



**International
Standard**

ISO/IEC 19763-7

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

**Part 7:
Metamodel for service model
registration**

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

Partie 7: Métamodèle pour l'enregistrement du modèle de service

**Second edition
2026-10**

Sample Document

get full document from standards.iteh.ai



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2026

All rights reserved.

This ISO publication is protected by copyright and is owned by ISO and/or its licensors.

The content of this ISO publication is provided under licence, not sold. Use is subject to the applicable licence terms issued by ISO, an ISO member body, or an authorized third-party distributor.

Except as required for implementation or expressly permitted by a separate licence, no part of this ISO publication may be reproduced, distributed, modified, used, or made available in any form or by any means – electronic or mechanical, including photocopying, scanning, recording, or posting on internal or external digital platforms.

Any use beyond the scope of the granted rights is prohibited and may result in legal action.

ISO copyright office
CP 401 • CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Abbreviated terms	3
5 Conformance	4
5.1 General	4
5.2 Degree of conformance	4
5.2.1 General	4
5.2.2 Strictly conforming implementation	4
5.2.3 Conforming implementation	4
5.3 Implementation Conformance Statement (ICS)	4
6 MFI Service model registration	5
6.1 Overview of MFI Service model registration	5
6.2 Associations between MFI Service model registration and other parts in MFI	6
6.3 Structure of MFI Service model registration	8
6.3.1 Service_Description_Language	8
6.3.2 Service_Model	9
6.3.3 Service	9
6.3.4 Service_Operation	11
6.3.5 Service_Type	12
6.3.6 Service_Evolution	12
6.3.7 Service_Change	13
6.3.8 Change_Impact	13
6.3.9 Change_Type	14
6.3.10 Invocation_Way	14
6.3.11 Protocol	15
6.3.12 Message_Exchange_Pattern	15
6.3.13 Message_Data_Format	15
6.3.14 Message_Type	16
6.3.15 Input_Message	16
6.3.16 Output_Message	17
6.3.17 Expression	18
6.3.18 Atomic_Expression	18
6.3.19 Composite_Expression	19
6.3.20 Composition_Type	19
6.3.21 Precondition	19
6.3.22 Postcondition	20
6.3.23 Exit_Condition	21
6.3.24 QoS_Type	21
6.3.25 QoS_Assertion	21
6.3.26 User_Tag	22
Annex A (informative) List of existing service description languages	23
Annex B (informative) Examples	24
Bibliography	52

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC have not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document 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 19763-7:2015), which has been technically revised.

The main changes are as follows:

- Additional metaclasses and enumerated datatypes have been introduced in the metamodel of MFI service model registration to support registration of heterogeneous services using multiple protocols, including *Invocation_Way*, *Message_Exchange_Pattern*, *Protocol*, and *Message_Data_Format*.
- Additional metaclasses and enumerated datatypes have been introduced in the metamodel of MFI service model registration to support registration of service evolution, including *Service_Evolution*, *Service_Change*, *Change_Type*, and *Change_Impact*.
- [Annex B](#) has been updated, providing registration examples, including an OpenAPI service registration example and a PROTO service registration example.

A list of all parts in the ISO/IEC 19763 series can be found on the ISO and IEC websites.

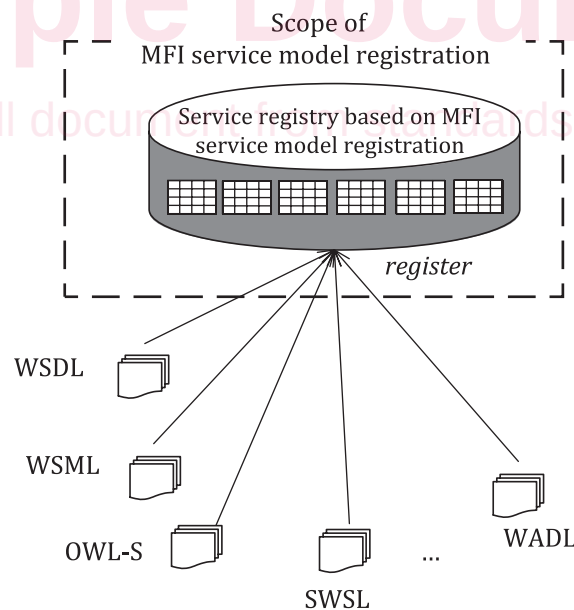
Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

With the rapid development of Service Oriented Computing (SOC), 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. A web service is a kind of Web based application that encapsulates one or more computing modules and is designed to support interoperable machine-to-machine interaction over a network.

