ISO/IEC JTC 1/SC 32 N 1791

Date: 2008-09-10

REPLACES: —

ISO/IEC JTC 1/SC 32

Data Management and Interchange

Secretariat: United States of America (ANSI)
Administered by Farance Inc. on behalf of ANSI

DOCUMENT TYPE	Working Draft Text (for information or comment)
TITLE	ISO/IEC WD 19763-5 Information technology - Metamodel framework for interoperability (MFI) Part 5: Metamodel for process model registration
SOURCE	WG2 - Keqing He - project editor
PROJECT NUMBER	1.32.22.01.05.00
STATUS	text for review and comment with the NB experts at the WG2 Portugal meeting in November for progression to CD ballot
REFERENCES	
ACTION ID.	FYI
REQUESTED ACTION	
DUE DATE	
Number of Pages	26
LANGUAGE USED	English
DISTRIBUTION	P & L Members SC Chair WG Conveners and Secretaries

Dr. Timothy Schoechle, Secretary, ISO/IEC JTC 1/SC 32
Farance Inc *, 3066 Sixth Street, Boulder, CO, United States of America
Telephone: +1 303-443-5490; E-mail: Timothy@Schoechle.org
available from the JTC 1/SC 32 WebSite http://www.jtc1sc32.org/
*Farance Inc. administers the ISO/IEC JTC 1/SC 32 Secretariat on behalf of ANSI

ISO/IEC JTC 1/SC 32 WG2

ISO/IEC JTC 1/SC 32 N1791

Date: 2008-05-27

ISO/IEC WD 19763-5:2008(E)-1

ISO/IEC JTC 1/SC 32/WG 2

Secretariat: ANSI

Information Technology –Metamodel Framework for Interoperability—

Part-5 : Metamodel for process models registration

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard. Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

He Keqing
SKLSE, Wuhan University, P.R.China

Document type: International Standard
Document subtype:
Document stage: (50) WorkingDraft
Document language: E

Copyright notice

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

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

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

 © ISO/IEC 2008 – All rights reserved

Contents

ISO/IEC JTC 1/SC 32 WG2 i
ISO/IEC JTC 1/SC 32 N i
ISO/IEC WD 19763-5:2008(E)-1 i
ISO/IEC JTC 1/SC 32/WG 2 i
Information Technology –Metamodel Framework for Interoperability— i
Part-5 : Metamodel for process models registration i
Warning i
He Keqing i
Copyright notice ii
Contents iii
Foreword v
Introduction vi
1 Scope 1
2 Normative references 2
3 Definitions and abbreviated terms 2
3.1 Definitions 2
3.2 Broad terms 2
3.2.1 process model 2
3.2.2 sub-process 2
3.3 Abbreviated terms 2
3.3.1 MFI Process registration 2
3.3.2 MFI Core 3
3.3.3 MFI-3 3
3.3.4 OWL-s 3
3.3.5 PSL 3
4 Structure of MFI Process registration 3
4.1 Overview of MFI Process registration 3
4.2 Relationship between MFI Core and MFI Process registration 5
4.3 MFI Process registration 5
4.3.1 Goal 5
4.3.2 Process 6
4.3.3 Process_Modeling_Language 7
4.3.4 Atomic_Process 7
4.3.5 Composite_Process 7

© ISO/IEC 2008 – All rights reserved

4.3.6 Input 8
4.3.7 Output 8
4.3.8 Artifact 9
4.3.9 Constraint 9
4.3.10 Artifact_Constraint 9
4.3.11 Control_Constraint 10
4.3.12 Condition 10
4.3.13 Precondition 10
4.3.14 Postcondition 10
4.3.15 Control_Construct 11
4.3.16 AnyOrder 11
4.3.17 Choice 11
4.3.18 Join 11
4.3.19 Sequence 11
4.3.20 Split 12

5 Conformance 13

5.1 General 13
5.2 Degree of conformance 13
5.2.1 General 13
5.2.2 Strictly conforming implementation 13
5.2.3 Conforming implementation 13
5.3 Implementation Conformance Statement (ICS) 13

Annex A (informative) Examples of MFI Process registration 14

Annex B (informative) Collaboration between MFI members 18

Annex C (informative) List of process modelling languages 19

Figures and tables

Figure 1 – Scope of MFI Process registration 1
Figure 2 – Base Model of MFI Process registration 3
Figure 3 – Process Control Model of MFI Process registration 4
Figure 4 – Relationship between MFI Core and MFI Process registration 5
Figure 5 – Registration information of BravoAir Reservation Service 15
Figure 6 – Registration Information of top level process for manufacturing GT350 16
Figure 7 – Registration information of some sub-processes of manufacturing GT350 17
Figure 8 – Semantic interoperation based on MFI Process registration and MFI Ontology registration 18
Table 1 – List of Process_Modelling_Languages 19

 © ISO/IEC 2008 – All rights reserved

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 part of ISO/IECWD 19763 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 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**:

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

 © ISO/IEC 2008 – All rights reserved

Introduction

Today, in the EB(E-Business) or EC(E-Commerce) through the internet, the effective interchange of business transactions or other related information across countries and cultures became the first concerns for people in both IT industry and other non-IT industries.

It is increasingly popular that resources with different representations and formats are organized to perform various practices within/across enterprises and domains. Process models, involving business process, workflow, Web services and etc., are deemed as a special kind of information resources with complex structure, rich semantics and behavioral features. They are the knowledge carriers of how to fulfill business cooperation, process integration, Web service composition and other web-based applications, including data exchange, state transfer and other behavioral details.

