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

**Part 9:
On demand model selection**

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

Partie 9: Sélection de modèle à la demande



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

Foreword	v
Introduction.....	vi
1 Scope.....	1
2 References	1
3 Terms, definitions and abbreviated terms	1
3.1 Terms and definitions	1
3.2 Abbreviated terms	2
4 Preliminaries of ODMS.....	3
4.1 Associations in RGPS.....	3
4.2 Semantic annotation	5
5 Framework of ODMS	6
5.1 Model selection approaches	6
5.2 General procedure of ODMS	7
6 Typical model selection cases.....	8
6.1 Model selection from goal to service	8
6.2 Model selection from process to service.....	9
Annex A (informative) Example of on demand model selection	10

Figures

Figure 1 – Associations in RGPS	4
Figure 2 – Semantic annotation in RGPS	6
Figure 3 – General procedure of ODMS represented in BPMN.....	7
Figure 4 – Model selection from goal to service	8
Figure A.1 – Graphical representation of the models to be registered.....	11
Figure A.2 – Example of role and goal model registration (Part 1 of 2)	12
Figure A.2 – Example of role and goal model registration (Part 2 of 2)	13
Figure A.3 – Example of process model registration	14
Figure A.4 – Example of service model registration	15

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.

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).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: [Foreword — Supplementary information](#).

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 19763-9 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

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

Industrial consortia have engaged in the standardization of domain-specific objects including business process models and software components using common modelling facilities 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.

ISO/IEC 19763 provides registration mechanisms for different kinds of information resources in business domains, such as ontologies, roles, goals, processes, and services. Faced with the abundant and heterogeneous models, how to select appropriate services and/or models to meet user-requests becomes an important issue. Based on the metamodels defined in parts 3, 5, 7 and 8 of ISO/IEC 19763, this technical report describes a framework and procedures for model selection so as to help users discover corresponding models or services that support their requests.

Information technology – Metamodel framework for interoperability (MFI) — Part 9: On demand model selection

1 Scope

This ISO/IEC Technical Report specifies a technical guideline on how to use the Role and Goal, Process, and Service (RGPS) metamodels to select appropriate combinations of models and/or services to support user-requests.

The scope of ISO/IEC TR 19763-9 is limited to model selection based on ISO/IEC 19763-5, ISO/IEC 19763-7, and ISO/IEC 19763-8.

2 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 19763-3, 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 19763-7, Information technology – Metamodel framework for interoperability (MFI) – Part 7: Metamodel for service 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

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

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this part, the terms and definitions contained in ISO/IEC 19763-1, 3, 5, 7, 8, 10 and the following shall apply.

3.1.1

goal

intended outcome of user interaction with a **process** (3.1.4) or **service** (3.1.10)

[ISO/IEC 19763-8, 3.1.1]

3.1.2

involvement type

statement that indicates the type of involvement of a **role** (3.1.8) with a **process** (3.1.4) or **service** (3.1.10)

NOTE Examples are performer, beneficiary, and customer

[ISO/IEC 19763-8, 3.1.4]

3.1.3

ontology

specification of concrete or abstract things, and the relationships among them, in a prescribed domain of knowledge

NOTE The specification should be computer processable

[ISO/IEC 19763-3:2010, 3.1.1.1]

3.1.4

process

collection of related, structured activities or tasks that achieve a particular **goal** (3.1.1)

NOTE The activities and tasks are represented by the Process metaclass in this part

[ISO/IEC 19763-5, 3.1.12]

3.1.5

process involvement

statement that specifies how a particular **role** (3.1.8) is engaged in or contributes in a particular **process** (3.1.4)

[ISO/IEC 19763-8, 3.1.6]

3.1.6

request type

target class in the MFI model to be used in the search, e.g., **goal** (3.1.1), **process** (3.1.4) or **service** (3.1.10)

3.1.7

return type

kind of models that the user would like to find in the search

3.1.8

role

named specific behaviour of an entity participating in a particular context

[ISO/IEC 19763-8, 3.1.7]

3.1.9

search term

term specified by the user in the search

3.1.10

service

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

[ISO/IEC 19763-7, 3.1.17]

3.1.11

service involvement

statement that specifies how a particular **role** (3.1.8) is involved in a particular **service** (3.1.10)

[ISO/IEC 19763-8, 3.1.8]

3.2 Abbreviated terms

BPMN

Business Process Model and Notation

[OMG BPMN version 2, formal/2011-01-03]

KAOS

Keep All Objects Satisfied

MFI

Metamodel framework for interoperability

[ISO/IEC 19763-1: 2007, 4.2]

ODMS

On Demand Model Selection

QoS

Quality of Service

RGPS

Role, Goal, Process, and Service

UML

Unified Modeling Language

[ISO/IEC 19505-1:2012] and [ISO/IEC 19505-2:2012]

WADL

Web Application Description Language

WSDL

Web Service Description Language

4 Preliminaries of ODMS

In order to show how to realize On Demand Model Selection (ODMS), some preliminaries need to be introduced first. The associations in RGPS classes and their semantic annotations form the basis for ODMS. The RGPS associations specify how the different models are related, and the ontology concepts used in the semantic annotations form the basis for matching user-requests with registered models.

4.1 Associations in RGPS

Since the scope of ISO/IEC TR 19763-9 is limited to model selection based on ISO/IEC 19763-5, ISO/IEC 19763-7, and ISO/IEC 19763-8, the three parts will be introduced first.

ISO/IEC 19763-5 specifies a metamodel to enable organizations to create a registry storing the administrative and descriptive information of process models. The process model registration metamodel is intended to promote semantic discovery and reuse of process models within/across organizations.

ISO/IEC 19763-7 specifies a metamodel to enable organizations to create a registry storing the administrative and descriptive information of service models. The service model registration metamodel is intended to promote semantic discovery and reuse of services within/across organizations.

ISO/IEC 19763-8 specifies a metamodel to enable organizations to create a registry storing the administrative and descriptive information of role and goal models. The role and goal model registration metamodel is intended to promote semantic discovery and reuse of role and goal models within/across organizations.