In Web Service registration and management, ebXML RegRep is a standard defining the Service interface, protocols and information model for an integrated registry and repository, which provides basic support for publishing and discovering Web Services within and across enterprises. Nevertheless, keyword matching is the basic service discovery method in ebXML RegRep, 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 web service discovery should be processed by machine. This shift necessitates the semantic description of administrative and evolution-related metadata of services, along with the provision of corresponding registration and management mechanisms. Moreover, with the rapid growth of service networks and service ecosystems, where large numbers of heterogeneous services coexist and dynamically evolve, the demand for interoperability becomes increasingly critical. Services characterized by more diverse communication protocols, description languages, message exchange patterns, and data formats must be capable of sharing and interaction. To support this, service registration frameworks must be enhanced to ensure that services can be discovered, composed, and evolved within highly dynamic and diverse digital ecosystems.

This document intends to provide a generic framework for registering administrative and evolution information about web services in an explicit way. The relationship between MFI service model registration, the service registry, and service models expressed in different service description languages is illustrated in [Figure 1](#).



Key



service model



metadata about service

NOTE

Not every model needs to exist in a repository before registration

Figure 1 — Scope of MFI Service model registration

Sample Document

get full document from standards.iteh.ai

Information technology — Metamodel framework for interoperability (MFI) —

Part 7: Metamodel for service model registration

1 Scope

The primary purpose of the ISO/IEC 19763 series is to specify a metamodel framework for interoperability. This document specifies a metamodel for registering administrative and evolution information of models of services, facilitating interoperability through the reuse of services.

This document is only applicable to web services whose capabilities are described by some web service description language (see [Annex A](#) for examples of such languages). [Figure 1](#) shows the scope of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 19763-3, *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*

ISO/IEC 19763-8, *Information technology — Metamodel framework for interoperability (MFI) — Part 8: Metamodel for role and goal model registration*

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

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19763-3, ISO/IEC 19763-5, ISO/IEC 19763-8, ISO/IEC 19763-10 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1 service

kind of application which encapsulates one or more computing modules and can be accessed through a specified interface

3.2

service operation

execution action of a *service* (3.1)

3.3

expression

sentence which is expressed using a logical notation to specify either a condition that applies to a *service operation* (3.2) or a *QoS assertion* (3.13) that applies to a *service* (3.1)

3.4

atomic expression

logical *expression* (3.3) that has the unit granularity

3.5

composite expression

logical *expression* (3.3) that comprises multiple *atomic expressions* (3.4) and (or) other *composite expressions* (3.5) by using connectives such as conjunction, disjunction and negation

3.6

precondition

constraint that must be true when an operation is invoked

Note 1 to entry: The operation can be a process or *service operation* (3.2).

3.7

postcondition

constraint that must be true at the completion of an operation

Note 1 to entry: The operation can be a process or *service operation* (3.2).

3.8

exit condition

constraint that, if true, causes an operation to finish unsuccessfully

Note 1 to entry: The operation can be a process or *service operation* (3.2).

3.9

message type

type of the message or a set of messages that is (are) consumed or generated during the execution of a *service operation* (3.2)

3.10

input message

information contained in the message that the *service operation* (3.2) consumes for its execution

3.11

output message

information contained in the message that the *service operation* (3.2) generates after its execution

3.12

QoS type

specified non-functional property for a *service* (3.1), such as availability, response time, etc

3.13

QoS assertion

specification of one or more *QoS types* (3.12) for the *service* (3.1)

3.14

web service

kind of *service* (3.1) that is designed to support interoperable machine-to-machine interaction over a network

3.15**entity service**

web service (3.14) that bases its functional boundary on one or more related organization entities, such as customer, employee, invoice and claim

3.16**task service**

web service (3.14) with a functional boundary directly associated with a process model

3.17**utility service**

web service (3.14) that is dedicated to providing reusable, cross-cutting utility functionality, such as event logging, notification, and exception handling

3.18**service evolution**

changes in the functional or non-functional properties of a *service* (3.1) that occur throughout the lifecycle of the *service* (3.1)

Note 1 to entry: Such changes include, but are not limited to, modifications to its computing modules, interfaces, communication protocols, message exchange patterns, or data formats.

