
MFI-9: On Demand Model Selection

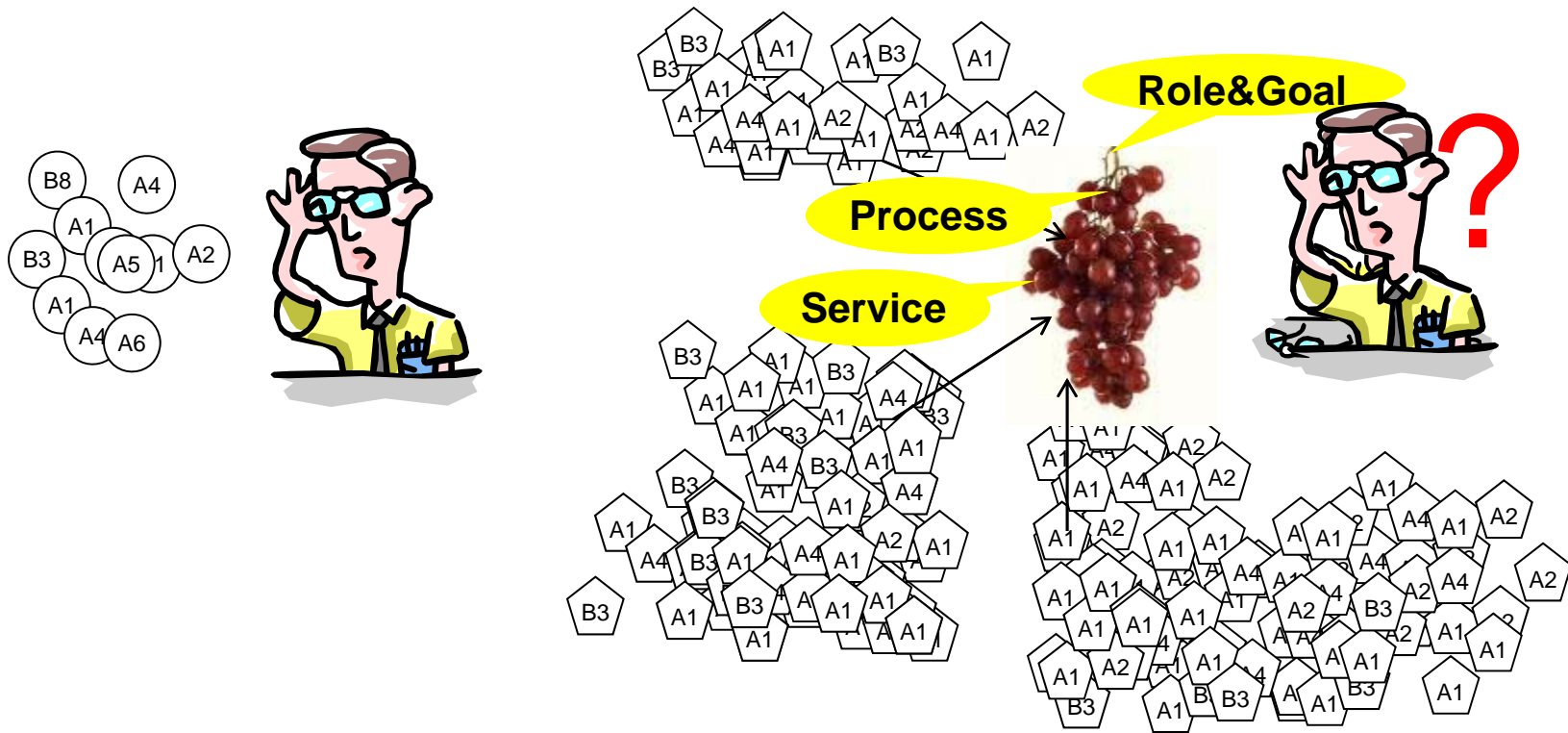
WANG Jian, HE Keqing, WANG Chong, LIANG Peng, FENG Zaiwen
State Key Lab of Software Engineering (SKLSE),
Wuhan University, China
2010-05-26

Motivation of ODMS

Objectives of RGPS

- RGPS provides a framework for the **management, discovery and sharing of service-related models**.
 - RGPS can be used to **describe services in different views(Role, Goal, Process,Service)**, and it can also be used to **express users' needs**.
 - According to the mappings between users' needs and registration information of models, appropriate models can be **selected to satisfy** users' needs.
 - **Semantic annotation**: some elements in these models can be annotated by registered domain ontologies.

Difficulty of Model Selection



Why **On Demand Model Selection** is important?

- Personalized and diverse requests
- Abundant and heterogeneous model resources

Basic idea of ODMS

- Faced with the personalized and diverse requests of users, how to select appropriate service/models from abundant and heterogeneous model resources
 - Register these models in MFI registries
 - Annotate the models based on registered ontology
 - According to users' request, select appropriate models based on the relationships among the models

