ISO/IEC JTC 1/SC 32/WG 2 N 1201

Date: 2008-11-18

ISO/IEC WD 19763-7

ISO/IEC JTC 1/SC 32/WG 2

Secretariat: ANSI

Information technology — Metamodel Framework for Interoperability (MFI) — Part 7: Metamodel for service registration

Document type: International Standard
Document subtype:
Document stage: (20) Preparatory
Document language: E

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 19763-7 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: Reference model

Part 2: Core model

Part 3: Metamodel for ontology registration

Part 4: Metamodel for model mapping

Part 5: Metamodel for process model registration

Part6: Registration Process

Part 7: Metamodel for service registration

Part 8: Metamodel for role and goal registration

Part 9: Registry of Registries

TR: Using RGPS for on demand model selection

Introduction

Due to the spread of e-Business and e-Commerce over the Internet, the effective interchange of business transactions or other related information across countries and cultures is an important concern for people in both the IT industry and other non-IT industries.

With the rapid development of SOA (Service Oriented Architecture) and SOC (Service Oriented Computing), more and more computing resources are presented in the form of Web services. Meanwhile, business integration based on Web services is becoming a popular application development method. Web service is a kind of Web based application, which encapsulates certain computing module and is designed to support interoperable machine-to-machine interaction over a network.

In Web service registration and management, UDDI is a widely applied specification, which provides basic support for publishing and discovering Web services within and across enterprises. Keyword matching is the basic service discovery method in UDDI, thus the discovery results will be inevitably inaccurate, and the discovery process will be time-consuming. When business information interchange and integration becomes increasingly frequent, major work in service discovery should be processed by machine, therefore, it is necessary to semantically describe service information including functional and non-functional information, and provide corresponding registration and management mechanism.

This part of ISO/IEC 19763 intends to provide a generic framework for registering functional and nonfunctional information of services in an explicit way.

Information technology — Metamodel framework for interoperability (MFI) — Part 7: Metamodel for service registration

1 Scope

The primary purpose of the multipart standard ISO/IEC 19763 is to specify a metamodel framework for interoperability. This part of ISO/IEC 19763 specifies a metamodel for registering services that can enable users to discover appropriate services.

The metamodel that this part specifies is intended to promote interoperation between various services.

It does not specify industry categorization of services and contact information of service providers, which are specified in UDDI.

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

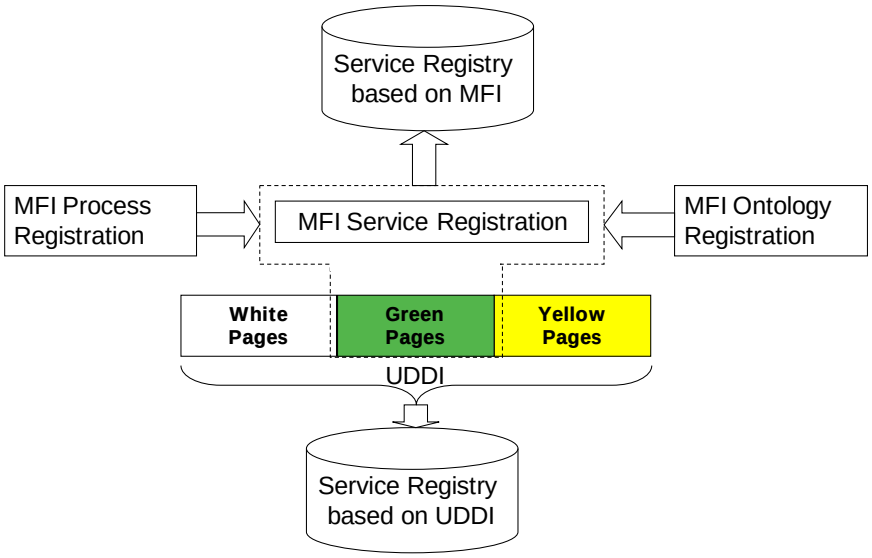

Figure 1 — Scope of MFI service registration

2 Conformance

2.1 General

An implementation claiming conformance with this part of ISO/IEC 19763 shall support the metamodel specified in 5.1, 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 implementation

A strictly conforming implementation

a) shall support the metamodel specified in 5.1;

b) shall not support any extensions to the metamodel specified in 5.1.

