

Available online at <http://www.mecspress.net/ijem>

## Research on Service Discovery Based on QoS in the Composition of Web Services

Zhou Min<sup>a</sup>, Tang Xiao-yan<sup>a</sup>, Li Jian-xin<sup>a</sup>, Cai Xiao-li<sup>b</sup>

<sup>a</sup> College of Changzhou Information Technology, Changzhou, Jiangsu, China

<sup>b</sup> Changzhou Institute of Technology, Changzhou, Jiangsu, China

---

### Abstract

Web services are applications deployed in network, can be used in some fields such as B2B e-commerce, the integration between portals and content providers. Web services discovery mechanism is the important part of integration. We extended traditional service description and proposed a service discovery mechanism and realized the mechanism in an agent-based Web services integration framework. Results of this study can be applied to a wide range of cross-enterprise business entity docking, thereby promoting the development of electronic commerce.

**Index Terms:** Web Service, Service Composition, Agent, Service Discovery, WSDL, UDDI

© 2012 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

---

### 1. Introduction

With the development of Web services technology and specifications, Web Service technology can be applied to any fields such as: the B2B e-commerce, the integration of portals and content providers, Internet terminal device control. Now, more and more businesses and enterprises have published their own services as Web services, in order to bring greater opportunities for themselves.

Web service integration is a common Web service application to integrate multiple Web services into one system, in order to achieve more complex functions or services. Description and discovery of Web services are the focus of Web services integration. Conventional description of Web services is the WSDL<sup>[1]</sup>, but the performance of Web services is an important aspect of the service measurement. In order to describe a service more completely and, dynamically select a service more conveniently at service runtime implementation, we extend regular services description and propose a solution and framework of service integration emphasize an Agent-based service discovery mechanism and its realization.

\* Corresponding author.

E-mail address:

## 2. Extension of the Web Services Description model

The conventional description of Web services is the WSDL, a language for describing network services providing a description of service auxiliary information. WSDL defined service description is composed of two parts: Service Interface Definition and Service Implementation Definition.

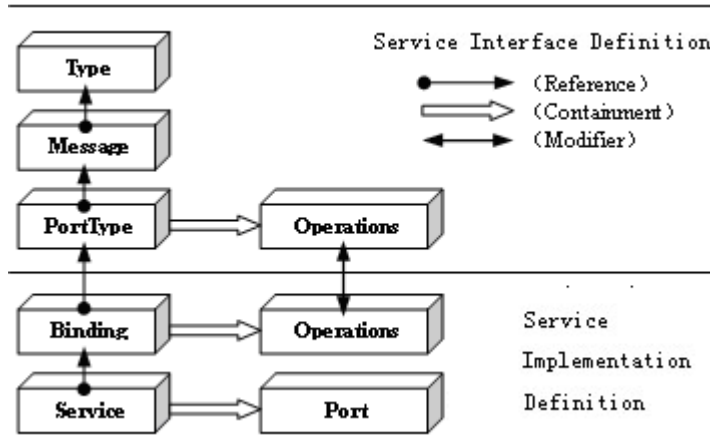


Figure1 The structure of WSDL

The construction of WSDL file is shown in Figure 1. The detailed meaning of each element is shown in [1]. WSDL has scalability which allows describing network services without considering the network status as well as the message protocol.

The QoS of Web service is the measure of the non-functionality that can be felt by users when they interact with the Web service, such as the price of service, duration, reliability, and network bandwidth etc. Many researchers emphasize the importance of describing the QoS of Web service, and give the description of many QoS attributes [2-3].

The measurement methods of above several important service quality attributes (non-functional characteristics) are described in [4].

We call the extended Web service description language as Q-WSDL (Quality based Web Services Description Language); its description model is shown in Figure 2.