Role of RGPS in ODMS

Top-Down

User

Who

What to do

How to do

Which service is needed

Matching Role

Matching Goal

Matching Process

Matching Service

RGPS

For Whom

What can do

How to do

How to use

Service

Bottom-Up

Scope of MFI-9

Scope

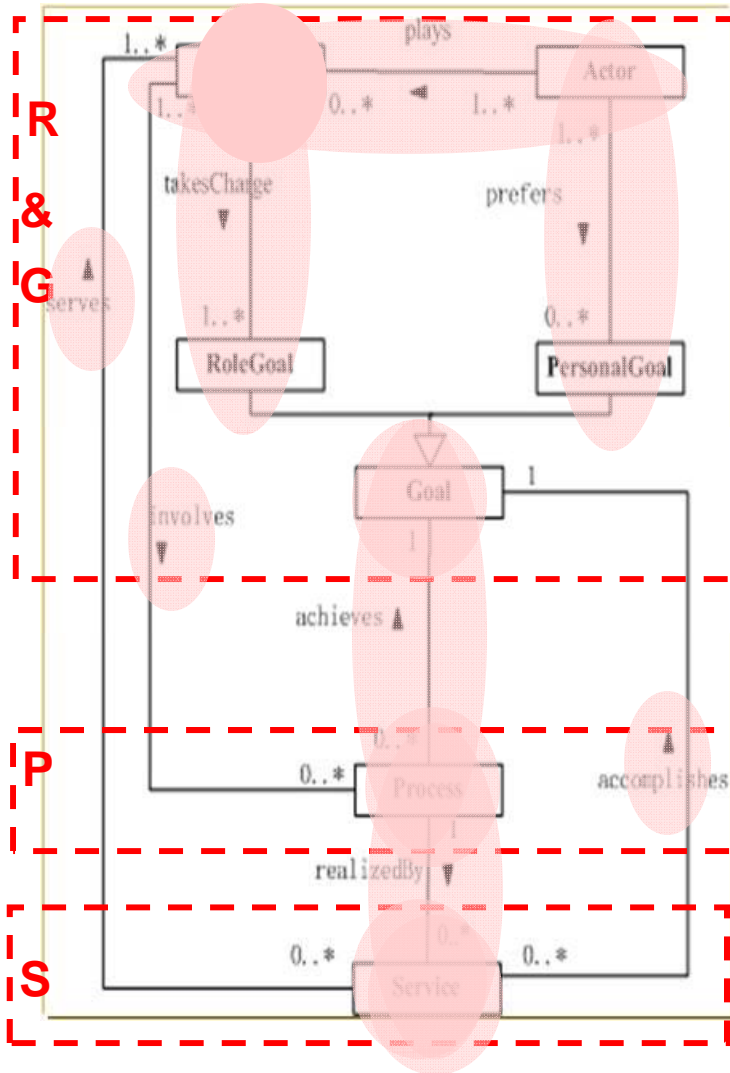
- This ISO/IEC Technical Report specifies a technical guideline on how to use R, G, P, and S metamodels to select appropriate combinations of models and/or services to meet users' goals.
- It also specifies how to use the RGPS infrastructure to support operational harmonization and interoperability within and between industries.

From resolution of Wuhan Meeting, September, 2008

Content of ODMS

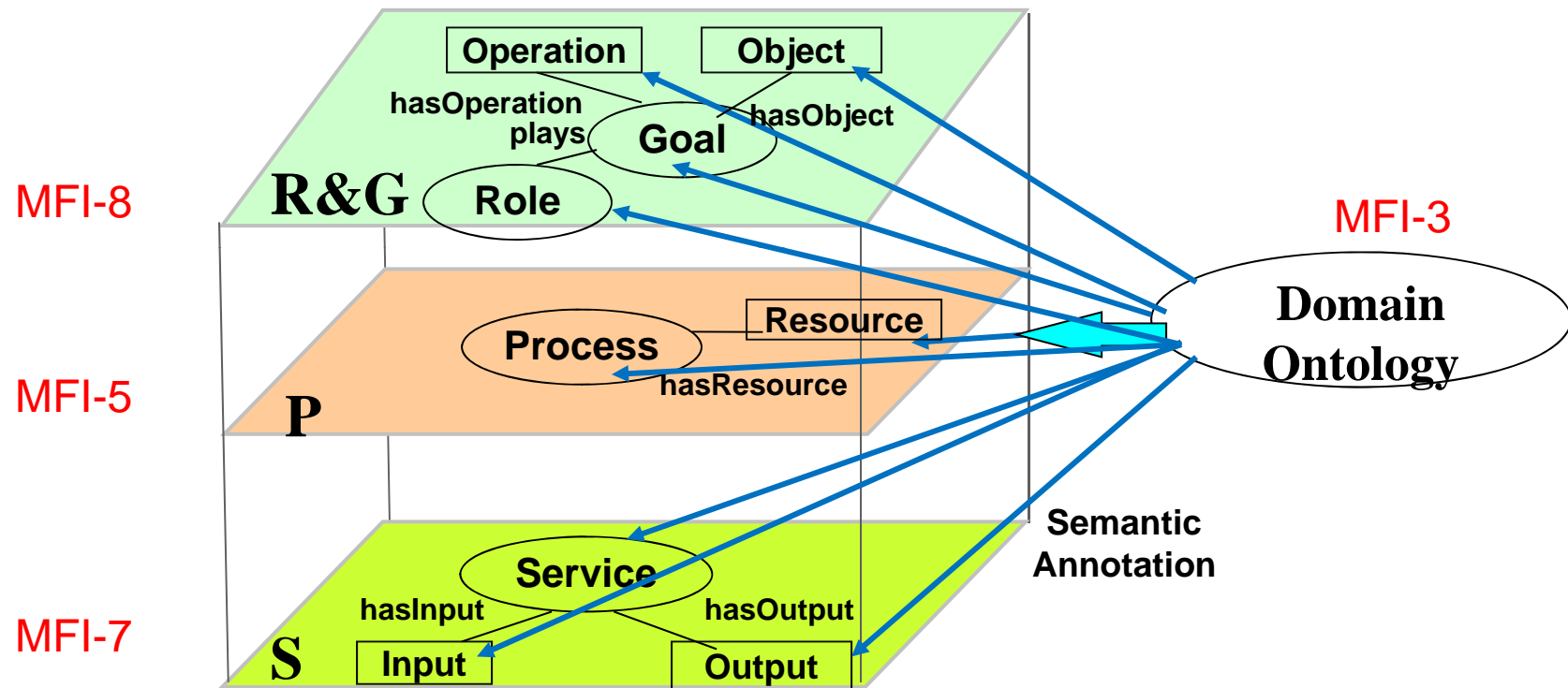
Relationships in RGPS

- An *actor* can *play* zero to many *roles*.
- A *role* can *take charge of* at least one *role goal*.
- An *actor* can *prefer* zero to many *personal goals*.
- A *role* can *involve* at least one *process*.
- A *service* can *serve* at least one *role*.
- A *process* can *achieve* one *goal*.
- A *process* can be *realized by* zero to many *services*.
- A *service* can *accomplish* one *goal*.



