QoS Management for SOA by Synchronizing Quality Context in UDDI

Youngkon Lee
e-Business Department, Korea Polytechnic University,
2121 Jeongwangdong, Siheung city, Korea
yklee777@kpu.ac.kr

Abstract

This paper presents the design principle for integrating quality management on Web service registry developed in UDDI specification and Web service quality management system (WSQMS). WSQMS, developed by NIA[1] *can measure and collect the quality information of Web services by its agency system installed on the Web service system. Web service registry is core system for registering and searching WSDL(Web Service Description Language). On the registry, there are no methods that users search Web services on the basis of quality information because the registry is operating without any relationship of WSQMS. So, it is required an integration method where WSQMS and registries share the quality data on the basis of classified quality data. For representing Web service quality information, we adopted the WSQDL(Web Service Quality Description Language), which published by WSQM technical committee in OASIS. In more detail, this paper also presents the scheme to compose the classification scheme for quality data and to modify the necessary data structure of the registry.*

1. Introduction

Web service is a core technology for sharing information resources and integrating processes in companies or organizations. Most applications depend on Web service as an interface for integrating applications, which as a result takes a major role of service oriented architecture (SOA) [1]. As the number of applications connected by Web service is increased, the importance of Web service quality tend to be the critical factor in determining the performance of IT system.

Web service quality factors could be classified as interoperability, manageability, security, business processing capability, business value, measurable quality, which have to be considered as applying Web service into real world. Web service quality model technical committee [2] established the Web service quality model and the XML schema for representing the specification.

Maintaining Web service quality above some critical level requires more effort to manage overall Web service framework than each of Web service. The Web service registry has a core position in overall Web service framework, implemented according to UDDI (Universal Description, Discovery, and Integration) specification [3]. The registry takes charge of mediating, registering, and searching Web service information, WSDL(Web Service Description Language) form in most cases. In the respect of the position of Web service

[1] NIA(National Information Society Agency) is an organization for Korea Government, sponsoring this research.

registry, as a major storage for Web service information, the registry could be best alternative as a facility for preserving Web service quality. For example, a Web service provider can use the registry for storing the address of Web service quality information, by which they describe the quality level for a Web service in a specification. A Web service consumer can select a proper Web service by the quality information.

In this paper, we provide the Web service registry management scheme for managing the quality information of Web services. For representing Web service quality information, we use the WSQDL (Web Service Quality Description Language) [4], which published by WSQM technical committee. This paper also presents how to compose the classification scheme for quality data in a Web service registry and how to modify the necessary data structure of the registry.

In section 2, we summarize the related works about registry classification scheme. Section 3 presents Web service management system and section 4 describes the sharing method of quality data. In section 5, we suggest the way to organize the classification scheme by using Web service quality data.

2. Related Works

A study for classification scheme for UDDI [5] tried to modify UNSPSC, NAICS, and ISO 3166 of industrial classification schemes to apply Web service registry. These schemes have been developed for analysis of industrial statistics and national code framework. The schemes had been modified for adjusting Web service registration, publication, and search because there are no other proper classification schemes for the Web service registry.

This study started at the problem that those classification schemes are so complex and broad that they are not appropriate as the classification scheme for Web services. Thus, they tried to arrange the classifications schemes so that the classifications may cover all the feasible Web services. For the purpose, they first analyzed the framework of Web services and domestic and foreign classification schemes for UDDI service and use case scenarios of UNSPSC, NAICS, ISO 3166, and Korean industrial classification. This study, however, focused on just minor update of the classification schemes, as a result their result still lacks in applying Web service area.

A study of registry interoperability [6] analyzed the status of Web service registries operating now and provided the national registry blueprint in the respect of policy and technology. Especially, they provided the various policies which enable the registries to be interoperable by regulating some major factors and by collaboration scheme. The study also induced the normative usage pattern of Web service registries. For example, the usage pattern could be classified as dynamic and static according to the binding time of Web service. In this way, they tried to give the guideline of Web service registry by classifying the usage pattern. They also designed the topology of Web service registries for their connection and interoperability and defined the role of each registry in the topology. They, however, did not present any scheme for service quality.

