

---

---

**Information technology —  
Metamodel framework for  
interoperability (MFI) —**

**Part 6:  
Registry Summary**

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

*Partie 6: Résumé Registry*



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2015, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Ch. de Blandonnet 8 • CP 401  
CH-1214 Vernier, Geneva, Switzerland  
Tel. +41 22 749 01 11  
Fax +41 22 749 09 47  
copyright@iso.org  
www.iso.org

# Contents

Page

Foreword .....	v
Introduction.....	vi
<b>1 Scope .....</b>	<b>1</b>
<b>2 Conformance .....</b>	<b>1</b>
2.1 General .....	1
2.2 Degree of conformance .....	1
2.2.1 General .....	1
2.2.2 Strictly conforming.....	2
2.2.3 Conforming implementation .....	2
2.3 Implementation Conformance Statement (ICS).....	2
<b>3 Normative references .....</b>	<b>2</b>
<b>4 Terms, definitions and abbreviated item .....</b>	<b>2</b>
4.1 Terms and definitions .....	2
4.2 Terms for concepts used in this part of 19763.....	2
4.3 Abbreviate terms .....	3
<b>5 Background and purpose .....</b>	<b>4</b>
5.1 General .....	4
5.2 Role of the Registry Summary .....	4
<b>6 Struct of Registry Summary .....</b>	<b>5</b>
6.1 Overview of Registry Summary .....	5
6.2 Associations between Registry Summary and Core_Model package from MFI-10.....	6
6.3 Metaclasses in the Registry_Summary package .....	7
6.3.1 Classification metaclass.....	7
6.3.2 Contact metaclass.....	8
6.3.3 Contents metaclass.....	8
6.3.4 Document metaclass.....	8
6.3.5 Interface metaclass .....	9
6.3.6 Organization metaclass .....	10
6.3.7 Other_Standard_Model metaclass .....	11
6.3.8 Registry metaclass.....	11
6.3.9 Registry_Component metaclass.....	13
6.3.10 Related_Registry metaclass.....	14
6.3.11 Registry_Summary_Language.....	15
<b>Annex A (informative) Types of a registry system for Registry Summary.....</b>	<b>16</b>
A.1 Introduction.....	16
A.2 Types of a registry system for Registry Summary .....	16
A.2.1 General .....	16
A.2.2 Independent Registry.....	16
A.2.3 Replicated Registry .....	16
A.2.4 Federated Registry .....	17
<b>Annex B (informative) Example of using models of other standard in Registry Summary .....</b>	<b>18</b>
B.1 Introduction.....	18
B.2 Adaptation method of model of other standard .....	18
B.2.1 General .....	18
B.2.2 Example of using ISO/IEC 11179-3:2013 classes.....	19
B.2.3 Example of using UN/CEFACT Core Component Library classes .....	19
<b>Annex C (informative) Example of XML Schema for Registry Summary .....</b>	<b>20</b>

<b>C.1</b>	<b>Introduction .....</b>	<b>20</b>
<b>C.2</b>	<b>XML Schema definition .....</b>	<b>20</b>
<b>C.2.1</b>	<b>General.....</b>	<b>20</b>
<b>C.2.2</b>	<b>ComplexType definition .....</b>	<b>20</b>
<b>C.2.3</b>	<b>Sample element definition .....</b>	<b>24</b>
<b>Annex D</b>	<b>(informative) Example of indicator of Registry Summary.....</b>	<b>25</b>
<b>D.1</b>	<b>Introduction .....</b>	<b>25</b>
<b>D.2</b>	<b>Location of Registry Summary in META tag.....</b>	<b>25</b>

## 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 19763-6 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data Management and Interchange*.

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

- *Part 1: Framework*
- *Part 3: Metamodel for ontology registration*
- *Part 5: Metamodel for process model registration*
- *Part 6: Registry Summary (this document)*
- *Part 7: Metamodel for service model registration*
- *Part 8: Metamodel for role and goal model registration*
- *Part 9: On demand model selection [Technical Report]*
- *Part 10: Core model and basic mapping*
- *Part 12: Metamodel for information model registration*
- *Part 13: Metamodel for form design registration*

## **Introduction**

The effective interchange of information across business domains, countries and cultures is an important concern for people in both the IT industry and non-IT industries.

To follow the current trends, industrial consortia have engaged in the standardization of domain-specific business objects, including business process models and software components using common modelling tools and interchange facilities such as UML and XML. They are very active in standardizing domain-specific business process models and standard modelling constructs such as data elements, entity profiles and value domains.

The ISO/IEC 19763 family of standards defines normative metamodels for the registration of models (including information models and process models), ontologies, services and roles & goals. Items or objects specified by those metamodels have been registered into particular registry systems. In order to perform effective exchange of pertinent information smoothly, individual registry systems need to interoperate with other registry systems.

This part of ISO/IEC 19763 defines a metamodel for the use case in which registry systems of different kinds must share information.

# Information Technology — Metamodel framework for interoperability (MFI) — Part 6: Registry Summary

## 1 Scope

The ISO/IEC 19763 family of standards defines normative metamodels for the registration of models (including information models and process models), ontologies, services and roles & goals. Currently a lot of metadata registries or model registries were constructed and utilized in many different business domains, such as e-business, healthcare, automobile, electronics devices and civil construction.

One of the key issues for the cross domain data or services integration must be enabling the easy discovery of metadata that are stored in the different registries that were scattered over different domains. Therefore, it is necessary to provide specific metadata that describes the registry itself in order to enable interoperation among different registries that were built following different standards.

This part of the ISO/IEC 19763 family of standards specifies an information artefact called the Registry Summary. The Registry Summary consists of information that describes administrative aspects, the summary of contents and the technical access method of the registry.

A collection of multiple Registry Summary information is called a "Registry of Registries" or RoR, however, this standard does not mandate a particular implementation. Also, any specific protocol between Registry Summaries and RoR, such as creation of RoR and synchronization of RoR, would not be specified in this standard. Those are to be specified by other standards.

The Registry Summary and RoR concept should be applied to all Metamodel Framework for Interoperability (MFI) registries, but its use may be applied to any kind of registries.

## 2 Conformance

### 2.1 General

An implementation claiming conformance with this part of ISO/IEC 19763 shall support the metamodel specified in clause 6, depending on a degree of conformance as described below.