The relationships are the foundation for model selection in MFI-9

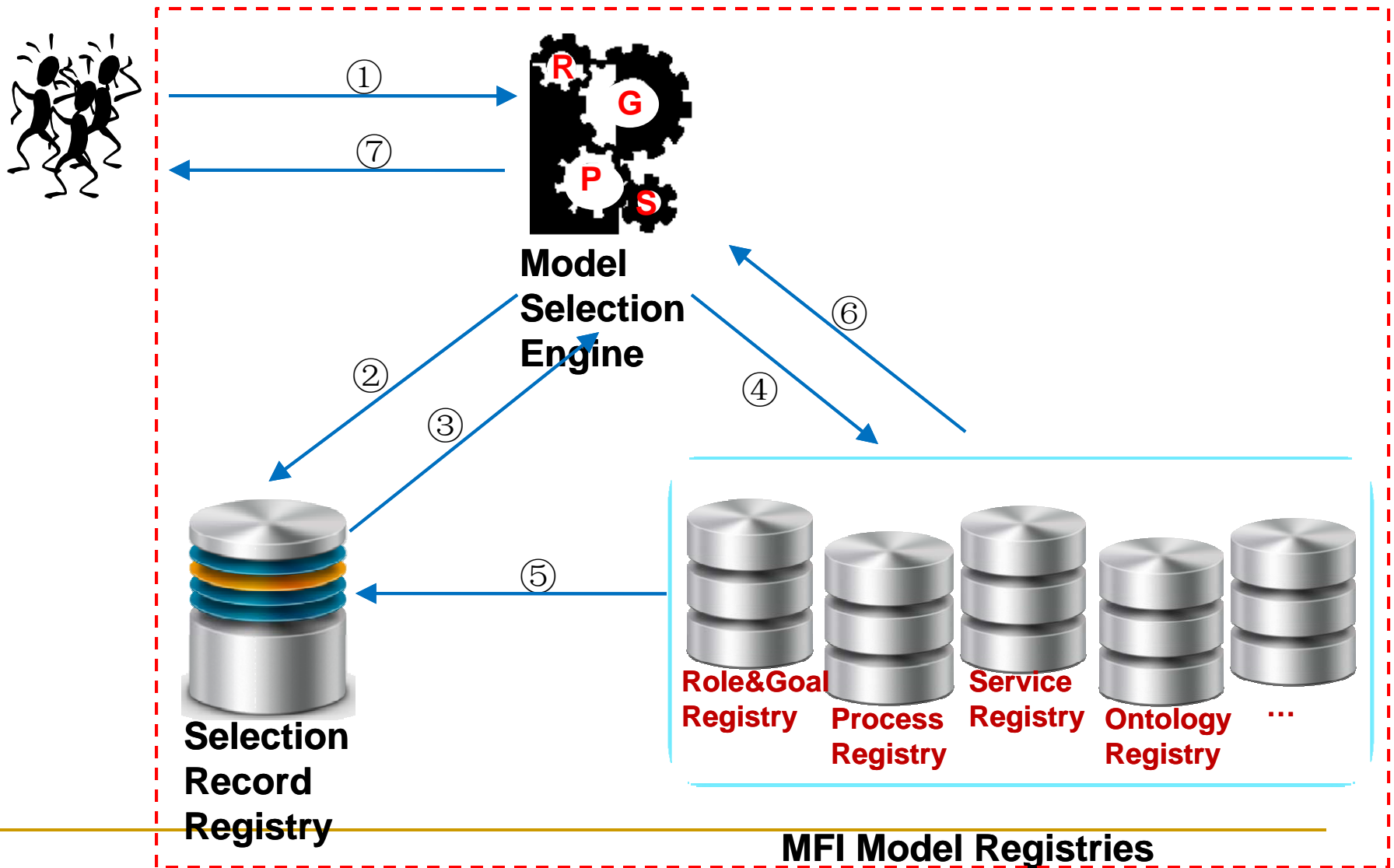
Semantic annotation of models



Contribute to

- mapping between models
- semantic discovery of models

Architecture of ODMS



Model Selection Strategies (1 / 5)

- Users' request can be expressed as **role/goal/process/service**.
- The whole model selection process consists of many steps
 - Each step may follow the **relationships** in RGPS
 - When there is no model/service that satisfies users, the registries will be revisited to find some **possibly useful** models/services.