To follow the current trends of EB or EC and utilize scattered resources in an effective manner, industrial consortia mainly contributed to standardization of domain specific process models using various representation notations and description languages for specific domains, such as BPMN (Business Process Modeling Notation) for business process and OWL-s for Web services. They are very active to standardize expressions and modeling constructs of process models in some specified domains. Especially, ISO 18629 PSL (Process Specification Language) prepared common a set of lexicon and axioms for processes, independently of the behaviors and capabilities of the processes

However, these specifications mainly concentrate on how to unify the representation of process models, rather than share their reusable pieces and semantic constraints for meaningful interoperation as well as combine them for web-based integration applications. What's more, not all of them are insufficient to handle semantics and support necessary semantic annotation. More specifically, ISO 18629 was developed to exchange and share discrete manufacturing process information within one industrial department or through several ones. Neither input/output messages nor semantics of the participants is taken into account when creating specification of relevant process information, which might hinder semantic interoperability and coordination of process models across or even beyond manufacturing domain. For this purpose, a unified metamodel is needed to standardize registration and management of heterogeneous process models on the semantic web and facilitate process integration across organizations or domains based on their domain-independent administrative information.

This part of ISO/IEC 19763 intends to provide a unified framework to register administrative structural information and meaningful semantics of various kinds of process models, based on the ISO/IEC 19763-2 Metamodel Framework for Interoperability Part-2 Core model and ISO/IEC 19763-3 Metamodel Framework for Interoperability Part-3 Metamodel for ontology registration.

© ISO/IEC 2008 – All rights reserved

Information Technology–Metamodel Framework for Interoperability –Part 5: Metamodel for process models registration

1 Scope

The primary purpose of the multipart standard ISO/IEC 19763 is to specify the framework for metamodel interoperability. This part of ISO/IEC 19763 specifies the metamodel that provides a facility to register administrative structural information and meaningful semantics of process models, including workflows, business processes, Web services, software processes, etc.

Figure1 shows the scope of this part of ISO/IEC 19763. The metamodel intends to promote the interoperation among varied kinds of process models and application systems, and finally support process models integration within/across organizations or domains. The objective can be achieved by effectively registering domain-independent administrative information of process models, such as their constitution and corresponding semantic relations.

Since this part of ISO/IEC 19763 is identified as a common and abstract facility for process model, process model registry based on this metamodel can be used to register and manage administration information of process models, which are widely applied into different domains and application systems.

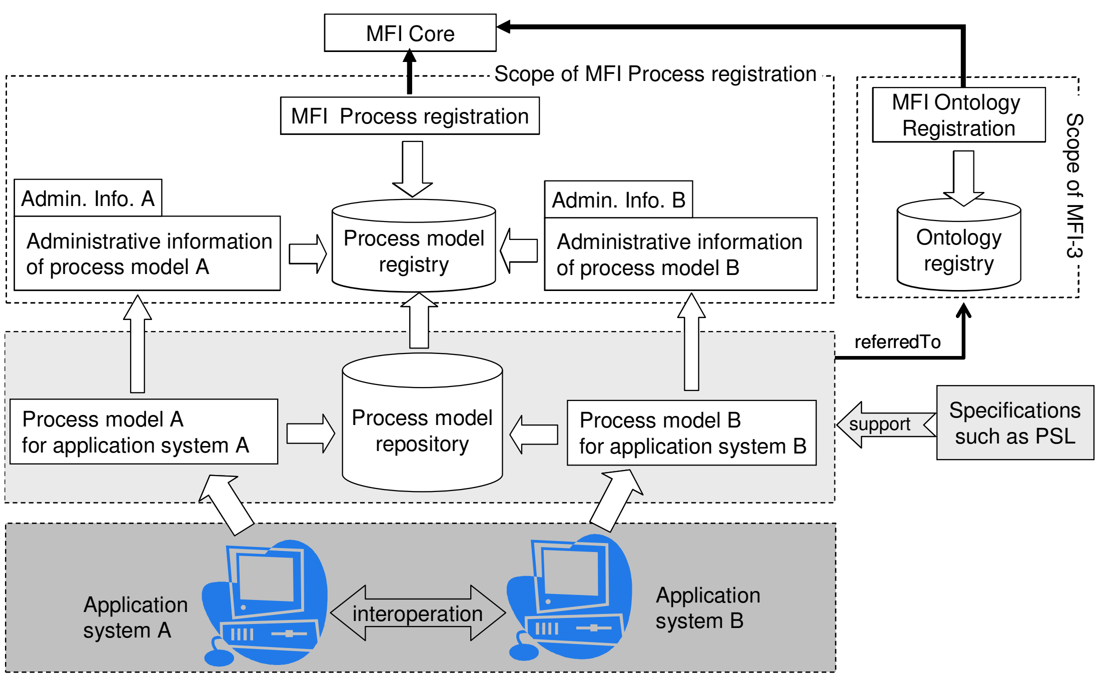

Figure 1 – Scope of MFI Process registration

Additionally, many specifications (such as PSL, OWL-s, etc.) have been developed for different aspects of process information. And as Figure 1 suggests, they can be treated as supplement facilities to enable syntax and semantic interoperability of process models adopted in application systems. Meanwhile, ontology has been widely accepted as the key to realize semantic interoperability of heterogeneous information resources and models. Since the ontology registration metamodel proposed in MFI-3 (Ed1) benefits semantic interoperation based on ontologies, the semantics

©ISO/IEC 2008 – All rights reserved

captured in registered ontologies can be used to promote semantic interoperation between those process models. That is to say, these two members of MFI family can coordinate with each other to support information sharing, exchange and integration among applications on the semantic web.

