



软件工程
国家重点实验室(武汉大学)
STATE KEY LAB OF SOFTWARE ENGINEERING (WUHAN UNIVERSITY)

Discussion about MFI-7: Metamodel for Service Registration

Wang Jian, He Keqing,
He Yangfan, Wang Chong
SKLSE, Wuhan University, China
2009.8.21

Outline

- Background
- Content of MFI-7
- Future Work

Content

- Background
 - What is Service?
 - Why We need MFI-7?
- Content of MFI-7
- Future Work

Service is Flourishing...

- New Business Model - Cloud Computing

- SaaS (Software as a Service)
- PaaS (Platform as a Service)
- IaaS (Infrastructure as a Service)
- RaaS (Registry as a Service)
- ...

} XaaS

- New Software Paradigm – SOA

- ...

Characteristics:

- Encapsulation of functionality
- Transparency to users

Definitions of Service

- In Information Science (e-service)
 - an **interactive, content-centered and Internet-based customer service**, driven by the customer and integrated with related organizational customer support processes . [Ruyter, 2001]
 - **modular**, nimble, electronic services that perform work, achieve tasks, or complete transactions. [Piccinelli, 2001]
- In Computer Science (Web service)
 - loosely coupled, **reusable software components** that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols. [The Stencil Group, 2001]

Definition of Service in MFI-7

- Service is defined as
 - a modular functionality entity with provision of value that can be invoked by accessing its interface

- Key points

- Provision of value
- Transparency
- Interaction/negotiation

Capability

Interface

Interaction

Web Service: SOAP, RESTful, Atom,...

■ SOAP Web Service

- Operation-centric

- For example, to query the detail information of a book with the ISBN 0321357890, the URI will be

<http://www.bookstore.com/action/query?t=b&id=11117645532&qp=0321357890>.

- SOAP+WSDL

■ RESTful (Representational State Transfer) Web Service

- Resource-centric

- In the example, the URI will be

<http://www.bookstore.com/books/0321357890/>

- Light-weight Web service

- HTTP-based

- Accessed by a fixed URI using Put, Post, Get, Delete

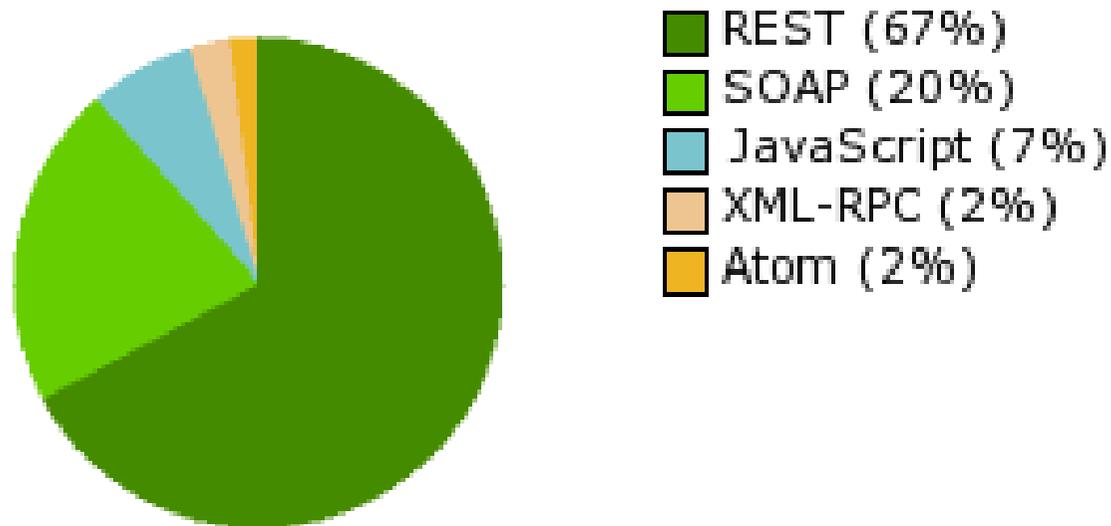
■ Atom Data Service

- Encapsulation of database or other data sources

Current Usage Status of Web Service API

- From Programmableweb
(www.programmableweb.com)