For the purposes of this technical report, RGPS is viewed as a generic term referring to the method of applying associations between RGPS models to support ODMS.

As shown in Figure 1, there are associations between the metamodels defined in parts 5, 7 and 8 of ISO/IEC 19763.

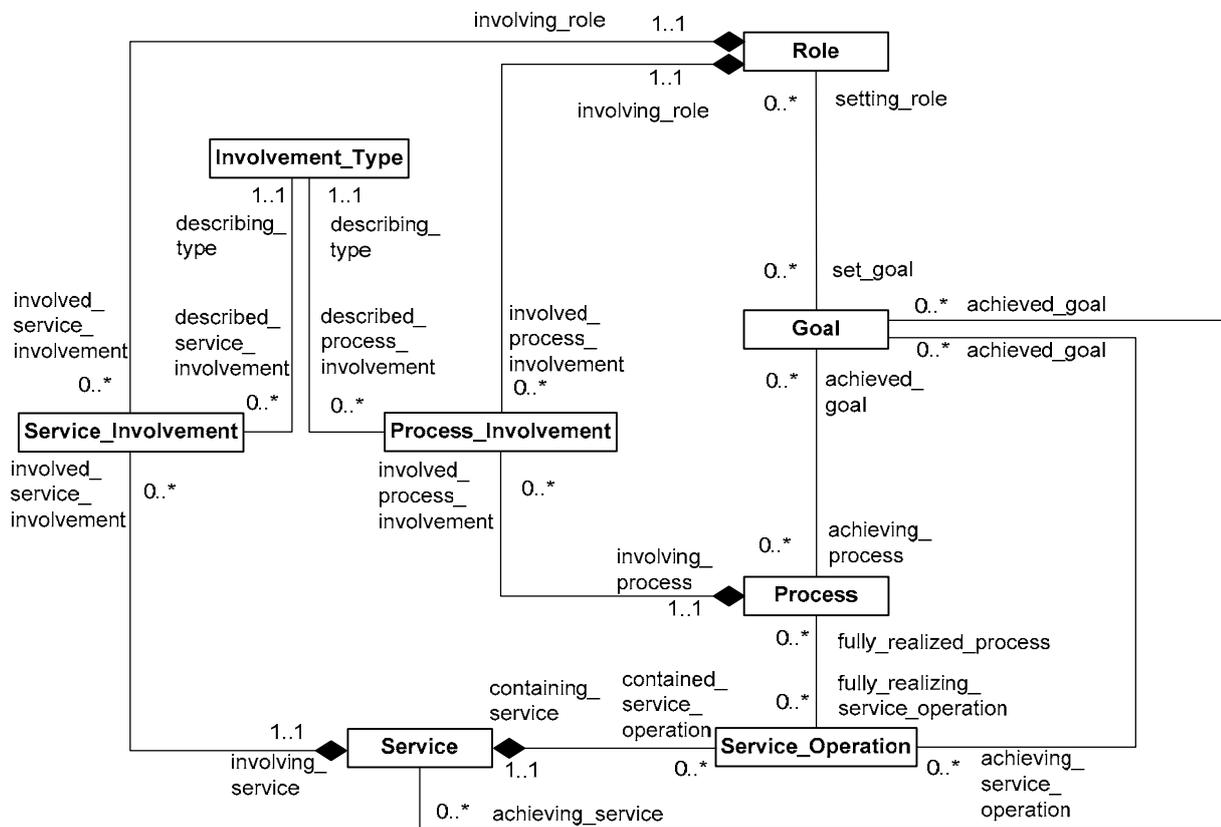


Figure 1 – Associations in RGPS

The associations in ISO/IEC 19763-8 are:

Each role sets zero, one or more goals.
 Each goal is set by zero, one or more roles.

Each role is involved in processes through zero, one or more process involvements.
 Each process involvement represents the involvement in processes of one and only one role.

Each role is involved in services through zero, one or more service involvements.
 Each service involvement represents the involvement in services of one and only one role.

Each involvement type is used to describe zero, one or more process involvements.
 Each process involvement is described by one and only one involvement type.

Each involvement type is used to describe zero, one or more service involvements.
 Each service involvement is described by one and only one involvement type.

The associations in ISO/IEC 19763-7 are:

Each service contains zero, one or more service operations.
 Each service operation is contained by one and only one service.

The associations that associate ISO/IEC 19763-5 with ISO/IEC 19763-8 are:

Each process is used to achieve zero, one or more goals.
Each goal is achieved by zero, one or more processes.

Each process is represented in zero, one or more process involvements.
Each process involvement is involved in one and only one process.

The associations that associate ISO/IEC 19763-7 with ISO/IEC 19763-8 are:

Each service is used to achieve zero, one or more goals.
Each goal is achieved by zero, one or more services.

Each service operation is used to achieve zero, one or more goals.
Each goal is achieved by zero, one or more service operations.

Each service is represented in zero, one or more service involvements.
Each service involvement is involved in one and only one service.

The associations that associate ISO/IEC 19763-7 with ISO/IEC 19763-5 are:

Each service operation is used to fully realize zero, one or more processes.
Each process is fully realized by zero, one or more service operations.

NOTE 1 The instance of involvement type can be performer, beneficiary, customer, and so on.

NOTE 2 In the case that a process is fully realized by a set of service operations, the process should be decomposed into a certain level such that each sub-process of the process can be fully realized by a service operation.

To facilitate ODMS within an organization's set of registries based on the metamodels defined in the various parts of ISO/IEC 19763, the associations in RGPS should be recorded. However, it is not necessary to maintain a separate registry to record these associations. In order to record these associations, the following strategies might be adopted. The associations between processes with their roles and goals will be registered in a registry based on the metamodel defined in ISO/IEC 19763-5; the associations between services with their roles, goals, and processes will be registered in a registry based on the metamodel defined in ISO/IEC 19763-7. Note that Figure 1 only shows the associations among roles, goals, processes and services, not all associations in the metamodels defined in parts 5, 7 and 8 of ISO/IEC 19763.