### 2.2 Degree of conformance

#### 2.2.1 General

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.

### 2.2.2 Strictly conforming

A strictly conforming implementation

- a) shall support the metamodel specified in clause 6;
- b) shall not support any extensions to the metamodel specified in clause 6.

### 2.2.3 Conforming implementation

A conforming implementation

- a) shall support the metamodel specified in clause 6;
- b) may support extensions to the metamodel specified in clause 6 that are consistent with the metamodel specified in clause 6.

## 2.3 Implementation Conformance Statement (ICS)

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

- a) whether it is a strictly conforming implementation or a conforming implementation (see 2.2)
- b) what extensions are supported if it is a conforming implementation.

## 3 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 19763-1, *Information technology — Metamodel framework for interoperability (MFI) — Part 1: Framework*

ISO/IEC 19763-10, *Information technology — Metamodel framework for interoperability (MFI) — Part 10: Core model and basic mapping*

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

ISO/IEC 11179-6, *Information technology — Metadata registries (MDR) — Part 6: Registration*

## 4 Terms, definitions and abbreviated item

### 4.1 Terms and definitions

For the purposes of this document, the items and definitions given in ISO/IEC 19763-1, ISO/IEC 11179-3, ISO/IEC 11179-6 and the following apply.

### 4.2 Terms for concepts used in this part of 19763

#### 4.1.1 attribute

⟨metamodel⟩ characteristic of an object or set of objects



[ISO/IEC 11179-3:2013, 3.1.4]

#### 4.1.2

##### **class**

⟨metamodel⟩ description of a set of objects that share the same attributes, operations, methods, relationships, and semantics

[ISO/IEC 11179-3:2013, 3.1.5]

#### 4.1.3

##### **metadata**

data that defines and describes other data

[ISO/IEC 11179-3:2013, 3.2.74]

#### 4.1.4

##### **metadata registry**

##### **MDR**

##### **registry system**

information system for registering metadata,

NOTE 1 The associated information store or database is known as a metadata register.

[ISO/IEC 11179-3:2013, 3.2.77]

NOTE 2 registry system has been added as a new synonym,

#### 4.1.5

##### **registry of registries**

##### **RoR**

registry that stores Registry Summary data showing an individual registry system

#### 4.1.6

##### **registry summary**

registry summary is metadata which refers to an individual registry system or an aggregate of registry systems

### 4.3 Abbreviate terms

MDR	Metadata Registry
MFI	Metamodel Framework for Interoperability
RoR	Registry of Registries
RS	Registry Summary
SLA	Service Level Agreement
UML	Unified Modeling Language
URL	Uniform Resource Locator
XML	Extensible Markup Language
WSDL	Web Service Description Language

## 5 Background and purpose

### 5.1 General

The purpose of this part of ISO/IEC 19763 family of standards is to define the metamodel of information required in order to enable interoperability between heterogeneous metadata registries.

Currently, many metadata registries have been developed and enforced in various types of business domains. Most of them have been developed conforming to international standards, such as e-business, healthcare, or library domains. However, those standards themselves have been incompatible each other, they are developed primarily according to requirements that came from their particular domains. This means that a single company or user who belongs to a particular domain has difficulties in accessing registries that are built according to requirements from other domains.

Sharing registry information across different domains can be important for system interoperability. This part addresses registry interoperation with regard to the problems mentioned above. This standard specifies a set of information called the Registry Summary to be attached to each domain-specific registry. Furthermore, Registry Summary is premised on exchanging between registries. The Registry Summary is represented by a metamodel using UML.

### 5.2 Role of the Registry Summary

The Registry Summary is a small set of metadata for the Registry that is attached in the registry system. If every registry would attach this registry summary, it would be possible to capture information on the nature of another registry as well as technical information for accessing the registry.

Figure 1 illustrates a typical use case for RS (Registry Summary) and ROR (Registry of Registries). Each registry prepared in a particular domain might be able to disclose RS at its access point as entry information. Different domains, such as Manufacturing or Retailing, would be able to prepare their own dedicated registry that is a collection of RS. It should be called the Registry of Registries (ROR).

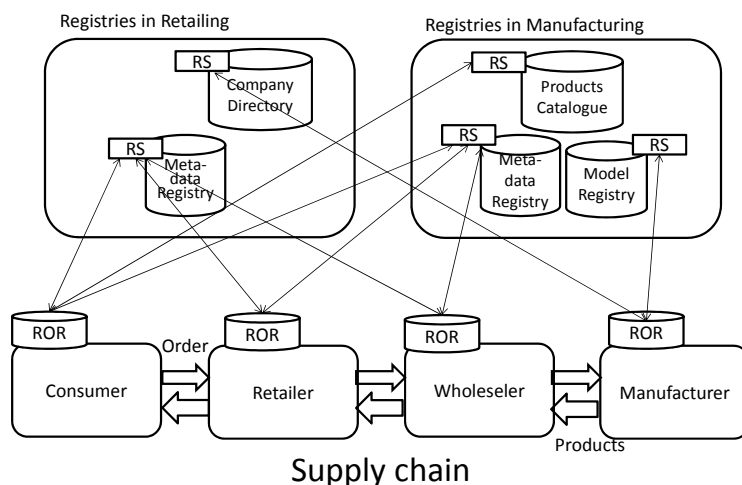


Figure 1 — Illustration of Registry Interoperability using Registry Summary

## 6 Struct of Registry Summary

### 6.1 Overview of Registry Summary

The Registry Summary is information which describes a particular registry system itself. Figure 2 shows the metamodel of Registry Summary. This metamodel consists of three parts.

The first part is "registry entity information" that forms a core part of the metadata in the Registry Summary. This core part represents the registry system, such as the name of the registry, and information about the contents. Furthermore, when registry system is constituted of two or more registries, the constitution of the registry and relationship among the parts can be expressed with this part.

NOTE 1 Registry entity information consists of Registry metaclass (see 6.3.8), Registry\_Component metaclass (see 6.3.9), Related\_Registry metaclass (see 6.3.10), Contact metaclass (see 6.3.2), Classification metaclass (see 6.3.1) and Document metaclass (see 6.3.4).

NOTE 2 Refer to Annex A for the detailed explanation about the form of the registry which Registry Summary expresses.

The second part is "registry administrator information". Registry administrator information identifies the administrator of a represented registry. Such information includes the name of organization that is managing and operating the registry and the points of contact.

