Reference number of working document: **ISO/IEC JTC 1/SC 32/WG 2 N1513**

Date: 2011-03-01

Reference number of document: **ISO/IEC WD 19763-9**

Committee identification: ISO/IEC JTC 1/SC 32/WG 2

Secretariat: ANSI

Information Technology –Metamodel Framework for Interoperability (MFI) –Part 9: On Demand Model Selection

Élément introductif — Élément principal — Partie n: Titre de la partie

Document type: Technical Report
Document subtype: 3
Document stage: Working Draft (WD)
Document language: E

Contents

1 Scope 1
1.1 Purpose....1
1.2 Scope 1
2 References....1
3 Terms, definitions and abbreviated terms 1
4 Preliminaries of ODMS....2
4.1 Relationships in RGPS....2
4.2 Semantic annotation 3
5 Framework of the ODMS....4
5.1 Model selection strategies....4
5.2 Architecture of ODMS 5
6 A typical model selection case 7

Figures

FIGURE 1 — RELATIONSHIPS IN RGPS.....2
FIGURE 2 — SEMANTIC ANNOTATION IN RGPS.....3
FIGURE 3 — ARCHITECTURE OF ODMS.....5
FIGURE 4 — THE COMMON TEMPLATE FOR ODMS.....6
FIGURE 5 — THE PROCESS-BASED TEMPLATE.....6
FIGURE 6 — THE SERVICE-BASED TEMPLATE.....7
FIGURE 7 — MODEL SELECTION FROM GOAL TO SERVICE (STEP 1).....7
FIGURE 8 — MODEL SELECTION FROM GOAL TO SERVICE (STEP 2).....7
FIGURE 9 — MODEL SELECTION FROM GOAL TO SERVICE (STEP 3).....8
FIGURE 10 — FLOW-CHART OF THE TYPICAL MODEL SELECTION CASE.....8

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 ISO/IEC Technical Report may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC Technical Report XXXXX, Information Technology –On Demand Model Selection, was prepared by Joint Technical Committee ISO/IEC JTC 1, Information Technology, Subcommittee SC 32, Data management and Interchange.

Introduction

ISO/IEC Technical Report XXXXX, Information Technology –Using RGPS for on demand model selection, was prepared by Joint Technical Committee ISO/IEC JTC 1, Information Technology, Subcommittee SC 32, Data management and Interchange.

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

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

Each part in ISO/IEC 19763 provides corresponding registration mechanism for different kinds of information resources in business domain, such as ontology, role, goal, process, and service. Users in a specific domain may express their requests in various ways since they have different background. Faced with the abundant and heterogeneous model resources, how to select appropriate services or models to satisfy users' requests becomes an important issue. Based on the registration metamodels in ISO/IEC 19763, this technical report describes a framework and procedures for the on demand model selection so as to satisfy users' requests.

Information Technology –Metamodel Framework for Interoperability (MFI) —On demand model selection

1 Scope

1.1 Purpose

Faced with the abundant and heterogeneous model resources in business domain, how to select appropriate services or models to satisfy users' requests becomes an important issue. The purpose of ISO/IEC TR 19763-9 is to describe a framework and a set of procedures for the on demand model selection based on MFI registries.

This ISO/IEC Technical Report specifies a technical guideline on how to use Role&Goal, Process, and Service metamodels to select appropriate combinations of models and/or services to meet users' goals.

1.2 Scope

The scope of ISO/IEC TR 19763-9 is limited to the model selection based on the Role&Goal models, Process models, and Service models registered according to MFI-8, MFI-5, and MFI-7, respectively.

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: Reference model

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

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

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

ISO/IEC 19763-6, Information technology – Metamodel framework for interoperability (MFI) – Part 6: Registration procedure

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

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

3 Terms, definitions and abbreviated terms

For the purposes of ISO/IEC 19763-9, the definitions contained in ISO/IEC 19763, Part 2, 3, 5, 7, and 8, and the following shall apply.

