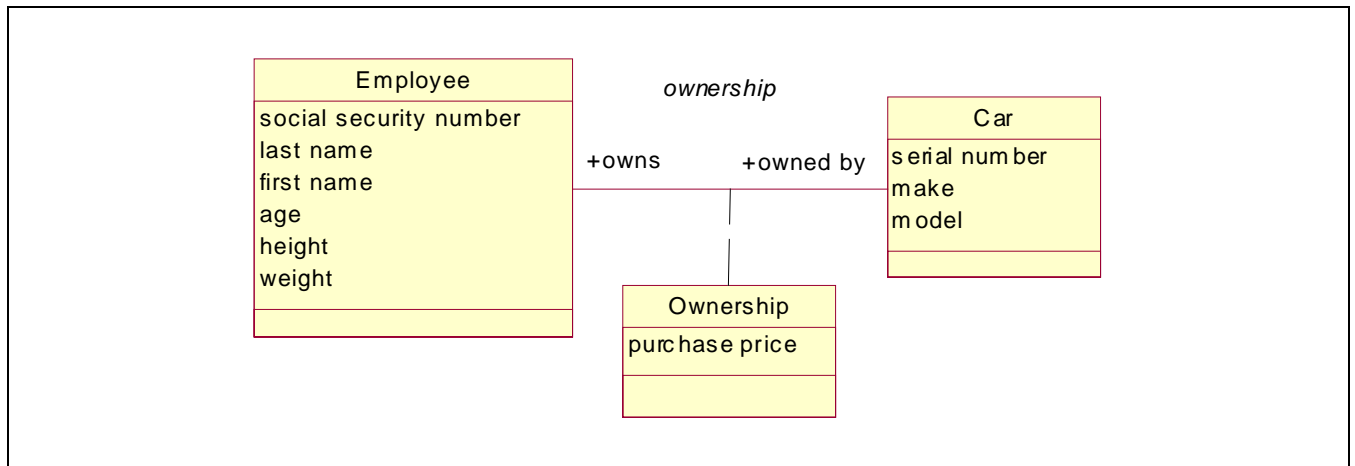


Figure A-1: Sample modelling diagram

At the current time operations are not shown.

#### A.1.1 Classes

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

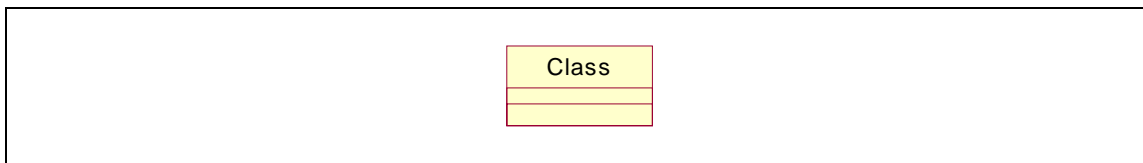


Figure A-2: Class modelling representation

As described below, entities may have attributes.