2.2.3 Conforming implementation

A conforming implementation

a) shall support the metamodel specified in 5.1;

b) may support extensions to the metamodel specified in 5.1 that are consistent with the metamodel specified in 5.1.

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 (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: Reference model

ISO/IEC 19763-2, Information technology – Metamodel framework for interoperability (MFI) – Part 2: Core model

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

ISO/IEC 19763-5, Information technology – Metamodel framework for interoperability (MFI) – Part 5: Metamodel for process model registration

4 Terms, definitions and abbreviated terms

4.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19763-1, ISO/IEC 19763-3 ISO/IEC 19763-5 and the following apply.

4.2 Broad terms

4.2.1
Service
a kind of Web based application, which encapsulates certain computing module and can be accessed by certain interface.

4.3 Abbreviated terms

MFI Core
ISO/IEC 19763-2, Information technology –Metamodel Framework for Interoperability – Part-2 : Core model

MFI Service registration
ISO/IEC 19763-7, Information technology – Metamodel framework for interoperability (MFI) – Part 7: Metamodel for service registration

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

MFI Process registration
ISO/IEC 19763-5, Information technology –Metamodel Framework for Interoperability (MFI) – Part-5: Metamodel for process models registration

UDDI
Universal Description, Discovery, and Integration

WSDL
Web Service Description Language

OWL-S
Web Ontology Language for Services

5 Structure of MFI Service registration

5.1 Overview of MFI Service registration

This part of MFI describes administrative information of services, including functional and nonfunctional description.

As Figure 2 shows, MFI service registration metamodel is provided to capture the common functional and nonfunctional information of various kinds of services. The nonfunctional description is depicted by **Quality_Property**, which can be used to represent the quantitative or qualitative value of service in certain aspect such as response time, cost, reliability, and so on. Each **Quality_Property** has a **Quality_Property_Type**, which should come from quality property concepts of domain ontology.

As for the functional description, since a service is an independent and modular component and it can be accessed only by interfaces, a service's functionality is embodied in two parts: **Message** and **Operation**.

There are two kinds of messages in a service: **Input_Message** and **Output_Message**. Each message has its **Message_Type**, which determines the range of values that can be assigned to the parameter. In order to add semantic information for service, **Message_Type** should come from noun concepts of domain ontology.

Operation is used to denote the execution action. Each operation has its **Operation_Type**. In order to add semantic information for service, **Operation_Type** should come from verb concepts of domain ontology. An operation has **Precondition** and **Postcondition**, which are essential to invoking and executing the service. **Precondition** is used to specify the state that should be satisfied before a service is invoked, while **Postcondition** is used to specify the state that should be satisfied after a service is invoked successfully.

Service_Description_Language is used to specify the description language that registered services expresses in.

