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.

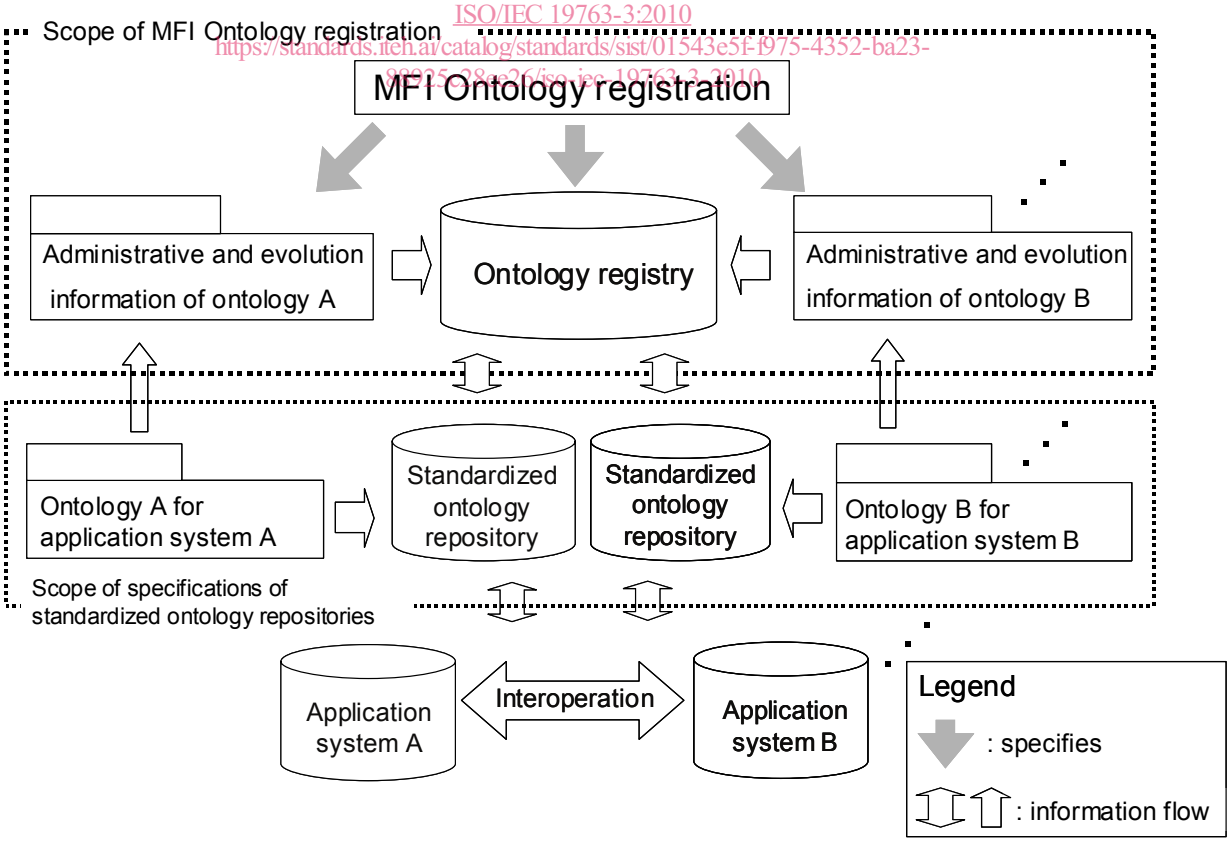


Figure 1 — Scope of MFI Ontology registration

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.

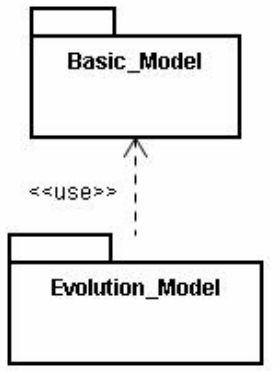


Figure 2 — Package structure of MFI Ontology registration

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**.

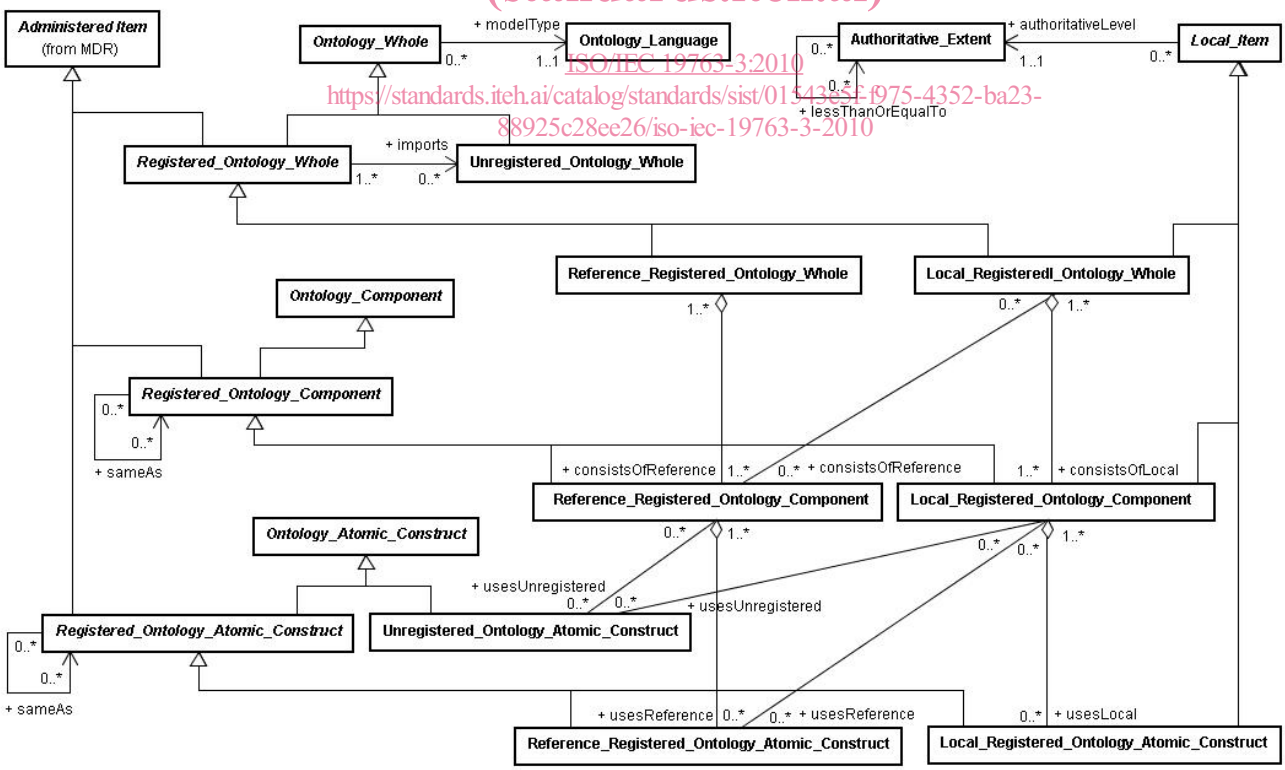
Ontology_Component is an abstract metaclass that represents a sentence contained in ontologies and contains the associated administrative information. The granularity of a sentence is not specified but is a user's choice. **Ontology_Atomic_Construct** is an abstract metaclass that represents a non-logical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