4.2 Semantic annotation

An essential issue in ODMS is how to match user-requests with registration information of the registered RGPS models. The use of semantic annotations of registered models based on domain specific ontologies can be used to bridge the gap between the registered RGPS models, as well as the gap between user-requests and the registration information.

In order to semantically annotate the registered RGPS models, two kinds of domain sub-ontologies, entity ontology and operation ontology, are considered (Figure 2). The entity ontology mainly describes the entity concepts and semantic relationships among them, and the operation ontology mainly describes the operational or functional concepts as well as semantic relationships among them. The domain ontology can be used to annotate the goal class with attributes <goal operation, goal object> in a registry based on the metamodel defined in ISO/IEC 19763-8. When registering a process in a registry based on the metamodel defined in ISO/IEC 19763-5, the goals achieved by the process can be defined by setting the reference `achieved_goal`, whose values are from goals registered in the registry based on the metamodel defined in ISO/IEC 19763-8, i.e., the same ontology is used to annotate the goals achieved by the process.

For example, given a transportation domain goal "Book ticket" with attributes <goal operation, goal object>, where the goal operation is annotated by the concept "Book" in the operation ontology of transportation domain, while the goal object would be annotated by the concept "Ticket" in the entity ontology of transportation domain. A user searching for a process that can achieve a goal "Reserve ticket" might find the concept in the transportation domain ontology with a synonym "Book ticket". Using the same ontology to annotate the RGPS models enables the ontology to provide support for semantic matching based on concept synonyms. Then the processes that are associated with the concept, regardless of whether the process is named "Book ticket" or "Reserve ticket", will be searched.

Provision is made in ISO/IEC 19763 for all models and model elements to be annotated with concepts within ontologies registered in a registry based on the metamodel defined in ISO/IEC 19763-3. Hence, in parts 5, 7 and 8 of ISO/IEC 19763, the meta-classes of role, goal, role and goal model, process, process model, event, resource, service model, service, service operation, input and output can be annotated by concepts from domain ontologies, and these annotations can be used to support model discovery across registries based on the metamodels defined in the various parts of ISO/IEC 19763.

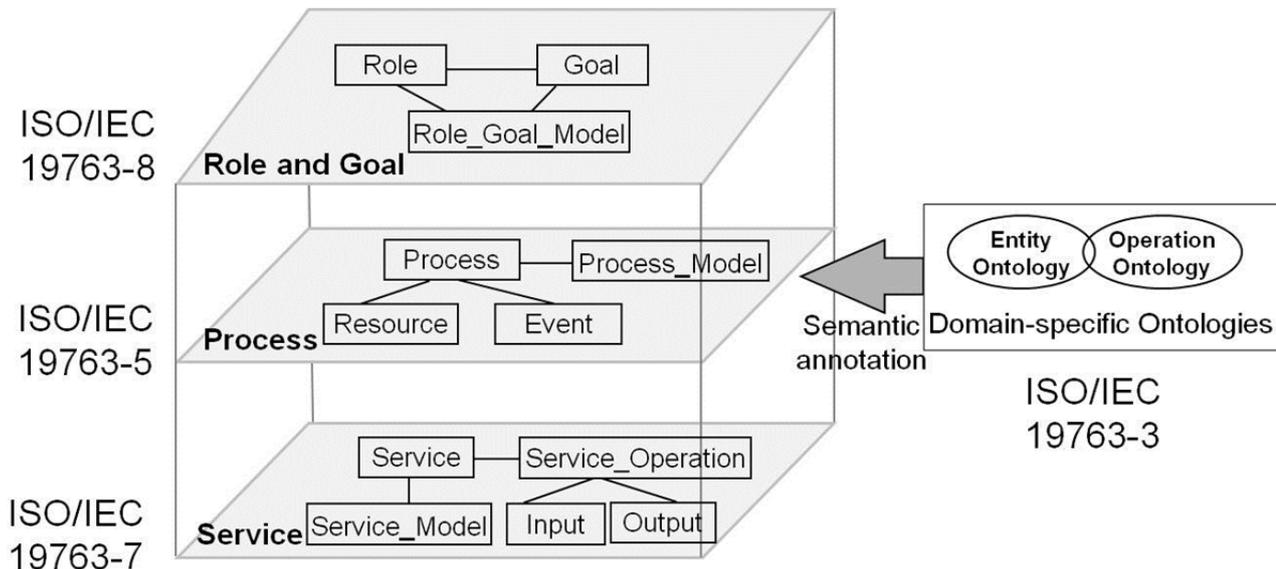


Figure 2 – Semantic annotation in RGPS

A registry based on the metamodels defined in the various parts of ISO/IEC 19763 does not hold all of the semantics of the registered models, but it is possible to retrieve all the semantics using the identifiers of the models registered in the registry. Figure 1 describes the associations that relate together various kinds of models. It is these associations that can be used to select models from the registries. Figure 2 presents the meta-classes from parts 5, 7 and 8 of ISO/IEC 19763 that may be annotated using domain ontologies to enable semantic matching during model selection.

5 Framework of ODMS

ODMS is achieved through the registration of models whose elements are annotated using domain ontologies. The metamodels in ISO/IEC 19763-5, 7, and 8 specify the classes and associations between them and the other parts of ISO/IEC 19763. For example, the metamodel for process model registration in ISO/IEC 19763-5 specifies a reference for the goal it achieves, and the goal class is annotated by concepts from the domain operation ontology and entity ontology such as “Book ticket”. Then, matching the user-requests with appropriate models can be accomplished using semantic annotations and the corresponding associations.

5.1 Model selection approaches

In addition to the semantic annotations, the associations between the models that are registered in a registry, or a set of interoperating registries, specified by the metamodels defined in parts 5, 7 and 8 of ISO/IEC 19763 play an important role during the ODMS process. The various model selection approaches available to users are based on these associations that are shown in Figure 1.

