INTERNATIONAL STANDARD

ISO/IEC 19763-3

Second edition
2010-08-15

Information technology — Metamodel framework for interoperability (MFI) —

Part 3: Metamodel for ontology registration

Technologies de l'information — Cadre du métamodèle pour l'interopérabilité (MFI) —

Partie 3: Métamodèle pour l'enregistrement de l'ontologie

Reference number
ISO/IEC 19763-3:2010(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Figures

Tables

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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

ISO/IEC 19763-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

This second edition cancels and replaces the first edition (ISO/IEC 19763-3:2007), which has been technically revised.

ISO/IEC 19763 consists of the following parts, under the general title *Information technology — Metamodel framework for interoperability (MFI)*:

— *Part 1: Reference model*

— *Part 2: Core model*

— *Part 3: Metamodel for ontology registration*

— *Part 4: Metamodel for model mapping*

The following part is under preparation:

— *Part 5: Metamodel for process model registration*

Registration procedure, metamodel for service registration, metamodel for role and goal registration, and on demand model selection will form the subjects of future parts.

Introduction

Interoperation among autonomous applications, such as Web services, is becoming important. To promote interoperation among application systems, unambiguous and formal specifications of the systems, especially of their inputs and outputs, are indispensable. Ontologies have a key role for that.

Several efforts to establish standards associated with ontologies have been made. But, most of them specify languages or are based on some particular language. To promote ontology-based interoperation, in addition to them, a generic framework for registering administrative and evolution information related to ontologies, independent of languages, is necessary.

This part of ISO/IEC 19763 intends to provide a generic framework for registering administrative and evolution information related to ontologies.

Information technology — Metamodel framework for interoperability (MFI) —

Part 3: Metamodel for ontology registration

1 Scope

ISO/IEC 19763 specifies a metamodel framework for interoperability. This part of ISO/IEC 19763 specifies the metamodel that provides a facility to register administrative and evolution information related to ontologies.

The metamodel specified in this part of ISO/IEC 19763 is intended to promote interoperation among application systems, by providing administrative and evolution information related to ontologies, accompanied with standardized ontology repositories that register ontologies themselves in specific languages.

This part of ISO/IEC 19763 does not specify the metamodels of ontologies expressed in specific languages and the mappings among them. They are specified in other specifications such as the Ontology Definition Metamodel from the Object Management Group (see bibliography item [1]).

Figure 1 shows the scope of this part of ISO/IEC 19763.

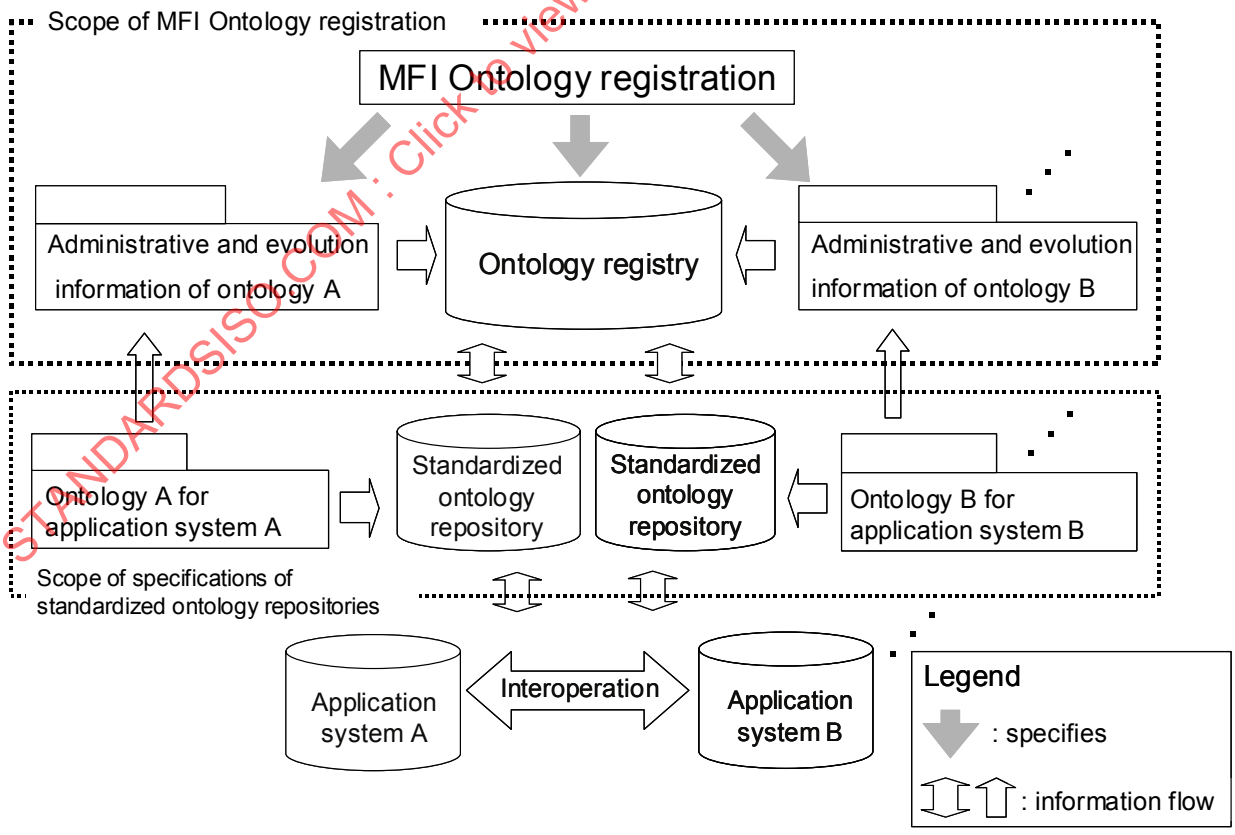


Figure 1 — Scope of MFI Ontology registration

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11179-3:2003, *Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes*

ISO/IEC 11179-3:2003/Cor.1:2004, *Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes — Technical Corrigendum 1*

ISO/IEC 19501:2005, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

ISO/IEC 19502:2005, *Information technology — Meta Object Facility (MOF)*

ISO/IEC 19763-1:2007, *Information technology — Metamodel framework for interoperability (MFI) — Part 1: Reference model*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 11179-3:2003, ISO/IEC 11179-3:2003/Cor.1:2004, ISO/IEC 19501:2005, ISO/IEC 19502:2005, ISO/IEC 19763-1:2007 and the following apply.

3.1.1 Terms on ontology

3.1.1.1
ontology
specification of concrete or abstract things, and the relationships among them, in a prescribed domain of knowledge

NOTE The specification should be computer processable.

3.1.1.2
registered ontology
ontology that is registered in a registry that conforms to MFI Ontology registration

3.1.1.3
unregistered ontology
ontology that is not registered in a registry that conforms to MFI Ontology registration

3.1.1.4
reference registered ontology
registered ontology that is usable and sharable by a community of interest

3.1.1.5
local registered ontology
registered ontology that is specialized for defined applications

3.1.2 Other terms

3.1.2.1
sentence
statement that has a truth value

3.1.2.2
symbol
most primitive lexical construct that is a part of a sentence

3.1.2.3
logical symbol
symbol whose meaning is defined by its language

EXAMPLE In KIF, “not” and “or” are logical symbols.