RGPS
Role, Goal, Process, and Service

ODMS
On Demand Model Selection

4 Preliminaries of ODMS

In order to show how to realize on demand model selection, some preliminaries need to be introduced first. The relationships in RGPS and semantic annotation form the basis for ODMS, where the relationships in RGPS specify how different kinds of models are connected, and semantic annotation is the basis for semantically matching between users' requests and registered models based on domain ontologies.

4.1 Relationships in RGPS

For the purposes of this technical report, RGPS is viewed as a generic term referring to the method of applying metamodels in MFI and the relationships among them into ODMS.

There are complex relationships among the various kinds of models (Figure 1). To sum up, roles take charge of their corresponding role goals, and actors prefer their respective personal goals. The goal decomposition process ends until the leaf-level subgoals are operational goals. Processes can directly or collaboratively achieve operational goals. Services can perform certain processes. More specifically,

An actor can play zero to many roles, and a role can be played by at least one actor.
A role can take charge of at least one role goal, and a role goal can be taken charge by at least one role.
A service can be involved by at least one role, and a role can involve at least one service.
A process can be involved by at least one role, and a role can involve at least one process.
An actor can prefer zero to many personal goals, and a personal goal can be preferred by at least one actor.
A process can achieve one goal, and a goal can be achieved by zero to many processes.
A service can achieve one goal, and a goal can be achieved by zero to many services.
A process can be performed by zero to many services, and a service can perform one process.

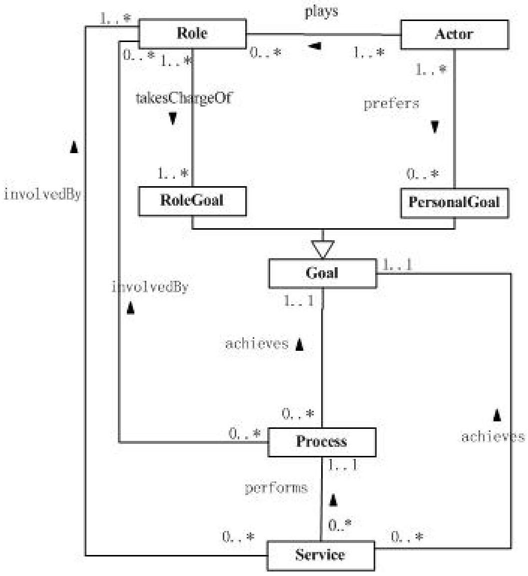

Figure 1 — Relationships in RGPS

To facilitate the model selection, the relationships in RGPS should be recorded by each registry. What's more, the corresponding multiplicity constraints in MFI-5, 7, 8 should also be maintained. These constraints can be used to check whether the registration information is in an "Approved" status. When the corresponding cardinality constraint is satisfied, the registry will be in an "Approved" status. For example, a service achieves one and only one goal. When a service is to be registered in MFI registries, although the corresponding goal instance does not exist, the service instance can still be registered in MFI registries with an "Unapproved" status. Only when there is a corresponding goal instance and the relationship between them is setup, it can be in an “Approved” status.

4.2 Semantic annotation

An essential issue in ODMS is how to match users’ requests with registration information of models in MFI registries. Semantic annotation for these registered models based on domain ontologies can be used to bridge the gap between them.

In order to annotate the registered models, two kinds of domain ontologies, entity ontology and operation ontology, are considered (Figure 2). Entity ontology mainly describes entity concepts and semantic relationships among them in a specific domain, while operation ontology mainly describes operational or functional concepts as well as semantic relationships among them in the domain. For example, as for the key attributes <*operation*, *object* > of a goal, the *operation* can be annotated by concepts in the operation ontology while the *object* can be annotated by concepts in the entity ontology. In this way, these two ontologies can provide semantic support for goal model matching. For example, a goal proposed by a user is “To book a ticket online”, and a goal that is identical to users’ request cannot be found in the model registry. However, a goal named “Reserve Ticket” can be matched. Since according to concept matching based on operation ontology, “Book” is equivalent to “Reserve”. So there is an equivalent relationship between these two goals. In other words, the goal “Reserve Ticket” can also satisfy the user’s request. Similarly, the concept of role, goal, process, resource, service, input and output in RGPS can also be annotated by domain ontologies.