Model Selection Strategy (2/5)

■ Role

- Role->Role
- Role->Goal
- Role->Process
- Role->Service

	Role	Goal	Process	Service
Role	interaction	takesCharge	involves	servedBy

Model Selection Strategy (3/5)

- Goal
 - Goal->Role
 - Goal->Goal
 - Goal->Process
 - Goal->Service

	Role	Goal	Process	Service
Goal	takenChargeBy	decomposes	achievedBy	accomplishedBy

Model Selection Strategy (4/5)

■ Process

- Process->Role
- Process->Goal
- Process->Process
- Process->Service

	Role	Goal	Process	Service
Process	involvedBy	achieves	composedBy	realizedBy

Model Selection Strategy (5/5)

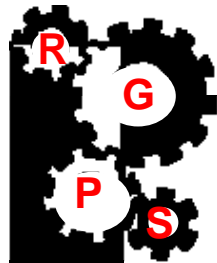
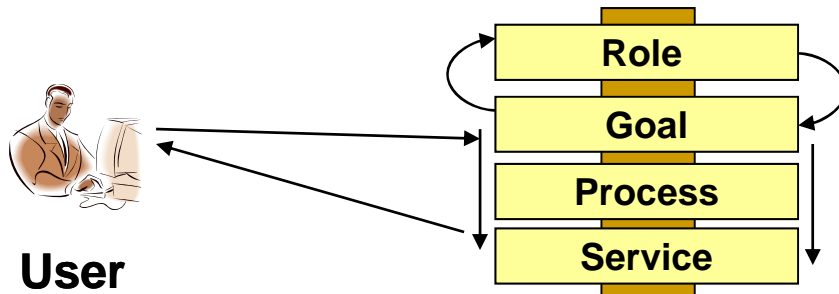
- Service
 - Service->Role
 - Service->Goal
 - Service->Process

	Role	Goal	Process	Service
Service	serves	accomplishes	realizes	

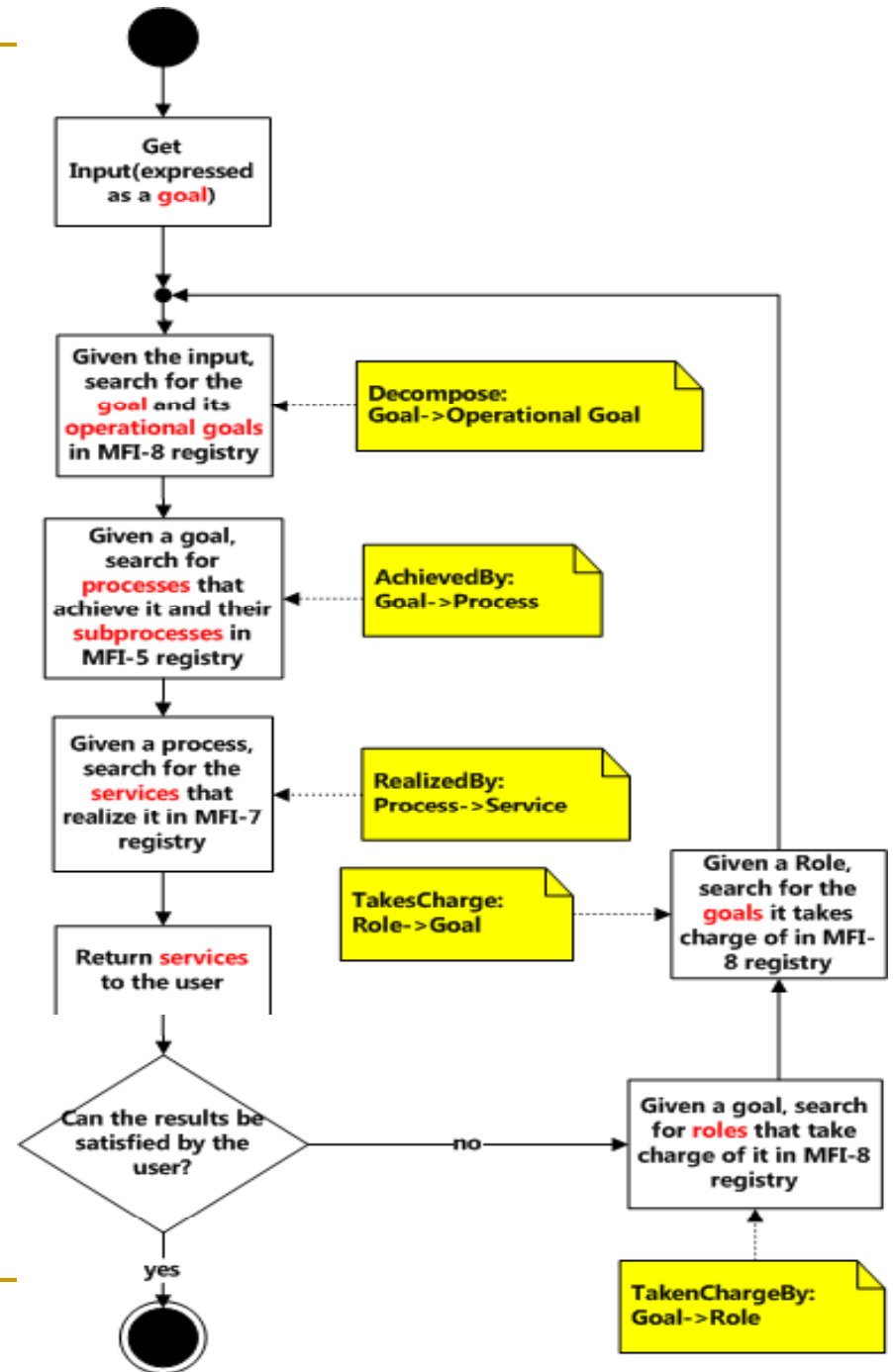
Template of Model Selection

- Model Selection
 - Input
 - Expected Output
 - Criterion
 - Similarity
 - Stopping criterion