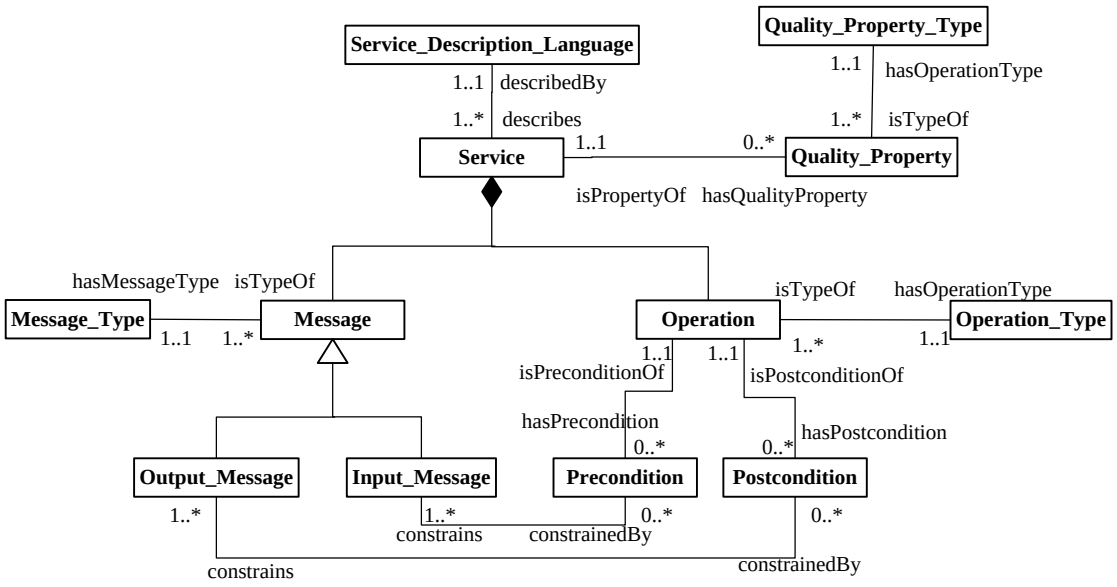

Figure 2 — Metamodel of MFI service registration

5.2 Relationship between MFI service registration and other parts in MFI

Figure 3 shows the relationship between MFI service registration and other parts in MFI. That is, **Service** and **Service_Description_Language** will inherit ModelComponent and ModelSpecification in MFI Core, respectively, **Message_Type**, **Operation_Type**, and **Quality_Property_Type** will inherit Ontology_Atomic_Construct in MFI ontology registration, service will realize process specified in MFI process registration.

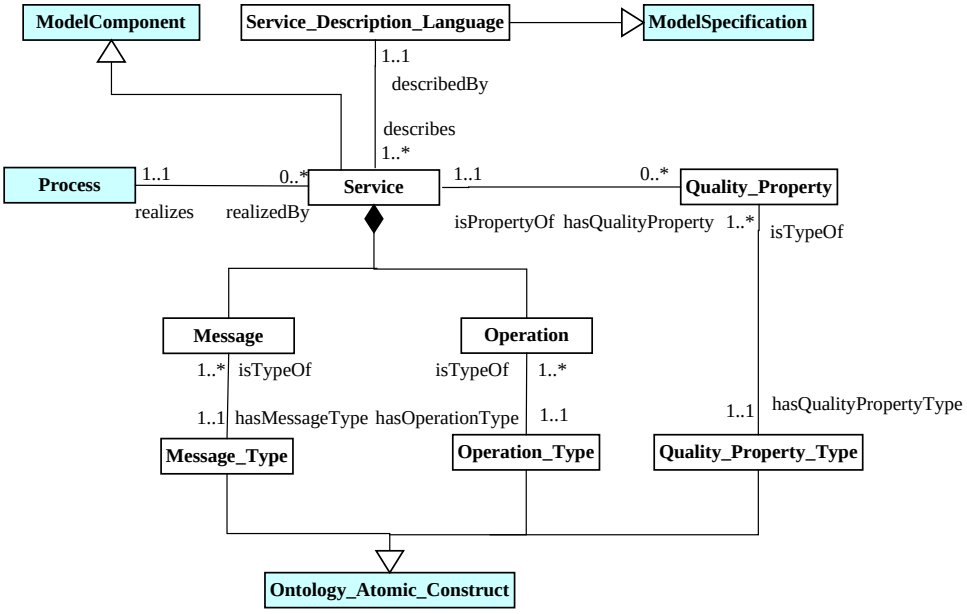

Figure 3 – Relationship between MFI service registration and other parts in MFI

5.3 MFI Service registration

5.3.1 Service

Service is a metaclass representing an independent and modular component.

Attribute	DataType	Multiplicity	Description
name	String	1..1	Name of the corresponding service
URI	String	1..1	URI where the corresponding service exists

Reference	Class	Multiplicity	Description
describedBy	Service_Description_Language	1..*	A description language that the corresponding service is described with
hasQualityProperty	Quality_Property	0..*	Nonfunctional description of the corresponding service
consistsOf	Message	1..*	Messages to be transferred or generated by the service
consistsOf	Operation	1..*	Operations that denote the execution actions of the service

Constraints

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

5.3.2 Service_Description_Language

Service_Description_Language is a metaclass representing the description language of a service.