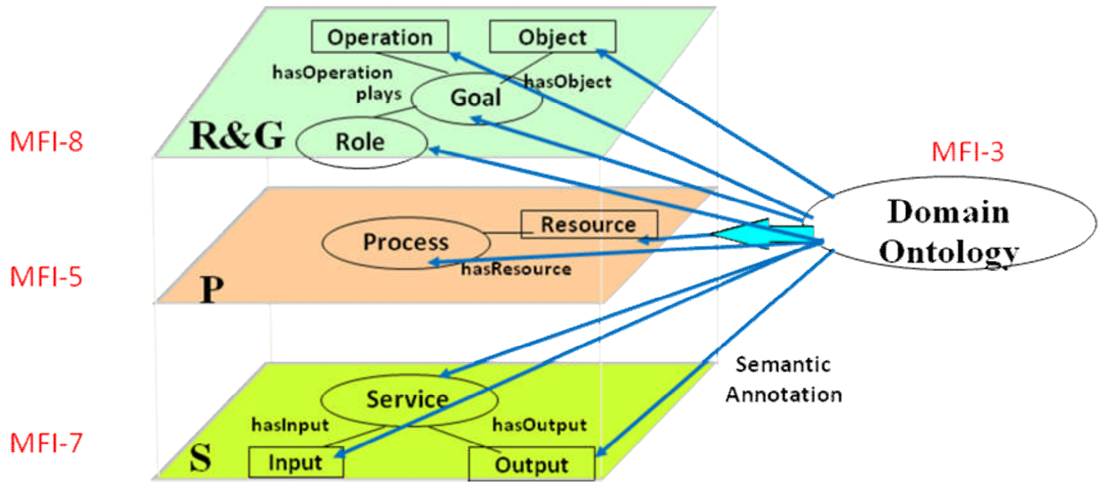

Figure 2 — Semantic annotation in RGPS

During the model selection, the concepts in models that are annotated by domain ontologies, such as role, goal, process, service, input, output, operation, and object, can be viewed as the matching items between user’s request and the candidate model. Based on the subsumption relationship among concepts in domain ontologies, the matching level for model selection can be defined, as shown in Table 1. When users submit their requests, they can also submit the expected matching level, such as exact matching (equivalentTo, subclassOf) or fuzzy matching (superclassOf / partOf). For example, the matching level between “Book Ticket” and “Reserve Ticket” mentioned above belongs to the “equivalentTo” matching level.

Table 1 — The Level of semantic matching

Matching level between the users’ request and candidate models		Related description
Exact Matching	equivalentTo	The concept in the users’ request is equivalent to that of the candidate model.
	subclassOf	The concept in the users’ request is subclass of that

		of the candidate model.
Fuzzy Matching	superclassOf	The concept in the users' request is superclass of that of the candidate model.
	partOf	The concept in the users' request is part of that of the candidate model.
Mismatching	misMatch	The concept in the users' request cannot be matched with the candidate model.

5 Framework of the ODMS

Faced with the personalized and diverse requests of users, how to select appropriate models from abundant and heterogeneous model resources becomes an important issue. To realize ODMS, the following steps are necessary to be taken. First, the registered models should be semantically annotated by domain ontologies and associated by the relationships in RGPS. And then, according to users' requests and corresponding selection strategies, appropriate models can be selected using the registration information of models to satisfy users' requests.

5.1 Model selection strategies

As mentioned before, the registered relationships among models play an important role during the model selection process. The model selection strategies are defined based on the relationships in RGPS, as shown in Table 2.

Table 2 — The relationships in RGPS

	Role	**Goal**	**Process**	**Service**
Role	interaction	takesChargeOf		
Goal		decomposition/ constraint		
Process	involvedBy	achieves		
Service	involvedBy	achieves	performs	

During the model selection, users' request can be expressed by means of a goal, a process, or a service. When users' request is matched to a goal in a goal model in MFI registries, the following steps can be taken, such as querying the subgoals that the goal can be decomposed into, querying the upper goal that the goal is derived from, querying the goals that the goal depends on, querying the role that takes charge of the goal, querying the process that achieves the goal, and querying the service that achieves the goal.