### A.1.2 Associations

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

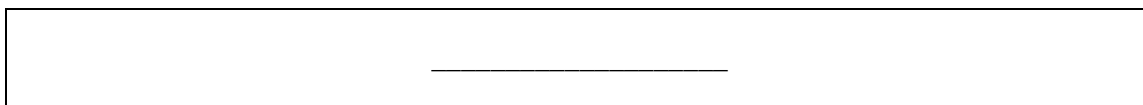


Figure A- 3: Association modelling representation

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

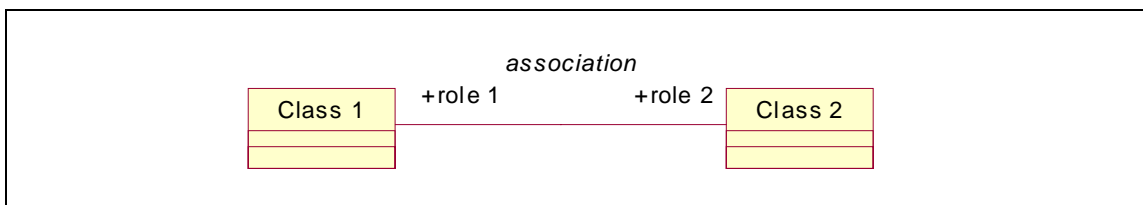


Figure A- 4: Class relationship modelling representation

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

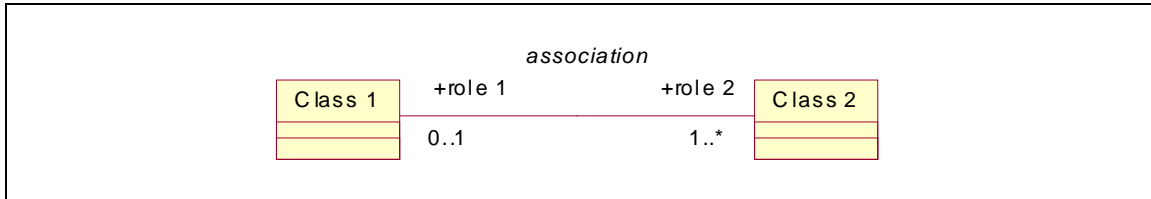


Figure A- 5: Class relationship cardinality modelling representation

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

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

### A.1.3 Associative entity

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

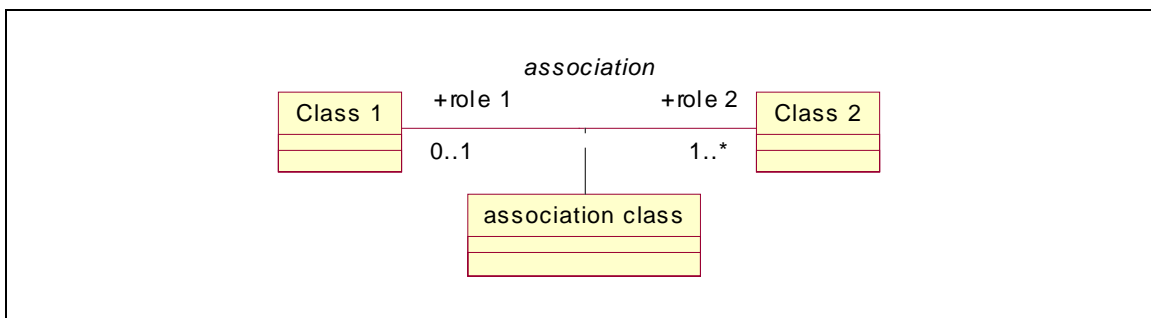


Figure A- 6: Associative class modelling representation

### A.1.4 Subtypes

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

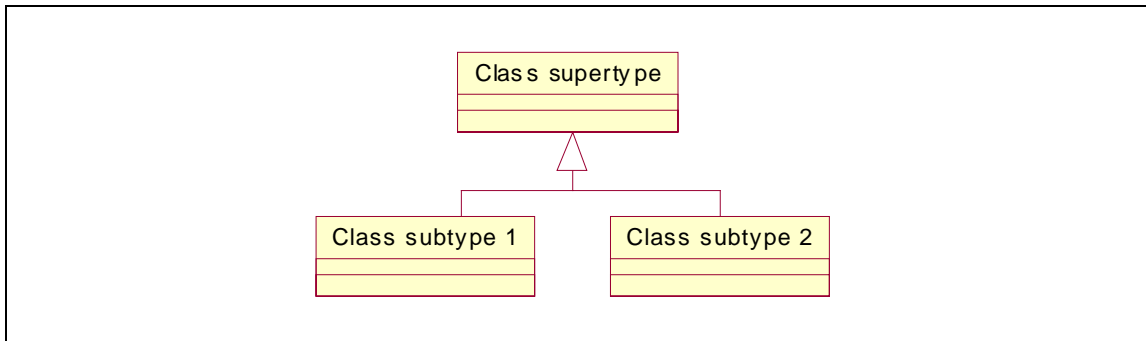


Figure A- 7: Subtype modelling representation

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