Attribute	DataType	Multiplicity	Description
name	String	1..1	Name of the service description language. It is advisable that its value should be one of the values in column “name” of Table1 at Annex A.

Reference	Class	Multiplicity	Description
describes	Service	1..*	Services described with the Service_Description_Language

Constraints

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

5.3.3 Quality_Property

Quality_Property is a metaclass that is used to represent the quantitative or qualitative value of a service in certain non-functional aspect.

Attribute	DataType	Multiplicity	Description
name	String	1..1	Name of the Quality_Property

Reference	Class	Multiplicity	Description
isPropertyOf	Service	1..1	The service that owns this Quality_Property
hasQualityPropertyType	Quality_Property_Type	1..1	The range of the quality_property_type

Constraints

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

5.3.4 Quality_Property_Type

Quality_Property_Type is a metaclass that denotes the type of quality property, which should come from quality property concepts of domain ontology.

Reference	Class	Multiplicity	Description
isTypeOf	Quality_Property	1..*	The Quality_Property that has this corresponding Quality_Property _type

5.3.5 Message

Message is an abstract metaclass that is a superClass of Input_Message and Output_Message.

Attribute	DataType	Multiplicity	Description
name	String	1..1	Name of the corresponding message

Reference	Class	Multiplicity	Description
hasMessageType	Message_Type	1..1	The range of the message, which should come from noun concepts of domain ontology

Constraints

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

5.3.6 Operation

Operation is a metaclass that denotes the execution actions of the service.

Reference	Class	Multiplicity	Description
hasOpertaionType	Operation_Type	1..1	The type of operations, which should come from verb concepts of domain ontology
hasPrecondition	Precondition	0..*	The state that should be satisfied before a service is invoked
hasPostcondition	Postcondition	0..*	The state that should be satisfied after a service is invoked successfully.

5.3.7 Precondition

Precondition is a metaclass that specifies the state that should be satisfied before a service is invoked.

Reference	Class	Multiplicity	Description
isPreconditionOf	Operation	1..1	The operation that has the Precondition
constrains	Input_Message	1..*	The Input_Messages constrained by the Precondition

5.3.8 Postcondition

Postcondition is a metaclass that specifies the state that should be satisfied after a service is invoked successfully.

Reference	Class	Multiplicity	Description
isPostconditionOf	Operation	1..1	The Operation that has the Precondition
constrains	Output_Message	1..*	The Output_Messages constrained by the Postcondition

5.3.9 Operation_Type

Operation_Type is a metaclass that denotes the type of operations, which should come from verb concepts of domain ontology.

Reference	Class	Multiplicity	Description
isTypeOf	Operation	1..*	The operation that has this corresponding Operation_Type

5.3.10 Message_Type

Message_Type is a metaclass that denotes the type of messages, which should come from noun concepts of domain ontology.

Reference	Class	Multiplicity	Description
isTypeOf	Message	1..*	The message that has this corresponding Message_type

5.3.11 Input_Message

Input Message is a metaclass that will be transferred by the service.

SuperClass Message			
Reference	**Class**	**Multiplicity**	**Description**
constrainedBy	Precondition	0..*	The Precondition that constrains the Input_Message

5.3.12 Output_Message

Output_Message is a metaclass that will be generated by the service.

SuperClass Message			
Reference	**Class**	**Multiplicity**	**Description**
constrainedBy	Postcondition	0..*	The Postcondition that constrains the Output_Message

Annex A (informative) **List of service description languages**

It is advisable that the value of attribute “name” of “Service_Description_Language” can be one of the values in column “name” of Table 1.

Table 1 – List of Service_Description_Languages

Name	**Description**
OWL-S	A language that conforms to “OWL Web Ontology Language for Web Service”, which specifying Semantic Markup for Web Services, 2004-11-02, W3C Member Submission.
WSDL	Web Services Description Language, version 1.1, 2001-03-15, W3C Member Submission.
BPEL	Business Process Execution Language for Web Service (BPEL/BPEL4WS), 2003-05-03, Version 1.1.
UML	A language that conforms to ISO/IEC 19501 Information technology – Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2. UML Activity Diagram is the focus of this part.
Other	