When users' request is matched to a process in a process model in MFI registries, the following steps can be taken, such as querying the roles that involve in the process, querying the goals that can be achieved by the process, querying the subprocesses that the process can be decomposed into, and querying the services that can perform the process.

When users' request is matched to a service in a service model in MFI registries, the following steps can be taken, such as querying the roles that involve in the service, querying the goals achieved by the service, and querying the process performed by the service.

Based on the model selection strategies, the whole model selection process may consist of several iteration steps, and each step will follow the relationships in RGPS.

5.2 Architecture of ODMS

In this section, we will discuss the architecture of ODMS. As shown in Figure 3, the architecture of ODMS consists of three parts: model selection engine, model selection record registry, and MFI model registries, where the model selection engine is used to receive and analyze users' requests, and find corresponding models or services according to the requests, the model selection record registry is used to store the history selection records, and the MFI registries stores the registration information of RGPS.

The following steps describe the general procedure of the model selection based on the architecture. As shown in Figure 3, the user's request is firstly submitted to the model selection engine. Then the request will be sent to the model selection record registry to search the related history records. If there is a similar request, the recorded models will be returned to the user. Otherwise, the model selection engine will search the candidate models from MFI registries. Based on the relationships among RGPS and semantic annotation provided by domain ontology, the appropriate models will be found and returned to the user. Meanwhile, the corresponding selection record will also be registered in the selection record registry.

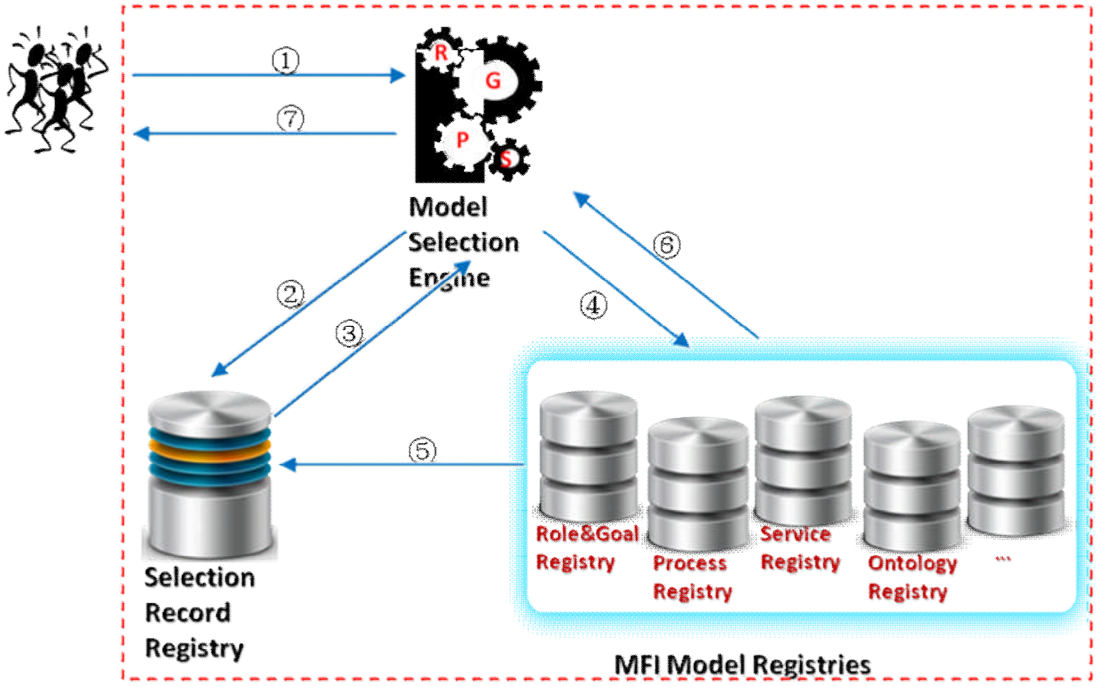

Figure 3 — Architecture of ODMS

In the process of ODMS, users need to be involved in the process. In order to elicit users' request, it is necessary to define a model selection template to collect users' requests. For example, a model selection template may consist of two parts: a mandatory part and an optional part, as shown in Figure 4. The mandatory part is mainly used to describe requests that must be submitted by the user, which may include users' request type (expressed as a goal, a process, or a service), request description and expected return type (a goal model, a process model, or a service model), while the optional part is used to describe the constraints such as the expected matching level.