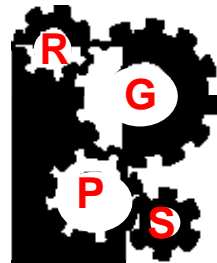
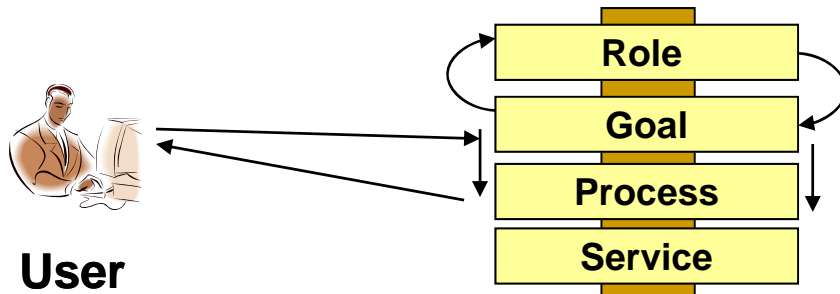
Typical Model Selection Process (Case 1)



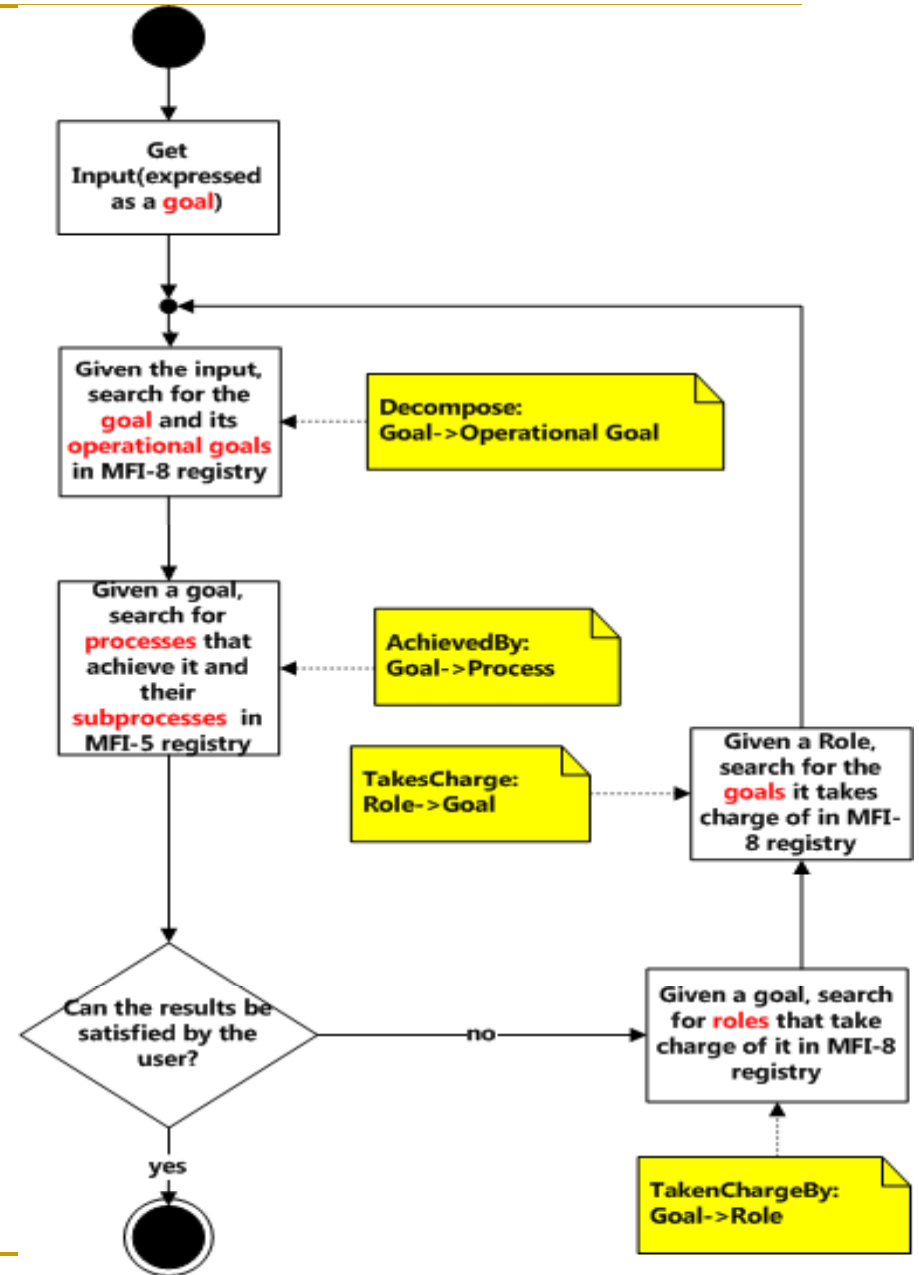
Input: Goal
Expected Output: Service



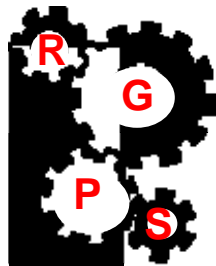
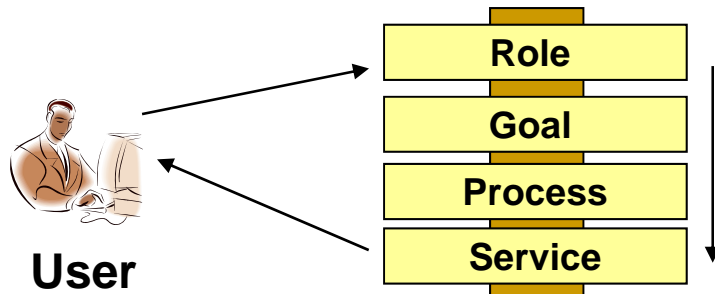
Typical Model Selection Process (Case 2)