3.1.2.4
non-logical symbol
symbol that is not a logical symbol

3.1.2.5
authoritative extent
extent that indicates authoritativeness

NOTE 1 An authoritative extent is used to determine whether a thing may use another thing. A thing may use another thing if and only if the authoritative extent of the former is less than or equal to the authoritative extent of the latter because the usage of the latter by the former does not affect the authoritative extent of the former since the authoritative level of the latter is greater than or equal to the one of the former.

EXAMPLE A product with some authoritative extent can only use as its component a part with an authoritative extent which is greater than or equal to the one of it, to keep its authoritative extent.

NOTE 2 In this part of ISO/IEC 19763, an authoritative extent is used to determine whether a Local_Item can consist of or use another Local_Item. See 5.4.2 Local_Item.

3.2 Abbreviated terms

IRI
Internationalized Resource Identifier (see bibliography item [2])

KIF
Knowledge Interchange Format (see bibliography item [3])

MDR
Metadata Registry

[ISO/IEC 11179-3:2003, 3.4.5]

MFI
Metamodel framework for interoperability

[ISO/IEC 19763-1:2007, 4.2]

MFI Ontology registration
ISO/IEC 19763-3, *Information technology — Metamodel framework for interoperability (MFI) — Part 3: Metamodel for ontology registration*

ODM
Ontology Definition Metamodel (see bibliography item [1])

OWL
Web Ontology Language (see bibliography item [4])

UML
Unified Modeling Language (see ISO/IEC 19501:2005)

4 Conformance

4.1 General

An implementation claiming conformance to this part of ISO/IEC 19763 shall support one or both of the metamodels specified in this part of ISO/IEC 19763 and may or shall not support any extensions, depending on which level of conformance and which degree of conformance it claims.

4.2 Levels of conformance

4.2.1 General

An implementation may conform to either of the two levels of conformance to this part of ISO/IEC 19763, depending on what packages it supports.

4.2.2 Conformance level 1

The metamodel specified in 5.4 Basic_Model package is supported.

4.2.3 Conformance level 2

The metamodels specified in 5.4 Basic_Model package and 5.5 Evolution_Model package are supported.

4.3 Degree of conformance

4.3.1 General

In each conformance level, the distinction between “strictly conforming” and “conforming” implementations is necessary to address the simultaneous needs for interoperability and extensions. This part of ISO/IEC 19763 describes specifications that promote interoperability. Extensions are motivated by needs of users, vendors, institutions, and industries, but are not specified by this part of ISO/IEC 19763.

A strictly conforming implementation may be limited in usefulness but is maximally interoperable with respect to this part of ISO/IEC 19763. A conforming implementation may be more useful, but may be less interoperable with respect to this part of ISO/IEC 19763.

4.3.2 Strictly conforming implementation

A strictly conforming implementation for some conformance level:

a) shall support the metamodels required in the conformance level;

b) shall not support any extensions to the metamodels required in the conformance level.

4.3.3 Conforming implementation

A conforming implementation:

a) shall support the metamodels required in the conformance level;

b) may support extensions to the metamodels required in the conformance level, and the extensions shall be consistent with the metamodels required in the conformance level.

4.4 Implementation Conformance Statement (ICS)

An implementation claiming conformance to this part of ISO/IEC 19763 shall include an Implementation Conformance Statement stating:

a) which conformance level it claims(4.2);

b) whether it is a strictly conforming implementation or a conforming implementation (4.3);

c) what extensions are supported if it is a conforming implementation.

5 Structure of MFI Ontology registration

5.1 Overview of MFI Ontology registration

MFI Ontology registration consists of two packages: **Basic_Model** package and **Evolution_Model** package. Figure 2 shows the package structure of MFI Ontology registration.

Basic_Model package is used to register administrative information related to ontologies, independently of the languages that are used to express them. The basic idea is that almost any ontology consists of several sentences and that each sentence uses several non-logical symbols. The metaclasses in **Basic_Model** package include **Ontology_Whole**, **Ontology_Component** and **Ontology_Atomic_Construct**, which have administrative information of ontologies, sentences and non-logical symbols respectively. Since an ontology evolves, **Basic_Model** package can register as many versions of an ontology as necessary. But, in **Basic_Model** package, each version of an ontology is treated as a different ontology.

Evolution_Model package is used to register information on how an ontology evolves from one version to another. **Evolution_Model** package basically consists of three metaclasses, **Registered_Ontology_Whole_Evolution**, **Registered_Ontology_Component_Evolution** and **Registered_Ontology_Atomic_Construct_Evolution**, which have evolution information on **Registered_Ontology_Whole**, **Registered_Ontology_Component** and **Registered_Ontology_Atomic_Construct** respectively.

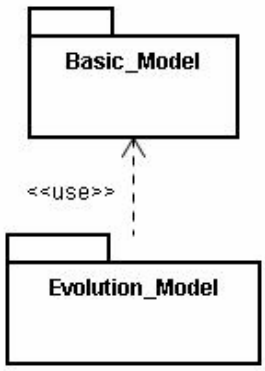


Figure 2 — Package structure of MFI Ontology registration

5.2 Overview of Basic_Model package

Figure 3 shows the metamodel in **Basic_Model** package.

Ontology_Whole is an abstract metaclass that represents an ontology and contains the associated administrative information. **Ontology_Language** is used as the reference "modelType" of **Ontology_Whole** to indicate a language that is used to express an ontology that is represented by **Ontology_Whole**.