### A.1.5 Aggregation

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

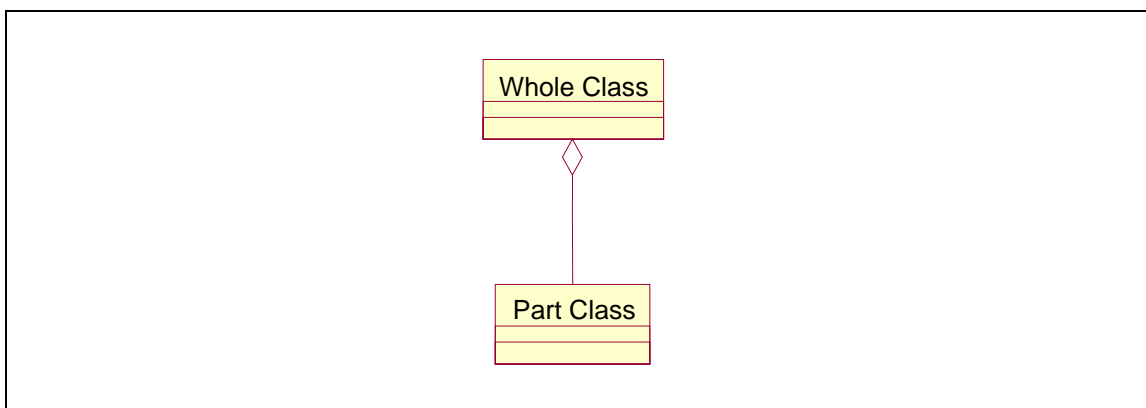


Figure A- 8: Aggregation modelling representation

### A.1.6 Composite Aggregation

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

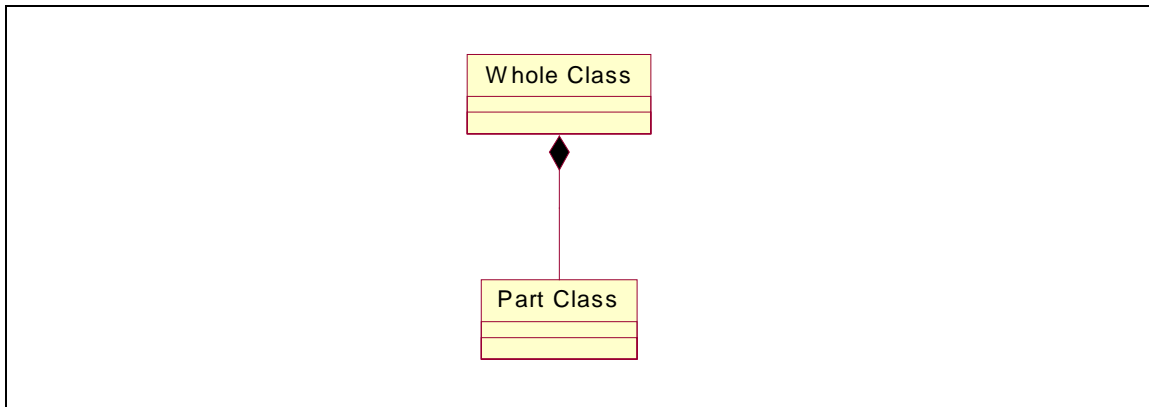


Figure A- 9: Composite Aggregation modelling representation

### A.1.7 Attribute

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

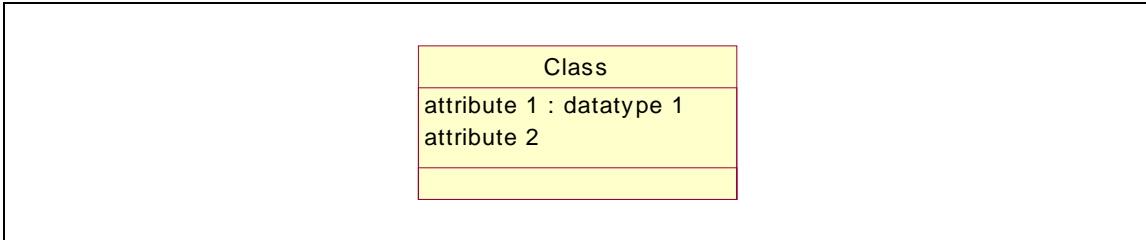


Figure A- 10: Class-attribute modelling representation

## **Annex B (Was Annex F) (informative) Alternative modeling representations**

**(Editors Note –The figures in this Annex will be re-generated when the normative UML model is approved.)**

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

**IDEF1X** a relational modeling paradigm, and  
**ORM or NIAM** a natural language 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 Section 8. The metamodel is represented in five diagrams. These diagrams are:

- 1. Administration, 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 Administration, 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**

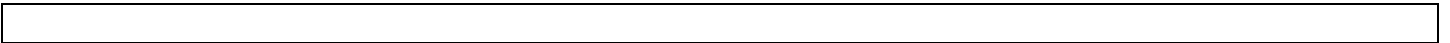
**Figure B4 – IDEF1X data Element administration metamodel region**

**Figure B5 – IDEF1X type hierarchy**

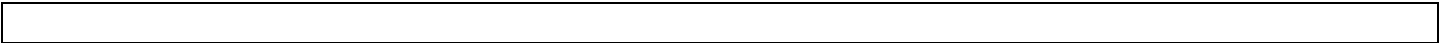
## B.2 Object Role Modeling (ORM) – Natural language Information Analysis Method (NIAM) representation model

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:

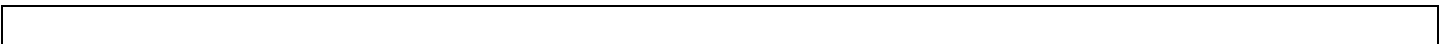
1. *Administration region,*
2. *Naming and identification region,*
3. *Classification region,*
4. *Data element concept administration region,*
5. *Conceptual domain and value domain administration region,*
6. *Data element administration region, and*
7. *Type hierarchy.*



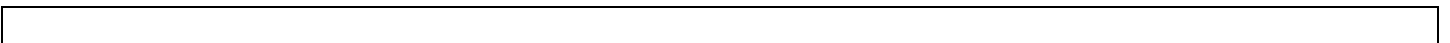
*Figure B- 1: ORM Administration region*



*Figure B- 2: ORM naming and identification region*



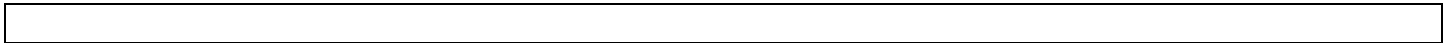
*Figure B- 3: ORM classification region*



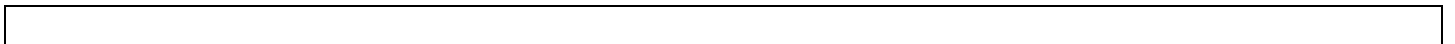
*Figure B- 4: ORM data element concept administration region*



*Figure B- 5: ORM conceptual domain and value domain administration region*



*Figure B- 6: ORM data element administration region*



*Figure B- 7: ORM type hierarchy*

## ~~Annex C (Was Annex A)(Informative) Additional attributes (for deletion)~~

~~In addition to the basic attributes specified in this Part of the International Standard the list of attributes can be extended depending on local data administration requirements.~~

~~The additions may refer to any of the categories identifying, definitional, relational, representational or administrative.~~

### ~~A.1. Additions to the category: identifying attributes~~

#### ~~A.1.1 Symbol (Reference: IS 12620)~~

~~A designation of a concept by letters, numerals, pictograms or any combination thereof e.g. the characters standing for chemical elements in the periodic system (Ag = silver) or the characters standing for electrical concepts (V =Voltage).~~

### ~~A.2. Additions to the category: definitional attributes~~

#### ~~A.2.1 Formula~~

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

~~Example: Data element 'inductance factor of soft magnetic part' L formula:  $AL = \mu N^2$~~

~~where L is the inductance of the coil when placed on the core, and N is the number of turns of the coil.~~

#### ~~A.2.2 Additional information~~

~~Information that describes a data element by giving useful details that can not adequately appear in the definition. Additional information can consists of an enumeration, a subset, a note, a remark or the like.~~

### ~~A.3. Additions to the category: relational attributes~~

~~Dictionaries can be used to control structures of application systems, message types or data models. Specific attributes for the control of structural relations can be defined to support 'where-used' or 'what-uses' queries.~~

#### ~~A.3.1 Data element associations~~

~~In a data management environment it may be required to control the relation between 'composite data elements' and the 'component data elements' which form the 'composite data element'.~~

~~Example: The composite data element: 'address' may be composed of the component data elements: 'name of addressee', 'street name', 'street number', 'city name', 'postal code', 'country name'.~~

~~(editor's note: this appears to only partially cover relationships between items, and as 'related data references provide for the specification of any type of relationships between data elements, the need to retain this additional attribute is questioned)~~

#### ~~A3.2 Data collection methods~~

To support the correct interpretation of the meaning and use of a data element it may be useful to provide comments and/or advice concerning the actual capture of data, including notes on the question(s) used to obtain the data, data collection protocols used and the treatment of 'not stated' responses or non-response.