3.19**user tag**

tag annotated by an individual or organization in order to describe the *service* (3.1) according to the understanding of the creator of the tag

4 Abbreviated terms

ebXML RegRep	ebXML Registry and Repository
GraphQL	Graph Query Language
gRPC	Google Remote Procedure Call
IRI	Internationalized Resource Identifier
MFI Core and mapping	Core model and basic mapping (see ISO/IEC 19763-10)
MFI Process model registration	Metamodel for process model registration (see ISO/IEC 19763-5)
MFI Role and Goal model registration	Metamodel for role and goal model registration (see ISO/IEC 19763-8)
MFI Service model registration	Metamodel for service model registration (see ISO/IEC 19763-7)
OWL-S	Web Ontology Language for Services
QoS	Quality of Service
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
SWRL	Semantic Web Rule Language
SWSL	Semantic Web Service Language

WADL	Web Application Description Language
WSDL	Web Service Description Language
WSML	Web Service Modelling Language

5 Conformance

5.1 General

An implementation claiming conformance with this document shall support the metamodel specified in [Clause 6](#), depending on a degree of conformance as described in [5.2](#).

5.2 Degree of conformance

5.2.1 General

The distinction between 'strictly conforming' and 'conforming' implementations is necessary to address the simultaneous needs for interoperability and extensions. This document describes specifications that promote interoperability. Extensions are motivated by needs of users, vendors, institutions and industries, but are not specified by this document.

A strictly conforming implementation may be limited in usefulness but is maximally interoperable with respect to this document. A conforming implementation may be more useful, but may be less interoperable with respect to this document.

5.2.2 Strictly conforming implementation

A strictly conforming implementation:

- shall support the metamodel specified in [Clause 6](#);
- shall not use, test, access, or probe for any extension features nor extensions to the metamodel specified in [Clause 6](#).

5.2.3 Conforming implementation

A conforming implementation:

- shall support the metamodel specified in [Clause 6](#);
- as permitted by the implementation, may use, test, access, or probe for any extension features or extensions to the metamodel specified in [Clause 6](#).

NOTE 1 All strictly conforming implementations are also conforming implementations.

NOTE 2 The use of extensions to the metamodel can cause undefined behaviour.

5.3 Implementation Conformance Statement (ICS)

An implementation claiming conformance with this document shall include an Implementation Conformance Statement stating:

- whether it is a strictly conforming implementation in [5.2.2](#) or a conforming implementation in [5.2.3](#);
- what extensions, if any, are supported or used if it is a conforming implementation.

6 MFI Service model registration

6.1 Overview of MFI Service model registration

This part of MFI specifies the metamodel that can be used to register functional and non-functional information about web services (hereafter referred to simply as “services”). Examples of some service description languages that can be registered using this metamodel are listed in [Annex A](#).

[Figure 2](#) shows the metamodel for the registration of services. This metamodel allows the registration of the common functional and non-functional features of services described using a number of service description languages. Each service model, expressed using a specific service description language, may describe one or more services. Each service is described by zero or one QoS assertion. This QoS assertion is used to represent the quantitative or qualitative non-functional features of the service, such as response time, cost, reliability, etc. Each QoS assertion is defined using one and only one expression, which may be a composite expression or an atomic expression. Each QoS assertion is of one or more QoS types. A service may undergo multiple service evolutions throughout its lifecycle. Each service evolution comprises one or more service changes, with each change describing a specific modification and its corresponding impact on service consumers.