During model selection, user-requests can be expressed by means of a goal, a process, or a service. When user-requests are matched to a goal in a role and goal model registered in a registry based on the metamodel defined in ISO/IEC 19763-8, the following steps can be taken:

- querying the subgoals that the matched goal can be decomposed into,
- querying the upper goal that the matched goal is derived from,
- querying the goals that the matched goal depends on,
- querying the role that undertakes the matched goal,

- querying the process that achieves the matched goal, and
- querying the service that achieves the matched goal.

When user-requests are matched to a process registered in a registry based on the metamodel defined in ISO/IEC 19763-5, the following steps can be taken:

- querying the roles that are involved in the matched process,
- querying the goals that can be achieved by the matched process,
- querying the subprocesses that the matched process can be decomposed into, and
- querying the services that can fully realize the matched process.

When user-requests are matched to a service registered in a registry based on the metamodel defined in ISO/IEC 19763-7, the following steps can be taken:

- querying the roles that are involved in the matched service,
- querying the goals achieved by the matched service,
- querying the processes fully realized by the matched service, and
- querying the services used by the matched service.

Based on the model selection approaches, the whole model selection process may consist of several iteration steps to obtain appropriate models that satisfy the user-requests.

5.2 General procedure of ODMS

As shown in Figure 3, the framework of ODMS consists of three parts: user interface, model selection engine, and MFI model registries, where the user interface is used to elicit user-requests; the model selection engine is used to analyze user-requests, and find corresponding candidate models or services according to the user-requests; and the MFI model registries store the registration information and associations of RGPS models.

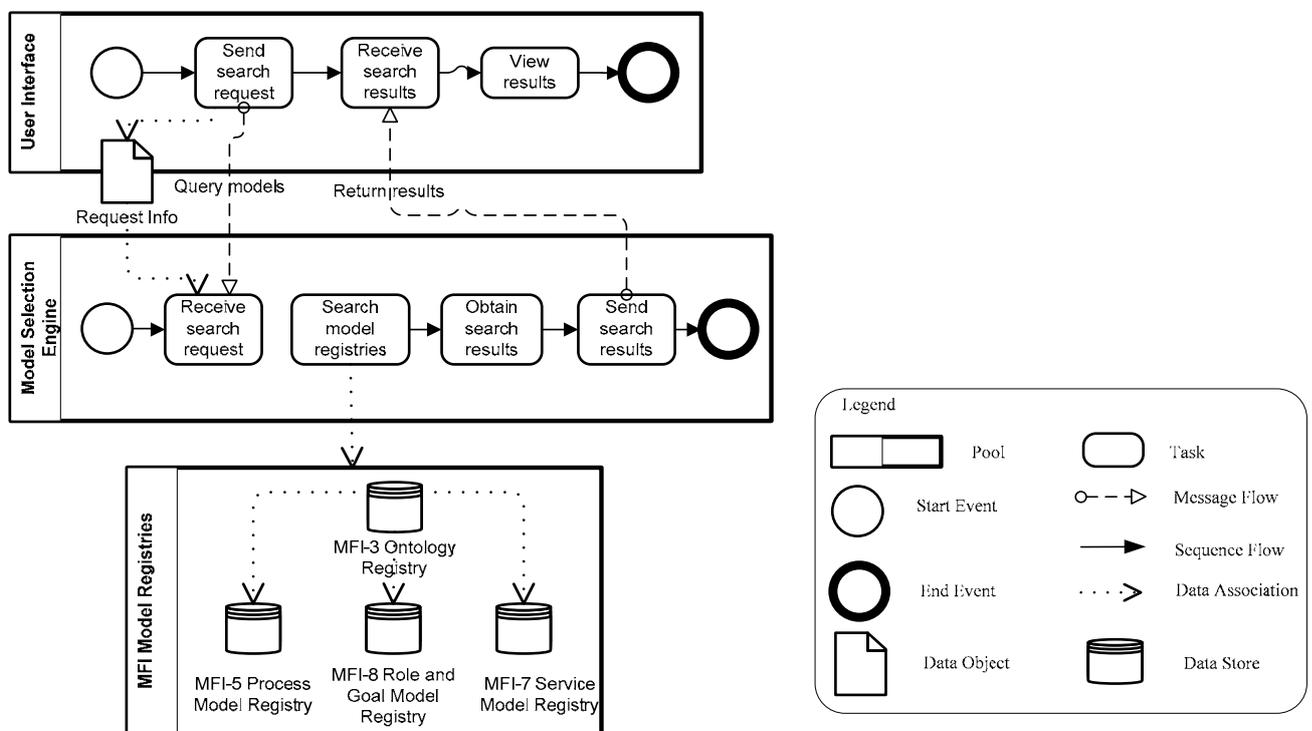


Figure 3 – General procedure of ODMS represented in BPMN

The following steps describe the general procedure of ODMS. As shown in Figure 3, user-requests are submitted to the model selection engine through the user interface. The model selection engine will search the registry based on the metamodel defined in ISO/IEC 19763-3 to find concepts that match the user-requests.

The model selection engine will perform a search for models using these concepts to match against the semantic annotations of the models registered in registries based on the metamodels defined in ISO/IEC 19763-5, 7, and 8.

A user interface to elicit user-requests might consist of three input parameters: the request type, specifying the target class in the MFI model to be used in the search (goal, process or service), the search term, and the result type, specifying the kind of model they would like to find. For example, they might enter the search term “Book Ticket” as a goal, and tell the model selection engine to return services that achieve the goal.

The user interface may be designed to have slight differences according to the request type of the search, as different attributes for each target class could be used based on the metamodels in ISO/IEC 19763-5, 7 and 8. When users select the goal as the request type, they need to specify the desired goal name and other elements that are annotated by domain ontologies as the search terms. When users select the process as the request type, they can specify the desired process name, resource and other elements that are annotated by domain ontologies as the search terms. When users select the service as the request type, users can specify service name, input and output of the service, and other elements that are annotated by domain ontologies as the search terms. In the optional part, users can also specify the corresponding QoS requests on their desired services. A query against the QoS target might need to be expressed in a qualitative manner or a quantitative manner in order to query against QoS_Assertion in ISO/IEC 19763-7. In a qualitative manner, the type (e.g., “security” or “performance”) and the expression of the assertion (e.g., “security is high” or “performance is low”) could be specified. In a quantitative manner, the type (e.g., “cost” or “response time”), might use a comparison operator (e.g., “equalsTo” or “lessThan”) in an expression containing its unit and value.