Notes that the following are outside the scope of ISO 19763-5:
- modeling notations or descriptive languages for process models;
- runtime environment or implementation platforms for the execution of process models;
- protocol and formats details for interoperation between process models.

2 Normative references

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

ISO/IEC 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 (Edition 1), Information technology – Metamodel framework for interoperability (MFI) – Part 3: Metamodel for ontology registration

3 Definitions and abbreviated terms

3.1 Definitions

The definitions provided in ISO/IEC 19763-1, ISO/IEC 19763-2 and ISO/IEC 19763-3 shall apply to this part of ISO/IEC 19763.

3.2 Broad terms

3.2.1 process model

the result of process modeling, carrying the process knowledge of how to do things in a given context.

3.2.2 sub-process

component process of composite process models. The instances of sub-processes can be process models at different level of granularity.

3.3 Abbreviated terms

3.3.1 MFI Process registration

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

© ISO/IEC 2008 – All rights reserved

3.3.2 MFI Core

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

3.3.3 MFI-3

ISO/IEC 19763-3 (Edition 1), Information technology –Metamodel Framework for Interoperability – Part-3 : Metamodel for ontology registration

3.3.4 OWL-s

OWL Web Ontology Language for Web services

3.3.5 PSL

PSL Process Specification Language

4 Structure of MFI Process registration

4.1 Overview of MFI Process registration

MFI provides basic information to facilitate registry of various process models and describes administrative information with respect to structure and inherent constraints within process models, including the corresponding constitution, sequence constraints and other semantic restrictions. In MFI Process registration, Base Model and Process Control Model are designed to record structural information and relevant constraints of process models respectively.

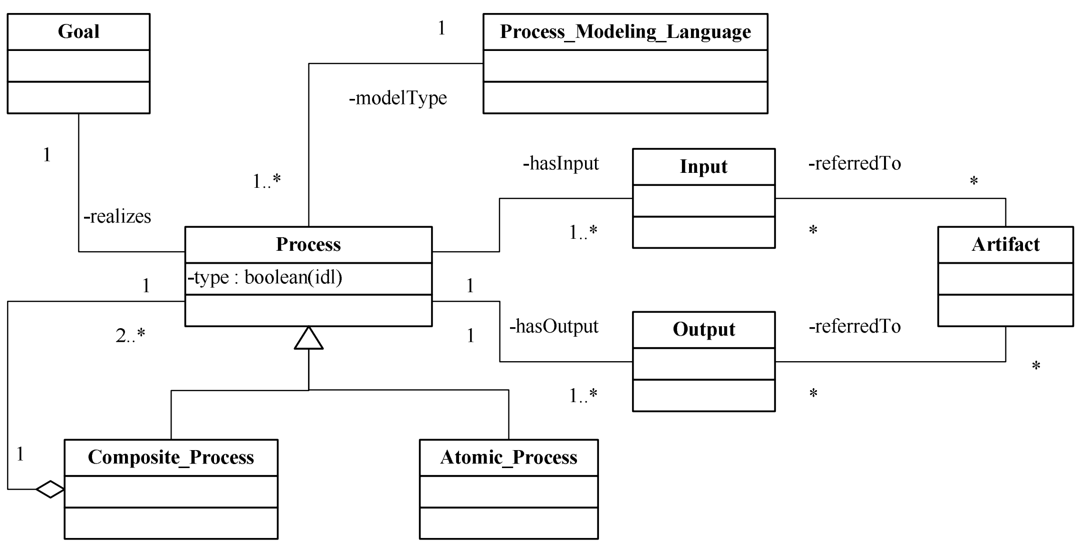

Figure 2 – Base Model of MFI Process registration

As Figure 2 suggests, Base Model is provided to capture the common structural information of various kinds of process models, such as which sub-processes are involved in the process model, and which artifacts participate in fulfilling the expected purpose. In Figure 2, **Process_Modeling_Language** is used to specify the modeling language that registered process model expresses in. **Goal** states the purpose that should be achieved by fulfilling the process model, i.e. its main functions. Generally speaking, one process model will have one or more **Input** to generate one or more **Output** as desirable products. If each input or output is taken as an information deliverer, then the involved objects or

© ISO/IEC 2008 – All rights reserved

resources can be treated as corresponding information carriers. So in Base Model, all the objects, data and resources used in the process model are the instances of **Artifact**. One referent instance might play different roles specified by different communities in different cases, so it is acceptable that referents respectively referred to the **Input** of one process and the **Output** of another process are the same.

Concerning the construction of process models, **Atomic_Process** and **Composite_Process** are proposed to denote two kinds of process model. **Atomic_Process** is the simplest process model and corresponds to one-step execution. In contrast to **Atomic_Process, Composite_Process** comprises at least two sub-processes, which can be atomic processes or other composite processes.