In the mandatory part, request type denotes that there are three choices for users when they propose their requests, that is, their request can be expressed as a goal, a process, or a service. Request description shows the content of users' requests. There are also three kinds of the expected return results including a goal model, a process model, or a service model, and users can select any of them or a combination of them. In the optional part, users can also specify the matching level such as exact matching or fuzzy matching between their requests and the registration information. Moreover, in the service request, QoS request may also need to be defined. Please note that if there are multiple requests, a slash ("/") can be used to separate them.

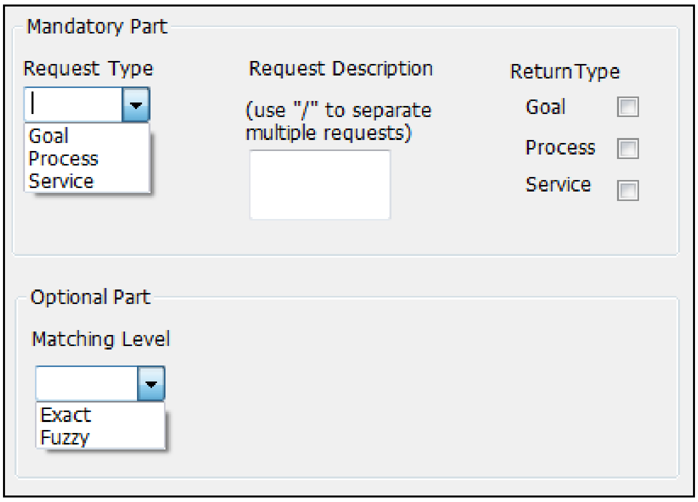

Figure 4 — The common template for ODMS

When users select the goal as the request type, they need to input the expected goal name in the request description part. For other parts, they just need to follow the instruction on the common template.

When users select the process as the request type, the template is slightly different. As shown in Figure 5, users can either input the expected process name to describe their requests, or fill the expected input and output of the process to describe their requests. For other parts, they just need to follow the instruction on the process-oriented template.

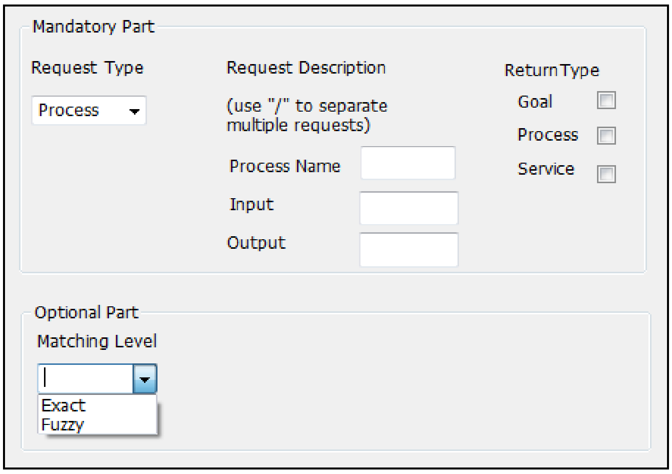

Figure 5 — The process-oriented template

As shown in Figure 6, the template for service request is similar to that of process, users can either input the expected service name, or fill the expected input and output of the service to describe their requests. Users can also input their QoS request on the expected services. The QoS request can be expressed as a qualitative manner or a quantitative manner. In a qualitative manner, the QoS type and degree should be defined, where the QoS type can be response time, security, performance, and so on, and the degree can be high, low or medium. In a quantitative manner, the QoS type, comparison operator, value, and unit should be defined, where the comparison operator can be *equalsTo*, *lessThan*, and so on, the value denotes a specific numeric value such as an integer or a decimal, and the unit is related to the QoS type, e.g., the second for the response time. For other parts, they just need to follow the instruction on the service-oriented template.