Protocol Usage by APIs

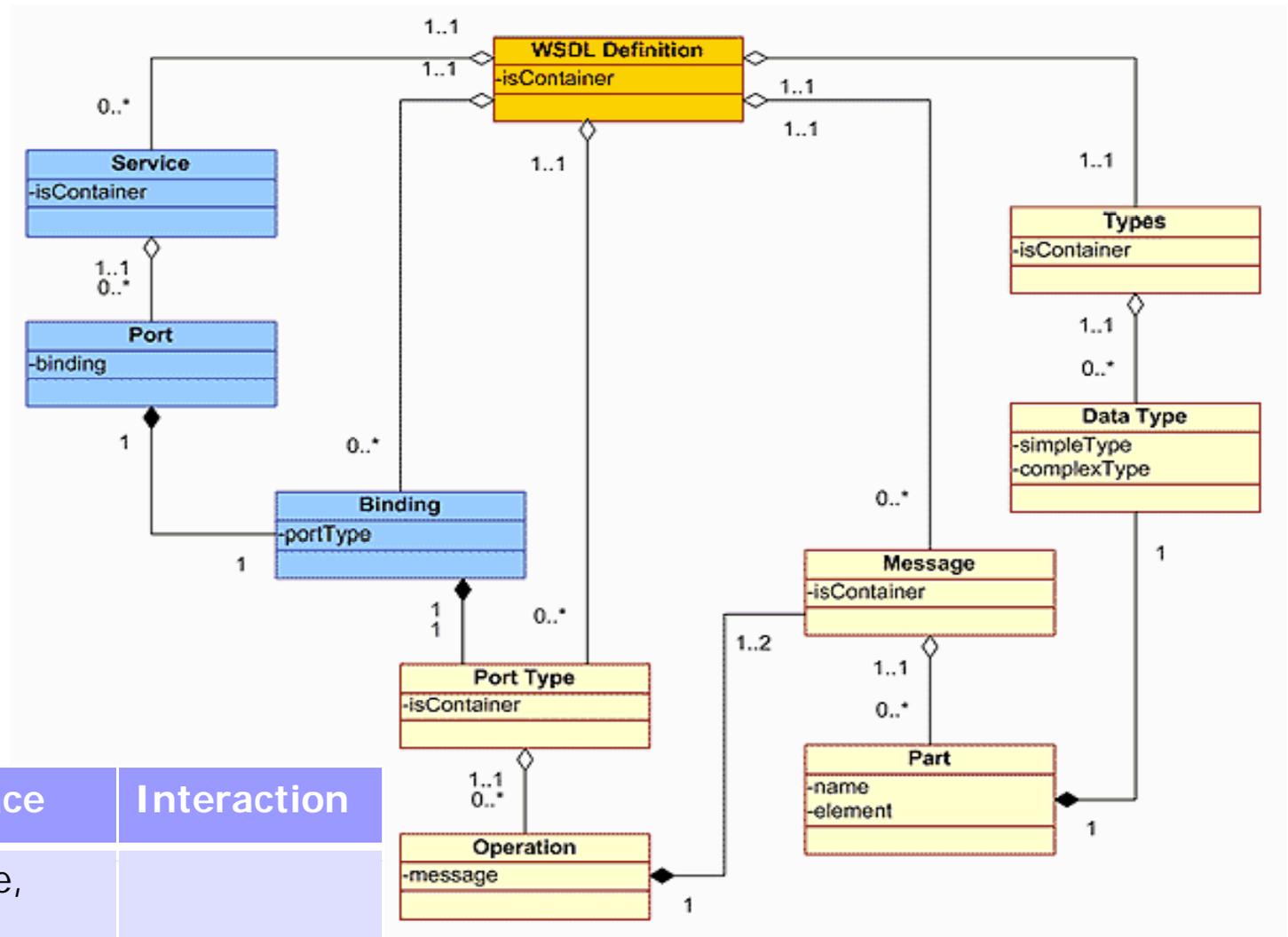


ProgrammableWeb.com 08/17/09

Related Work - Service Description

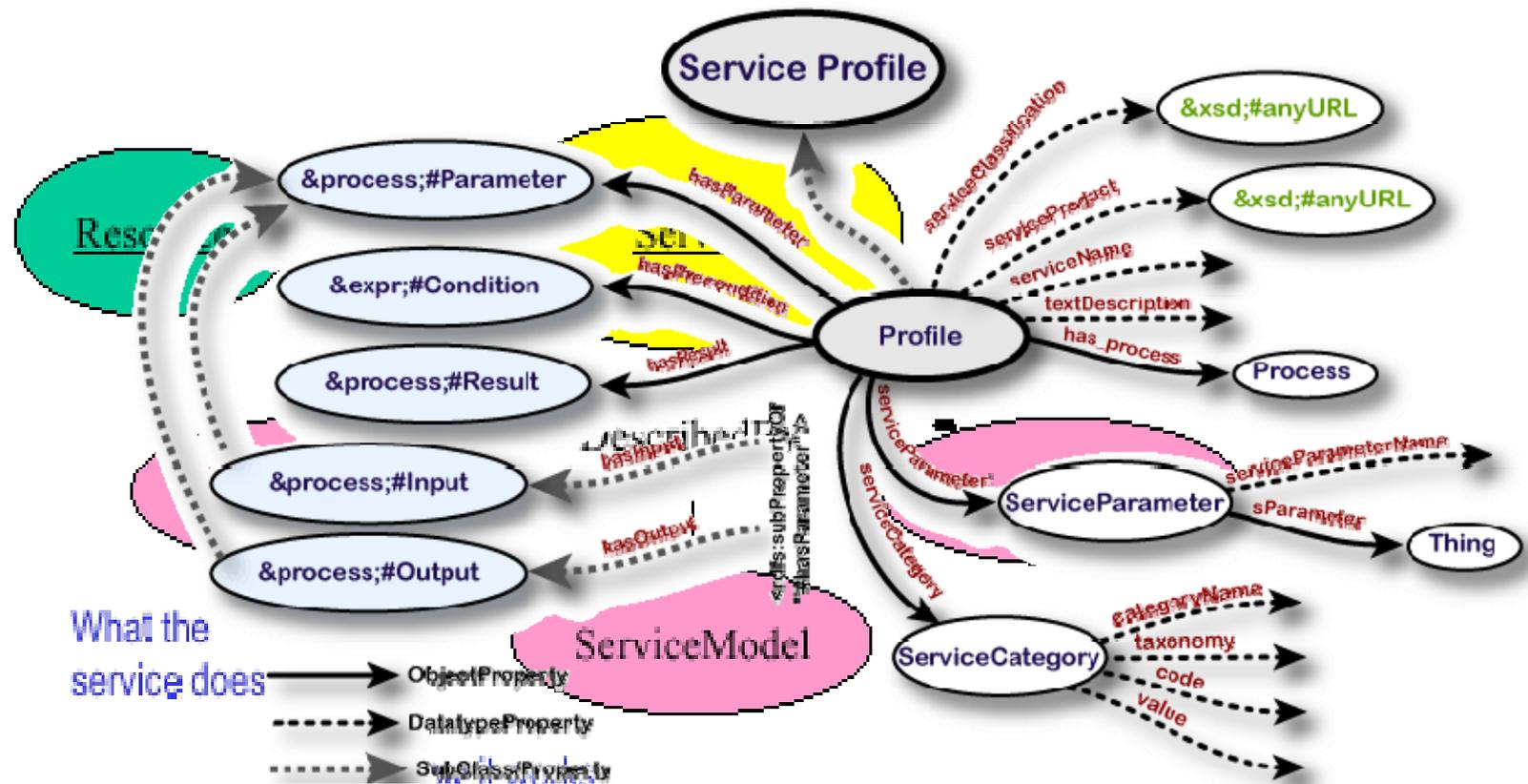
- WSDL
- OWL-S
- WSMO

WSDL(Web Service Description Language)



Capability	Interface	Interaction
Operation	Message, Binding	

OWL-S (Web Ontology Language for Services)



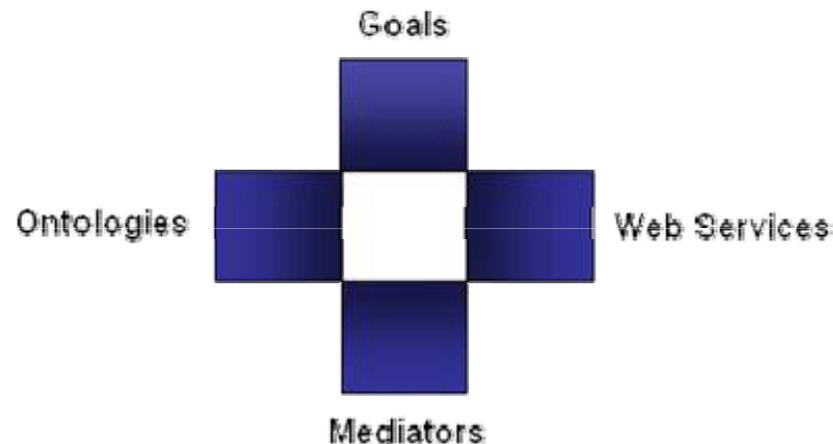
What the service does

Capability	Interface	Interaction
condition, Result, serviceCategory	Input, Output	ServiceModel

WSMO (Web Service Modeling Ontology)

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components



Semantic description of Web Services:
- **Capability** (*functional*)
- **Interfaces** (*usage*)

Connectors between components with mediation facilities for handling heterogeneities

Definition of Web Service in WSMO

■ Class `webService`

- `hasNonFunctionalProperties` **type** `nonFunctionalProperties`
- `importsOntology` **type** `ontology`
- `usesMediator` **type** `{ooMediator, wwMediator}`
- `hasCapability` **type** `capability` *multiplicity = single-valued*
- `hasInterface` **type** `interface`

Class `capability`

`hasNonFunctionalProperties` **type** `nonFunctionalProperties`
`importsOntology` **type** `ontology`
`hasPrecondition` **type** `axiom`
`hasAssumption` **type** `axiom`
`hasPostcondition` **type** `axiom`
`hasEffect` **type** `axiom`