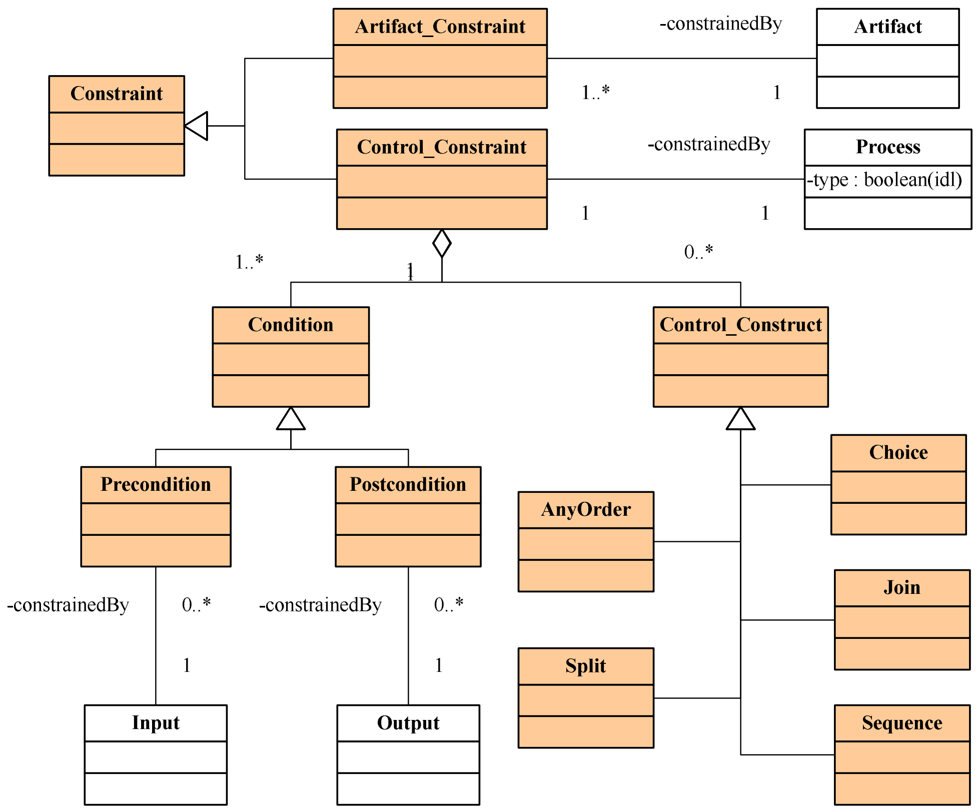

Figure 3 – Process Control Model of MFI Process registration

Process Control Model is defined in Figure 3 to record control constraints within process models. It concentrates on inherent semantics within process models, which can be classified into two categories, one is **Artifact_Constraint**, and the other is **Control_Constraint**.

In general, **Artifact_Constraint** is used to record relationship between **Artifacts**, which can be derived from knowledge base of domain or ontologies that artifacts are contained, such as equivalence relation between two concepts. It also can be used to add semantics to referred resources and connect process models semantically or semi-automatically.

Process is constrained by **Control_Constraint**. According the complexity of registered process models, two types of strategies are implies in Process Control Model. As for **Atomic_Process**, **Condition** is the only mandatory constraint. It has two subclasses, **Precondition** and **Postcondition**. **Precondition** is referred to **Input** from Base Model to

© ISO/IEC 2008 – All rights reserved

specify the information state that should be satisfied before execution, while **Postcondition** is restricted to **Output** to represent desirable outcomes when process is completed successfully. Considering **Composite_Process**, **Control_Constraint** becomes more complicated. It comprises **Condition** and **Control_Construct** because its sub-processes are connected with each other through at least one instance of **Control_Construct**. Specifically, **Control_Constructs** are generalized as **AnyOrder, Choice**, **Join**, **Split** and **Sequence** in Process Control Model. **AnyOrder** allows sub-processes to be executed in an unspecified order. **Choice** invokes one component of process model from a given collection. **Join** works when all of its components have been completed; **Sequence** means execution in order. **Split** produces at least two branches when the previous process model is executed successfully. Notice that inherent operation semantics of **Control_Construct** should be considered when specifying **Precondition** and **Postcondition** of **Composite_Process**.

4.2 Relationship between MFI Core and MFI Process registration

Some metaclasses from Base Model in MFI Process registration inherit the basic structure from MFI Core. Figure 4 shows the relationship between MFI Core and MFI Process registration. That is, **Artifact, Process** and **Process_Modeling_Language** will inherit ModelClassifier, Model Component and ModelSpecification in MFI Core repectively.

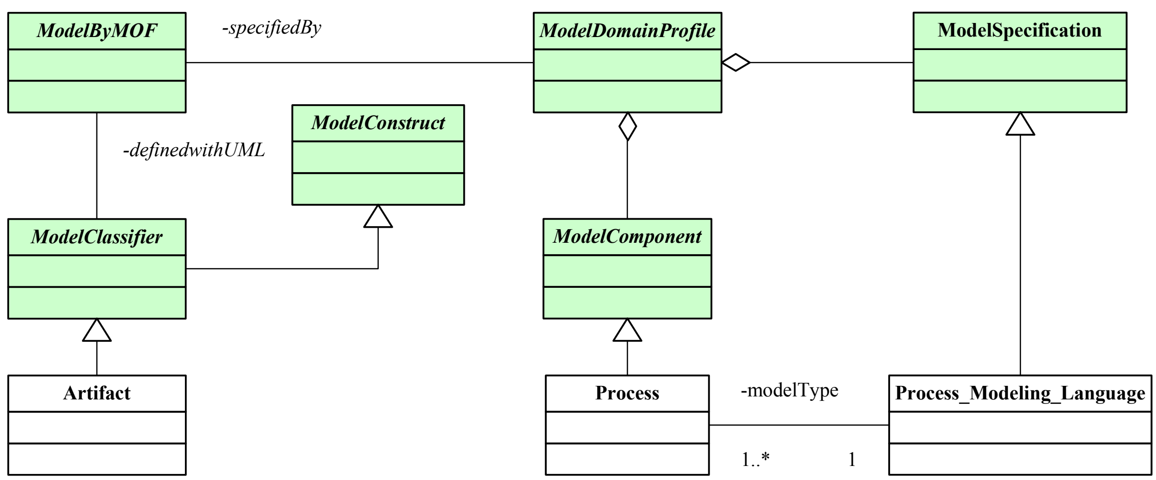

Figure 4 – Relationship between MFI Core and MFI Process registration

4.3 MFI Process registration

4.3.1 Goal

Goal is a metaclass addressing the common purpose that a process model should achieve. It implies the main function of the process model.			
Attribute	**DataType**	**Multiplicity**	**Description**
description	String	1..1	The description addressing the main objective that a process model is designed for.
Constraints			
The value of attribute "description" is not requested to be unique in this metaclass.			