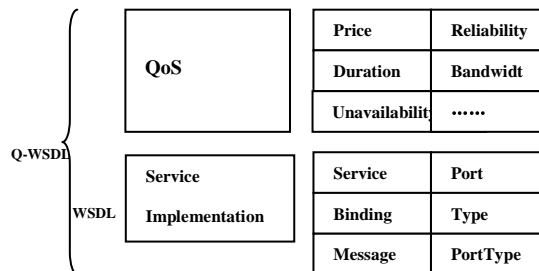


Figure 2: Q-WSDL elements

Based on the above QoS attribute, the quality of a service *s* can be expressed as follows:

$$Q(s) = \{ qprice(s), qdu(s), qrel(s), qnav(s), qbw(s) \}.$$

Note that the value of some attributes should be measured from the perspective of service requestors but not from that of providers because they may be influenced by communication links. Meanwhile, the above quality computing criteria is just a reference and the other systems can provide some other computing criteria. So a Web service has two parts: the conventional description of the service class and the quality of service description. The detailed description of the components and their inter-relationship are shown in figure 3. Extended WSDL are registered in UDDI, these described qualities are dynamic and their values can be regularly updated, so it is easy to search the latest quality of service.

### 3.3. An agent-based framework for Web services composition

Software agent technology<sup>[5]</sup> is an active research area in the current software technology. The software agent is characterized by the following properties: autonomy, social ability, intelligence, etc. And now, it is a technical basis of the new generation software middleware platform. Agent technology provides a new solution for Web services integration. Web services composition architecture based on multi-agent system adopts a hierarchical structure for Web service discovery, integration and configuration. Figure 3 shows the main components of this architecture.

(1) The client application: The user layer includes the service integrated language compiler environment. In our system, the user should provide the service integration demand, the functions need to be called and the structure of the functions.

(2) Service integration process: Web Service integration needs to use an effective modeling language to describe service integration and use an integration services definition language to define it. Here we use BPEL4WS as it supports static and dynamic integration. Before service discovery, the roughly service integration process is needed. After service discovery, the process is instantiated into real binding service processes. The client applications integrate services and provide visual graphics integrated flow according to user requests.

(3) Service Discovery: in the run-time process, Discovery Agent lookups service dynamically based on the expansion of services description in UDDI<sup>[6]</sup>.

(4) Integration process instantiation: binding Web services to integration process which can achieve the desired functionality of the services.

(5) Multi-Agent Systems: Agents work collaboratively in the system to realize service integration process dynamically.

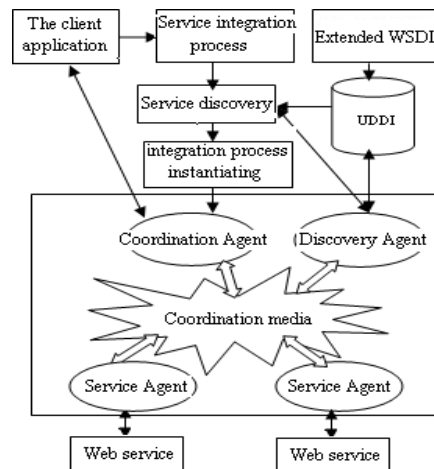


Figure 3 an agent-based framework for Web services composition

Multi-Agent system consists of three types of Agent: Discovery Agent, the Collaborative Agent and Service Agent. The eventual realization of the Agent Communication is ACL. The dynamic interaction between the major parts of the framework is shown in Figure 4.

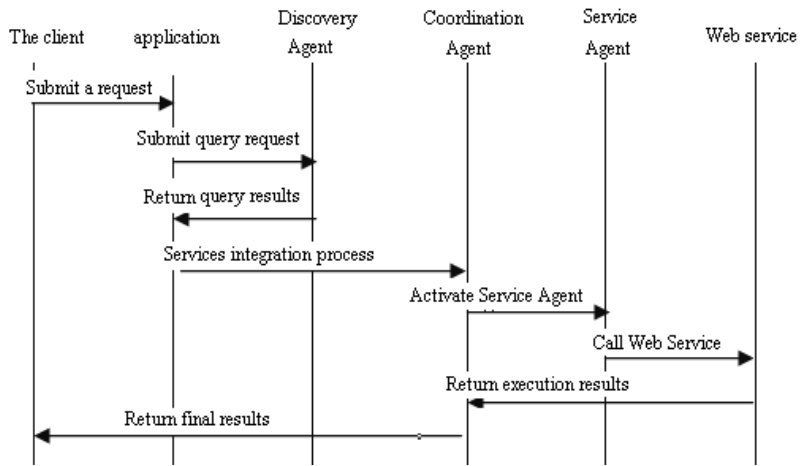


Figure 4 Dynamic interaction diagram

#### 4. Service Discovery

Web service discovery mechanism is an important part of the entire Web service integration. Service discovery mechanism allows Web service providers publishing their Web services, Web service requester searching for their required services under certain conditions and selecting the most appropriate results according to a certain standard.