Class `interface`

`hasNonFunctionalProperties` **type** `nonFunctionalProperties`
`importsOntology` **type** `ontology`
`usesMediator` **type** `ooMediator`
`hasChoreography` **type** `choreography`
`hasOrchestration` **type** `orchestration`

Capability

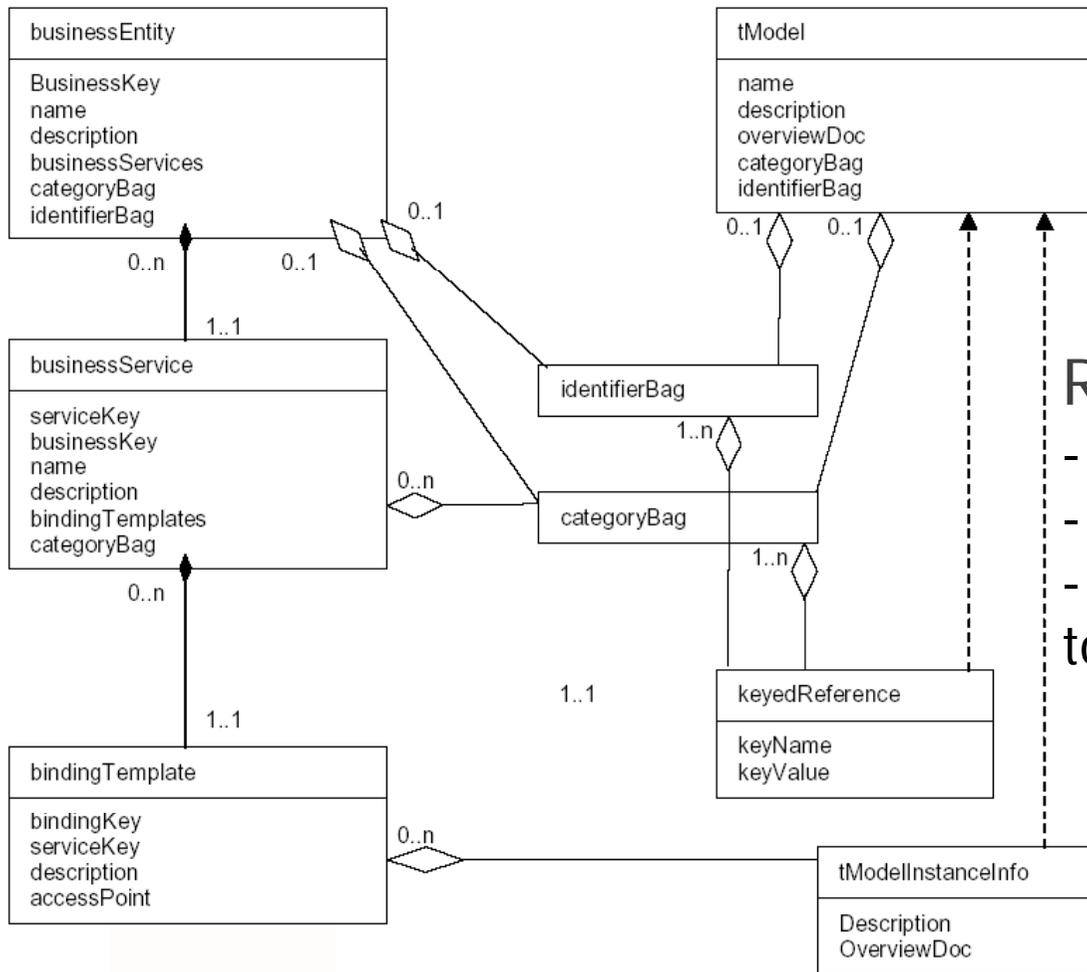
Precondition, Postcondition, Assumption, Effect

Interface

Orchestration, choreography

Interaction

UDDI(Universal Description, Discovery, and Integration)



Registry for Web Services:

- OASIS
- technical access
- does not provide a mechanism to store generic service semantics