6 Typical model selection cases

Two cases are explained at 6.1 and 6.2 to illustrate the selection of appropriate models and/or services to support user-requests. In the first case at 6.1, the user-requests are represented as a goal, and the desired result is the service. In the second case at 6.2, the user-requests are represented as a process and the service as the desired result. The steps show how the user can select the desired models and/or services based on ODMS. Please note that there are other model selection cases, such as model selection from process to process and model selection from service to goal, besides these two example cases.

6.1 Model selection from goal to service

In this case, the user selects the goal as the request type, and the service as the result type. In order to find the appropriate services, the following steps are taken based on the associations in RGPS and semantic annotations provided by domain ontologies.

Step 1: the user-requests can be matched to a semantic annotation of a goal in a goal model registered in the registry based on the metamodel defined in ISO/IEC 19763-8 (the sub-step *input* in Figure 4). As an optional stage, the subgoals decomposed by the matched goal can be returned to the user for selection. According to the “goal-service” association, if corresponding services that can achieve the matched goal can be found in the registry based on the metamodel defined in ISO/IEC 19763-7(the sub-step *gs* in Figure 4), then the result is directly returned to the user (the sub-step *return* in Figure 4). If the returned results can meet the user-requests, then the model selection process will end, otherwise the following steps are taken.

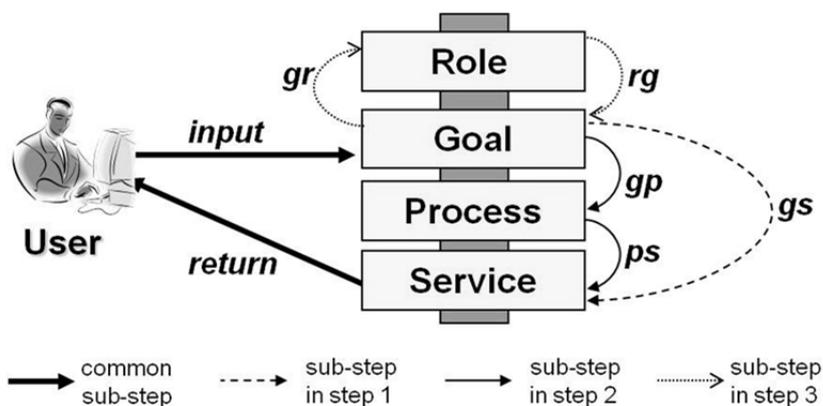


Figure 4 – Model selection from goal to service

Step 2: the model selection engine will visit the registry based on the metamodel defined in ISO/IEC 19763-5 according to the known “goal-process” association to find a process with a goal that matches the user goal (the sub-step *gp* in Figure 4). It will then retrieve the process semantic annotation and use it to revisit the registry based on the metamodel defined in ISO/IEC 19763-7, searching for matching processes, and then use the “process-service operation” association to retrieve candidate services (the sub-step *ps* in Figure 4). If the returned results can meet the user-requests, then the model selection process will end. Otherwise, the following step is taken.

Step 3: the model selection engine will visit the registry based on the metamodel defined in ISO/IEC 19763-8 to find roles that can undertake the matched goal by the “role-goal” association (the sub-step *gr* in Figure 4), and then find and supplement other goals undertaken by the role (the sub-step *rg* in Figure 4). In this way, the candidate goal set can be expanded. For these candidate goals, the subsequent selection process follows Step 1 and Step 2. If the returned results cannot meet the user-requests, then the model selection process will end with a status that no suitable models or services can satisfy the user.

Please note that during the model selection process, the match between the user-requests and registration information, as well as the match between different registration information from RGPS models are based on the corresponding domain ontologies registered in the registry based on the metamodel defined in ISO/IEC 19763-3.

6.2 Model selection from process to service

In this case, the user selects the process as the request type, and selects the service as the result type. In order to find the user’s desired services, the following steps are taken based on the associations in RGPS and semantic annotations provided by domain ontologies registered in the registry based on the metamodel defined in ISO/IEC 19763-3.

Step 1: the user-requests can be matched to a process in a process model registered in the registry based on the metamodel defined in ISO/IEC 19763-5 by the semantic match, which may occur between the user-requests with the process name, resource and other semantically annotated classes in the process model. Then, according to the “process-service operation” association, if corresponding services that contain service operations which can fully realize the matched process can be found in the registry based on the metamodel defined in ISO/IEC 19763-7, and then the result is directly returned to the user. If the returned results can meet the user-requests, then the model selection process will end. Otherwise, the following step is taken.

Step 2: the selection engine will visit the registry based on the metamodel defined in ISO/IEC 19763-8 to find the goal that the matched process can achieve according to the “goal-process” association. After finding the matched goal, the subsequent selection process from goal to service is similar to the case mentioned in 6.1 and depicted in Figure 4.