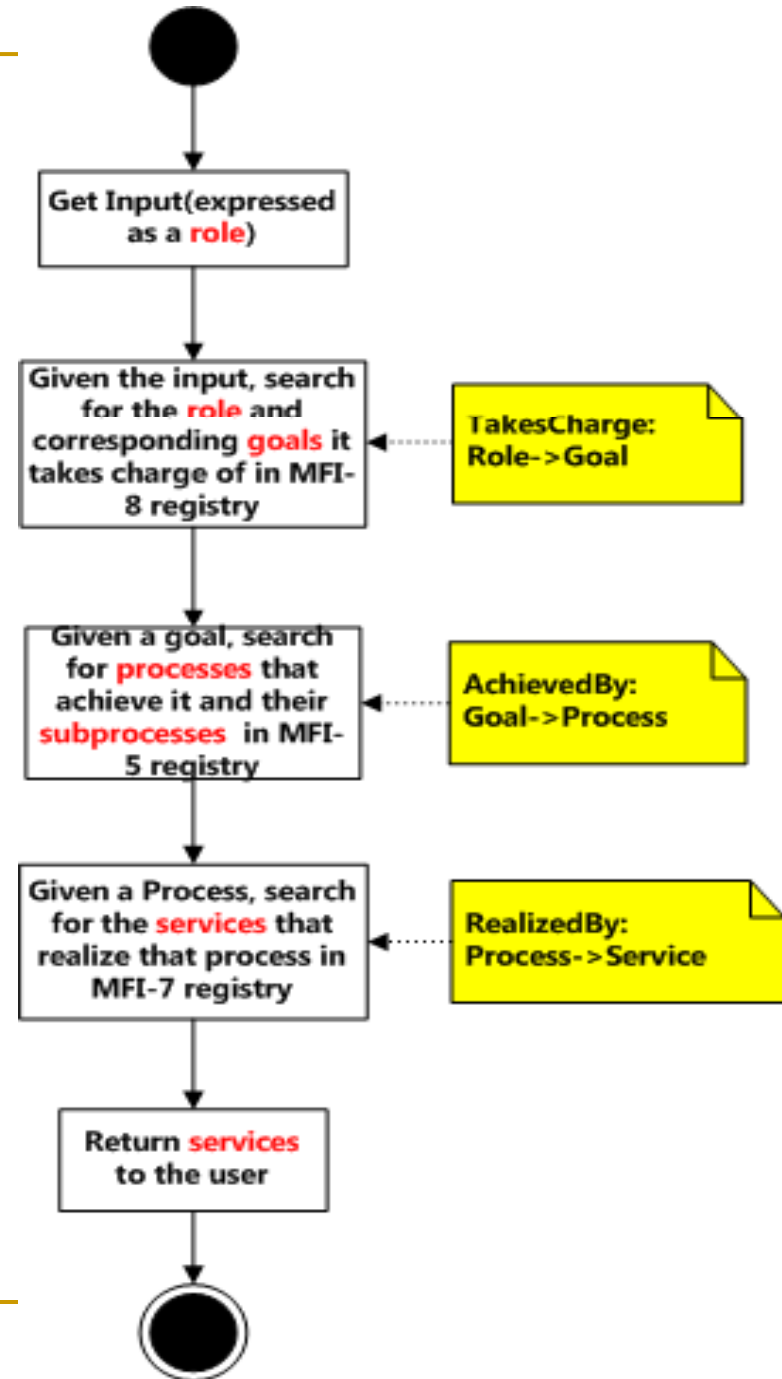
Input: Goal
Expected Output: Process



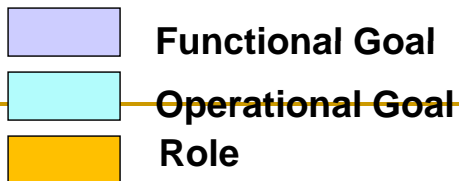
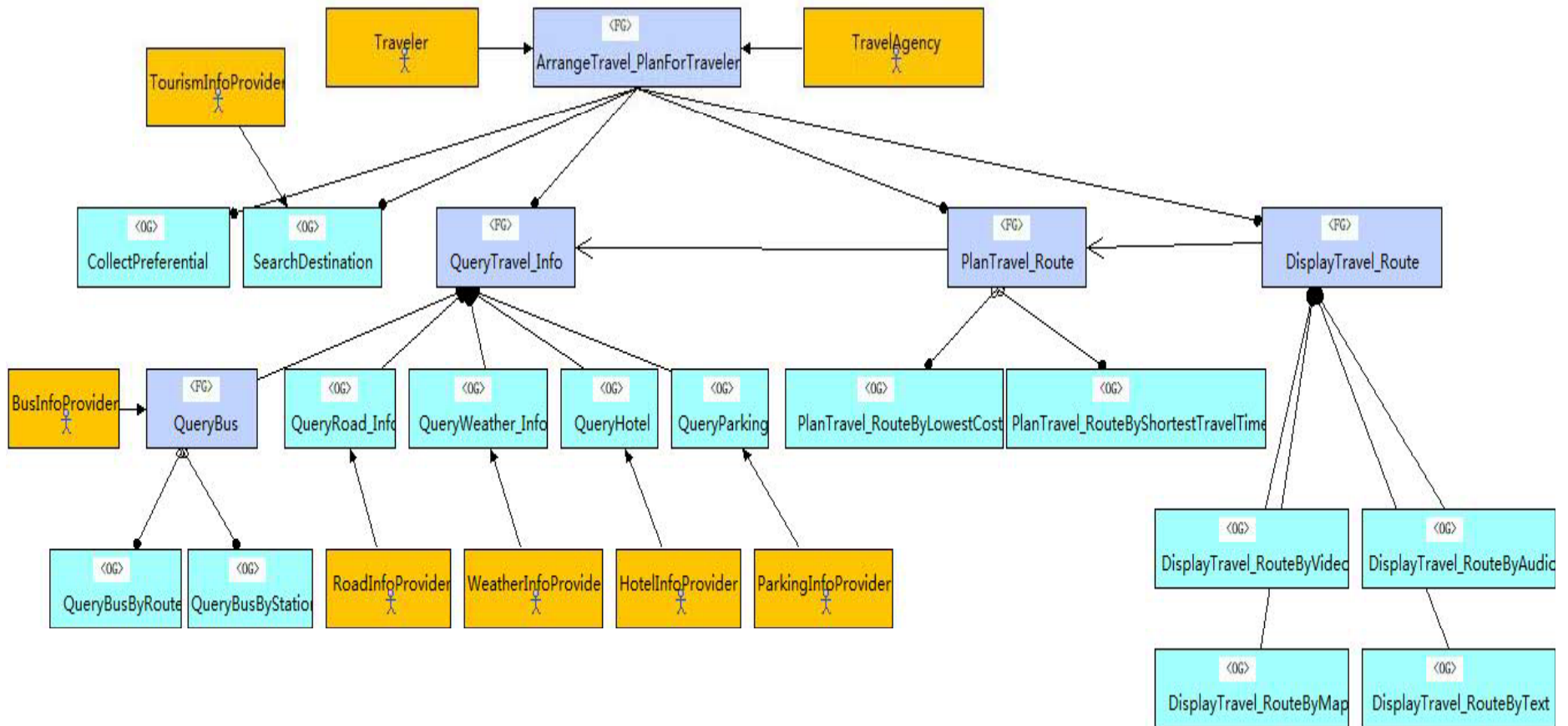
Typical Model Selection Process (Case 3)