NOTE 3 Registry administrator information consists of Organization metaclass (see 6.3.6) and Contact metaclass (see 6.3.2).

The third part is "registry interface information". "Registry interface information" is the information which indicates the interface through which to access the registry system by an individual or other information system.

NOTE 4 Registry interface information consist Interface metaclass (see 6.3.5) Document metaclass (see 6.3.4).

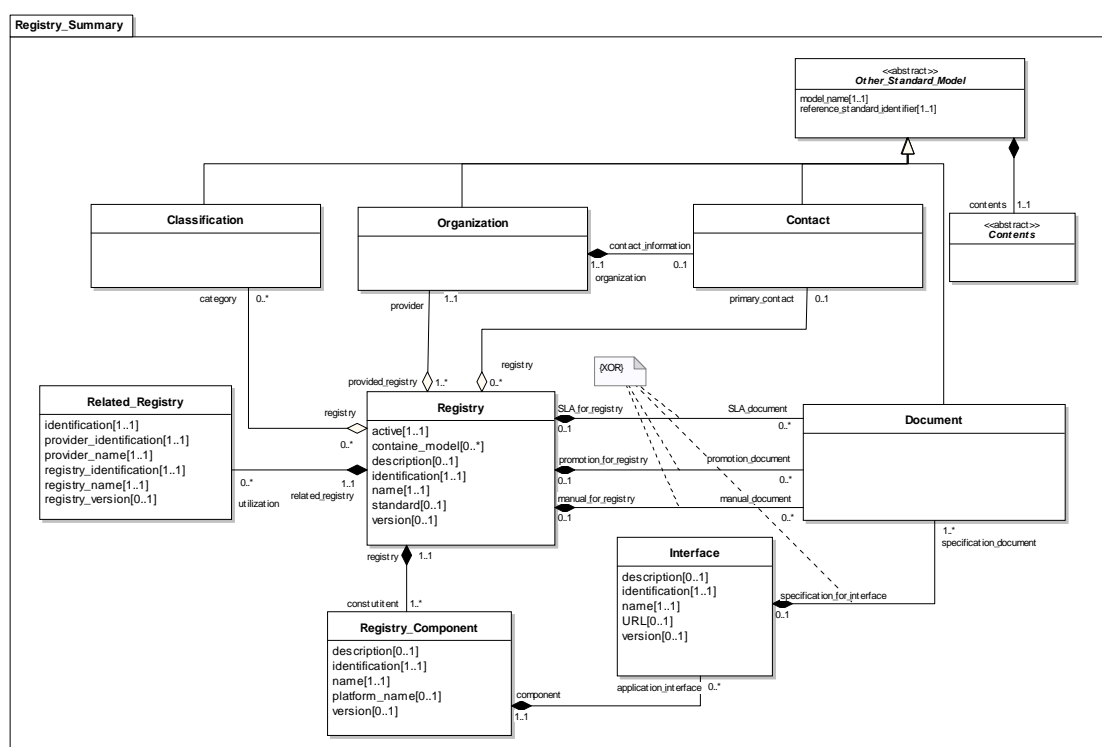


Figure 2 — Metamodel of Registry Summary

NOTE 5 Classification metaclass (see 6.3.1), Contact metaclass (see 6.3.2), Document metaclass (see 6.3.4) and Organization metaclass (see 6.3.6) are subclass of Other\_Standard\_Model metaclass (see 6.3.7) which is abstract classes. The models of information which these metaclasses represent are already defined by other standards. Therefore, this part reuses the models which other standards defined. Annex B shows an example of reuse of the models of other standard in Registry Summary.

Each Registry contains Related\_Registry. Each Registry utilizes zero, one or more Related\_Registry. Each Related\_Registry is utilized by one and only one Registry.

Each Registry contains Registry\_Component. Each Registry is constituted by one or more Registry\_Component. Each Registry\_Component constitutes one and only one Registry.

Each Registry contains three types document, each type document does not double as other type document.

Each Registry has zero, one or more Document as the SLA document. Each Document which is the SLA document is related by one and only one Registry which is the target of SLA.

Each Registry has zero, one or more Document as the promotion document. Each Document which is the promotion document is related by one and only one Registry which is the target of promotion.

Each Registry has zero, one or more Document as the manual document. Each Document which is the manual document is related by one and only one Registry which is explained by the manual.

Each Registry is categorised by zero, one or more Classification. Each Classification categorises zero, one or more Registry.

Each Registry is provided by one and only one Organization. Each Organization provides one or more Registry.

Each Registry has zero or one Contact as the primary contact point. Each Contact represents the primary contact point of zero, one or more Registry.

Each Organization contains Contact. Each Organization has zero or one Contact as contact information. Each Contact represents the contact information of one and only one Organization.

Each Contact is contained by Registry and/or Organization.

Each Interface has one or more Document as the specification document. Each Document which is the specification document is related by zero or one Interface. When the Document is the specification, each Document is not had by any Registry.

Each Registry\_Component contains Interface. Each Registry\_Component has zero, one or more Interface as the application interface. Each Interface is had by one and only one Registry\_Component.

Each Other\_Standard\_Model contains Contents. Each Other\_Standard\_Model has only one Content as the content. This relation is inherited to subclasses (Classification, Contact, Document and Organization).

## **6.2 Associations between Registry Summary and Core\_Model package from MFI-10**

Figure 3 shows the associations between metaclass in the Registry Summary and metaclass in the Core\_Model package from ISO/IEC 19763-10 (MFI-10).

Registry\_Summary\_Language in Registry Summary is the subclass of Modelling\_Language in Core\_Model package. Registry in Registry Summary is the subclass of Model in Core\_Model package. Interface, Registry\_Component, Related\_Registry and Other\_Standard\_Model are the subclass of Model\_Element in Core\_Model package.