The study of B2B integrated classification scheme [7] tried to provide the effective search method on the service classification represented by topic map. They also provide the way that benefits the merit of topic map on the connection and the interoperability issues. They also tried to get flexibility and expansibility of Web service registries by registry service classification and to guarantee interoperability at the time of changing or integrating classification schemes. The major issue, however, is that there is no effective way for applying conceptual polymorphism of classification items to the topic map instance until now. That is, it's very difficult to give guideline for making the relationship between topics. As a result, according to inclination of author and context, relationship between topic map and configuration could have propensity or be subjective.

3. Web Services Quality Management System

The purpose of Web service quality management system (WSQMS) [8] is to provide the integrated framework, called One-stop service, for processing the quality information on the Web service quality information life cycle: test, evaluation, registration, search, and abolition. For providing quality data during all the lifecycle of Web service, it is necessary integration of a Web service registry and WSQMS, enabling Web service users to easily get the Web service quality information gathered by the WSQMS. When a Web service is registered and a registry notifies it to WSQMS, the WSQMS automatically tested the Web service and stored quality data at its local storage. The WSQMS, after then, continue to collect the quality data of the Web service by periodic monitoring.

The WSQMS operating now is based on subscription mechanism. After installing an agent, WSQMS requests the operation information (OS (Operating System), platform, etc) of target Web service system to the agent. The agent has been developed according to the Web Services Distributed Management V1.0 (WSDM) [9].

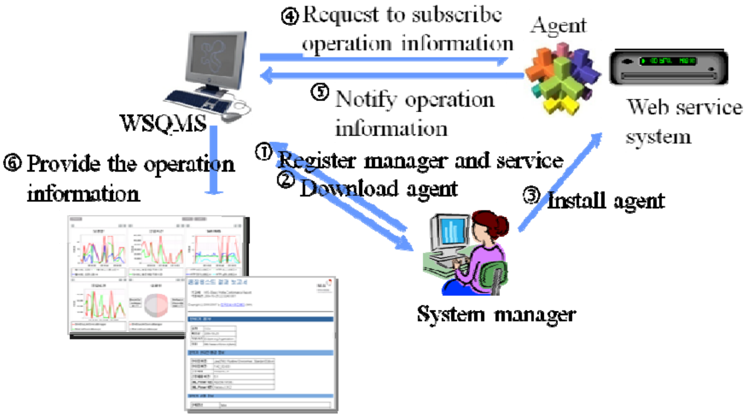
④Request to subscribe operation information
Agent
⑤ Notify operation information
Web service system
WSQMS
⑥ Provide the operation information
① Register manager and service
② Download agent
③ Install agent
System manager

Figure 1. Web Service Monitoring Process

WSDM (Web Service Distributed Management) is composed of MUWS and MOWS: MUWS (Management Using Web services) which is a specification for managing various computing resources by Web service and MOWS (Management of Web services) which is a specification for managing Web services. The agent is developed in conformance form with MUWS for managing the OS and service container and MOWS for Web services.

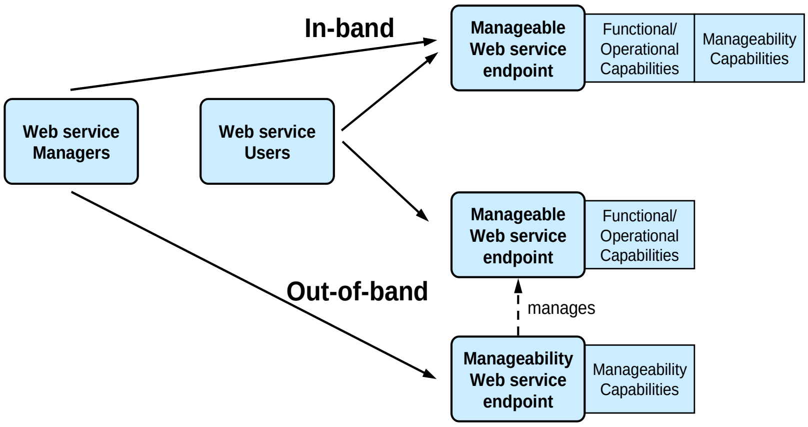
In-band
Manageable Web service endpoint
Functional/ Operational Capabilities
Manageability Capabilities
Web service Managers
Web service Users
Manageable Web service endpoint
Functional/ Operational Capabilities
Out-of-band
manages
Manageability Web service endpoint
Manageability Capabilities

Figure 2. WSQMS Agent Type

WSQMS supports basically out-of-band type agent for managing Web service, OS, and Web service container and in-band type agent for general Web service management. WSQMS conforms to completely WSDM specification so that any Web service system with management function may be connected with the WSQMS without any additional process. The agent installed at Web service system notifies the operation information to WSQMS depending on the type of the OS and container. As a result, the Web service messages sent from or to the Web service system are forwarded to the WSQMS by the agent. The WSQMS analyzes the quality of the Web service on the basis of the messages with referring TAD (Test Assertion Document) and provides the quality information to Web service associates if they require it.