A service is an independent and modular component and it can be accessed only by interfaces. For this reason, the functional capability of a service is expressed using service operations, where each service operation denotes an execution action of the service. Each service is comprised of zero, one or more service operations. Each service operation is described with zero or one precondition, with zero or one postcondition and with zero or one exit condition. A precondition specifies a constraint that must be true when a service operation is invoked. A postcondition specifies a constraint that must be true at the completion of a service operation, and an exit condition specifies a constraint that, if true, causes an operation to finish unsuccessfully. Each precondition, each postcondition and each exit condition is defined using one and only one expression, which may be a composite expression or an atomic expression. Each service operation is also described with zero, one or more input messages and with zero, one or more output messages. Each input message specifies information that the operation needs for its execution. Each output message specifies information that the operation generates after its successful execution. Each message type provides a description of a message or a set of messages, where each of these messages is consumed or generated during the execution of a service operation. Each input message is constrained by zero, one or more preconditions and each output message is constrained by zero, one or more postconditions. Each service operation may have an invocation way, which involves the communication protocol, message exchange pattern, and message data format. Each service can be annotated by zero, one or more user tags, each of which may be created by any person using the service.

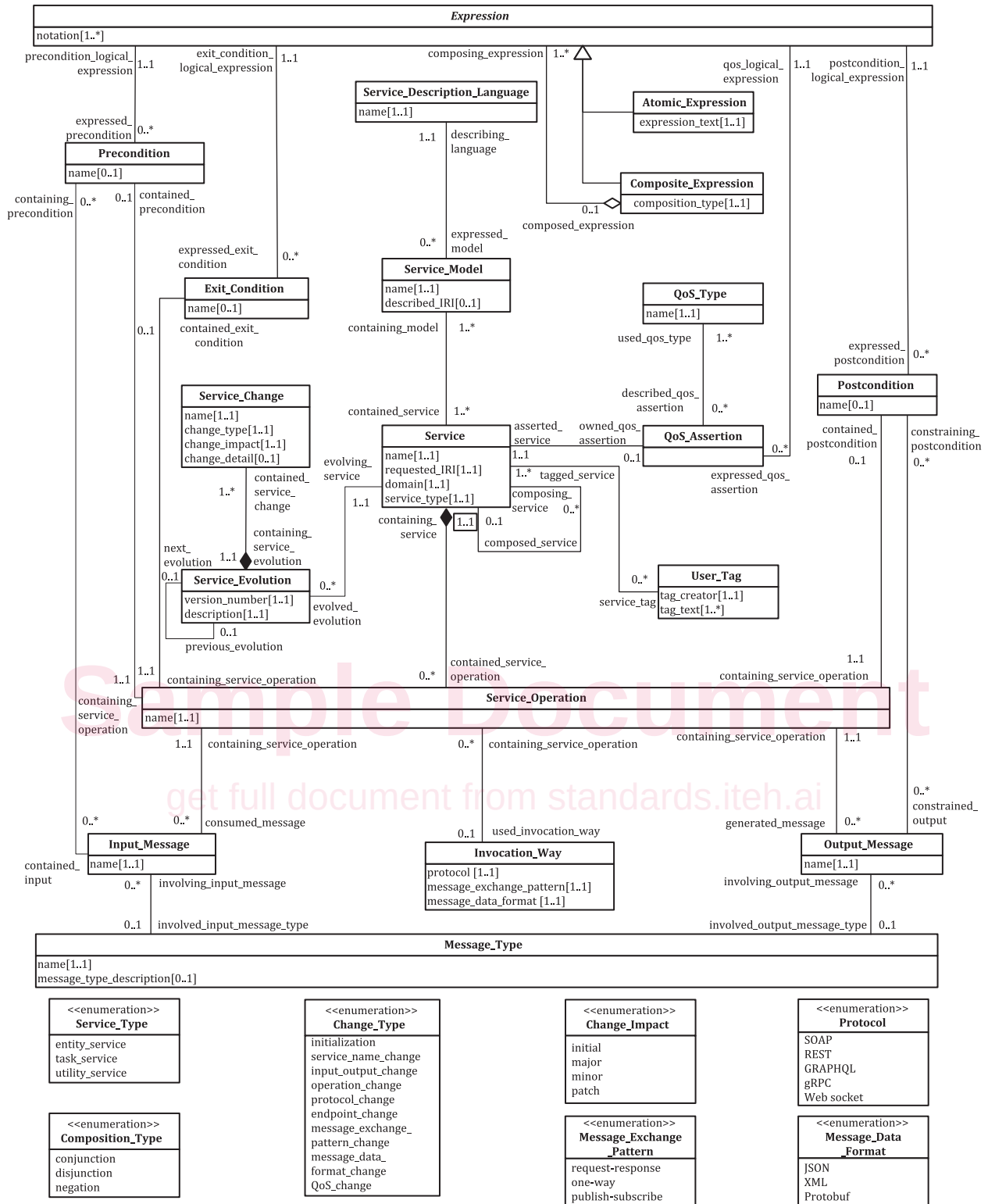
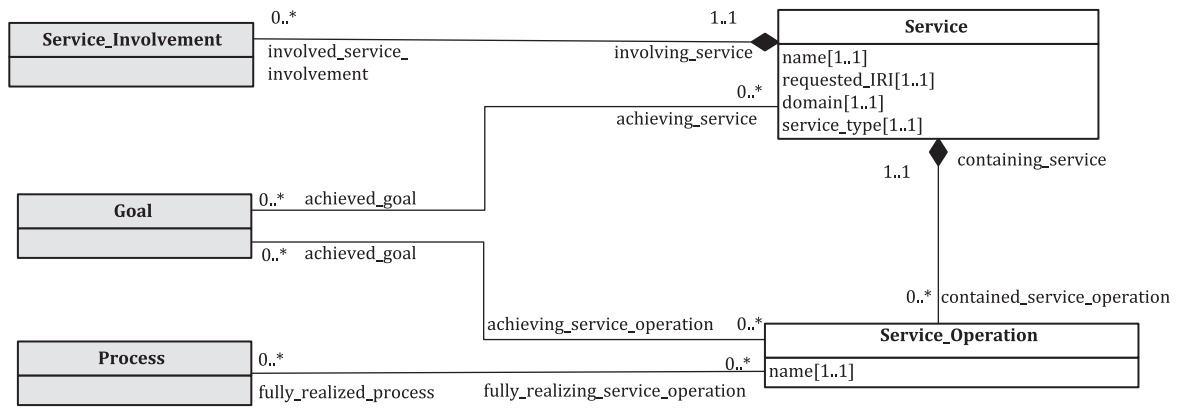