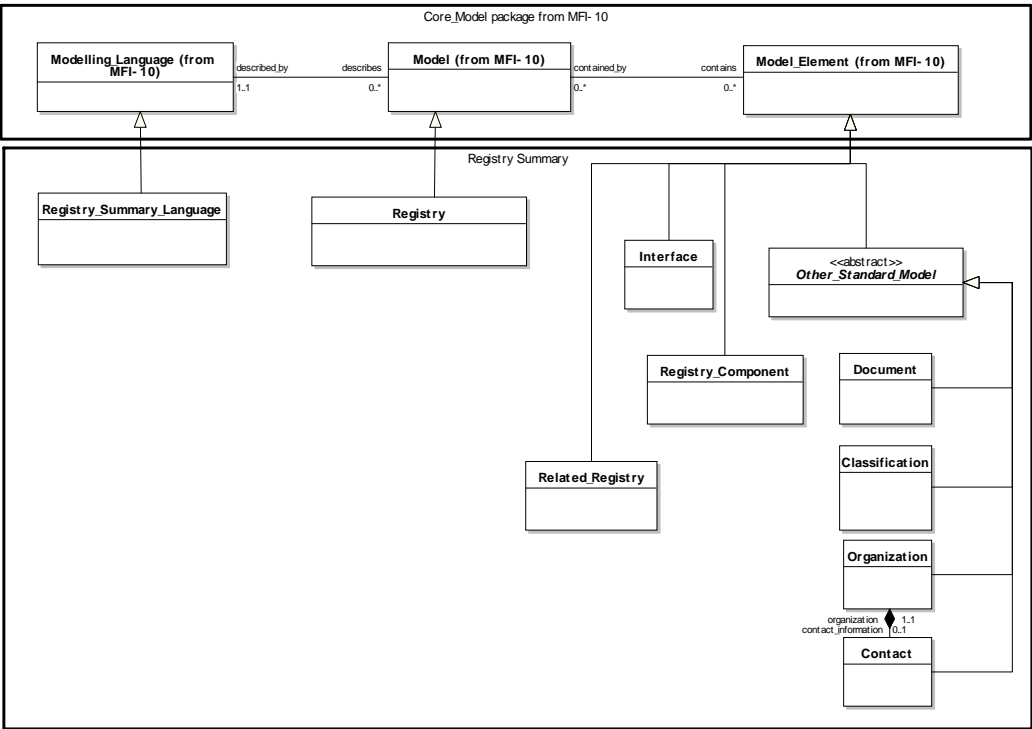


Figure 3 — Associations between Registry Summary and Core\_Model package from MFI-10

All subclasses have the association which is inherited from superclass. Some inherited associations are specialised in this part of ISO/IEC 19763 family of standards. The detail of specialization is defined in clause 6.3.

6.3 Metaclasses in the Registry\_Summary package

6.3.1 Classification metaclass

Classification is a metaclass each instance of which represents data about the categorization of Registry (see 6.3.8). When the Classification metaclass is implemented, the models which are defined by other standard are reused. The purpose of this class is to express various classifications to the registry which the instance of registry class represents. For example, the classification of any particular domain categorizes the registry which exhibits the contents of information in the specific domain.

Superclass  
Other\_Standard\_Model (see 6.3.7)

Attribute	Datatype	Multiplicity	Description	Inverse	Precedence
None					
Reference registry	Class Registry	Multiplicity 0..*	Description One of a set of registries belonging to a specific instance of this metaclass. This reference specialize the "contained_by" reference which is inherited from the superclass.	Inverse category	Precedence Yes

### 6.3.2 Contact metaclass

Contact is a metaclass each instance of which represents data about the Point of Contact (POC) of the registry and the organization described in Registry metaclass (see 6.3.8) and Organization metaclass (see 6.3.6) respectively. When this metaclass is implemented, the models which are defined by other standard are reused. The purpose of this class is to express the information on contacts, such as a telephone number, a mail address, and so on.

#### Superclass

*Other\_Standard\_Model* (see 6.3.7)

Attribute	Datatype	Multiplicity	Description		
None					
Reference	Class	Multiplicity	Description	Inverse	Precedence
organization	Organization	1..1	The organization for which this contact is the POC.	contact_organization	No
registry	Registry	0..*	The registries for which this contact is the POC. This reference specialize the "contained_by" reference which is inherited from the superclass	primary_contact	No

### 6.3.3 Contents metaclass

Contents is an abstract metaclass for quoting model which is defined by other standard specification. This metaclass is equivalent to an "any" element of an XML schema. The how to use of this metaclass is shown in Annex B.

#### Superclass

*Model\_Element* (from MFI-10)

Attribute	Datatype	Multiplicity	Description		
None					
Reference	Class	Multiplicity	Description	Inverse	Precedence
-	<i>Other_Standard_Model</i>	1..1	This reference is inverse relation from metaclasses of Registry_Summary Package to class defined by other standard. (see <i>Other_Standard_Model</i> metaclass (see 6.3.7) and Annex B)	contents	No

NOTE In this metaclass, the inherited reference from *Model\_Element* (from MFI-10) does not refer to any *Model* (from MFI-10) and subclasses of *Model* (from MFI-10).

### 6.3.4 Document metaclass

Document is a metaclass each instance of which represents data about the detail information of Registry metaclass (see 6.3.8) and Interface metaclass (see 6.3.5). When this metaclass is implemented, the models which are defined by other standard are reused.

#### Superclass

*Other\_Standard\_Model* (see 6.3.7)

Attribute None	Datatype	Multiplicity	Description		
<b>Reference</b> SLA_for_ registry	<b>Class</b> Registry	<b>Multiplicity</b> 0..1	<b>Description</b> When the instance of Document metaclass (see 6.3.4) is an SLA document that designates the registry system. This reference specialize the "contained_by" reference which is inherited from the superclass.	<b>Inverse</b> SLA_ document	<b>Precedence</b> No
promotion_for_ registry	Registry	0..1	When the instance of Document metaclass (see 6.3.4) is a promotion document. The purpose of a promotion document is to provide an introduction to the registry system, such as a brochure, a booklet and a white paper and so on. This reference specialize the "contained_by" reference which is inherited from the superclass.	promotion_ document	No
manual_for_ registry	Registry	0..1	When the instance of Document metaclass (see 6.3.4) is a manual or guide, the document is a set of instructions for operation of the registry system. This reference specialize the "contained_by" reference which is inherited from the superclass.	manual_ document	No
specification_ for_interface	Interface	0..1	When the instance of Document metaclass (see 6.3.4) is an interface specification for access to the registry, this item references the target interface of the registry.	specification_ document	No

### Constraints

Four associations which Document metaclass (see 6.3.4) has have constraint of XOR.