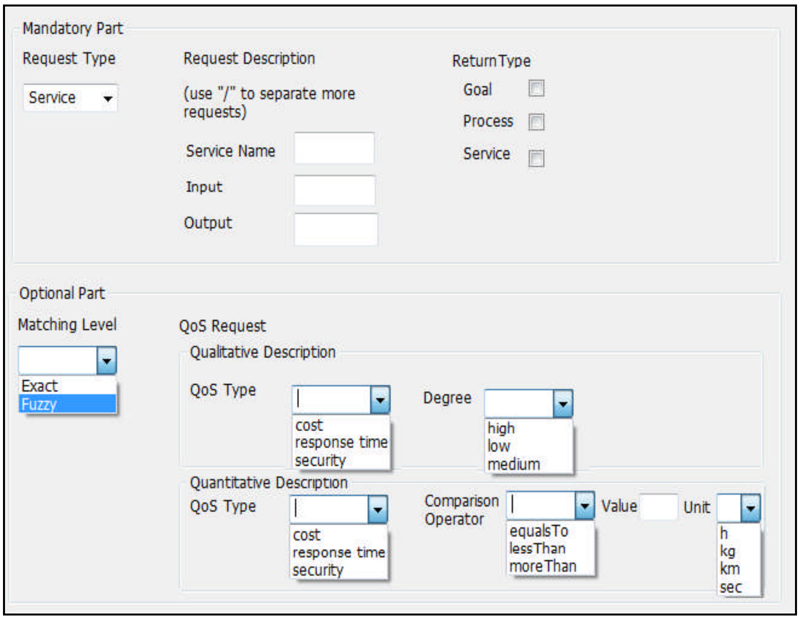

Figure 6 — The service-oriented template

6 A typical model selection case

According to the ODMS architecture, a typical model selection case is illustrated in this section. In this case, the user's request can be represented as a goal, and the service is the expected output. The following steps will be taken according to the relationships in RGPS.

In Step 1, the user's request can be matched to a goal in MFI-8 registry by ontology-based semantic matching. According to the Goal-Service relationship (*achieves*), if corresponding services that can achieve the goal can be found in MFI-7 registry, then the result will be directly returned to the user, as shown in Figure 7.

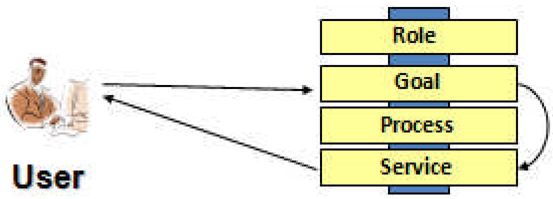

Figure 7 — Model selection from goal to service (Step 1)

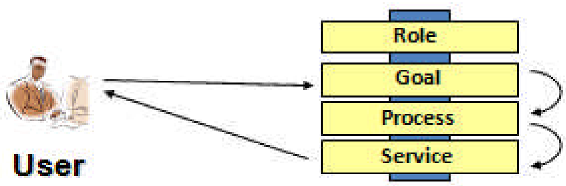

Figure 8 — Model selection from goal to service (Step 2)

In Step 2, if the corresponding services cannot be found in MFI-7 registry, the selection engine will visit MFI-5 registry to find which process can achieve the goal by the Goal-Process relationship (*achieves*), and then revisit MFI-7 registry to search the services that can perform the processes by the Process-Service relationship (*performs*) , as shown in Figure 8.

If the returned results from the two steps cannot satisfy the user's request, then Step 3 and Step 4 will be taken.

In Step 3, the model selection engine will visit MFI-8 registry to find which role can take charge of the goal by the Role-Goal relationship (*takesChargeOf*), and find and supplement other goals that are related to the role by the Role-Goal relationship (*takesChargeOf*). The following selection process is similar to Step 1 and Step 2, as shown in Figure 9.

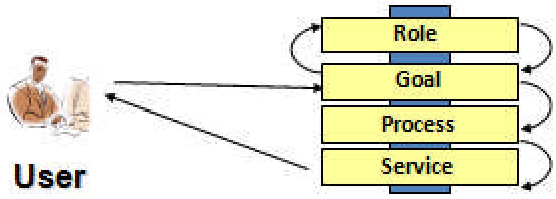

Figure 9 — Model selection from goal to service (Step 3)

In Step 4, the subgoals decomposed by the goal according to Goal-Goal relationship are also considered. Please note that the process of decomposition will not finish until the leaf-level goals are operational goals. The following selection process is similar to Step 1 and Step 2.