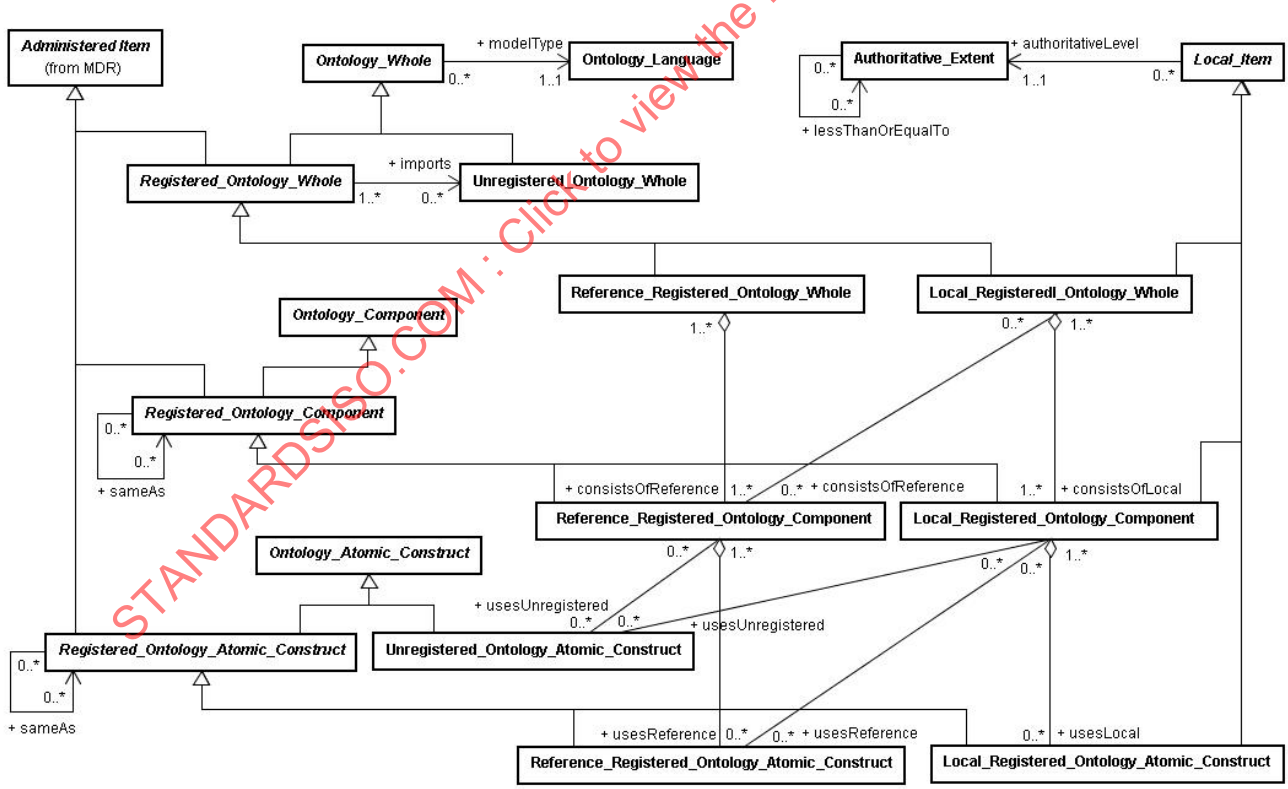
Ontology_Component is an abstract metaclass that represents a sentence contained in ontologies and contains the associated administrative information. The granularity of a sentence is not specified but is a user's choice. **Ontology_Atomic_Construct** is an abstract metaclass that represents a non-logical symbol that is used in sentences and contains the associated administrative information.

Ontology_Whole has two direct subclasses, **Registered_Ontology_Whole** and **Unregistered_Ontology_Whole**. **Registered_Ontology_Whole** is an abstract metaclass that represents an ontology that is registered in a registry that conforms to MFI Ontology registration and is therefore also a subclass of **Administered Item** of MDR. **Unregistered_Ontology_Whole** is a metaclass that represents an ontology that is not registered but imported by an ontology that is registered in a registry that conforms to MFI Ontology registration.

Ontology_Component has only one direct subclass **Registered_Ontology_Component**. **Registered_Ontology_Component** is an abstract metaclass that represents an ontology that is registered in a registry that conforms to MFI Ontology registration and is therefore also a subclass of **Administered Item** of MDR. Any instance of **Ontology_Component** is a **Registered_Ontology_Component** since a sentence that is not registered in a registry that conforms to MFI Ontology registration is out of scope.

Similar to **Ontology_Whole**, **Ontology_Atomic_Construct** has two direct subclasses, **Registered_Ontology_Atomic_Construct** and **Unregistered_Ontology_Atomic_Construct**. **Registered_Ontology_Atomic_Construct** is an abstract metaclass that represents a non-logical symbol that is registered in a registry that conforms to MFI Ontology registration. **Unregistered_Ontology_Atomic_Construct** is a metaclass that represents a non-logical symbol that is not registered but used by a sentence that is registered in a registry that conforms to MFI Ontology registration.

Registered_Ontology_Whole has two direct subclasses, **Reference_Registered_Ontology_Whole** and **Local_Registered_Ontology_Whole**. **Reference_Registered_Ontology_Whole** represents a reference registered ontology and **Local_Registered_Ontology_Whole** represents a local registered ontology.



NOTE Metaclasses whose names are italicized are abstract metaclasses.

Figure 3 — Metamodel in Basic_Model package

Similarly, **Registered_Ontology_Component** has two direct subclasses, **Reference_Registered_Ontology_Component** and **Local_Registered_Ontology_Component**. **Reference_Registered_Ontology_Component** represents a sentence contained in ontologies that are represented by **Reference_Registered_Ontology_Whole**. **Local_Registered_Ontology_Component** represents a sentence contained in ontologies that are represented by **Local_Registered_Ontology_Whole**. A sentence that is represented by **Reference_Registered_Ontology_Component** may also be contained in ontologies that are represented by **Local_Registered_Ontology_Whole**, but a sentence that is represented by **Local_Registered_Ontology_Component** cannot be contained in ontologies that are represented by **Reference_Registered_Ontology_Whole**.

Similarly again, **Registered_Ontology_Atomic_Construct** also has two direct subclasses, **Reference_Registered_Ontology_Atomic_Construct** and **Local_Registered_Ontology_Atomic_Construct**. **Reference_Registered_Ontology_Atomic_Construct** represents a non-logical symbol that is used in sentences that are represented by **Reference_Registered_Ontology_Component**. **Local_Registered_Ontology_Atomic_Construct** represents a non-logical symbol that is used in sentences that are represented by **Local_Registered_Ontology_Component**. A non-logical symbol that is represented by **Reference_Registered_Ontology_Atomic_Construct** may also be used in sentences that are represented by **Local_Registered_Ontology_Component**, but a non-logical symbol that is represented by **Local_Registered_Ontology_Atomic_Construct** cannot be used in sentences that are represented by **Reference_Registered_Ontology_Component**.

Local_Item is an abstract metaclass that is a collectively exhaustive SuperClass of **Local_Registered_Ontology_Whole**, **Local_Registered_Ontology_Component** and **Local_Registered_Ontology_Atomic_Construct**. **Authoritative_Extent** is used as the reference "authoritativeLevel" of **Local_Item** to indicate whether a **Local_Item** can consist of or use another **Local_Item**. A sentence that is represented by **Local_Registered_Ontology_Component** may also be contained in ontologies that are represented by **Local_Registered_Ontology_Whole** whose value of "authoritativeLevel" is less than or equal to the value of "authoritativeLevel" of this **Local_Registered_Ontology_Component**. A non-logical symbol that is represented by **Local_Registered_Ontology_Atomic_Construct** may also be used in sentences that are represented by **Local_Registered_Ontology_Component** whose value of "authoritativeLevel" is less than or equal to the value of "authoritativeLevel" of this **Local_Registered_Ontology_Atomic_Construct**.