Figure 2 — Metamodel of MFI service model registration

6.2 Associations between MFI Service model registration and other parts in MFI

Figure 3 shows the associations between MFI Service model registration (this document) and MFI Role and Goal model registration and MFI Process model registration.



Key

metaclasses from other parts of MFI

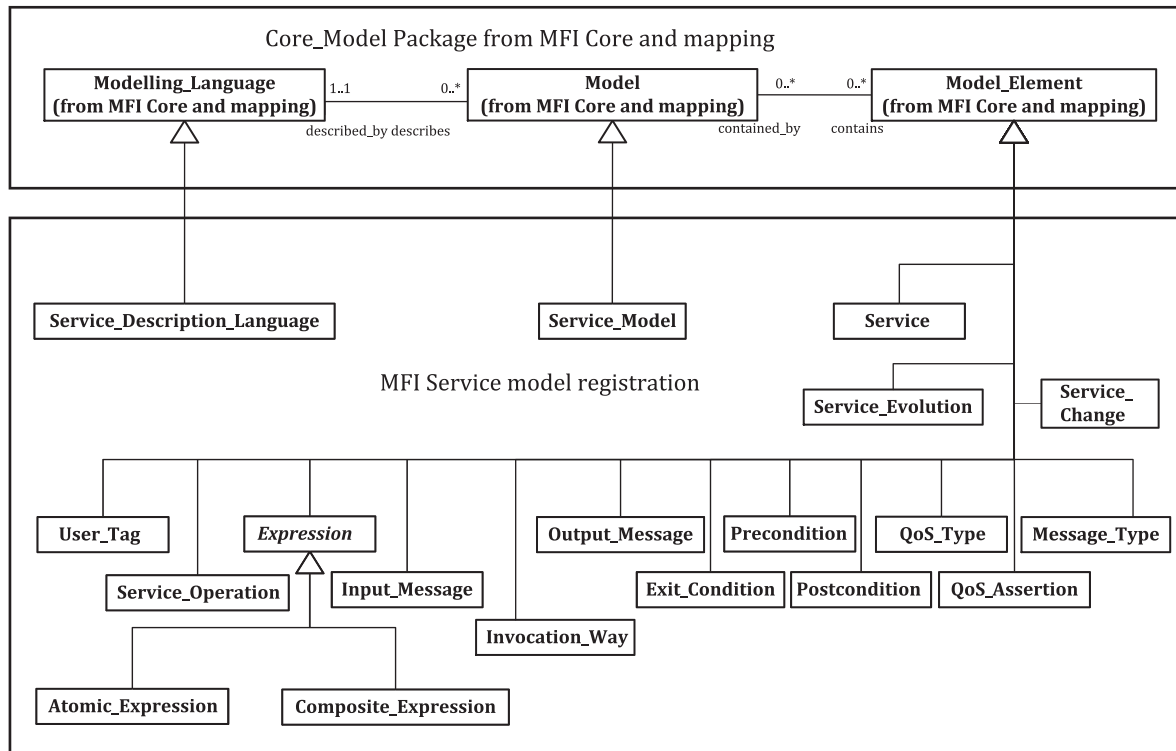
Figure 3 — Associations between MFI Service model registration and MFI Process model registration and MFI Role and Goal model registration