Example: It is important to note how questions about cultural identity are completed – for example, whether self-reported, by administered questionnaire, etc.

#### *A.3.3 See also*

To support the correct interpretation of the meaning and use of a data element it may be useful to refer to (one or more) other data elements, which are semantically related or are derived from the same data element concept and have different representations.

Example 1: In the specification of data element: 'shipment number' the attribute 'see also' is used to refer to the data element 'consignment number', to emphasise that the data management environment makes a distinction between the concepts 'shipment' and 'consignment'.

Example 2: In the specification of the data element 'product description' the attribute 'see also' is used to refer to the data elements: 'product code' and 'trade name of product'.

#### *A.3.4 Procedural information*

A statement (or reference to a documented statement) about the procedural rules or constraints for the application of the data element in a certain application environment.

Example: In the specification of data element: 'currency code', the attribute 'procedural information' is used to state that in international data interchange this data element shall always be used in conjunction with data elements specifying 'financial amounts' and 'prices'.

### *A.4. Additions to the category: representation attributes*

#### *A.4.1 Validity category for usage*

Application for which the representation attributes are valid, e.g. electronic data interchange, database, user-interface.

Example: In the specification of data element 'currency code' the attribute 'validity category for usage' is used to indicate that this data element is used in electronic data interchange and in databases. In the specification of the data element 'currency name', the attribute 'validity category of usage' is used to indicate that this data element is used in a user-interface.

### *A.5. Additions to the category: administrative attributes*

#### *A.5.1 Expiry date*

Date at which the use of a data element expires in an application area. The date can be represented in the layout YYMMDD or CCYYMMDD according to ISO 8601.

#### *A.5.2 Source document*

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

#### *A.5.3 Date of approval*

The date when the data element is accepted for inclusion in the data element dictionary. The date can be represented in the layout YYMMDD or CCYYMMDD according to ISO 8601.

*A.5.4 Date of latest modification*

The date of the latest change in one or more attributes of the data element being specified in the data element dictionary. The date can be represented in the layout YYMMDD or CCYYMMDD according to ISO 8601.

*A.5.5 Date of receipt*

The date of receipt of a maintenance request for a data element from the submitting organization. The date can be represented in the layout YYMMDD or CCYYMMDD according to ISO 8601.

*A.5.6 Protection category and method*

The indication of the level of information protection and the method used, e.g. 'confidential, encrypted in data interchange' or 'exclusively for internal use, encrypted in external communication'.

**A.6. Additional category: internal storage attributes**

A data administration can require data element attributes that describe the properties for storing data elements in databases (internal level).

*A.6.1 Programming name*

A name of the data element that conforms to programming language conventions and is used in programs. Example: The data element name 'Article number' may be shortened to 'ARTNUC'.

*A.6.2 Database format*

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

*A.7. Attributes used for cross dictionary identification purposes*

To control data elements in a data element dictionary one or more basic attributes of the category 'identifying attributes' can be required to guarantee uniqueness when referring to data elements registered in other data element dictionaries or in the exchange among data element dictionaries.

The following basic attributes can be used for composing a unique cross dictionary identification:

- name (see Clause 7, section 7.1.1)
- context (see Clause 7, section 7.1.6)
- identifier (see Clause 7, section 7.1.2)
- version (see Clause 7, section 7.1.3)
- Registration Authority (see Clause 7, section 7.1.4)

For further recommendations see also Parts 5 and 6 of this International Standard.