© ISO/IEC 2008 – All rights reserved

4.3.2 Process

Process is an abstract metaclass, which is the superclass of Atomic_Process and Composite_Process.			
SuperClass ModelComponent			
Attribute	**DataType**	**Multiplicity**	**Description**
URI	String	1..1	URI where the corresponding process model exists
name	String	1..1	Name of the corresponding process model
type	Boolean	1..1	The type of registered process model. "0" means the process model is Composite_Process and "1" means the process model is Atomic_Process.
Reference	**Class**	**Multiplicity**	**Description**
modelType	Process_Modeling_ Language	1..1	Process modeling language that the process model is described with
realizes	Goal	1..1	The purpose that the process model should achieve
hasInput	Input	1..*	Input messages that will be transferred by the process model
hasOutput	Output	1..*	Output messages that are generated as desirable results after successful execution of the process model.
constrainedBy	Control_Constraint	1..1	The constraints that should be satisfied to perform the process model successfully.
Constraints			
The value of attribute "URI" has to be unique in this metaclass.			

 © ISO/IEC 2008 – All rights reserved

4.3.3 Process_Modeling_Language

Process_Modeling_Language is a metaclass representing the modeling language of a process model..			
SuperClass ModelSpecification			
Attribute	**DataType**	**Multiplicity**	**Description**
name	String	1..1	Name of the process modeling language. It is advisable that its value be one of the values in column "name" of Table 1 in Annex C.
Constraints			
The value of attribute "namel" has to be unique in this metaclass.			

4.3.4 Atomic_Process

Atomic_Process is a metaclass designating the simplest process model and corresponding to one-step execution. Its instance is the smallest blocks during process integration and composition.
SuperClass Process

4.3.5 Composite_Process

Composite_Process is a metaclass designating more complex process model with multiple-steps performance. It consists of at least two Processes, which can be Atomic_Process or Composite_Process.			
SuperClass Process			
Reference	**Class**	**Multiplicity**	**Description**
consistsOf	Process	2..*	Composite_Process should involve at least two sub-processes.

© ISO/IEC 2008 – All rights reserved

4.3.6 Input

Input is a metaclass specifying the messages that will be transferred by the process model			
Attribute	**DataType**	**Multiplicity**	**Description**
name	String	1..1	Name of the input message
Reference	**Class**	**Multiplicity**	**Description**
referredTo	Artifact	1..*	The data, concept or other resources that are carried by input message as participants of the process model
constrainedBy	Precondition	0..*	The condition should be satisfied before the execution of a process model.
Constraints			
The value of attribute “name” has to be unique in this metaclass.			

4.3.7 Output

Output is a metaclass specifying the messages that are generated by the process model			
Attribute	**DataType**	**Multiplicity**	**Description**
name	String	1..1	Name of the output message
Reference	**Class**	**Multiplicity**	**Description**
referredTo	Artifact	1..*	The data, concept or other kind of resources that are carried by output message as the results of the process model
constrainedBy	Postcondition	0..*	The condition should be satisfied after the execution of a process model.
Constraints			
The value of attribute “name” has to be unique in this metaclass.			

© ISO/IEC 2008 – All rights reserved

4.3.8 Artifact

Artifact is a metaclass designating the resources that participate in the process model. It is carried by Input or Output.			
SuperClass ModelClassifier			
Attribute	**DataType**	**Multiplicity**	**Description**
URI	String	1..1	URI where the corresponding artifact exists
Name	String	1..1	Name of the corresponding artifact
Reference	**Class**	**Multiplicity**	**Description**
constrainedBy	Artifact_Constraint	1..*	The constraint used to record semantic restriction adhered to the artifact
Constraints			
The value of attribute "URI" has to be unique in this metaclass.			

4.3.9 Constraint

Constraint is a metaclass designating various kinds of control constraints and semantic constraints within process models.			
Attribute	**DataType**	**Multiplicity**	**Description**
name	String	1..*	Name of the constraints within the process model

4.3.10 Artifact_Constraint

Artifact_Constraint is a metaclass designating the inherent semantic relations between objects as participants in process models. It is used for artifacts.			
SuperClass Constraint			
Reference	**Class**	**Multiplicity**	**Description**
constrains	Artifact	1..1	The participants in the execution of process model

© ISO/IEC 2008 – All rights reserved

4.3.11 Control_Constraint

Control_Constraint is a metaclass designating mandatory condition that should be satisfied for successful execution of a process model.			
SuperClass Constraint			
Reference	**Class**	**Multiplicity**	**Description**
constrainedBy	Process	1..1	The process model that the constraints restrict.
consistsOf	Condition	1..*	The condition that should be satisfied before or after the execution of a process model.
consistsOf	Control_Construct	0..*	The process construct used to connect sub-processes of a composite process.

4.3.12 Condition

Condition is a metaclass designating the conditional state that should be satisfied before or after the execution of a process model.

4.3.13 Precondition

Precondition is a metaclass designating the information space and state that should be satisfied to trigger the process models.			
SuperClass Condition			
Reference	**Class**	**Multiplicity**	**Description**
constrains	Input	1..1	The Input triggered by the corresponding Precondition

4.3.14 Postcondition

Postcondition is a metaclass designating the information space and state that should be satisfied after the successful execution of the process model.			
SuperClass Condition			
Reference	**Class**	**Multiplicity**	**Description**
constrains	Output	1..1	The Output restricted by the corresponding Postcondition

 © ISO/IEC 2008 – All rights reserved

4.3.15 Control_Construct

Control_Construct is a metaclass designating constraint relationship between process models.			
Attribute	**DataType**	**Multiplicity**	**Description**
type	String	1..1	Type of the corresponding Control_Contruct
Constraints			
The value of attribute “type” has to be one of the follows: Merge, Choice, Sequence, Join and AnyOrder.			