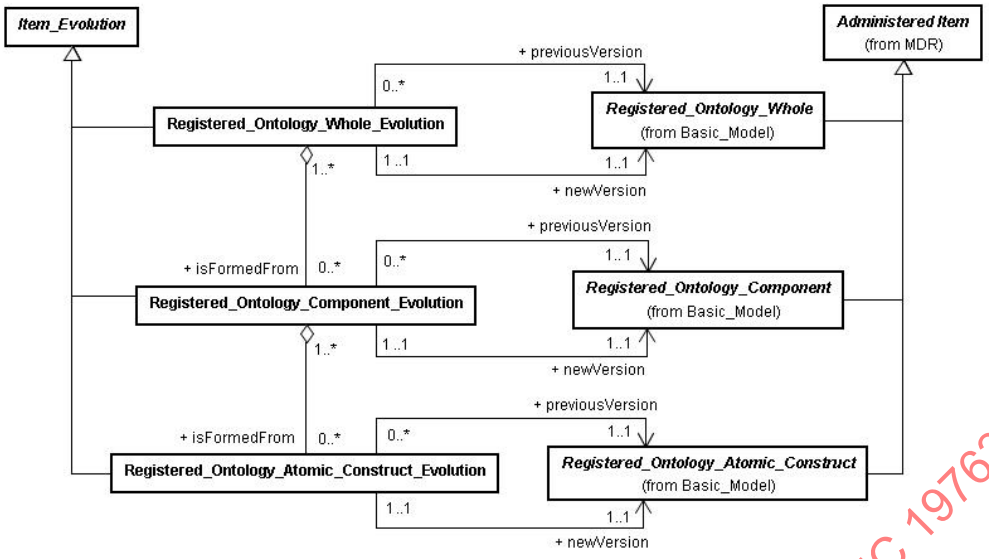
The exact specification of each metaclass in **Basic_Model** package is given in 5.4.

5.3 Overview of Evolution_Model package

Figure 4 shows the metamodel in **Evolution_Model** package.

Item_Evolution is an abstract metaclass that is a collectively exhaustive SuperClass of **Registered_Ontology_Whole_Evolution**, **Registered_Ontology_Component_Evolution** and **Registered_Ontology_Atomic_Construct_Evolution**. **Registered_Ontology_Whole_Evolution** is a metaclass that indicates what a **Registered_Ontology_Whole** evolves to and is possibly composed of **Registered_Ontology_Component_Evolution**. **Registered_Ontology_Component_Evolution** is a metaclass that indicates what a **Registered_Ontology_Component** evolves to and is possibly composed of **Registered_Ontology_Atomic_Construct_Evolution**. **Registered_Ontology_Atomic_Construct_Evolution** is a metaclass that indicates what a **Registered_Ontology_Atomic_Construct** evolves to. Evolution information on **Unregistered_Ontology_Whole** and **Unregistered_Ontology_Atomic_Construct** is out of scope since they are not registered.

The exact specification of each metaclass in **Evolution_Model** package is given in 5.5.



NOTE Metaclasses whose names are italicized are abstract metaclasses.

Figure 4 — Metamodel in Evolution_Model package

5.4 Basic_Model package

5.4.1 Authoritative_Extent

Authoritative_Extent is a metaclass whose instances indicate authoritative extent and compose a partially ordered set in terms of the reference "lessThanOrEqualTo".

Attribute	DataType	Multiplicity	Description
authoritativeExtent	String	1..1	String representing the authoritative extent of this Authoritative_Extent

Reference	Class	Multiplicity	Description
lessThanOrEqualTo	Authoritative_Extent	0..*	Authoritative_Extents that this Authoritative_Extent is less than or equal to

Constraints

The value of the attribute "authoritativeExtent" has to be unique in this metaclass.

NOTE Authoritative_Extent is used by Local_Item. See NOTE at 5.4.2.

5.4.2 Local_Item

Local_Item is an abstract metaclass that is a collectively exhaustive SuperClass of Local_Registered_Ontology_Whole, Local_Registered_Ontology_Component and Local_Registered_Ontology_Atomic_Construct.

Reference	Class	Multiplicity	Description
authoritativeLevel	Authoritative_Extent	1..1	Authoritative extent of this Local_Item

NOTE Local_Item X can consist of or use Local_Item Y if and only if the authoritativeLevel of X is less than or equal to the authoritativeLevel of Y. For more details, see Constraints, 5.4.8 Local_Registered_Ontology_Whole and 5.4.12 Local_Registered_Ontology_Component.

5.4.3 Ontology_Language

Ontology_Language is a metaclass that represents an ontology expression language.

Attribute	DataType	Multiplicity	Description
name	String	1..1	Name of the ontology expression language.

Constraints

The value of the attribute “name” has to be unique in this metaclass.

NOTE Annex A suggests a means of deciding the value of the attribute.

5.4.4 Ontology_Whole

Ontology_Whole is an abstract metaclass that is a collectively exhaustive SuperClass of Registered_Ontology_Whole and Unregistered_Ontology_Whole.

Attribute	DataType	Multiplicity	Description
IRI	String	1..1	IRI that identifies the corresponding ontology
ontologyName	String	1..1	Name of the corresponding ontology

Reference	Class	Multiplicity	Description
modelType	Ontology_Language	1..1	Ontology_Language that is used to express the corresponding ontology

Constraints

The value of the attribute “IRI” has to be unique in this metaclass.

5.4.5 Registered_Ontology_Whole

Registered_Ontology_Whole is an abstract metaclass that is a collectively exhaustive SuperClass of Reference_Registered_Ontology_Whole and Local_Registered_Ontology_Whole.

SuperClass

Administered Item (from MDR), Ontology_Whole

Reference	Class	Multiplicity	Description
imports	Unregistered_Ontology_Whole	0..*	Unregistered_Ontology_Wholes representing an unregistered ontology that is imported by the corresponding ontology

Constraints

Exists exactly one Registered_Ontology_Whole_Evolution whose “newVersion” is this Registered_Ontology_Whole.

5.4.6 Unregistered_Ontology_Whole

Unregistered_Ontology_Whole is a metaclass representing an unregistered ontology.

SuperClass
Ontology_Whole

Constraints
Exists at least one Registered_Ontology_Whole whose “imports” is this Unregistered_Ontology_Whole.

5.4.7 Reference_Registered_Ontology_Whole

Reference_Registered_Ontology_Whole is a metaclass representing a reference registered ontology.

SuperClass
Registered_Ontology_Whole

Reference	Class	Multiplicity	Description
consistsOfReference	Reference_ Registered_ Ontology_ Component	1..*	Reference_Registered_Ontology_Components representing the sentence contained in the ontology represented by this Reference_Registered_ Ontology_Whole

5.4.8 Local_Registered_Ontology_Whole

Local_Registered_Ontology_Whole is a metaclass representing a local registered ontology.

SuperClass
Registered_Ontology_Whole, Local_Item

Reference	Class	Multiplicity	Description
consistsOfReference	Reference_ Registered_ Ontology_ Component	0..*	Reference_Registered_Ontology_Components representing the sentence contained in the ontology represented by this Local_Registered_Ontology_ Whole
consistsOfLocal	Local_ Registered_ Ontology_ Component	1..*	Local_Registered_Ontology_Components representing the sentence contained in the ontology represented by this Local_Registered_Ontology_ Whole

Constraints
The value of the attribute "authoritativeLevel" (inherited from Local_Item) of this Local_Registered_Ontology_ Whole has to be "lessThanOrEqualTo" the value of the attribute "authoritativeLevel" (inherited from Local_Item) of the Local_Registered_Ontology_Component that is the value of the reference “consistsOfLocal” of this Local_Registered_Ontology_Whole.

NOTE It is good practice that a Local_Registered_Ontology_Whole consists of some Reference_Registered_ Ontology_Components as it evolves.