Discovery Agent is responsible for receiving the services flow provided by User application, and returning found services included in the flow to user application. User application binds found services to the flow and then submit the instantiated flow to Process Agent and its structure is shown in Figure 5.

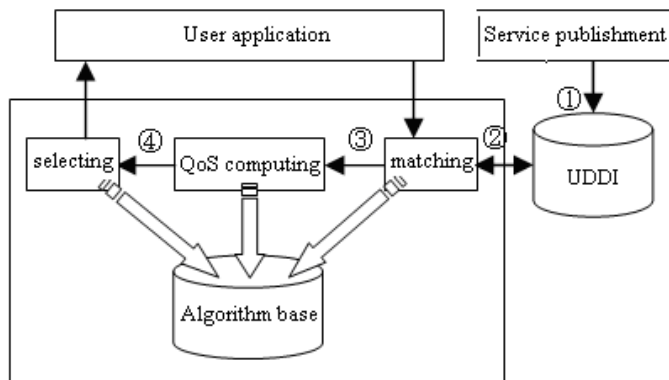


Figure 5 The structure of Discovery Agent (DA)

In order to discover and select services dynamically, we design the Discovery Agent composed of three modules: services match, services QoS computation and services selection. The algorithms that will be used in the process of services discovery are all stored in the algorithm base.

#### 4.1. The abstract model of registered service

After the users have permission to access registration services, they will see the corresponding service description information of all registered services. Users can publish new services, browse or select services. They should provide the regular WSDL text description, in addition to service QoS attributes. All these information are stored in UDDI for other applications to find.

Registered service (RS) can be abstracted into a triple, on behalf of each available Web service for service queries and matching. The formal definition of a registered service is:  $RS = \langle sn, sd, QoS \rangle$ .

RS consists of three elements:  $sn$ ,  $sd$ , and QoS.  $sn$  represents the Web service name to be registered, the  $sd$  and QoS respectively corresponding text description of the registered Web services and QoS attributes.

The following example is a registered service structure:  $RS = \langle \text{"Airline Booking"}, \text{"A service that provides flight ticket reservations based on the specification of a flight request"}, \{price = 50, duration = 5\} \rangle$ .

#### 4.2. The service discovery mechanism

Based on the above service registration information, when service integration process needs to call Web services, it is not feasible for applications to individually visit these Web services and then pick a suitable service from numerous services, because the potential available Web services are all registered Web services in the Internet. An automatic discovery mechanism is necessary.

Here we introduce a service matching algorithm to solve the above problems.

Before we find Web services to be integrated, we should create a Service Model (SM). The service model is decided by integration functions. Service model can be abstracted into a triple which represents a brief description of the characteristics of the required Web services.

Model includes three aspects: the name of the Web service to find, text descriptions and QoS.  $SM = \langle sn, sd, QoS \rangle$ ,  $sn$  means the name of the Web service to find, which does not have to exactly syntax match the name of the Web services.  $sd$  and QoS are respectively corresponding to the text description of Web service to find and QoS metrics. The individual elements have the same meaning as defined  $RS$ , to facilitate  $RS$  and  $SM$  matching.  $sn$ ,  $sd$  and QoS metrics of  $SM$  are specified by the designer based on the functional objectives and operating requirements of the task.

##### 4.2.1 Service description matching

A key issue in service discovery is service description matching algorithm. Matching is to compare a service model and a registration service and calculate their coincidence through a certain way. During the service discovery phase, the  $SM$  and multiple  $RS$  should be matched, and then sort them according to their respective compliance. The user can select a  $RS$  service with higher degree compliance and integrate it into the service integration process.

We only consider the syntax information of service name and service description, without considering the semantic problem. Syntax compliance is calculated based on the service name and service description of the  $SM$  and  $RS$ .

Function  $Anal(SM, RS)$  is to calculate the syntax compliance of service model ( $SM$ ) and Registration Services ( $RS$ ). This function is mainly based on the other two functions  $AnalN(SM, RS)$  and  $AnalD(SM, RS)$  and the weight value.

$$Anal(SM, RS) = w_1 AnalN(SM, RS) + w_2 AnalD(SM, RS)$$

Where  $w_1, w_2 \in [0, 1]$ , and  $w_1 + w_2 = 1$ .