Annex A

(informative)

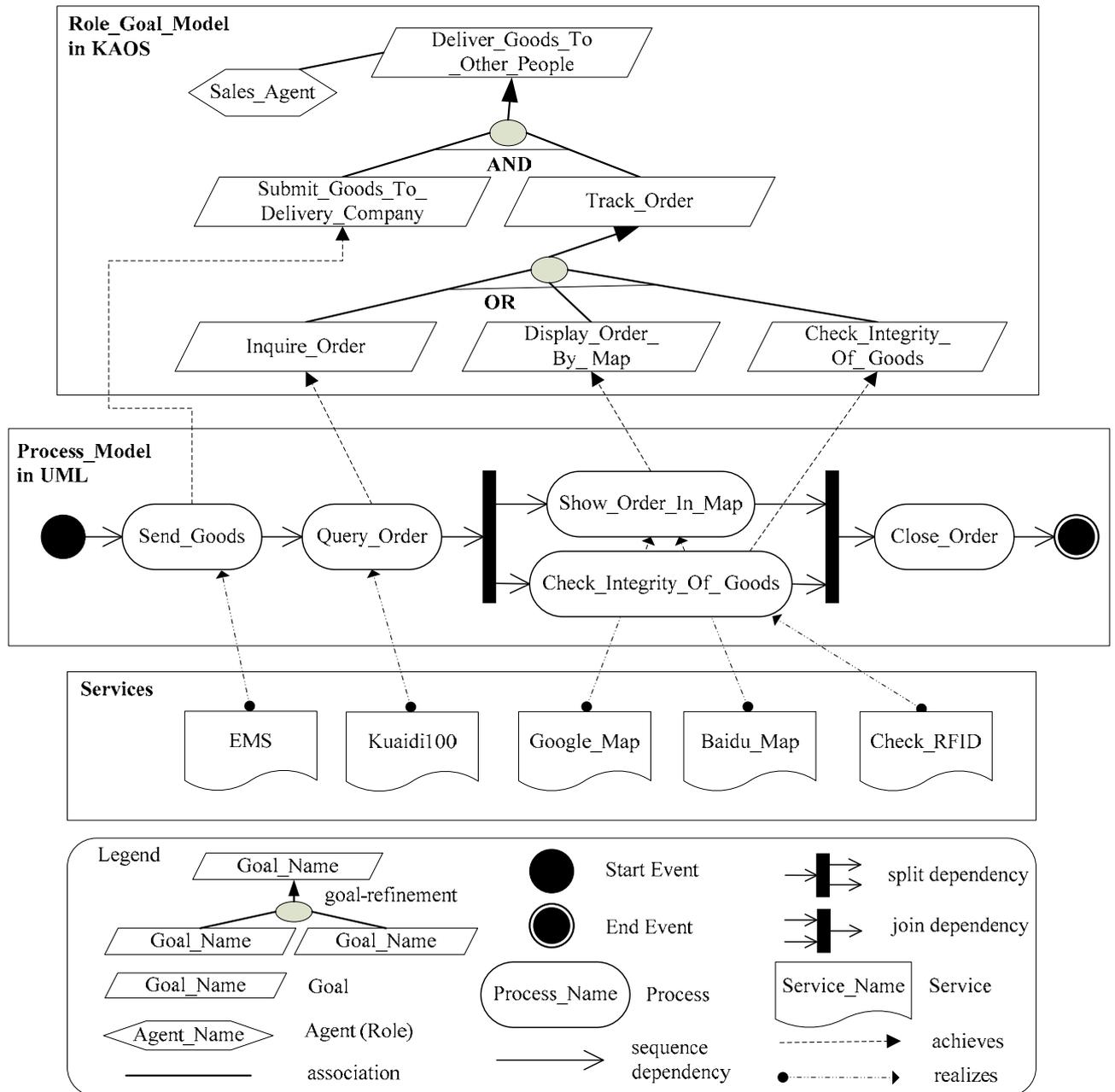
Example of on demand model selection

The following example is provided to illustrate how to find appropriate models according to user-requests based on ODMS. In this example, user-requests are represented as goals, and services are their desired result type. Please note that there are also some other cases that take other kinds of models in RGPS as request type or result type.

In this example, suppose that a user plans to deliver goods to people in another city and he also wants to inquire the order information for tracking the goods. The user-requests can be expressed as “to deliver goods to another city”, and then it will be submitted to the model selection engine. In order to find the user’s desired models, the selection engine searches the candidate models from MFI model registries based on associations in RGPS and semantic annotations provided by domain ontologies registered in the registry based on the metamodel defined in ISO/IEC 19763-3. Figure A.1 depicts the graphical representation of the registered models as well as the associations among them. Please note that the role and goal model is represented in KAOS, the process model is represented in UML, and the services are described in WADL or WSDL. Figure A.2, Figure A.3, and Figure A.4 show partial registration information of role and goal model, process model, and service model registered in registries based on the metamodels defined in ISO/IEC 19763-8, ISO/IEC 19763-5, and ISO/IEC 19763-7, respectively.

The details of the model selection process are described as follows. Firstly, the model selection engine will search a registry based on the metamodel defined in ISO/IEC 19763-3 to find concepts that match the user-requests. In this example, the concept “Deliver_Goods” in a domain ontology can be matched. Then the model selection engine will search models using the concept to match against the semantic annotations of models registered in the registry based on the metamodel defined in ISO/IEC 19763-8, as shown in Figure A.2. The goal “Deliver_Goods_To_Other_People” which is annotated by the concept “Deliver_Goods” can be matched. Then the subgoals of the matched goal are returned to the user. If all the subgoals of this matched goal are selected, then according to the goal-service association, the selection engine visits a registry based on the metamodel defined in ISO/IEC 19763-7 to find the corresponding services that achieve the selected subgoals. However, there is no related service that can achieve the corresponding subgoals. Then according to the goal-process association, the selection engine visits a registry based on the metamodel defined in ISO/IEC 19763-5 to find the corresponding processes that can achieve the selected goals. As shown in Figure A.3, the processes in process model “Goods_Delivery_Model” can be matched with these selected goals. Then according to the process-service association, the selection engine visits a registry based on the metamodel defined in ISO/IEC 19763-7 to find the appropriate service operations that can fully realize the corresponding processes. As shown in Figure A.4, the service operation in service “EMS” can fully realize the process “Send_Goods”, the service operation in service “Kuaidi100” can fully realize the process “Query_Order”, both the service operations in service “Google_Map” and service “Baidu_Map” can fully realize the process “Show_Order_In_Map”, and the service operation in service “Check_RFID” can fully realize the process “Check_Integrity_Of_Goods”. Finally, the selected services as well as the corresponding process model are returned to the user.