5.4.9 Ontology_Component

Ontology_Component is an abstract metaclass that is a collectively exhaustive SuperClass of Registered_ Ontology_Component.

Attribute	DataType	Multiplicity	Description
namespace	String	1..1	IRI where the value of the sentenceIdentifier is uniquely identified
sentenceIdentifier	String	1..1	Identifier of the corresponding sentence within the namespace

Constraints

a) The value of the attribute “sentenceIdentifier” prefixed by the value of the attribute “namespace” has to be unique in this metaclass.

b) The value of the attribute “sentenceIdentifier” prefixed by the value of the attribute “namespace” has to identify the corresponding sentence.

5.4.10 Registered_Ontology_Component

Registered_Ontology_Component is an abstract metaclass representing a sentence contained in a registered ontology that is represented by Registered_Ontology_Whole. Granularity of a sentence is not specified in this part of the standard.

SuperClass

Administered Item (from MDR), Ontology_Component

Reference	Class	Multiplicity	Description
sameAs	Registered_ Ontology_ Component	0..*	Registered_Ontology_Components representing the sentence that is interpreted exactly the same as the sentence represented by this Registered_ Ontology_Component

Constraints

Exists exactly one Registered_Ontology_Component_Evolution whose “newVersion” is this Registered_ Ontology_Component.

5.4.11 Reference_Registered_Ontology_Component

Reference_Registered_Ontology_Component is a metaclass representing a sentence contained in a reference registered ontology that is represented by Reference_Registered_Ontology_Whole.

SuperClass

Registered_Ontology_Component

Reference	Class	Multiplicity	Description
usesUnregistered	Unregistered_ Ontology_ Atomic_Construct	0..*	Unregistered_Ontology_Atomic_Constructs representing the non-logical symbol that is used in the sentence represented by this Reference_ Registered_Ontology_Component
usesReference	Reference_ Registered_ Ontology_ Atomic_Construct	0..*	Reference_Registered_Ontology_Atomic_ Constructs representing the non-logical symbol that is used in the sentence represented by this Reference_Registered_Ontology_Component

Constraints

a) Exists at least one Reference_Registered_Ontology_Whole whose "consistsOfReference" is this Reference_Registered_Ontology_Component.

b) The range of the attribute "sameAs" (inherited from Registered_Ontology_Component) has to be within Reference_Registered_Ontology_Component.

5.4.12 Local_Registered_Ontology_Component

Local_Registered_Ontology_Component is a metaclass representing a sentence contained in a local registered ontology that is represented by Local_Registered_Ontology_Whole.

SuperClass

Registered_Ontology_Component, Local_Item

Reference	Class	Multiplicity	Description
usesUnregistered	Unregistered_ Ontology_ Atomic_Construct	0..*	Unregistered_Ontology_Atomic_Constructs representing the non-logical symbol that is used in the sentence represented by this Local_ Registered_Ontology_Component
usesReference	Reference_ Registered_ Ontology_ Atomic_Construct	0..*	Reference_Registered_Ontology_Atomic_ Constructs representing the non-logical symbol that is used in the sentence represented by this Local_ Registered_Ontology_Component
usesLocal	Local_ Registered_ Ontology_ Atomic_Construct	0..*	Local_Registered_Ontology_Atomic_Constructs representing the non-logical symbol that is used in the sentence represented by this Local_ Registered_Ontology_Component

Constraints

a) Exists at least one Local_Registered_Ontology_Whole whose “authoritativeLevel” (inherited from Local_Item) is the same as the one of this Local_Registered_Ontology_Component and whose “consistsOfLocal” is this Local_Registered_Ontology_Component.

b) The value of the attribute "authoritativeLevel" (inherited from Local_Item) of this Local_Registered_Ontology_Component has to be "lessThanOrEqualTo" the value of the attribute "authoritativeLevel" (inherited from Local_Item) of the Local_Registered_Ontology_Atomic_Construct that is the value of the reference “usesLocal” of this Local_Registered_Ontology_Component.

c) If a value of the attribute "sameAs" (inherited from Registered_Ontology_Component) of this Local_Registered_Ontology_Component is an instance of Local_Registered_Ontology_Component, the value of the attribute "authoritativeLevel" (inherited from Local_Item) of this Local_Registered_Ontology_Component has to be "lessThanOrEqualTo" the value of the attribute "authoritativeLevel" (inherited from Local_Item) of the instance of Local_Registered_Ontology_Component.

NOTE It is good practice that a Local_Registered_Ontology_Component uses some Reference_Registered_Ontology_Atomic_Constructs as it evolves

5.4.13 Ontology_Atomic_Construct

Ontology_Atomic_Construct is an abstract metaclass that is a collectively exhaustive SuperClass of Registered_Ontology_Atomic_Construct and Unregistered_Ontology_Atomic_Construct.

Attribute	DataType	Multiplicity	Description
namespace	String	1..1	IRI where the corresponding non-logical symbol is uniquely identified
nonLogicalSymbol	String	1..1	The corresponding non-logical symbol

Constraints

The value of the attribute “nonLogicalSymbol” prefixed by the value of the attribute “namespace” has to be unique in this metaclass.

5.4.14 Registered_Ontology_Atomic_Construct

Registered_Ontology_Atomic_Construct is an abstract metaclass representing a non-logical symbol that is defined in a sentence that is represented by Registered_Ontology_Component.

SuperClass

Administered Item (from MDR), Ontology_Atomic_Construct

Reference	Class	Multiplicity	Description
sameAs	Registered_Ontology_Atomic_Construct	0..*	Registered_Ontology_Atomic_Constructs representing the non-logical symbol that is interpreted exactly the same as the non-logical symbol represented by this Registered_Ontology_Atomic_Construct

Constraints

Exists exactly one Registered_Ontology_Atomic_Construct_Evolution whose “newVersion” is this Registered_Ontology_Atomic_Construct.

5.4.15 Unregistered_Ontology_Atomic_Construct

Unregistered_Ontology_Atomic_Construct is a metaclass representing a non-logical symbol that is not defined but used in a sentence that is represented by Registered_Ontology_Component.

SuperClass

Ontology_Atomic_Construct

5.4.16 Reference_Registered_Ontology_Atomic_Construct

Reference_Ontology_Atomic_Construct is a metaclass representing a non-logical symbol that is defined in a sentence that is represented by Reference_Registered_Ontology_Component.

SuperClass

Registered_Ontology_Atomic_Construct

Constraints

a) Exists at least one Reference_Registered_Ontology_Component whose “usesReference” is this Reference_Registered_Ontology_Atomic_Construct.

b) The range of the attribute "sameAs" (inherited from Registered_Ontology_Atomic_Construct) has to be within Reference_Registered_Ontology_Atomic_Construct.

5.4.17 Local_Registered_Ontology_Atomic_Construct

Local_Registered_Ontology_Atomic_Construct is a metaclass representing a non-logical symbol that is defined in a sentence that is represented by Local_Registered_Ontology_Component.