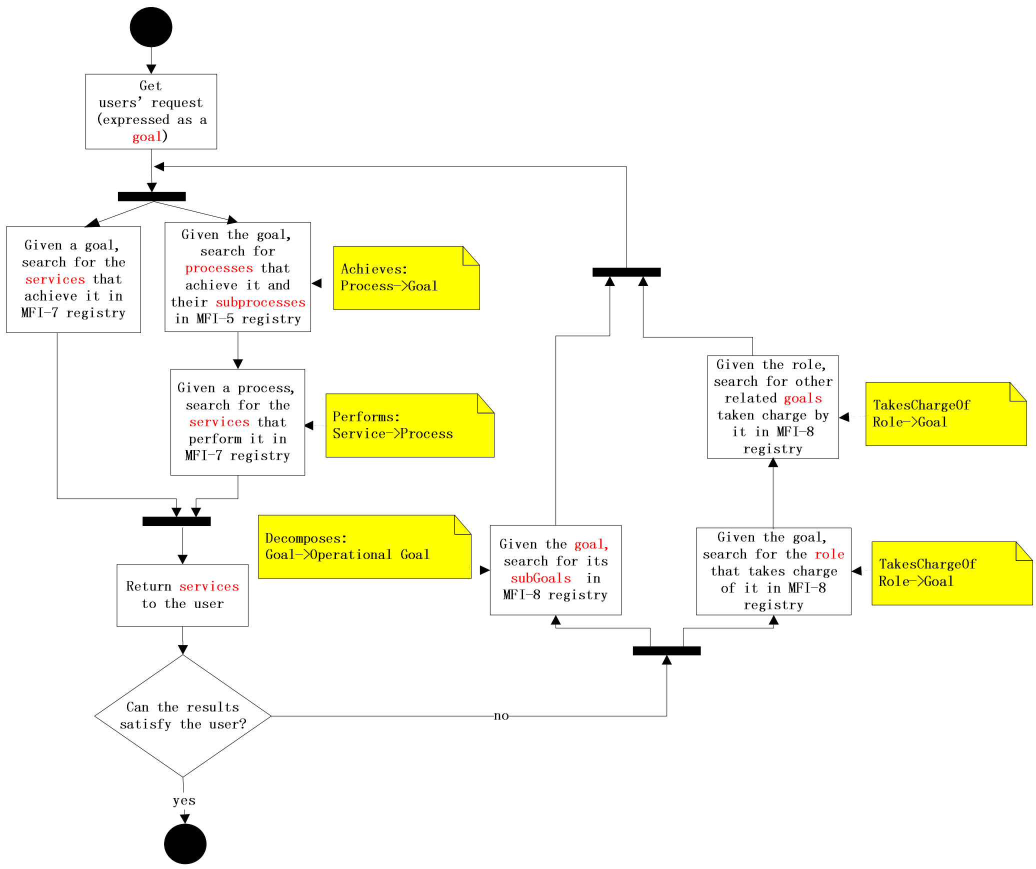

Figure 10 — Flow-chart of the typical model selection case

The whole procedure for model selection from goal to service is shown in Figure 10. Please note that during the model selection process, semantic matching between users' request and registration information, as well as between the different registration information is used based on domain ontology.

Annex A
(informative)
Examples of On Demand Model Selection

In this section, the following examples are used to illustrate how to find appropriate models according to users' request based on MFI-9. In these two examples, users' requests are represented as goals, and services are their expected result. Please note that there are also some other cases that take other kinds of models in RGPS as input or expected result.

For a simple example, suppose a user wants to find a service model on weather information. The user's request can be expressed as a goal "to query the weather information", which can be submitted to the model selection engine. The request will be firstly sent to the selection record registry. If there is a similar request, the recorded models will be returned to the user. Otherwise, the model selection engine will search for candidate models from MFI registries.