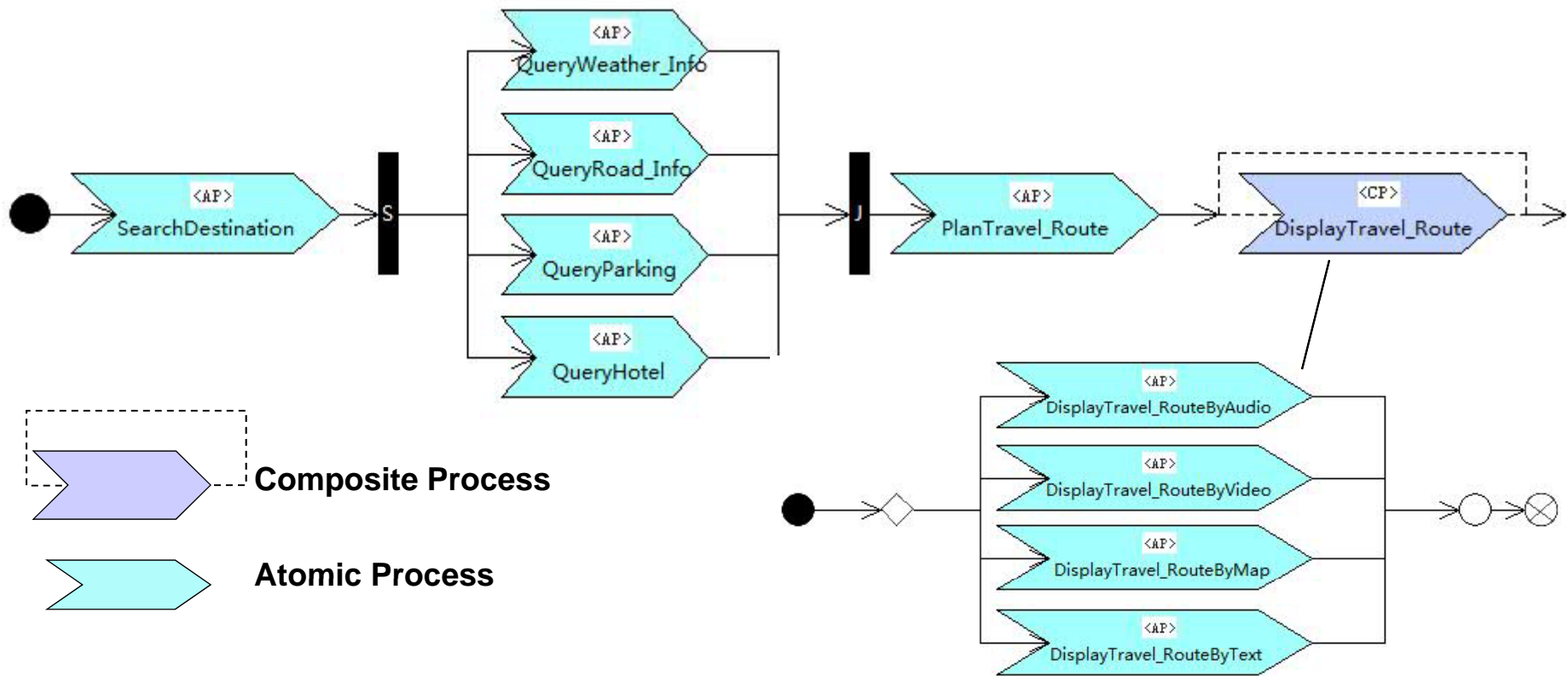


Input: Role
Expected Output: Service



An example





Service01: QueryWeather_Info		Service02: QueryHotel	
URI	http://www.daml.org/services/owl/1.2/Service.owl#WeatherInforSe	URI	http://www.daml.org/services/WSMO/QueerHotelervice
Provider	Wuhan weather station	Provider	...
Description_Language	OWL-S	Description_Language	WSML