Capability	Interface	Interaction
category	Binding	

MFI-7 vs. UDDI

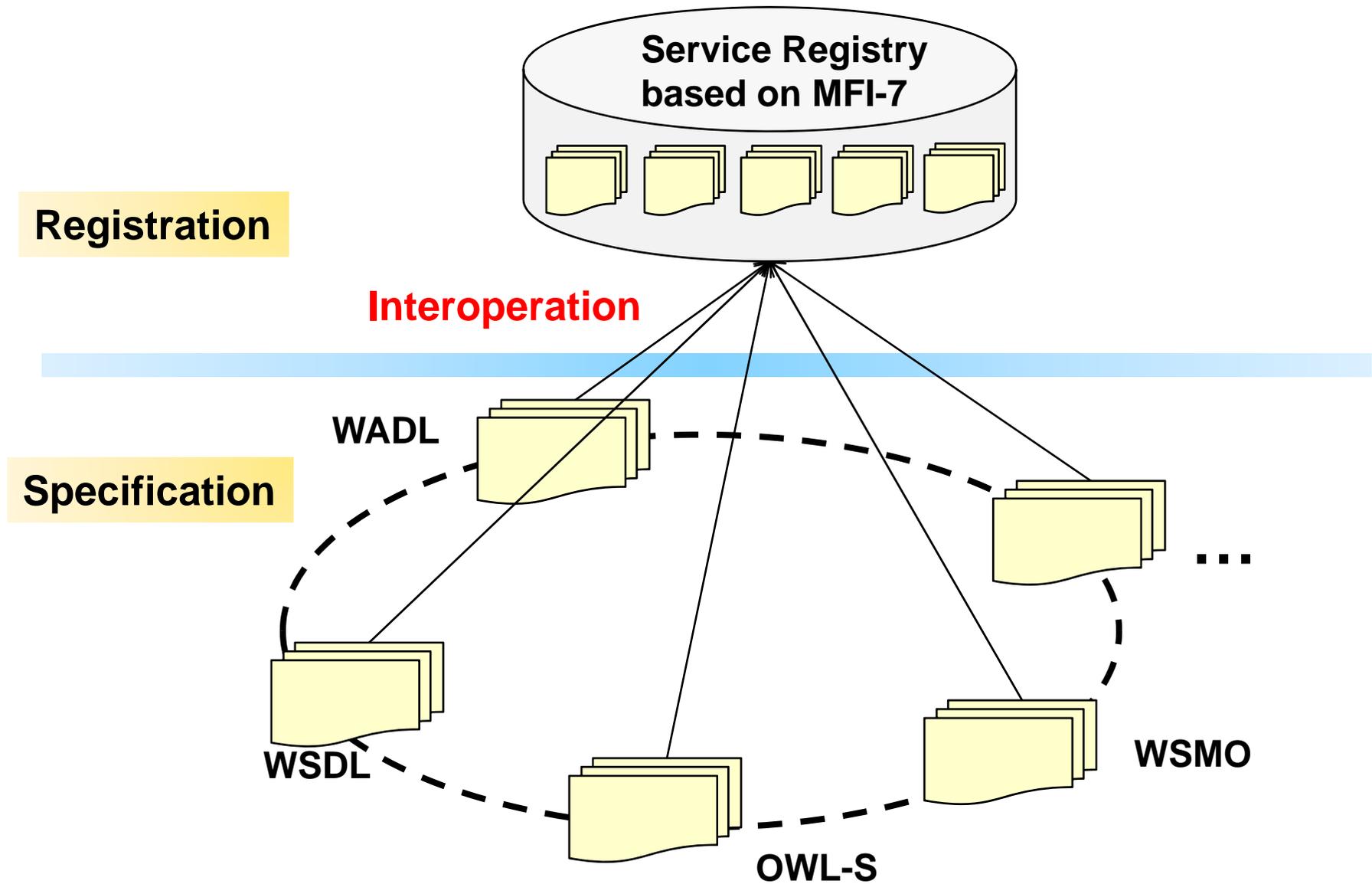
■ UDDI

- service semantics is not explicitly provided
- mainly considers SOAP Web services

■ MFI-7

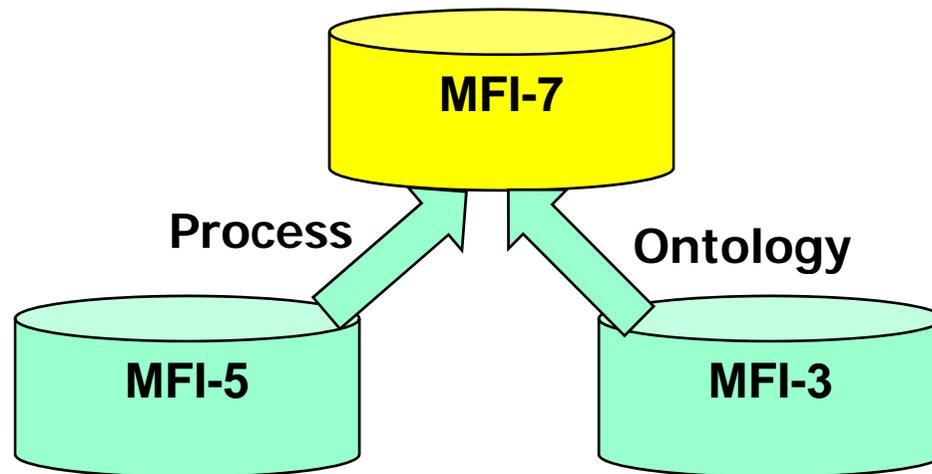
- Provides a mechanism to register service semantics based on MFI-3 and MFI-5
- also considers other kinds of services
 - XaaS in cloud computing