4.3.16 AnyOrder

AnyOrder is a metaclass that allows sub-processes to be executed in an unspecified order.
SuperClass Control_Construct

4.3.17 Choice

Choice is a metaclass that invokes one component of process model from a given collection.
SuperClass Control_Construct

4.3.18 Join

Join is a metaclass that works when all of its components have been completed.
SuperClass Control_Construct

4.3.19 Sequence

Sequence is a metaclass that means execution in order.
SuperClass Control_Construct

© ISO/IEC 2008 – All rights reserved

4.3.20 Split

Split is a metaclass that produces at least two branches when the previous process model is executed successfully.
SuperClass Control_Construct

© ISO/IEC 2008 – All rights reserved

5 Conformance

5.1 General

An implementation claiming conformance with this part of ISO/IEC 19763 shall support the metamodel specified in 5.3, depending on a degree of conformance as described below.

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 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.

5.2.2 Strictly conforming implementation

A strictly conforming implementation

a) shall support the metamodel specified in 5.3;

b) shall not support any extensions to the metamodel specified in 5.3.

5.2.3 Conforming implementation

A conforming implementation

a) shall support the metamodel specified in 5.3;

b) may support extensions to the metamodel specified in 5.3 that are consistent with the metamodel specified in 5.3.

5.3 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.

© ISO/IEC 2008 – All rights reserved

Annex A (informative) **Examples of MFI Process registration**

In this section, two cases will be studied to illustrate how to register various kinds of process models based on MFI-5 and enable semantic interoperation between them.

Case 1: BravoAir reservation service (http://www.daml.org/services/owl-s/1.0/examples.html)
BravoAir reservation service is expressed in OWL-s to designate the processes of online flight booking. More specifically, BravoAir process consists of a sequence of sub-processes, involving two atomic processes respectively called GetDesiredFlightDetails and SelectAvailableFlight, and a composite process named BookFlight.

```
   -->
 - <process:CompositeProcess rdf:ID="BravoAir_Process">
     <rdfs:label>This is the top level process for BravoAir</rdfs:label>
   - <process:composedOf>
     - <process:Sequence>
       - <process:components rdf:parseType="Collection">
           <process:AtomicProcess rdf:about="#GetDesiredFlightDetails" />
           <process:AtomicProcess rdf:about="#SelectAvailableFlight" />
           <process:CompositeProcess rdf:about="#BookFlight" />
         </process:components>
       </process:Sequence>
     </process:composedOf>
   </process:CompositeProcess>
```

BookFlight comprise two atomic processes Login and ConfirmReservation.

```
 - <process:CompositeProcess rdf:ID="BookFlight">
   - <process:composedOf>
     - <process:Sequence>
       - <process:components rdf:parseType="Collection">
           <process:AtomicProcess rdf:about="#Login" />
           <process:AtomicProcess rdf:about="#ConfirmReservation" />
         </process:components>
       </process:Sequence>
     </process:composedOf>
   </process:CompositeProcess>
```

Each instance of Process has several Input and Output, carrying Artifacts to fulfill a common purpose. Meanwhile, corresponding control constraints will also be added to Input and Output respectively to restrict the execution of the process model and obtain the desirable results.

```
- <process:AtomicProcess rdf:ID="ConfirmReservation">
    <process:hasInput rdf:resource="#ReservationID_In" />
    <process:hasInput rdf:resource="#Confirm_In" />
    <process:hasOutput rdf:resource="#PreferredFlightItinerary_Out" />
    <process:hasOutput rdf:resource="#AcctName_Out" />
    <process:hasOutput rdf:resource="#ReservationID_Out" />
    <process:hasEffect rdf:resource="#HaveSeat" />
  </process:AtomicProcess>
- <process:Input rdf:ID="ReservationID_In">
    <process:parameterType rdf:resource="http://www.daml.org/services/owl-s/1.0/Concepts.owl#ReservationNumber" />
  </process:Input>
- <process:Input rdf:ID="Confirm_In">
    <process:parameterType rdf:resource="http://www.daml.org/services/owl-s/1.0/Concepts.owl#Confirmation" />
  </process:Input>
- <process:UnConditionalOutput rdf:ID="PreferredFlightItinerary_Out">
    <process:parameterType rdf:resource="http://www.daml.org/services/owl-s/1.0/Concepts.owl#FlightItinerary" />
  </process:UnConditionalOutput>
- <process:UnConditionalOutput rdf:ID="AcctName_Out">
    <process:parameterType rdf:resource="http://www.daml.org/services/owl-s/1.0/Concepts.owl#AcctName" />
  </process:UnConditionalOutput>
- <process:UnConditionalOutput rdf:ID="ReservationID_Out">
    <process:parameterType rdf:resource="http://www.daml.org/services/owl-s/1.0/Concepts.owl#ReservationNumber" />
  </process:UnConditionalOutput>
- <process:UnConditionalEffect rdf:ID="HaveSeat">
    <process:ceEffect rdf:resource="http://www.daml.org/services/owl-s/1.0/Concepts.owl#HaveFlightSeat" />
  </process:UnConditionalEffect>
</rdf:RDF>
```


© ISO/IEC 2008 – All rights reserved

Then the registration information of Composite_Process , Atomic_Process and Artifact will be illustrated in Figure 5.