**NOTE** Registry Summary limits the type of the document relevant to Registry metaclass (see 6.3.8) to three, the promotion documents, the manual documents, and the SLA documents. It is because Registry Summary aims at expressing the information showing registry by the smallest possible set. Furthermore, these three types of document is a minimum document which the user of registry needs.

### 6.3.5 Interface metaclass

Interface is a metaclass each instance of which represents data about the access interface of the registry. The purpose of this metaclass is to represent the interface for external access to the registry described by the Registry Summary. Moreover, when Registry Summary represents a virtual registry, the interface which each registry which constitutes the virtual registry also can be represented.

### Superclass

Model\_Element (from MFI-10)

Attribute	Datatype	Multiplicity	Description
identification	string	1..1	The unique character string which identifies this interface.
name	string	1..1	The name of this interface.
description	text	0..1	The descriptive information about this interface.
URL	string	0..1	The URL for accessing the interface which this class represents. In the interface which cannot be expressed by URL, there is no value to this item. In that case, it defines by related specification documents.
version	string	0..1	The version of this interface.

Reference component	Class	Multiplicity	Description	Inverse	Precedence
	Registry_Component	1..1	When the instance of this metaclass represents the interface of the registry component which constitutes a virtual registry, this association designates the registry component with an interface.	application_for_interface	No
specification_document	Document	1..*	Relation to the instance of the Document metaclass showing the technical specification document which contains an explanation of and information on usage of the interface.	specification_for_interface	Yes

NOTE In this metaclass, the inherited reference from Model\_Element (from MFI-10) does not refer to any Model (from MFI-10) and subclasses of Model (from MFI-10).

### 6.3.6 Organization metaclass

Organization is a metaclass each instance of which represents data about the provider of the registry. When this metaclass is implemented, the models which are defined by other standard are reused.

### Superclass

Other\_Standard\_Model (see 6.3.7)

Attribute	Datatype	Multiplicity	Description
None			

Reference component	Class	Multiplicity	Description	Inverse	Precedence
contact_information	Contact	0..1	The Point of Contact (POC) of this organization.	organization	Yes



provided_registry	Registry	1..*	The registries which this organization provides. This reference specialize the "contained_by" reference which is inherited from the superclass.	provider	No
-------------------	----------	------	---	----------	----

### 6.3.7 Other\_Standard\_Model metaclass

Other\_Standard\_Model is an abstract metaclass for quoting to the Registry Summary the model defined by other standard specification. This metaclass is superclass for Classification metaclass (see 6.3.1), Contact metaclass (see 6.3.2), Document metaclass (see 6.3.4) and Organization metaclass (see 6.3.6). The how to use this metaclass is shown in Annex B.

#### Superclass

Model\_Element (from MFI-10)

Attribute	Datatype	Multiplicity	Description	Inverse	Precedence
reference_standard_identifier	string	1..1	The identification is assigned to the standard containing the model to be used.		
model_name	string	1..1	The name of the model used as a concrete class defined in the associated reference standard.		
Reference contents	Class Contents	1..1	This reference is relation from metaclasses of Registry_Summary Package to class defined by other standard. (see Annex B)	-	Yes

### 6.3.8 Registry metaclass

Registry is a metaclass each instance of which represents data about the particular registry instance. This metaclass is a core of the Registry Summary, and when the Registry Summary represents a particular registry, it becomes one instance.

#### Superclass

Model (from MFI-10)

Attribute	Datatype	Multiplicity	Description
identification	string	1..1	The unique character string which identifies this registry.
name	string	1..1	The name of this registry
contain_model	Model (from MFI-10)	0..*	The list of the model which this registry stores. The Model (from MFI-10) has three references, but the Registry Summary use only "described_by" reference.
description	string	0..1	The descriptive information about this registry.
version	string	0..1	The version showing the version of this registry.
standard	boolean	0..1	The standard means whether registry is based on a certain standard.
active	boolean	1..1	The state of operation of this registry.

Reference category	Class Classification	Multiplicity 0..*	Description The relation to the instance showing the Classification metaclass (see 6.3.1) to which the instance registry belongs. This reference specialize the "contains" reference which is inherited from the superclass.	Inverse registry	Precedence Yes
constituent	Registry_ Component	1..*	The relation to the instance of Registry_Component metaclass (see 6.3.9) showing the actual condition of the particular registry which Registry Summary represents. When the Registry Summary represents a virtual registry, in cooperation with the instance of Registry_Component metaclass (see 6.3.9), it designates the registry of each component. This reference specialize the "contains" reference which is inherited from the superclass.	registry	Yes
primary_ contact	Contact	0..1	The relation to the instance showing the main Points of Contact (POC) for this registry as concrete Contact metaclass (see 6.3.2). This reference specialize the "contains" reference which is inherited from the superclass.	registry	Yes
provider	Organization	1..1	The relation to the instance showing the organization which provides registry of concrete Organization metaclass (see 6.3.6). This reference specialize the "contains" reference which is inherited from the superclass.	provided_registry	Yes

utilization	Related_Registry	0..*	The relation to the instance of Related_Registry metaclass (see 6.3.10) which indicates a registry which is related by business domain although it is not a component of a virtual registry. This reference specialize the "contains" reference which is inherited from the superclass.	related_registry	Yes
SLA_document	Document	0..*	The relation to the instance of concrete Document metaclass (see 6.3.4) showing the document which indicates the SLA of this registry. This reference specialize the "contains" reference which is inherited from the superclass.	SLA_for_registry	Yes
promotion_document	Document	0..*	The relation to the instance of concrete Document metaclass (see 6.3.4) showing the document which provides promotion information for this registry. This reference specialize the "contains" reference which is inherited from the superclass.	promotion_for_registry	Yes
manual_document	Document	0..*	The relation to the instance of manual documents, such as how to uses the particular registry, of concrete Document metaclass (see 6.3.4). This reference specialize the "contains" reference which is inherited from the superclass.	manual_for_registry	Yes

### 6.3.9 Registry\_Component metaclass

Registry\_Component is a metaclass which shows each component registry, when the aggregate Registry in which two or more Registries cooperate and constitute is shown. However, when Registry Summary represents a single registry, only one instance of this class exists.