SuperClass

Registered_Ontology_Atomic_Construct, Local_Item

Constraints

a) Exists at least one Local_Registered_Ontology_Component whose “authoritativeLevel” (inherited from Local_Item) is the same as the one of this Local_Registered_Ontology_Atomic_Construct and whose “consistsOfLocal” is this Local_Registered_Ontology_Atomic_Construct.

b) If a value of the attribute "sameAs" (inherited from Registered_Ontology_Atomic_Construct) of this Local_Registered_Ontology_Atomic_Construct is an instance of Local_Registered_Ontology_Atomic_Construct, the value of the attribute "authoritativeLevel" (inherited from Local_Item) of this Local_Registered_Ontology_Atomic_Construct has to be "lessThanOrEqualTo" the value of the attribute "authoritativeLevel" (inherited from Local_Item) of the instance of Local_Registered_Ontology_Atomic_Construct.

5.5 Evolution_Model package

5.5.1 Item_Evolution

Item_Evolution is an abstract metaclass that is a collectively exhaustive SuperClass of Registered_Ontology_Whole_Evolution, Registered_Ontology_Component_Evolution and Registered_Ontology_Atomic_Construct_Evolution.

5.5.2 Registered_Ontology_Whole_Evolution

Registered_Ontology_Whole_Evolution is a metaclass that has information on evolution of Registered_Ontology_Wholes.

SuperClass
Item_Evolution

Reference	Class	Multiplicity	Description
previousVersion	Registered_Ontology_Whole	1..*	Registered_Ontology_Wholes that the evolution represented by this Registered_Ontology_Whole_Evolution is from
newVersion	Registered_Ontology_Whole	1..1	Registered_Ontology_Whole that the evolution represented by this Registered_Ontology_Whole_Evolution is to
isFormedFrom	Registered_Ontology_Component_Evolution	0..*	Registered_Ontology_Component_Evolutions representing the evolution of Registered_Ontology_Components that form the evolution represented by this Registered_Ontology_Whole_Evolution

5.5.3 Registered_Ontology_Component_Evolution

Registered_Ontology_Component_Evolution is a metaclass that has information on evolution of Registered_Ontology_Components.

SuperClass
Item_Evolution

Reference	Class	Multiplicity	Description
previousVersion	Registered_Ontology_Component	1..*	Registered_Ontology_Components that the evolution represented by this Registered_Ontology_Component_Evolution is from
newVersion	Registered_Ontology_Component	1..1	Registered_Ontology_Component that the evolution represented by this Registered_Ontology_Component_Evolution is to
isFormedFrom	Registered_Ontology_Atomic_Construct_Evolution	0..*	Registered_Ontology_Atomic_Construct_Evolutions representing the evolution of Registered_Ontology_Atomic_Construct that form the evolution represented by this Registered_ Ontology_Component_Evolution

Constraints

Exists at least one Registered_Ontology_Whole_Evolution whose “isFormedFrom” is this Registered_Ontology_Component_Evolution.

5.5.4 Registered_Ontology_Atomic_Construct_Evolution

Registered_Ontology_Atomic_Construct_Evolution is a metaclass that has information on evolution of Registered_Ontology_Atomic_Constructs.

SuperClass
Item_Evolution

Reference	Class	Multiplicity	Description
previousVersion	Registered_ Ontology_ Atomic_Construct	1..*	Registered_Ontology_Atomic_Constructs that the evolution represented by this Registered_Ontology_ Atomic_Construct_Evolution is from
newVersion	Registered_ Ontology_ Atomic_Construct	1..1	Registered_Ontology_Atomic_Construct that the evolution represented by this Registered_Ontology_ Atomic_Construct_Evolution is to

Constraints

Exists at least one Registered_Ontology_Component_Evolution whose “isFormedFrom” is this Registered_Ontology_Atomic_Construct_Evolution.

Annex A
(informative)

List of Ontology_Languages

Annex A suggests that the value of the attribute “name” of “Ontology_Language” be one of the values in column “name” of Table A.1 if there is an appropriate one. If there is not an appropriate one, a new name may be used as the value of the attribute “name” of “Ontology_Language”, but it has to be managed in accordance with ISO/IEC 19763-6 Registration procedure.[1)]

Table A.1 — List of Ontology_Languages

name	Description
CGIF	The dialect of Common Logic specified in Annex B (normative) Conceptual Graph Interchange Format (CGIF), ISO/IEC 24707:2007 (see bibliography item [5])
CL	A language other than CLIF, CGIF or XCL that conforms to ISO/IEC 24707:2007 (see bibliography item [5])
CLIF	The dialect of Common Logic specified in Annex A (normative) Common Logic Interchange Format (CLIF), ISO/IEC 24707:2007 (see bibliography item [5])
KIF	Knowledge Interchange Format specified at http://www-ksl.stanford.edu/knowledge-sharing/kif/
NIAM	Natural language Information Analysis Method specified at http://www.essentialstrategies.com/publications/modeling/niam.htm
ORM	Object Role Modeling specified at http://www.orm.net/
OWL	Any of the sublanguages specified by “OWL Web Ontology Language Semantics and Abstract Syntax”, W3C Recommendation (see bibliography item [6])
OWL2	Any of the sublanguages specified by “OWL 2 Web Ontology Language Conformance”, W3C Recommendation (see bibliography item [7])
RDFS	A language that conforms to “Resource Description Framework (RDF): Concepts and Abstract Syntax” and “RDF Vocabulary Description Language 1.0: RDF Schema”, W3C Recommendations (see bibliography items [8] [9])
SBVR	A language that conforms to “Semantics of Business Vocabulary and Business Rules (SBVR), v1.0”, formal/2008-01-02 (see bibliography item [10])
TM	A language that conforms to ISO/IEC 13250:2007 (see bibliography item [11])
UML	ISO/IEC 19501, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*
XCL	The dialect of Common Logic specified in Annex C (normative) eXtended Common Logic Markup Language (XCL), ISO/IEC 24707:2007 (see bibliography item [5])

1) To be published.

Annex B
(informative)

Example of Basic_Model

B.1 Example of a reference registered ontology

Suppose that some organization establishes a formalized ontology called “RO1” about kernel units in OWL. A kernel unit is a unit with its own name without prefix, such as “metre”. In “RO1”, there are several sentences about kernel units. Figure B.1 shows three examples of them. Suppose that these sentences are named “RC1”, “RC2”, and “RC3” as shown in Figure B.1.

RC1

```
<owl:ObjectProperty rdf:ID="dimensionality">
  <rdfs:domain rdf:resource="#Unit" />
  <rdfs:range rdf:resource="#Dimensionality" />
</owl:ObjectProperty>
```

RC2

```
<owl:Class rdf:ID="KernelUnit">
  <rdfs:subClassOf rdf:resource="#Unit"/>
</owl:Class>
```

RC3

```
<KernelUnit rdf:ID="metre">
  <dimensionality>
    <Dimensionality rdf:ID="length"/>
  </dimensionality>
</KernelUnit>
```

where xmlns:owl=http://www.w3.org/2002/07/owl#
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#

Figure B.1 — Three examples of the sentences in RO1

Then, “RC1” means “Dimensionality is a property of a unit.", “RC2” means “A kernel unit is a unit.”, and “RC3” means “Metre is a kernel unit whose dimensionality is length.”.

Figure B.2 shows how “RO1” is registered as a reference registered ontology in accordance with MFI Ontology registration.