Process	
type of Process	*Composite Process*
name	*BravoAir_ProcessModel*
URI	*URI_BravoAir_ProcessModel*
administration_Record	*#*
hasGoal	*Online flight booking*
modelType	*OWL-S*
hasInput	*Input: DepartureAirport_In*
	Input: ArrivalAirport_In
	Input: RoundTrip_In
	……
hasOutput	*Output: ReservationID_Out*
	Output: PreferredFlightItinerary_Out
	……
consistsOf	*AtomicProcess: GetDesiredFlightDetails*
	AtomicProcess: SelectAvailableFlight
	CompositeProcess: BookFlight

Process	
type of Process	*Atomic Process*
name	*Login*
URI	*URI_Login*
administration_Record	*#*
hasGoal	*Input username and password*
modelType	*OWL-S*
hasInput	*Input: AcctName_In*
	Input: Password_In

Artifact	
name	*AcctName*
referredTo	*Input:AcctName_In*
URI	*http://www.daml.org/services/owl-s/1.0/ Concepts.owl#AcctName*

Figure 5 – Registration information of BravoAir Reservation Service

Case 2: process for manufacturing a GT350 *(from Annex C of ISO 18629-12)*

The GT-350 manufacturing process is divided into six main areas. They are “make interior”, “make drive”, “make trim”, “make engine”, “make chassis” and “final assembly”. Particularly, the first five tasks are all unordered with respect to each other but they must all be complete before final assembly takes place.

The following is the fragment top level process for manufacturing a GT350

```
(subactivity make-chassis make_gt350)
(subactivity make-interior make_gt350)
(subactivity make-drive make_gt350)
(subactivity make-trim make_gt350)
(subactivity make-engine make_gt350)
(subactivity final-assembly make_gt350)
```

```
(forall (?s1 ?s2 ?s3 ?s4 ?s5 ?s6)
(implies     (and    (leaf_occ ?s1 ?occ1)
                     (leaf_occ ?s2 ?occ2)
                     (leaf_occ ?s3 ?occ3)
                     (leaf_occ ?s4 ?occ4)
                     (leaf_occ ?s5 ?occ5)
                     (root_occ ?s6 ?occ6))
        (min_precedes ?s1 ?s6 make_gt350)
        (min_precedes ?s2 ?s6 make_gt350)
        (min_precedes ?s3 ?s6 make_gt350)
        (min_precedes ?s4 ?s6 make_gt350)
(min_precedes ?s5 ?s6 make_gt350))))))
```

Figure 6 suggests the registration information of top level process for manufacturing GT350.

© ISO/IEC 2008 – All rights reserved

Process	
type of Process	*Composite Process*
name	*Make_gt350*
URI	*URI_make_gt350*
administration_Record	*#*
hasGoal	*Manufacturing GT350*
modelType	*PSL*
consistsOf	*CompositeProcess: make_chassis*
	CompositeProcess: make_interior
	CompositeProcess: make_drive
	CompositeProcess: make_trim
	CompositeProcess: make_engine
	CompositeProcess: final_assembly

Control_Constraint	
name	*Order_Of_Manufacturing_GT350*
constrains	*Make_gt350*
consistsOf	*Control_Construct: C1*
	Control_Construct: C2
	Control_Construct: C3
	Control_Construct: C4
	Control_Construct: C5

Control_Construct	
name	*C1*
Type	*Anyorder*
before	*make_chassis*
after	*Make_interior*

Control_Construct	
name	*C2*
Type	*Join*
before	*make_chassis*
	make_trim
	……
after	*final_assembly*

Figure 6 – Registration Information of top level process for manufacturing GT350

The following is the Fragment of manufacturing subprocess.

```
(subactivity make_block make_engine)
(subactivity make-harness make_engine)
(subactivity make-wires make_engine)
(subactivity assemble_engine make_engine)

(forall (?occ)
 (⇔ (occurrence_of ?occ make_engine)
      (exists (?occ1 ?occ2 ?occ3 ?occ4)
         (and   (occurrence_of ?occ1 make_block)
                (occurrence_of ?occ2 make_harness)
                (occurrence_of ?occ3 make_wires)
                (occurrence_of ?occ4 assemble_engine)
                (subactivity_occurrence ?occ1 ?occ)
                (subactivity_occurrence ?occ2 ?occ)
                (subactivity_occurrence ?occ3 ?occ)
                (subactivity_occurrence ?occ4 ?occ)
                   (forall (?s1 ?s2 ?s3 ?s4)
                   (implies   (and   (leaf_occ ?s1 ?occ1)
                                     (leaf_occ ?s2 ?occ2)
                                     (leaf_occ ?s3 ?occ3)
                                     (root_occ ?s4 ?occ4))
                        (min_precedes ?s1 ?s4 make_engine)
                        (min_precedes ?s2 ?s4 make_engine)
                   (min_precedes ?s3 ?s4 make_engine)))))
```

```
(subactivity produce_molded_metal make_block)
(subactivity machine_block make_block)
(primitive machine_block)
(primitive produce_molded_metal)
(forall (?occ)
   (⇔      (occurrence_of ?occ make_block)
           (exists (?occ1 ?occ2)
              (and   (occurrence_of ?occ1 produce_molded_metal)
                     (occurrence_of ?occ2 machine_block)
                     (min_precedes ?occ1 ?occ2 make_block)))))
```

Figure 7 shows the registration information of some sub-processes of manufacturing GT350.