#### Superclass

Model\_Element (from MFI-10)

Attribute	Datatype	Multiplicity	Description		
identification	string	1..1	The unique character string which identifies this component registry.		
name	string	1..1	The name of this component registry.		
description	string	0..1	The descriptive information about this component registry.		
platform_name	string	0..1	The name of the platform where this component registry works.		
version	string	0..1	The character string indicating the version of this component registry.		
Reference	Class	Multiplicity	Description	Inverse	Precedence
application_interface	Interface	0..*	The interfaces for accessing this registry component.	component	Yes
registry	Registry	1..1	The registry which consists of this registry component. This reference specialize the "contained_by" reference which is inherited from the superclass.	constituent	No

### 6.3.10 Related\_Registry metaclass

Related\_Registry is metaclass shows the registry system which the other organization provides and which cooperates with other registries. The instance of this class represents the registry which cooperates in certain aspects; however it is not close a relationship as the component registry which constitutes a virtual registry.

#### Superclass

Model\_Element (from MFI-10)

Attribute	Datatype	Multiplicity	Description
identification	string	1..1	The unique character string which identifies this related registry in this Registry Summary data.
provider_identification	string	1..1	The identification of instance of Organization which provides the related registry.
provider_name	string	1..1	The name of instance of concrete Organization class which provides the related registry.
registry_identification	string	1..1	The identifier of the registry defined in other Registry Summary data showing the related registry.
registry_name	string	1..1	The name of the related registry.
registry_version	string	0..1	The version of the related registry.

Reference	Class	Multiplicity	Description	Inverse	Precedence
related_ registry	Registry	1..1	The registry which is referring to this registry. This reference specialize the "contained_by" reference which is inherited from the superclass.	utilization	No

### 6.3.11 Registry\_Summary\_Language

Registry\_Summary\_Language is a metaclass each instance of which represents a particular language or notation is used to express registry summary models.

#### Superclass

Modelling\_Language (from MFI-10)

Attribute	Datatype	Multiplicity	Description
None			

Reference	Class	Multiplicity	Description	Inverse	Precedence
None					

## Annex A (informative)

### Types of a registry system for Registry Summary

#### A.1 Introduction

The Registry Summary information is based on what the registry itself creates and publishes. However, in order to realize wide range Registry Interoperability which is the purpose of Registry Summary, many registries need to create and release Registry Summary information.

On the other hand, the registry cooperation function of the same kind is defined as the existing registry technical specifications, and it can be considered more than one registry of the same kind could operate as one virtual registry. The range of Registry Interoperability can be extended by representing the virtual registry by the one Registry Summary information.

This annex illustrates which registry should carry out Registry Summary information creation according to the cooperation situation of registry also including virtual registry.

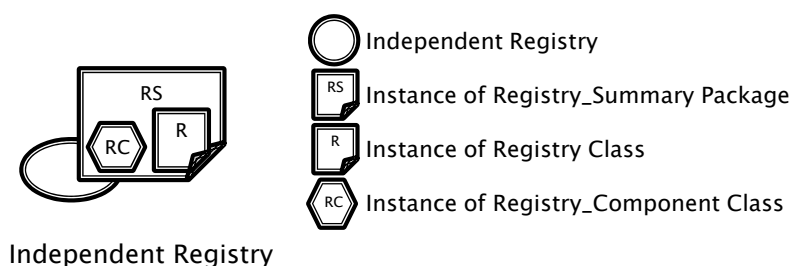
#### A.2 Types of a registry system for Registry Summary

##### A.2.1 General

This clause illustrates which registry creates and publishes Registry Summary information for every type of cooperation of the registry group, which are independent, replicated, and federated.

##### A.2.2 Independent Registry

An independent registry without cooperation with another registry is called an "Independent Registry" regardless of existing technical specifications or original specification. In this case, the registry itself will create and publish Registry Summary information, and the registry becomes the structure whereby the instance of one Registry\_Component metaclass (see 6.3.9) relates to the instance of a Registry class (Figure A.1).



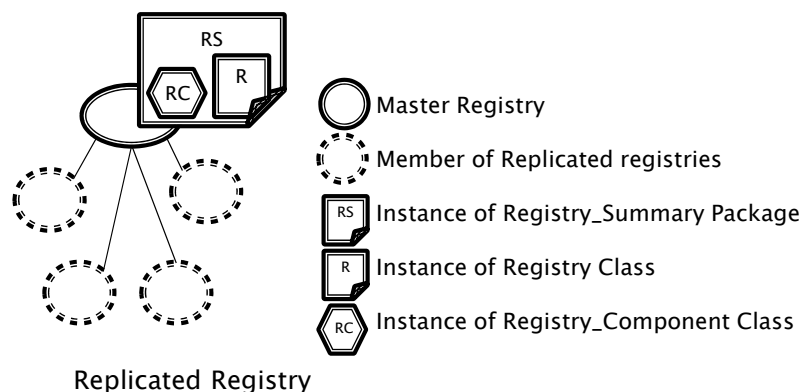
**Figure A.1 — Conceptual image of Independent Registry**

##### A.2.3 Replicated Registry

"Replicated Registry" is a form in which one in a group of registries has collected the contents of other registries as a master registry. The contents of each registry which has participated in the group are copied to a master registry; the registry system refers to the master registry. A registry user may refer or have access to all the contents in a group if business rules allow (Figure A.2). The caution in a replication registry is that

unless contents are updated dynamically when the registry system is updated, contents of the master registry may be out of date.

In this case, since the registry user can refer to all the contents in a group by accessing master registry. Since the master registry creates and publishes Registry Summary information for all the registries in the group, becomes equivalent to an Independent Registry. Therefore, in this case, the instance of a Registry class and the instance of a Registry\_Component metaclass (see 6.3.9) have the same relationship as Independent Registry for master registry.



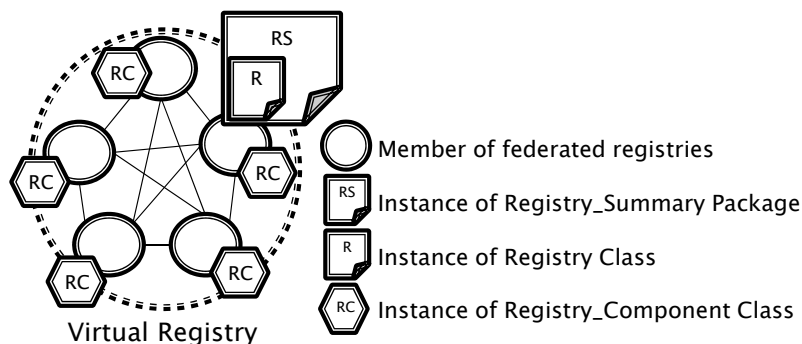
**Figure A.2 — Conceptual image of Replicated Registry**

#### A.2.4 Federated Registry

The third type, Federated Registry, is a group of registries constituted from two or more registries, which are likely from the same domain. The contents of each federated registry are not copied but are stored separately by each registry (i.e. there is no master registry in this type).