< Reference_Registered_Ontology_Whole>

Object01

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record01
IRI	http://ref1.org/kernel-unit
ontologyName	"RO1"
modelType	"OWL"
consistsOfReference	Object02
	Object03
	Object04
	…

<Reference_Registered_Ontology_Component>

Object02

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record02
namespace	http://ref1.org/kernel-unit
sentenceIdentifier	"RC1"
usesReference	Object05
	Object06
	Object07

Object03

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record03
namespace	http://ref1.org/kernel-unit
sentenceIdentifier	"RC2"
usesReference	Object06
	Object08

Object04

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record04
namespace	http://ref1.org/kernel-unit
sentenceIdentifier	"RC3"
usesReference	Object05
	Object07
	Object08
	Object09
	Object10

<Reference_Registered_Ontology_Atomic_Construct>

Object05

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record05
namespace	http://ref1.org/kernel-unit
nonLogicalSymbol	"dimensionality"

Object06

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record06
namespace	http://ref1.org/kernel-unit
nonLogicalSymbol	"Unit"

Object07

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record07
namespace	http://ref1.org/kernel-unit
nonLogicalSymbol	"Dimensionality"

Object08

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record08
namespace	http://ref1.org/kernel-unit
nonLogicalSymbol	"KernelUnit"

Object09

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record09
namespace	http://ref1.org/kernel-unit
nonLogicalSymbol	"metre"

Object10

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record10
namespace	http://ref1.org/kernel-unit
nonLogicalSymbol	"length"

Figure B.2 — Registration of RO1

NOTE 1 For simplicity, the sentences other than “RC1”, “RC2”, and “RC3” are ignored.

NOTE 2 Except “administered_item_administration_record”, the attributes and references that are inherited from the other standards are not shown for simplicity.

NOTE 3 Objectxx (xx= 01 to 10) are object identifiers introduced only for the descriptive purpose of this example. The detailed specifications of them are beyond the scope of this part of the standard.

B.2 Example of another reference registered ontology

Suppose that another organization establishes a formalized ontology called “RO2” about prefixed units in OWL. A prefixed unit is a unit with prefix, such as “kilometer”. In “RO2”, there are several sentences about prefixed units. Figure B.3 shows two examples of them. Suppose that these sentences are named “RC4” and “RC5” as shown in Figure B.3.

RC4

```
<owl:Class rdf:ID="PrefixedUnit">
 <rdfs:subClassOf rdf:resource="&ref1;Unit"/>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:about="#prefix"/>
   <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:onProperty rdf:about="#kernel"/>
   <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

RC5

```
<PrefixedUnit rdf:ID="micrometre">
 <prefix>
  <MetricPrefix rdf:ID="micro"/>
 </prefix>
 <kernel>
  <ref1:KernelUnit rdf:resource="&ref1;metre"/>
 </kernel>
</PrefixedUnit>
```

Figure B.3 — Two examples of the sentences in RO2

Then, “RC4” means “A prefixed unit is a unit, has exactly one prefix, and has exactly one kernel.” and “RC5” means “Micrometre is a prefixed unit whose prefix is micro as a metric prefix and whose kernel is metre as a kernel unit.”.

Figure B.4 shows how “RO2” is registered as a reference registered ontology in accordance with MFI Ontology registration.

< Reference_Registered_Ontology_Whole>

Object11

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record11
URI	http://ref2.org/prefixed-unit
ontologyName	"RO2"
modelType	"OWL"
consistsOfReference	Object12
	Object13
	…

<Reference_Registered_Ontology_Component>

Object12

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record12
namespace	http://ref2.org/prefixed-unit
sentenceIdentifier	"RC4"
usesReference	Object06
	Object14
	Object15
	Object16

Object13

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record13
namespace	http://ref2.org/prefixed-unit
sentenceIdentifier	"RC5"
usesReference	Object08
	Object09
	Object14
	Object15
	Object16
	Object17
	Object18
	Object19

<Reference_Registered_Ontology_Atomic_Construct>

Object14

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record14
namespace	http://ref2.org/prefixed-unit
nonLogicalSymbol	"PrefixedUnit"

Object15

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record15
namespace	http://ref2.org/prefixed-unit
nonLogicalSymbol	"prefix"

Object16

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record16
namespace	http://ref2.org/prefixed-unit
nonLogicalSymbol	"kernel"

Object17

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record17
namespace	http://ref2.org/prefixed-unit
nonLogicalSymbol	"micrometre"

Object18

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record18
namespace	http://ref2.org/prefixed-unit
nonLogicalSymbol	"MetricPrefix"

Object19

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record19
namespace	http://ref2.org/prefixed-unit
nonLogicalSymbol	"micro"

Figure B.4 — Registration of RO2

NOTE 1 For simplicity, the sentences other than “RC4” and “RC5” are ignored.

NOTE 2 Except “administered_item_administration_record”, the attributes and references that are inherited from the other standards are not shown for simplicity.

NOTE 3 Objectxx (xx= 06 to 19) are object identifiers introduced only for the descriptive purpose of this example. The detailed specifications of them are beyond the scope of this part of the standard.

B.3 Example of a local registered ontology

Suppose that some application system establishes its own ontology called “LO1” about units based on “RO1” and “RO2”. “LO1” is expressed in KIF and not in OWL and its authoritative level is “local1”. In “LO1”, there are several sentences about units for this application system. Figure B.5 shows three examples of them. Suppose that these sentences are named “LC1”, “LC2”, and “LC3” as shown in Figure B.5.

LC1

```
(and (KernelUnit metre) (Dimensionality length) (dimensionality metre length))
```

LC2

```
(and (PrefixedlUnit micron) (MetricPrefix micro) (KernelUnit metre)
(prefix micron micro) (kernel micron metre))
```

LC3