**Annex D (Was Annex B) (Informative) (Removed to Content Technical report)**

**Annex E (Was Annex C) (Informative) (Removed to Content Technical report)**

## **~~Annex F (Was Annex D) (Informative) Recommendations for the use of additional registration descriptors of attributes (for deletion)~~**

~~When implementing basic attributes of data elements in a data element dictionary additional arrangements have to be made on the registration of the attributes, depending on the data management rules and the constraints of a data element dictionary system.~~

~~This Annex gives recommendations for the use of the following additional registration descriptors of attributes:~~

- ~~— language (see D1);~~
- ~~— maximum size of attribute value (see D2).~~

### ~~D.1. Language~~

~~The language in which data element attributes are specified can be valid for the entire data element dictionary, can vary per data element, or can vary per specific data element attribute.~~

#### ~~D.1.1 Default language for entire data element dictionary~~

~~The default language is valid for the entire data element dictionary. All attribute descriptors and attribute values are registered in only one language.~~

#### ~~D.1.2 Language varies per data element~~

~~In cases that a data element dictionary can comprise data elements that are specified in various languages, it is recommended that 'language' is a separate attribute of the category 'definitional attributes' and forms part of or qualifies the 'identifier' of the data element. All attribute descriptors and their values are then specified in the language as specified in the 'identifier'. For control purposes the attributes 'related data reference' and 'type of relationship' can be used to refer to similar data elements that are specified in more languages.~~

~~The language can be specified according to ISO 639, part 1.~~

~~**Example:** A data element dictionary can contain data element specifications in the Dutch and the German language.~~

#### **F.1.1.1 data element 1:**

Naam:	Groep van hulpstoffen
Identificatie:	EC-1002-nl
Taal:	NL
Definitie:	De identificatie van een groep van hulpstoffen
Gerelateerde gegevens referentie:	EC-1002-de
Soort relatie:	Duitstalig equivalent

etc.	
------	--

**F.1.1.2 — data element 2:**

Naam:	Hilfstoffgruppe
Identificatie:	EC-1002-de
Taal:	De
Definitie:	Die Identifikation einer Gruppe von Hilfstoffen.
Datenverweisung:	EC-1002-nl
Art der Verweisung:	Niederländische Sprache
etc.	

**Note:**

The languages chosen can require that also the descriptor 'character set of attribute' be specified separately.

**D.1.3 Language varies per data element attribute**

In case 'language' can vary for each data element attribute, 'language' can be specified for each variant attribute. The language can be specified according to ISO 639, part 1.

*Example*

A data management function has decided that, in addition to the default language 'English (en)' valid for the complete data element dictionary (including attribute descriptors) the attribute: 'name' can be presented in the languages: 'Dutch (nl)', 'French (fr)' and 'German (de)'. The language variants of 'name' then become separate attribute pairs: 'name' and 'context'. The attribute descriptor and its value are presented in the specified language.

**F.1.1.3 — data element:**

Name:	Dispatch date
Naam:	Verzenddatum
Context:	nl: Dutch language
Nom:	Date d'expédition
Context:	fr: French language
Name:	Versanddatum
Context:	de: German language
Definition:	The date at which goods are dispatched from the senders premises.

etc.	
------	--

#### D.2. Maximum size of attribute values

The following table gives recommendations for the maximum size for attribute values (see Clause 5, section 5.9).

If no recommendation is given a hyphen (-) indicates that the maximum is 'infinite'.

**Table 3: Recommendations for the maximum size of attribute values**

<b>Attribute Category</b>	<b>Name of data element attribute</b>	<b>maximum size of attribute values (in characters)</b>
Identifying	Name	80
	Identifier	15
	Version	5
	Registration Authority	35
	Synonymous name	80
	Context	80
Definitional	Definition	-
Relational	Classification scheme	35
	Keyword(s)	-
	Related data reference	-
	Type of relationship	-
Representational	Representation category	35
	Form of representation	-
	Datatype data element value	35
	Maximum size data element value	5
	Minimum size data element value	5
	Layout of representation	15
	Value domain	-
Administrative	Responsible organization	80
	Registration status	15
	Submitting organization	80
	Comments	-

***Annex G (Informative) Principles of managing shared data (Removed to Content Technical Report)***

**Annex H (Informative) Meta data registry uses and users (Removed to Content Technical Report)**

**Annex I (Informative) Conceptual and logical data models (Removed to Content Technical Report)**

## Annex J (Informative) Bibliographic references

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 5127-6:1983, *Documentation and Information – Vocabulary — Part 6: Documentary languages*

ISO TR9007:1987 *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.