In the case of the inquiry from a registry user, when there are no contents of relevance in the registry itself, it asks other registry in a group from the registry, and returns the result to a registry user. It is the cooperation which is visible as the registry which received the inquiry has all the contents in a group in this cooperation Type.

In this case, the instance of a Registry metaclass (see 6.3.8) can be defined by making the whole group into one virtual registry (Figure A.3). Each registry can describe the whole federated group by single Registry Summary information by communicating the instance of a Registry\_Component metaclass (see 6.3.9) (i.e. the Registry\_Component for each component registry in the federation).



**Figure A.3 — Conceptual image of Federated Registry**

## Annex B (informative)

### Example of using models of other standard in Registry Summary

#### B.1 Introduction

This part of ISO/IEC 19763 defines two abstract metaclasses (Other\_Standard\_Model abstract metaclass (see 6.3.7) and Contents abstract metaclass (see 6.3.3)). And four metaclasses (Classification class (see 6.3.1), Contact class (see 6.3.2), Document class (see 6.3.4) and Organization class (see 6.3.6)) is subclass of Other\_Standard\_Model metaclass (see 6.3.7). The models with the same meaning as these four metaclasses are already defined in other standards. Specifying a new model with the same meaning in this part of ISO/IEC 19763 would cause confusion. Therefore, in this part of ISO/IEC 19763, the existing models are adopted for these classes. When using Registry Summary as described in this part, it is necessary to relate the existing models with these classes first. Annex B shows an example which relates the concrete model to these classes of Registry Summary.

#### B.2 Adaptation method of model of other standard

##### B.2.1 General

In order to use the existing model in a RegistrySummary model, the Contents metaclass (see 6.3.3) is used as glue. In using the class defined by other standards, it prepares the class which carries out the multiple inheritances of this class and Contents metaclass (see 6.3.3).

Figure B.1 shows an example which uses Organization class of ISO/IEC 11179-3:2013 for Organization metaclass (see 6.3.6) of Registry Summary. In this example, the instance of the Organization metaclass (see 6.3.6) in a Registry\_Summary package has all the attributes of the Organization class of MDR-3, and also has relations with other classes in a RegistrySummary package.

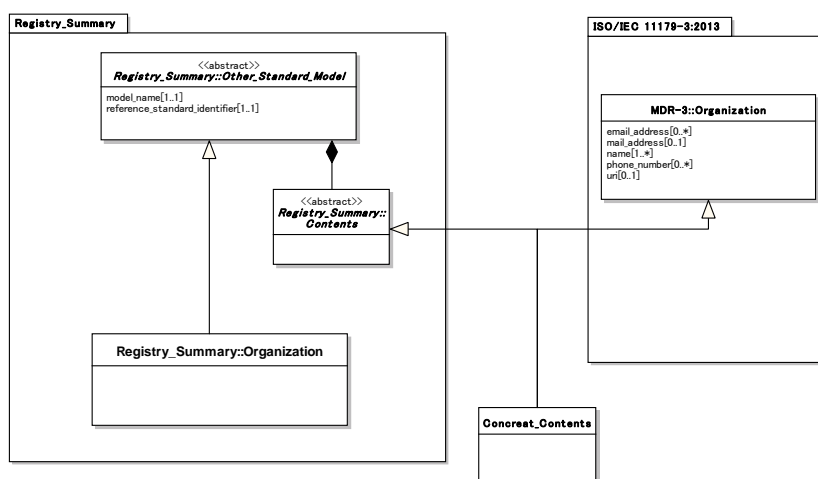


Figure B.1 — Example of Adopting Organization metaclass

The Classification metaclass (see 6.3.1), the Contact metaclass (see 6.3.2) and the Document metaclass (see 6.3.4) use the class defined by other standard by the same method.



### B.2.2 Example of using ISO/IEC 11179-3:2013 classes

This chapter shows a mapping table when using the class defined by ISO/IEC 11179-3:2013 (MDR-3) as extension of four metaclasses in the RegistrySummary model.

NOTE Refer to ISO/IEC 11179-3:2013 for the definition of the classes of MDR-3 quoted in this Annex.

**Table B.1 — Mapping table of MDR-3 and RegistrySummary**

Metaclass of Registry Summary	Class of MDR-3
Classification metaclass	Classification class
Contact metaclass	Contact class
Document metaclass	Reference_Document class
Organization metaclass	Organization class

### B.2.3 Example of using UN/CEFACT Core Component Library classes

This chapter shows a mapping table when using the class defined by UN/CEFACT Core Component Library 11A (CCL 11A) as extension of four metaclasses in the RegistrySummary model.

NOTE Refer to UN/CEFACT Core Component Library 11A for the definition of the classes of CCL 11A quoted in this Annex.

**Table B.2 — Mapping table of CCL 11A and RegistrySummary**

Metaclass of Registry Summary	Class of CCL 11A
Classification metaclass	Node_ Classification. Details
Contact metaclass	Contact. Details
Document metaclass	Universal_ Document. Details
Organization metaclass	Organization. Details

## Annex C (informative)

### Example of XML Schema for Registry Summary

#### C.1 Introduction

A Registry Summary is generated by a specific registry and collected by a RoR. It's necessary to give the form in which data of Registry Summary can be exchanged. In this annex, we select XML as an example of the form in which a Registry Summary could be exchanged with another registry. In addition, we show the XML Schema that would publish the Registry Summary as an XML document.

#### C.2 XML Schema definition

##### C.2.1 General

There are several ways to create an XML Schema from a UML class diagram. In this example, a class is defined as a ComplexType in XML Schema to be used as the type attribute for a corresponding element. It is possible to combine several elements in this way to define XML documents to support various scenarios.

NOTE 1 The "tns:" is used for the prefix of the name space of a RegistrySummary package. However, this is a tentative prefix for this example.