Firstly, the model selection engine will visit MFI-8 registry to match the request with the registered role_goal model, as shown in Figure A.1. Then according to the Goal-Service relationship, the selection engine will visit MFI-7 registry to find the corresponding candidate service model that achieves the goal. As shown in Figure A.1, the registered service01 can achieve the Role_Goal00 "QueryWeatherInformation_Goal". Finally, the service will be returned to the user, and the corresponding selection result will also be registered in the selection record registry.

Role_Goal00	
name	***QueryWeatherInformation_Goal***
goal_type	*Functional Goal*
is_operational	*true*
hasOperation	*Query*
hasObject	*Weather Information*
hasManner	

Service01	
URI	*http://example.org/ QueryWea_InfoWS*
name	*QueryWea_Info Web Service*
hasAggregatedOperation	*Operation00*
achieves	***QueryWeatherInformation_Goal***
performs	*QueryWeatherInformation_Process*

Figure A.1 — A simple Example of registered models in MFI registries

For a relatively complex example, suppose that a user plans to attend a conference in another city. The following tasks are needed to organize the trip, a) book a flight, b) book a hotel, and c) rent a car. The user's request can be expressed as a goal "to arrange a travel plan", and it can be submitted to the model selection engine. Assume that currently there is no related history record in the selection record registry, so the selection engine will search for candidate models from MFI registries.

Firstly, the model selection engine will visit MFI-8 registry to match the request with the registered role_goal model, as shown in Figure A.2. The goal "*ArrangeTravelPlan_Goal*" can be matched. Then according to the Goal-Service relationship, the selection engine will visit MFI-7 registry to find the corresponding service model that achieves the goal. Unfortunately there is no related service that can achieve the goal. Then according to the Goal-Process relationship, the selection engine will visit MFI-5 registry to find the corresponding process model that can achieve the goal. A composite process "*ArrangeTravelPlan_Process*" can be matched and the selection engine will also find the subprocesses that compose the composite process. Then according to the Process-Service relationship, the selection engine will visit MFI-7 registry to find the corresponding service models that can perform the processes. For example, the registered services service02, service03, and service04 can perform the processes "*BookTicket*", "*QueryHotel*" and "*RentCar*", respectively. Finally, the composite process as well as the corresponding services will be returned to the user, and the corresponding selection result will also be registered in the selection record registry.

Role_Goal01	
name	***ArrangeTravelPlan_Goal***
goal_type	*Functional Goal*
is_operational	*false*
hasOperation	*Arrange*
hasObject	*Travel_Plan*
hasManner	

Process01	
ID	*1*
name	*ArrangeTravelPlan_Process*
administration_Record	*#*
achieves	***ArrangeTravelPlan_Goal***
type	*Composite*
describedBy	*Process_Model01:* *ArrangeTravelPlan_ProcessModel*
restrictedBy	*Sequence01*

Sequence01	
constructType	*Sequence*
nodeSet	*Process02: SearchDestination*
	Process03: PlanTravelRoute
	Process04: DispayTravelRoute

Process03	
ID	*3*
name	*PlanTravelRoute_Process*
administration_Record	*#*
achieves	***PlanTravelRoute_Goal***
type	*Composite*
describedBy	*Process_Model01:* *ArrangeTravelPlan_ProcessModel*
restrictedBy	*Split01*

Split01	
constructType	*Split*
nodeSet	***Process05: BookTicket***
	Process06: QueryHotel
	Process07: RentCar

Service02	
URI	*http://example.org/BookingTicketWS*
name	*Booking Ticket Web Service*
hasAggregatedOperation	*Operation01*
achieves	*BookingTicket_Goal*
performs	***BookTicket_Process***

Service03	
URI	*http://example.org/ QueryWea_InfoWS*
name	*Query Hotel Web Service*
hasAggregatedOperation	*Operation02*
achieves	*QueryHotel _Goal*
performs	***QueryHotel _Process***

Service04	
URI	*http://example.org/ RentCarWS*
name	*Rent Car Web Service*
hasAggregatedOperation	*Operation03*
achieves	*RentCar_Goal*
performs	***RentCar _Process***

Figure A.2 — A complex example of registered models in MFI registries