Why We need MFI-7? (1/2)



Why We need MFI-7? (2/2)

- Semantic Interoperability of services
 - Common understanding of services
 - Integration of heterogeneous services
- Service Discovery and Selection
 - Register functional and non-functional information



Based on MFI-3 and MFI-5, MFI-7 provides a mechanism of registering service semantics

Outline

- Background
- Content of MFI-7
 - Introduction and Scope
 - Metamodel for Service Registration
 - Service Registration Prototype
- Future Work

Introduction of MFI-7

- When business information interchange and integration become increasingly frequent, major work in service discovery should be processed by machine, it is necessary to
 - semantically describe service information including functional and non-functional information
 - provide corresponding registration and management mechanism
- MFI-7 intends to provide a generic framework for registering functional and nonfunctional information of services in an explicit manner.

Scope of MFI-7

- Specify a metamodel for registering services that can enable users to discover appropriate services.
- Define the functional and nonfunctional description of services.
- Promote semantic interoperation between various services.

Important Decisions from ROR/ODMS/SMMP Study Period Meeting,
Wuhan, 2008.9.6

- It does not specify
 - language specific details
 - the composition of services

Elements of Metamodel for Service Registration

■ Capability

- Precondition, Postcondition
- Quality_Property

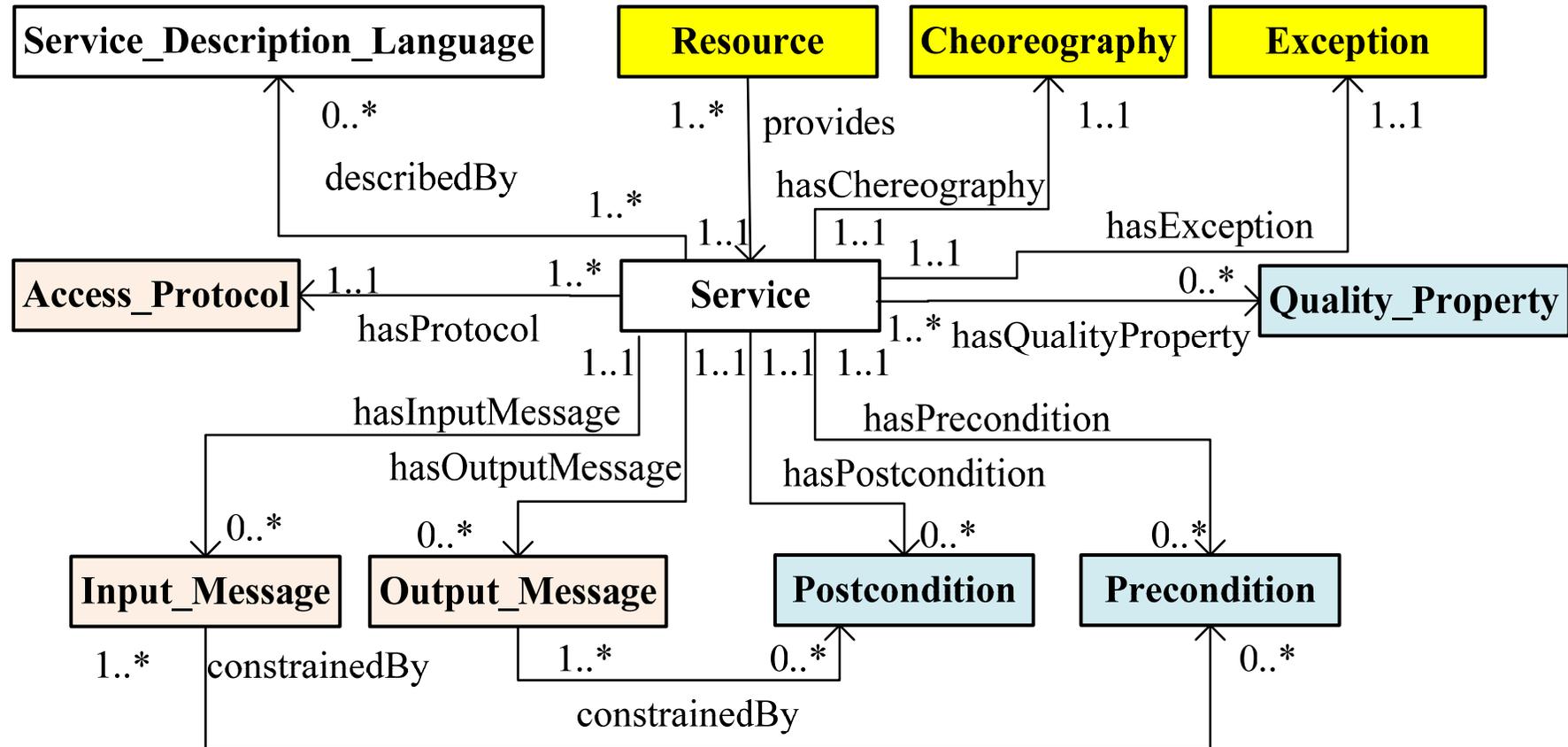
■ Interface

- Access_Protocol
- Input_Message, Output_Message

■ Interaction

- Choreography
 - Value exchange: Provider, Requester
- Resource
- Exception

Metamodel of MFI Service Registration



Capability_Related
 Interface_Related
 Interaction_Related

Capability-related Metaclasses

■ Quality_Property

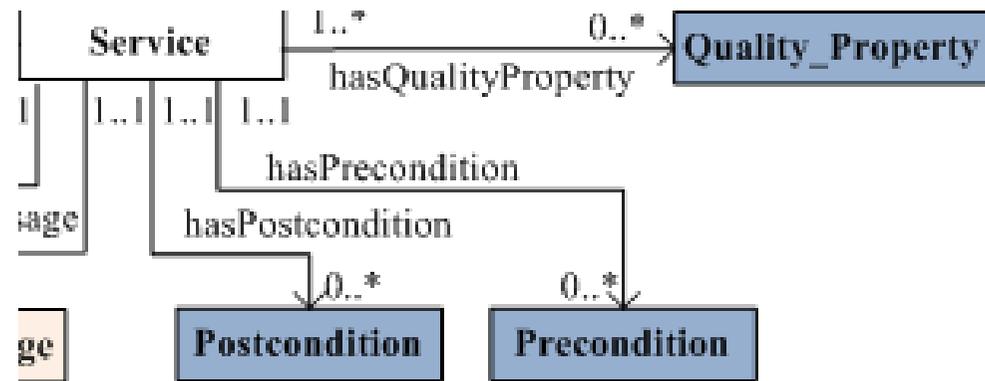
- depicts the nonfunctional description, which can be used to represent the quantitative or qualitative value of service in certain aspect such as response time, cost, reliability, and so on

■ Precondition

- specify the state that should be satisfied before a service is invoked

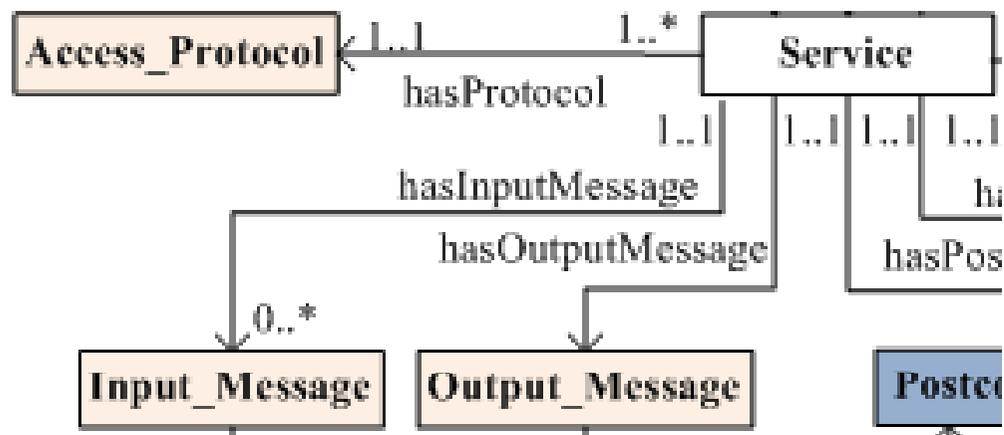
■ Postcondition

- specify the state that should be satisfied after a service is invoked successfully



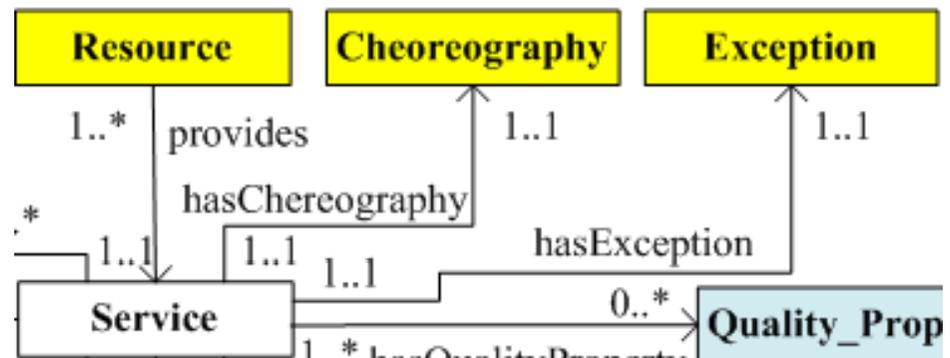
Interface-related Metaclasses

- Access_protocol
 - a requester can access the service by obeying the protocol
- Input_message, output_message
 - the link between service and outer space