NOTE 2 This example uses "text" data type. But W3C XML Schema specification has no "text" data type. If the text data type is required when using this example, please define in the implementation.

##### C.2.2 ComplexType definition

###### C.2.2.1 Registry

RegistryType is an example which defines Registry metaclass (see 6.3.8) as ComplexType in XML Schema.

```
<complexType name="Registry">
  <sequence>
    <element name="active" type="boolean" minOccurs="1" maxOccurs="1"/>
    <element name="description" type="tns:text" minOccurs="0" maxOccurs="1"/>
    <element name="contain_model" type="tns:Model" minOccurs="0" maxOccurs="unbounded"/>
    <element name="identification" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="standatd" type="boolean" minOccurs="0" maxOccurs="1"/>
    <element name="version" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="category" type="tns:Classification" minOccurs="0" maxOccurs="unbounded"/>
    <element name="constituent" type="tns:Registry_Component" minOccurs="1" maxOccurs="unbounded"/>
    <element name="primary_contact" type="tns:Contact" minOccurs="0" maxOccurs="1"/>
    <element name="provider" type="tns:Organization" minOccurs="1" maxOccurs="1"/>
    <element name="utilization" type="tns:Related_Registry" minOccurs="0" maxOccurs="unbounded"/>
    <element name="SLA_document" type="tns:Document" minOccurs="0" maxOccurs="unbounded"/>
    <element name="promotion_document" type="tns:Document" minOccurs="0" maxOccurs="unbounded"/>
    <element name="manual_document" type="tns:Document" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

### C.2.2.2 Related\_Registry

Related\_Registry is an example which defines Related\_Registry metaclass (see 6.3.10) as ComplexType in XML Schema.

```
<complexType name="Related_Registry">
  <sequence>
    <element name="identification" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="provider_identification" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="provider_name" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="registry_identification" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="registry_name" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="registry_version" type="string" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

### C.2.2.3 Registry\_Component

Registry\_Component is an example which defines Registry\_Component metaclass (see 6.3.9) as ComplexType in XML Schema.

```
<complexType name="Registry_Component">
  <sequence>
    <element name="description" type="tns:text" minOccurs="0" maxOccurs="1"/>
    <element name="identification" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="platform_name" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="version" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="application_interface" type="tns:Interface" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

### C.2.2.4 Interface

Interface is an example which defines Interface class (see 6.3.5) as ComplexType in XML Schema.

```
<complexType name="Interface">
  <sequence>
    <element name="description" type="tns:text" minOccurs="1" maxOccurs="1"/>
    <element name="identification" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="URL" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="version" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="specification_document" type="tns:Document" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

### C.2.2.5 Complex type for Abstract metaclasses

#### C.2.2.5.1 Other\_Standard\_Model

Other\_standard\_Model is an example which defines Other\_Standard\_Model abstract metaclass (see 6.3.7) as ComplexType in XML Schema.

```
<complexType name="Other_Standard_Model" abstract="true">
  <sequence>
    <element name="model_name" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="reference_stadard_idenifier" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="contents" type="tns:Contets"/>
  </sequence>
</complexType>
```

#### C.2.2.5.2 Contents

Contents is an example which defines Contents abstract metaclass (see 6.3.3) as ComplexType in XML Schema.

```
<complexType name="Contets" abstract="true">
  <sequence>
    <any namespace="##any" processContents="lax" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

### C.2.2.6 Complex type for subclasses of Other\_Standard\_Model metaclass

#### C.2.2.6.1 Classification

Classification is an example which defines Classification metaclass (see 6.3.1) as ComplexType in XML Schema.

```
<complexType name="Classification">
  <complexContent>
    <extension base="tns:Other_Standard_Model"/>
  </complexContent>
</complexType>
```

#### C.2.2.6.2 Contact

Contact is an example which defines Contact metaclass (see 6.3.2) as ComplexType in XML Schema.

```
<complexType name="Contact">
  <complexContent>
    <extension base="tns:Other_Standard_Model"/>
  </complexContent>
</complexType>
```

**C.2.2.6.3 Document**

Document is an example which defines Document metaclass (see 6.3.4) as ComplexType in XML Schema.

```
<complexType name="Document">
  <complexContent>
    <extension base="tns:Other_Standard_Model"/>
  </complexContent>
</complexType>
```

**C.2.2.6.4 Organization**

Organization is an example which defines Organization metaclass (see 6.3.6) as ComplexType in XML Schema.

```
<complexType name="Organization">
  <complexContent>
    <extension base="tns:Other_Standard_Model">
      <sequence>
        <element name="contact_information" type="tns:Contact" minOccurs="0" maxOccurs="1"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

**C.2.2.7 Model (from MFI-10)**

Model (from MFI-10) is an example which defines Model metaclass (from MFI-10) as ComplexType in XML Schema.

```
<complexType name="Model">
  <sequence>
    <element name="identifier" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="described_by" type="tns:Modelling_Language" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

**C.2.2.8 Modelling\_Language (from MFI-10)**

Modelling\_Language (from MFI-10) is an example which defines Modelling\_Language metaclass (from MFI-10) as ComplexType in XML Schema.

```
<complexType name="Modelling_Language">
  <sequence>
    <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

### C.2.3 Sample element definition

This example shows a "root" element based on Registry (see C.2.2.1).

```
<element name="root" type="Registry"/>
```

## Annex D (informative)

### Example of indicator of Registry Summary

#### D.1 Introduction

ISO/IEC 19763 Part 6 specifies that an RoR collect Registry Summaries from other registry systems, but does not specify a normative way of doing so. This Annex provides one possible way as an example.

#### D.2 Location of Registry Summary in META tag

Some registries may be accessible via a web site using http or https protocols, but other will have restricted access for security reasons. Even if access to the registry itself is restricted, it may be possible to make the Registry Summary information more freely available. A META tag on a web page could be used to identify it as a Registry Summary.

```
<!DOCUMENT HTML PUBLIC "-//W3C//DTD HTML .....>
:
<head>
:
  <metaname="registrysummary" content=" http://somedomain/some-location-path/registrysummary.xml">
:
</head>
```

An RoR which collects registry summaries can access a web page, and with this META tag, the RoR can know the place from which it can obtain a Registry Summary. Then, RoR can obtain a Registry Summary from the location designated in the META tag.