```
((forall ?Unit1 ?Unit2 ?Dimensionality1 ?Dimensionality2)
(implies (and (PrefixedUnit ?Unit1) (kernel ?Unit1 ?Unit2)
(dimensionality ?Unit1 ?Dimensionality1) (dimensionality ?Unit2 ?Dimensionality2))
(equal ?Dimensionality1 ?Dimensionality2)))
```

Figure B.5 — Three examples of the sentences in LO1

Then, “LC1” has the same meaning as “RC3” in “RO1” and uses the same non-logical symbols as “RC3” in “RO1”.

“LC2” has the same meaning as “RC5” in “RO2” and uses the same non-logical symbols as “RC5” in “RO2”, except that “micron” is used instead of “micrometer” for this application system.

“LC3” is a new sentence that is not in “RO1” or in “RO2”. “LC3” means “Dimensionality of a kernel unit and the dimensionality of the prefixed unit whose kernel is the kernel unit are equal.”.

Figure B.6 shows how “LO1” is registered as a local registered ontology in accordance with MFI Ontology registration.

<Local_Registered_Ontology_Whole>

Object20

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record20
authoritativeLevel	Object25
IRI	http://local1.org/unit
ontologyName	"LO1"
modelType	"KIF"
consistsOfLocal	Object21
	Object22
	Object23
	…

<Local_Registered_Ontology_Component>

Object21

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record21
authoritativeLevel	Object25
namespace	http://local1.org/unit
sentenceIdentifier	"LC1"
usesReference	Object05
	Object07
	Object08
	Object09
	Object10
sameAs	Object04

Object22

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record22
authoritativeLevel	Object25
namespace	http://local1.org/unit
sentenceIdentifier	"LC2"
usesReference	Object08
	Object09
	Object14
	Object15
	Object16
	Object18
	Object19
usesLocal	Object24
sameAs	Object13

Object23

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record23
authoritativeLevel	Object25
namespace	http://local1.org/unit
sentenceIdentifier	"LC3"
usesReference	Object05
	Object14
	Object16

<Local_Registered_Ontology_Atomic_Construct>

Object24

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record24
authoritativeLevel	Object25
namespace	http://local1.org/unit
nonLogicalSymbol	"micron"
sameAs	Object17

<Authoritative_Extent>

Object25

Attribute/Reference	Literal/Instance
authoritativeExtent	"local1"
lessThanOrEqualTo	…

Figure B.6 — Registration of LO1

NOTE 1 For simplicity, the sentences other than “LC1”, “LC2”, and “LC3” are ignored.

NOTE 2 Except “administered_item_administration_record”, the attributes and references that are inherited from the other standards are not shown for simplicity.

NOTE 3 Objectxx (xx= 04 to 25) are object identifiers introduced only for the descriptive purpose of this example. The detailed specifications of them are beyond the scope of this part of the standard.

B.4 Example of another local registered ontology

Suppose that another application system establishes its own ontology called “LO2” about units based on “LO1”, “RO1” and “RO2”. “LO2” is also expressed in KIF and its authoritative level is “local2”, which is less than or equal to “local1”, which is the authoritative level of “LO1”. “LO2” reuses “LC1” and “LC2”, sentences in “LO1” and have several other sentences about units for this application system. Figure B.7 shows an example of them. Suppose that this sentence is named “LC4” as shown in Figure B.7.

LC4

```
(and (PrefixedIUnit angstrom) (MetricPrefix 100pico) (KernelUnit metre)
(prefix angstrom 100pico) (kernel angstrom metre))
```

Figure B.7 — An example of the sentences in LO2

Then, “LC4” is a new sentence that is not in either “RO1”, “RO2” or “LO1”. “LC4” means “Angstrom is a prefixed unit whose prefix is 100pico as a metric prefix and whose kernel is metre as a kernel unit.”

Figure B.8 shows how “LO1” is registered as a local registered ontology in accordance with MFI Ontology registration.

<Local_Registered_Ontology_Whole>

Object26

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record27
authoritativeLevel	Object30
IRI	http://local2.org/unit
ontologyName	"LO2"
modelType	"KIF"
consistsOfLocal	Object21
	Object22
	Object27
	…

<Local_Registered_Ontology_Component>

Object27

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record28
authoritativeLevel	Object30
namespace	http://local2.org/unit
sentenceIdentifier	"LC4"
usesReference	Object08
	Object09
	Object14
	Object15
	Object16
usesLocal	Object28
	Object29

<Local_Registered_Ontology_Atomic_Construct>

Object28

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record29
authoritativeLevel	Object30
namespace	http://local2.org/unit
nonLogicalSymbol	"angstrom"

Object29

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record30
authoritativeLevel	Object30
namespace	http://local2.org/unit
nonLogicalSymbol	"100pico"

<Authoritative_Extent>

Object30

Attribute/Reference	Literal/Instance
authoritativeExtent	"local2"
lessThanOrEqualTo	object25

Figure B.8 — Registration of LO2

NOTE 1 For simplicity, the sentences other than “LC4” are ignored.

NOTE 2 Except “administered_item_administration_record”, the attributes and references that are inherited from the other standards are not shown for simplicity.

NOTE 3 Objectxx (xx= 08 to 30) are object identifiers introduced only for the descriptive purpose of this example. The detailed specifications of them are beyond the scope of this part of the standard.

Annex C
(informative)

Example of Evolution_Model

C.1 Example of evolution

Suppose that “LO1” evolves to “LO3” so that “LO3” uses non-logical symbol “micrometer” rather than “micron”. Since “LO2” is still based on “LO1”, “LO3” is identified by different IRI from the one of “LO1”. Figure C.1 shows three examples of the sentences in “LO3”. These sentences are named “LC5”, “LC6”, and “LC7”, as shown in Figure C.1, corresponding to “LC1”, “LC2” and “LC3” in “LO1” respectively.

LC5

```
(and (KernelUnit metre) (Dimensionality length) (dimensionality metre length))
```

LC6

```
(and (PrefixedIUnit micrometre) (MetricPrefix micro) (KernelUnit metre)
(prefix micrometre micro) (kernel micrometre metre))
```

LC7

```
((forall ?Unit1 ?Unit2 ?Dimensionality1 ?Dimensionality2)
(implies (and (PrefixedUnit ?Unit1) (kernel ?Unit1 ?Unit2)
(dimensionality ?Unit1 ?Dimensionality1) (dimensionality ?Unit2 ?Dimensionality2))
(equal ?Dimensionality1 ?Dimensionality2)))
```

Figure C.1 — Three examples of the sentences in LO3

Then, “LC5” has the same meaning as “RC3” in “RO1” and uses the same non-logical symbols as “RC3” in “RO1”.

“LC6” has the same meaning as “RC5” in “RO2” and uses the same non-logical symbols as “RC5” in “RO2”.

“LC7” has the same meaning as “LC3” in “LO1” and uses the same non-logical symbols as “LC3” in “LO1”.

Figure C.2 shows how “LO3” is registered as a local registered ontology in accordance with MFI Ontology registration.

<Local_Registered_Ontology_Whole>

Object31

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record32
authoritativeLevel	Object25
URI	http://local1.org/unit/evo
ontologyName	"LO3"
modelType	"KIF"
consistsOfLocal	Object21
	Object22
	Object23
	…

<Local_Registered_Ontology_Component>

Object32

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record33
authoritativeLevel	Object25
namespace	http://local1.org/unit/evo
sentenceIdentifier	"LC5"
usesReference	Object05
	Object07
	Object08
	Object09
	Object10
sameAs	Object04

Object33

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record34
authoritativeLevel	Object25
namespace	http://local1.org/unit/evo
sentenceIdentifier	"LC6"
usesReference	Object08
	Object09
	Object14
	Object15
	Object16
	Object17
	Object18
	Object19
sameAs	Object13

Object34

Attribute/Reference	Literal/Instance
administered_item_administration_record	Administration_Record35
authoritativeLevel	Object25
namespace	http://local1.org/unit/evo
sentenceIdentifier	"LC7"
usesReference	Object05
	Object14
	Object16
sameAs	Object23

Figure C.2 — Registration of LO3

NOTE 1 For simplicity, the sentences other than “LC5”, “LC6”, and “LC7” are ignored.

NOTE 2 Except “administered_item_administration_record”, the attributes and references that are inherited from the other standards are not shown for simplicity.

NOTE 3 Objectxx (xx= 04 to 34) are object identifiers introduced only for the descriptive purpose of this example. The detailed specifications of them are beyond the scope of this part of the standard.