Interaction-related Metaclasses

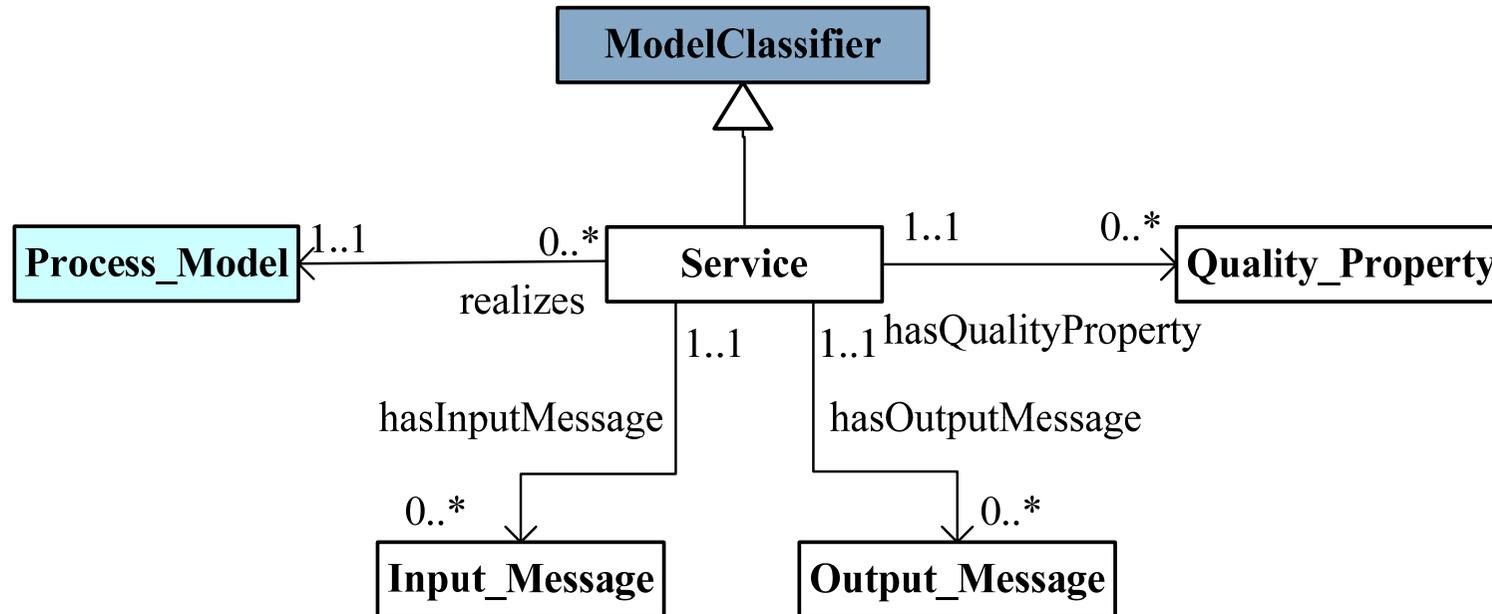
- Resource
 - can be used to provide service
- Choreography
 - provide the necessary information to enable communication with the service from the client point of view
- Exception
 - define the case that the service is not correctly invoked



Semantic Annotation

- Input_Message, Output_Message
 - Message_Type: concepts in Ontology
- Quality_Property
 - Quality_Property_Type: concepts in Ontology

Relationship with Other Parts in MFI



 Metaclass from MFI-2

 Metaclass from MFI-5

Service vs. Process in MFI (1/2)

- Services can be used to realize process
 - A process can be realized by zero or more services
 - A service can realize exactly one (atomic or composite) process
- Services emphasize the encapsulation of certain functionality
 - only exposes certain interface to outer space, while hide its internal execution order
- Processes focus on achieving certain functionality by composing certain activities

Service vs. Process in MFI (2/2)

- Process only considers implementation independent information;
- Service considers implementation specific information such as the invocation protocols.

Why we need two separate parts?

- The diversity of services (SOAP, RESTful, Atom,...)
 - Restful and Atom services are resource-centric, and only use simple operations such as Put, Post, Get, Delete, rather than complex operations which are often described by processes.
- Process can be realized by human beings or local software systems, etc, other than services.
- There is no tight relationships between services and processes.

Outline

- Background
- Content of MFI-7
- Future Work

Future Work

■ Summary

- MFI-7 provides a means to register functional and nonfunctional information and semantics of services
- The services supported in MFI-7 are more than Web services

■ Future work

- Refinement of the metamodel
- Harmonization with other standards

Thank You!
תודה רבה!