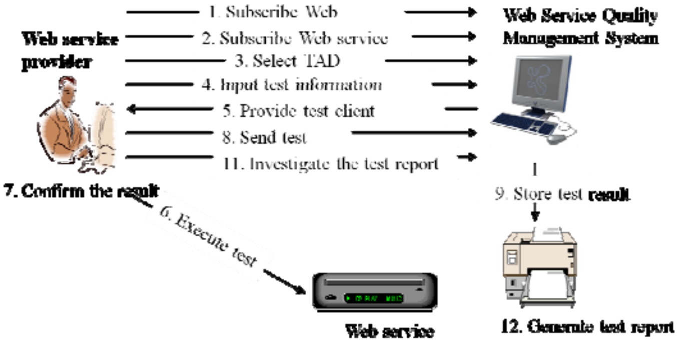

Figure 3. Quality Test Process

In the case that a Web service provider requests a quality test to WSQMS, a test agent is transferred and deployed automatically at the requester’s system. The test agent automatically starts test and notifies the test result to the WSQMS, which then will analyze the result and provide the quality data if the requester requires it. The detail process for quality test and result notification is shown in [Figure 3].

4. Principle of Quality Data Sharing

Without any restrictions, Web service associates could use the Web service quality information by connecting a Web service registry and WSQMS. For basic technical point, the connection between WSQMS and the registries generally depends on the SOAP (Simple Object Access Protocol) messaging. They also use WSDL (Web Service Description Language) for describing the Web service interface. Besides, we should consider following advance topics.

Firstly, how to store a reference to the quality information to the registry? As the previous section described, the quality data could be obtained from WSQMS and the quality data is described in a WSQDL document. The issue is how the registry refers the WSQDL document or related data. Another issue is whether the data object in the registry just refers to the WSQDL rather than stores the quality data. If it just refers to the quality data, it may be hard work for Web service consumer to search a Web service that satisfies his requirement. Therefore, it is strongly required the quality data classification scheme so that the Web service user can search easily a Web service satisfying required quality level.

Secondly, who is the initiator for requesting quality data? It could be WSQMS or a registry. According to the initiator type, system interface and the kind of APIs could be very different, thus we should deeply consider who the initiator is.

Last question is how the Web service user gets the quality information. The user gets the information from WSQMS and a registry. According to the source for getting quality information, the process could be different seriously.

4.1 Registration of Quality Information

We explain the registration process of Web service quality information where Web service registration induces to test the Web service and to generate its quality data. We assume that Web service user has been registered at a registry. The first scenario is the case that WSQMS requests the information of new Web service, as shown in [Figure 4]. In this case the registry has to provide the WSDL if it has the new registered Web service. The WSQMS will get the quality data by testing the Web service and provide the reference to the document to the registry, which stores the reference data to some data object, tModel.

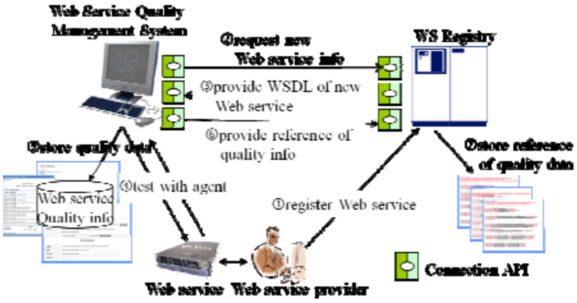
Web Service Quality Management System
②request new Web service info
③provide WSDL of new Web service
⑥provide reference of quality info
WS Registry
⑤store quality data
④test with agent
Web service Quality info
①register Web service
⑦store reference of quality data
Connection API
Web service
Web service provider

Figure 4. Scenario 1: Quality Data Registration

WSQMS requests actively the Web service information to a registry, resulting that WSQMS gets the initiative to manage the quality data and it can control its own overload. On the other hand, it should take care of all the quality data management and need more APIs for implementation.