Example

- User's Input: Goal(Arrange Travel Plan)
- User's Expected Output: Process
- Step1: Match users' input with the Goal in registry

Role_Goal00	
id	<i>RG00</i>
name	<i>ArrangeTravel_Plan</i>
goal_type	<i>Functional Goal</i>
is_operational	<i>false</i>
hasOperation	<i>Arrange</i>
hasObject	<i>Travel_Plan</i>
hasManner	

Process01	
ID	1
name	ArrangeTravel_Plan
administration_Record	#
achieves	ArrangeTravel_Plan
type	C
describedBy	Process_Model01: ArrangeTravel_Plan_ProcessModel
creates	Resource01: Audio_TravelRout/ Resource02: Vedio_TravelRout/ Resource03: Map_TravelRout/ Resource04: Text_TravelRout
restrictedBy	Sequence01

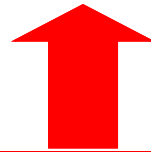
Sequence01	
constructType	Sequence
nodeSet	Process02: SearchDestination
	Process03: QueryTavellInfo
	Process04: PlanTravel_Rout
	Process05: DispayTravel_Rout

Process02	
ID	2
name	<u>SearchDestination</u>
<u>administration_Record</u>	#
achieves	<u>SearchDestination</u>
type	A
creates	Resource05: Destination

- Step2: Search for the process achieve that goal.
- Step3: Find the processes that compose the process
- Step4: Return the process to the user.

MFI-9 vs. the other parts of MFI

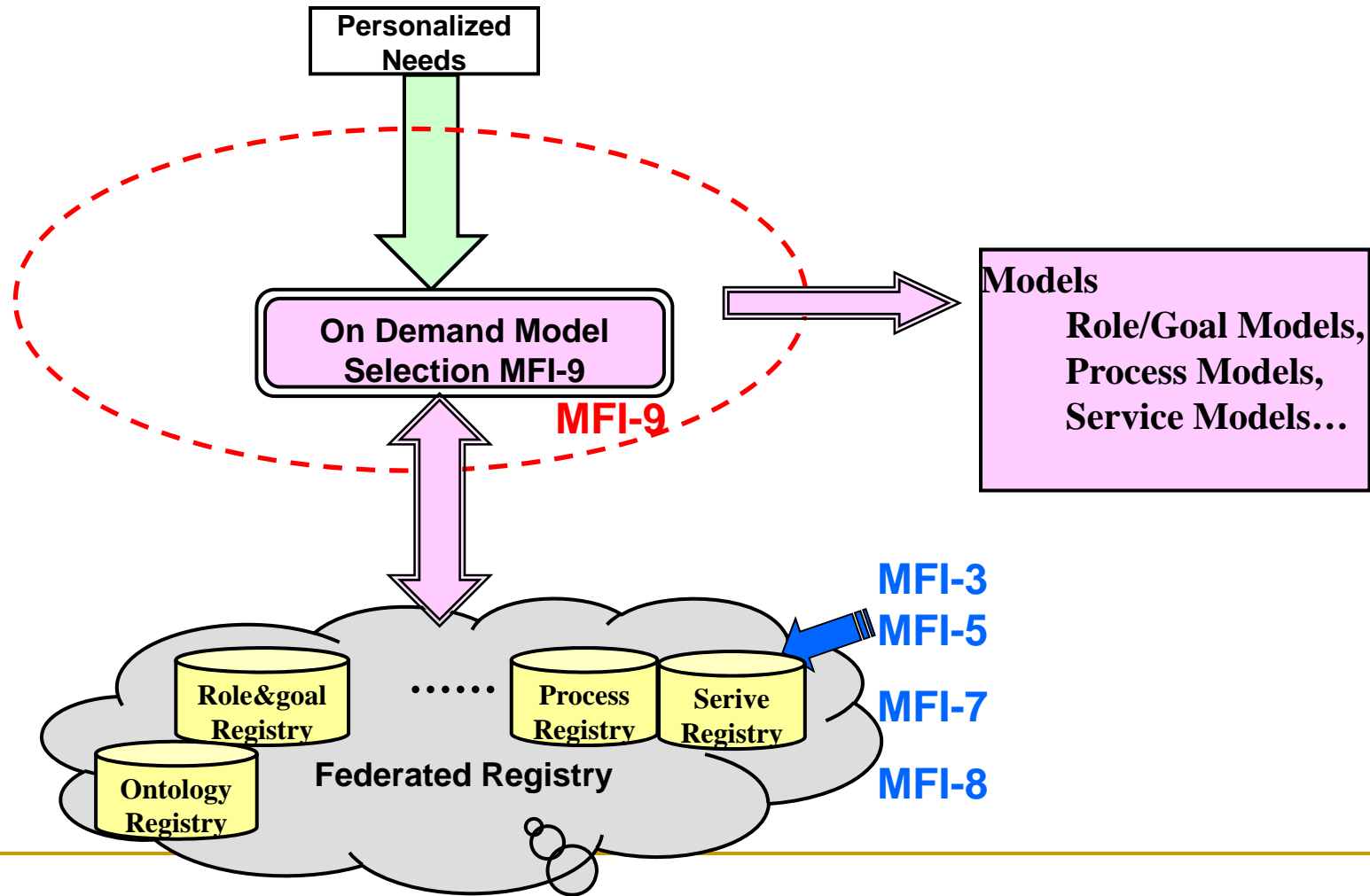
**Basic Interoperation Pattern
(MFI-1,2,3,4,6)**



Based on

**Service Oriented Interoperation
(MFI-5, 7, 8, 9)**

Relationship between ODMS and the other parts of MFI



Thank You!