*AnalN* (*SM*, *RS*) and *AnalD* (*SM*, *RS*) respectively calculate the compliance of the name and description of the two services, considering only syntax currently, without considering the semantic information. Both functions return a real value between 0 and 1.

In the fields of information extraction and natural language processing, there are many ways to calculate the compliance of Web service name and service description text. Herein, the compliance of service name and text description is calculated by the number of common Q-grams<sup>[7]</sup> of the two strings<sup>[8]</sup>.

4.2.2 QoS value computing

We should then compute QoS attribute values of the registration services found in the above matching results. QoS attribute values can be calculated based on the measurement criterion mentioned previously to choose the suitable service. The user needs to set the weights of each service attribute weights and the sum of all weights must be 1. If one of attributes is more important, the weight of it can be set higher.

An example of the matching results of Service model and registration services can be seen in table 1.

Table 1 the sample table of service matching

	Service model	registration service
<b>Service basic information</b>		
Name	Airline Booking	China Airline Booking
Text description	A flight ticket reservation service	A Chinese flight ticket reservation service
<b>QoS attributes</b>		
The average response time	50	55
Average overhead	200.00	220.00
Average reliability	0.95	0.96
<b>Matching results</b>		
Syntax compliance	0.97	
QoS value	The highest one	

The whole process of finding services is as follows:

Step 1: When the designer needs to add in a service to achieve a certain function, he should create a service model and call lookup service module to search for the most appropriate Web services. The display order of the search results is based on the value of registration services from higher to lower, so that when the first service is called fail a sub-optimal service can be called in turn to achieve the same function.

Step 2: After the service has be called, the Q-WSDL parser parses the service model, and extracts information, including the name and text description and QoS attributes of the service, and finally sends them to the service search engine as selection criteria.

Step 3: The service lookup engine extracts all registration services from the UDDI registry that meet the requirement.

Step 4: to syntax match registration services and service model one by one, and to calculate the corresponding QoS values, and sort them. The discovery service returns query results.

4.2.3 Optimization algorithm

Web service optimization mechanism and algorithm based QoS in the service integration can be referenced in [4].

## 5. Conclusion

The discovery and selection of services are closely related to the overall performance of the service integration. The main work of this paper is to establish a Web service description and discovery mechanisms to provide support for more efficient services integration. The service discovery mechanism provided in this paper has the following two significance aspects: (1) QoS based services selection can improve service integration efficiency and reduce the cost of integration. (2) the service provider using the selection mechanism can improve the quality of service, provide a variety of different services, and improve customer satisfaction while increasing the revenue of the service provider.

The Next work includes: Semantic Web Services such as DAML-S to achieve automatic lookup of services.

## References

- [1] Web Services Description Language (WSDL 2.0) [EB/OL]. <http://www.w3.org/TR/> .2007
- [2] T.C. Lau, QoS for B2B Commerce in the New Web Services Economy, ISEC 2001 Workshop on Performance and QoS for E-Commerce Applications, Hong-Kong, China, 2001.
- [3] A. Mani and A. Nagarajan, Understanding Quality of Service for Web Services, IBM DeveloperWorks Technical Paper, 2002
- [4] Tang Xiaoyan, Li Bin Web service integration and computer applications. QoS-based service selection [J]. 2006, Vol 26, 242-243
- [5] Oliveira Eugénio; Fischer Klaus; Stepankova Olga, Multi-Agent systems: which research for which applications [J], Robotics and Autonomous Systems, 1999, Volume 27: 91-106.
- [6] CurberaF, EhnebuskeD, RogersD.Using WSDL in a UDDI Registry 1.06[EB/OL]. <http://www.uddi.org/bestpractices.html>, 2006.
- [7] Sutinen E and Tarhio J. On Using Q-gram Locations in Approximate String Matching. In Proceedings of Third Annual European Symposium on Algorithms (ESA'95), 1995, 327-340.
- [8] He Wenli Discovery and Integration of Web services workflow system: a master's degree thesis, Beijing: Institute of Software, Chinese Academy of Sciences, 2003.