Each service achieves zero, one or more goals. Each goal is achieved by zero, one or more services. Each service operation achieves zero, one or more goals. Each goal is achieved by zero, one or more service operations. Each service operation can fully realize zero, one or more processes. Each process is fully realized by zero, one or more service operations. Each service involves zero, one or more service involvements, where each service involvement is the involvement of a role with a service, such as actor or beneficiary. Each service involvement indicates that a role is involved in the execution of one and only one service.

The association between the metaclasses in MFI Service model registration and the metaclasses in MFI Core and mapping is shown in [Figure 4](#).

Service_Description_Language in MFI Service model registration is a subclass of Modelling_Language in MFI Core and mapping. Service_Model in MFI Service model registration is a subclass of Model in MFI Core and mapping. All the remaining metaclasses are subclasses of Model_Element in MFI Core and mapping.

All subclasses have the associations which are inherited from their superclass. Some inherited associations are specifications of associations specialized in MFI Core and mapping. The details of specialization are defined in [6.3](#).



NOTE Metaclasses whose names are italicized are abstract metaclasses

Figure 4 — The associations between MFI Service model registration and MFI Core and mapping

6.3 Structure of MFI Service model registration

6.3.1 Service_Description_Language

Service_Description_Language is a metaclass each instance of which represents a language or a notation that is used to model a service.

Superclass

Modelling_Language (from MFI Core and mapping)

Attribute	DataType	Multiplicity	Description		
name	string	1..1	The name of this service description language		
Reference	Class	Multiplicity	Description	Inverse	Precedence
expressed_model	Service_Model	0..*	The set of service models that are described by this service description language. This reference specializes the 'describes' reference which is inherited from the superclass	describing_language	No

Constraints

[None]

6.3.2 Service_Model

Service_Model is a metaclass each instance of which represents a model that is used to model services that can be modelled.

Superclass

Model (from MFI Core and mapping)

Attribute	DataType	Multiplicity	Description		
name	string	1..1	The name of this service model		
described_IRI	string	0..1	The IRI that identifies the corresponding service model		
Reference	Class	Multiplicity	Description	Inverse	Precedence
describing_language	Service_Description_Language	1..1	The language or notation that is used to model this service. This reference specializes the 'described_by' reference which is inherited from the superclass	expressed_model	Yes
contained_service	Service	1..*	The set of services, each of which is modelled by this service model. This reference specializes the 'contains' reference which is inherited from the superclass	containing_model	Yes

Constraints

The value of attribute 'described_IRI' has to be unique in this metaclass.

6.3.3 Service

Service is a metaclass each instance of which represents a kind of web based application which encapsulates one or more computing modules and can be accessed through a specified interface.

Superclass

Model_Element (from MFI Core and mapping)

Attribute	DataType	Multiplicity	Description
name	string	1..1	The name of this service
requested_IRI	string	1..1	The IRI for invoking this service