If the user does not need to query the integrity of the goods, then the user will select other subgoals of the matched goal “Deliver_Goods_To_Other_People”. For example, the subgoals “Submit_Goods_To_Delivery_Company”, “Inquire_Order”, and “Display_Order_By_Map” may be selected. Then based on the matches of goal-process association and process-service association, all services mentioned above except the service “Check_RFID” will be returned to the user.



NOTE Concept annotations as described in parts 5, 7, 8 of ISO/IEC 19763 are used to facilitate searching models.

Figure A.1 – Graphical representation of the models to be registered

<Role_Goal_Modelling_Language>

Object101

Attribute/Reference	Literal/Instance
name	"KAOS"
expressed_model	Object102

<Role_Goal_Model>

Object102

Attribute/Reference	Literal/Instance
name	"Goods_Delivery"
describing_language	Object101
described_role	Object103
described_goal	Object104, Object106, Object107, Object109, Object110, Object111

<Role>

Object103

Attribute/Reference	Literal/Instance
name	"Sales_Agent"
describing_model	Object102
set_goal	Object104
annotation	Object112

<Functional Goal>

Object104

Attribute/Reference	Literal/Instance
name	"Deliver_Goods_To_Other_People"
is_operational	"FALSE"
goal_operation	"Deliver"
goal_object	"Goods"
describing_model	Object102
setting_role	Object103
decomposing_decomposition	Object105
annotation	Object113

<Decomposition>

Object105

Attribute/Reference	Literal/Instance
decomposition_type	'And'
decomposed_goal	Object104
decomposing_goal	Object106, Object107

<Functional Goal>

Object106

Attribute/Reference	Literal/Instance
name	"Submit_Goods_To_Delivery_Company"
is_operational	"TRUE"
goal_operation	"Submit"
goal_object	"Goods"
describing_model	Object102
decomposed_decomposition	Object105
achieving_process	Object204
annotation	Object114

<Functional Goal>

Object107

Attribute/Reference	Literal/Instance
name	"Track_Order"
is_operational	"FALSE"
goal_operation	"Track"
goal_object	"Order"
describing_model	Object102
decomposing_decomposition	Object108
decomposed_decomposition	Object105
annotation	Object115

<Decomposition>

Object108

Attribute/Reference	Literal/Instance
decomposition_type	'Or'
decomposed_goal	Object107
decomposing_goal	Object109, Object110, Object111

<Functional Goal>

Object109

Attribute/Reference	Literal/Instance
name	"Inquire_Order"
is_operational	"TRUE"
goal_operation	"Inquire"
goal_object	"Order"
describing_model	Object102
decomposed_decomposition	Object108
achieving_process	Object206
annotation	Object116

<Functional Goal>

Object110

Attribute/Reference	Literal/Instance
name	"Display_Order_By_Map"
is_operational	"TRUE"
goal_operation	"Display"
goal_object	"Order"
describing_model	Object102
decomposed_decomposition	Object108
achieving_process	Object210
annotation	Object117

<Functional Goal>

Object111

Attribute/Reference	Literal/Instance
name	"Check_Integrity_Of_Goods"
is_operational	"TRUE"
goal_operation	"Check"
goal_object	"Integrity_Of_Goods"
describing_model	Object102
decomposed_decomposition	Object108
achieving_process	Object211
annotation	Object118

Figure A.2 – Example of role and goal model registration (Part 1 of 2)

<Annotating_Registered_Ontology_Atomic_Construct>

Object112

Attribute/Reference	Literal/Instance
namespace	"http://onto.org/logistics"
nonLogicalSymbol	"Sales_Agent"
annotated_model_element	Object103

<Annotating_Registered_Ontology_Atomic_Construct>

Object116

Attribute/Reference	Literal/Instance
namespace	"http://onto.org/logistics"
nonLogicalSymbol	"Inquire_Order"
annotated_model_element	Object109, Object206, Object307

<Annotating_Registered_Ontology_Atomic_Construct>

Object113

Attribute/Reference	Literal/Instance
namespace	"http://onto.org/logistics"
nonLogicalSymbol	"Deliver_Goods"
annotated_model_element	Object104, Object201

<Annotating_Registered_Ontology_Atomic_Construct>

Object117

Attribute/Reference	Literal/Instance
namespace	"http://onto.org/logistics"
nonLogicalSymbol	"Display_Order_By_Map"
annotated_model_element	Object110, Object210, Object310, Object313

<Annotating_Registered_Ontology_Atomic_Construct>

Object114

Attribute/Reference	Literal/Instance
namespace	"http://onto.org/logistics"
nonLogicalSymbol	"Submit_Goods"
annotated_model_element	Object106, Object204, Object304

<Annotating_Registered_Ontology_Atomic_Construct>

Object118

Attribute/Reference	Literal/Instance
namespace	"http://onto.org/logistics"
nonLogicalSymbol	"Check_Integrity_Of_Goods"
annotated_model_element	Object111, Object211, Object317

<Annotating_Registered_Ontology_Atomic_Construct>

Object115

Attribute/Reference	Literal/Instance
namespace	"http://onto.org/logistics"
nonLogicalSymbol	"Track_Order"
annotated_model_element	Object107

Figure A.2 – Example of role and goal model registration (Part 2 of 2)

<Process_Model>

Object201

Attribute/Reference	Literal/Instance
name	"Goods_Delivery_Model"
describing_language	Object202
contained_process_model_element	Object204, Object205, Object206, Object207, Object210, Object211, Object212, Object215
annotation	Object113

<Process_Modelling_Language>

Object202

Attribute/Reference	Literal/Instance
name	"UML"
version	"2.1.2"
expressed_model	Object201

<Event>

Object203

Attribute/Reference	Literal/Instance
name	"Goods_Received_From_User"
triggered_process_model_element	Object204

<Process>

Object204

Attribute/Reference	Literal/Instance
name	"Send_Goods"
containing_model	Object201
trigger	Object203
successor	Object205
achieved_goal	Object106
fully_realizing_service_operation	Object304
annotation	Object114