 © ISO/IEC 2008 – All rights reserved

Process	
type of Process	*Composite Process*
name	*Make_block*
URI	*URI_make_block*
administration_Record	*#*
hasGoal	*Manufacturing 350-Block*
modelType	*PSL*
consistsOf	*AtomicProcess: machine_block*
	AtomicProcess: produce_molded_metal

Process	
type of Process	*Atomic Process*
name	*machine_block*
URI	*URI_machine_block*
administration_Record	*#*
hasGoal	*Manufacturing 350- machine block*
modelType	*PSL*

Control_Constraint	
name	*Order_Of_Manufacturing_GT350*
constrains	*Make_gt350*
consistsOf	*Control_Construct: C8*

Control_Construct	
name	*C8*
Type	*Sequence*
before	*produce_molded_metal*
after	*machine_block*

Figure 7 – Registration information of some sub-processes of manufacturing GT350

© ISO/IEC 2008 – All rights reserved

Annex B (informative) **Collaboration between MFI members**

Annex B shows the exemplary collaboration between MFI members, especially the semantic interoperation based on the cooperation between MFI Process registration and MFI Ontology registration.

There are two process models existing on the web. One is named Withdraw Deposit and described with BPEL. If we input valid AccountID and password, the output will be an amount of RMB. The other is Sell Goods expressed in OWL-s, which is used to get expected Goods by Currency. Now users intend to exchange information between these two process models and enable interoperation between them.

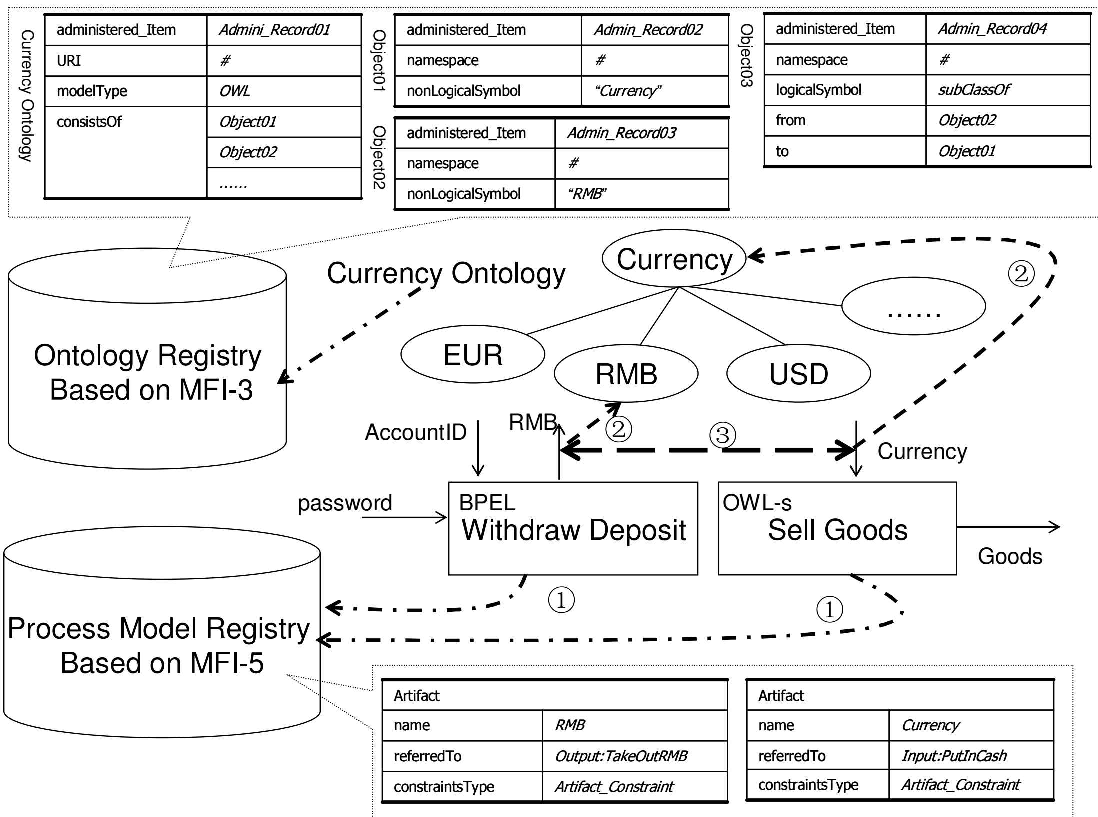

Figure 8 – Semantic interoperation based on MFI Process registration and MFI Ontology registration

For this purpose, it is required to match the output message of Withdraw Deposit process to the input message of Sell Goods process. First of all, we will search the process model registry based on MFI Process registration for semantic registration of both Withdraw Deposit and Sell Goods, as the bottom of Figure 8 suggests. Obviously, input/output of these two models are referred to designated artifacts which are interrelated semantically because RMB is a kind of Currency in Currency Ontology. Then registration information of Currency Ontology on the top of Figure 8 explains the inherent relation between these two concepts in a precise way. Therefore, it is feasible for agent to fill in the Artifact_Constraint within process models and establish semantic relation between them. That is, there is semantic relation between the output message of Withdraw Deposit process and the input of Sell Goods process, which will promote interoperation between these two process models.

 © ISO/IEC 2008 – All rights reserved

Annex C (informative) **List of process modelling languages**

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

Table 1 – List of Process_Modelling_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.
BPMN	Business Process Modeling Notation, Object Management Group, 2004.
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.
PSL	A language that conforms to ISO/IEC 18629 Process Specification Language.
IDEF-3	The IDEF (Integrated Definition Methods) Process Description Capture Method provides a mechanism for collecting and documenting processes, especially the behavioral aspects of an existing or proposed system.
Other	

© ISO/IEC 2008 – All rights reserved