<Sequence_Dependency>

Object205

Attribute/Reference	Literal/Instance
containing_model	Object201
preceding_process	Object204
following_process	Object206

<Process>

Object206

Attribute/Reference	Literal/Instance
name	"Query_Order"
containing_model	Object201
precedent	Object205
successor	Object207
achieved_goal	Object109
fully_realizing_service_operation	Object307
annotation	Object116

<Split_Dependency>

Object207

Attribute/Reference	Literal/Instance
containing_model	Object201
split_dependency_type	"Or"
is_synchronous	"FALSE"
preceding_element	Object206
following_option	Object208, Object209

<Split_Dependency_Option>

Object208

Attribute/Reference	Literal/Instance
guard_condition	"Display_Of_Order_In_Map_Required"
following_element	Object210
precedent	Object207

<Split_Dependency_Option>

Object209

Attribute/Reference	Literal/Instance
guard_condition	"Integrity_Check_Required"
following_element	Object211
precedent	Object207

<Process>

Object210

Attribute/Reference	Literal/Instance
name	"Show_Order_In_Map"
containing_model	Object201
preceding_option	Object208
following_option	Object213
achieved_goal	Object110
fully_realizing_service_operation	Object310, Object313
annotation	Object117

<Process>

Object211

Attribute/Reference	Literal/Instance
name	"Check_Integrity_Of_Goods"
containing_model	Object201
preceding_option	Object209
following_option	Object214
achieved_goal	Object111
fully_realizing_service_operation	Object317
annotation	Object118

<Join_Dependency>

Object212

Attribute/Reference	Literal/Instance
join_dependency_type	"Or"
is_synchronous	"FALSE"
preceding_option	Object213, Object214
following_element	Object215
containing_model	Object201

<Join_Dependency_Option>

Object213

Attribute/Reference	Literal/Instance
guard_condition	"Order_Shown_In_Map"
preceding_element	Object210
successor	Object212

<Join_Dependency_Option>

Object214

Attribute/Reference	Literal/Instance
guard_condition	"Integrity_Checked"
preceding_element	Object211
successor	Object212

<Process>

Object215

Attribute/Reference	Literal/Instance
name	"Close_Order"
containing_model	Object201
precedent	Object212
produced_event	Object216

<Event>

Object216

Attribute/Reference	Literal/Instance
name	"Order_Closed"
producer	Object215

Figure A.3 – Example of process model registration

<Service_Description_Language>

Object301

Attribute/Reference	Literal/Instance
name	"WADL"
expressed_model	Object302, Object305, Object308, Object311

<Service_Model>

Object302

Attribute/Reference	Literal/Instance
name	"EMS_Service_Model"
described_IRI	"http://www.ems.com.cn/EMSService"
describing_language	Object301
contained_service	Object303

<Service>

Object303

Attribute/Reference	Literal/Instance
requested_IRI	"http://www.ems.com.cn/EMSService"
domain	"Logistics"
service_type	"Entity_Service"
name	"EMS"
containing_model	Object302
contained_service_operation	Object304

<Service_Operation>

Object304

Attribute/Reference	Literal/Instance
name	"Goods_Delivery_Capability"
fully_realized_process	Object204
containing_service	Object303
annotation	Object114

<Service_Model>

Object305

Attribute/Reference	Literal/Instance
name	"Kuaiid100_Service_Model"
described_IRI	"http://www.kuaiid100.com/openapi"
describing_language	Object301
contained_service	Object306

<Service>

Object306

Attribute/Reference	Literal/Instance
requested_IRI	"http://www.kuaiid100.com/openapi"
domain	"Logistics"
service_type	"Entity_Service"
name	"Kuaiid100"
containing_model	Object305
contained_service_operation	Object307

<Service_Operation>

Object307

Attribute/Reference	Literal/Instance
name	"Order_Query"
fully_realized_process	Object206
containing_service	Object306
annotation	Object116

<Service_Model>

Object308

Attribute/Reference	Literal/Instance
name	"Google_Map_Service_Model"
described_IRI	"https://developers.google.com/maps"
describing_language	Object301
contained_service	Object309

<Service>

Object309

Attribute/Reference	Literal/Instance
requested_IRI	"https://developers.google.com/maps"
domain	"Maps"
service_type	"Entity_Service"
name	"Google_Map"
containing_model	Object308
contained_service_operation	Object310

<Service_Operation>

Object310

Attribute/Reference	Literal/Instance
name	"Google_Map_Route_Display"
fully_realized_process	Object210
containing_service	Object309
annotation	Object117

<Service_Model>

Object311

Attribute/Reference	Literal/Instance
name	"Baidu_Map_Service_Model"
described_IRI	"http://api.map.baidu.com"
describing_language	Object301
contained_service	Object312

<Service>

Object312

Attribute/Reference	Literal/Instance
requested_IRI	"http://api.map.baidu.com"
domain	"Maps"
service_type	"Entity_Service"
name	"Baidu_Map"
containing_model	Object311
contained_service_operation	Object313

<Service_Operation>

Object313

Attribute/Reference	Literal/Instance
name	"Baidu_Map_Route_Display"
fully_realized_process	Object210
containing_service	Object312
annotation	Object117

<Service_Description_Language>

Object314

Attribute/Reference	Literal/Instance
name	"WSDL"
expressed_model	Object315

<Service_Model>

Object315

Attribute/Reference	Literal/Instance
name	"Check_RFID_Service_Model"
described_IRI	"http://202.114.107.230:8080/En_CloudCrm/rfid"
describing_language	Object314
contained_service	Object316

<Service>

Object316

Attribute/Reference	Literal/Instance
requested_IRI	"http://202.114.107.230:8080/En_CloudCrm/rfid?wsdf"
domain	"Logistics"
service_type	"Entity_Service"
name	"Check_RFID"
containing_model	Object315
contained_service_operation	Object317

<Service_Operation>

Object317

Attribute/Reference	Literal/Instance
name	"RFID_Check_Capability"
fully_realized_process	Object211
containing_service	Object316
annotation	Object118

Figure A.4 – Example of service model registration