The other method is that a registry provides voluntarily the information of new registered Web service, WSDL. There are two ways for providing WSDL: whenever a new Web service is registered and periodic. Just after WSQMS obtains the WSDL of a new Web service, it tests and measures the Web service remotely by the agent. The method for storing quality information and referencing it is the same as the first method. This method provides the WSDL information to WSQMS more easily than the first method. The number of APIs for integration is less than that of the first method. The registry controls the period for sending WSDL to WSQMS flexibly. WSQMS, however, could be overloaded if the requests of quality test are concentrated at a specific time.

4.2 Use of Quality Data

For using the quality information, Web service user has to obtain the quality information by search. For the process, there could be 3 scenarios. First, when Web service user requests the quality data of a Web service, the registry searches its local storage to obtain the reference to the data from WSQMS. This method has the advantage that the user accesses only one interface of the Web service registry. He can also search the quality data in the context of Web service information. On the contrary, the data structure of the registry should be modified for including even quality data.

Next scenario is that the Web service user requests directly the quality data to WSQMS. In this time, WSQMS provides the WSQDL directly to the user, but it should get the WSDL of the Web service because it has no WSDL information. This method requires no modification of the previous registry architecture because users get quality data of Web service just from WSQMS. Web service users can search the quality data easily on the support of the WSQMS classification scheme. This method also requires no additional processes to guarantee the consistency of quality data between a registry and WSQMS. But, there are some restrictions in terms of Web service search because WSQMS cannot afford the full-fledged functionality of Web service registry.

Third scenario is that Web service user gets all the information of the Web service from WSQMS: WSQDL and WSDL. For the purpose, WSQMS should keep WSDL of all the Web services registered. The WSDL of registries and WSQMS should be synchronized for data consistency. WSQMS takes the charge of overall management of Web service quality and registries provide their Web service information unilaterally to the WSQMS. As a result, this method inevitably endures the overburden for synchronization between registries and WSQMS. The Web service user can get all the Web service information by just access only one WSQMS system. However, it requires WSQMS to be endowed with the similar functionalities of Web service registry and as a result brings overall performance degradation.

Generally speaking, it is most desirable that WSQMS collaborates with registries in preserving their peculiar duty. In other words, it is best scenario that each system preserves its role and shares the Web service quality information if necessary. In this respect, we conclude the first scenario is the best choice because according to the scenario, WSQMS takes doing its best for quality management and controls its own overload. In this case, there should be specific reference model for quality data between a registry and WSQMS.

5. Quality Classification Scheme

In a Web service registry, WSDL is referenced in a tModel, which is a container for a reference to the WSDL. Because tModel is devised to include the detail information about a Web service, it is natural conclusion that we modify tModel to be proper for including the reference to WSQDL. There are two choices. First is to make a new reference data object to WSQDL in <*overviewDoc*> as the form described as WSDL. This way is trivial, so it enables users to find out at once that WSDL and WSQDL describe characteristics for the same target Web service. This way, however, restricts severely the usage of reference to quality data. That is, user cannot search the quality data rapidly because there are no classification schemes for quality data. [Figure 5] shows <*wsqrlURL*> in <*overviewDoc*> and XML schema for WSQDL complex type.

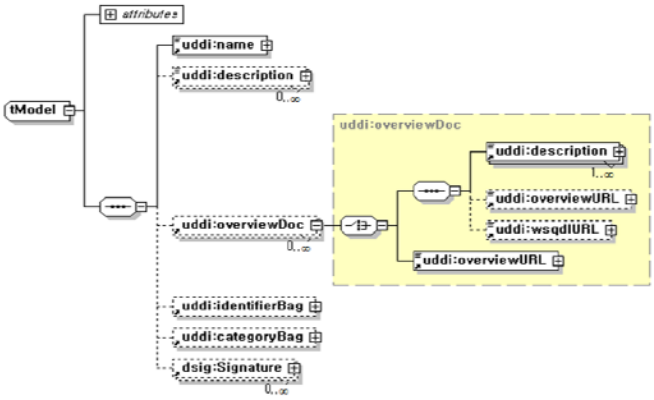

Figure 5. tModel and its Schema including <wsqdlURL> component

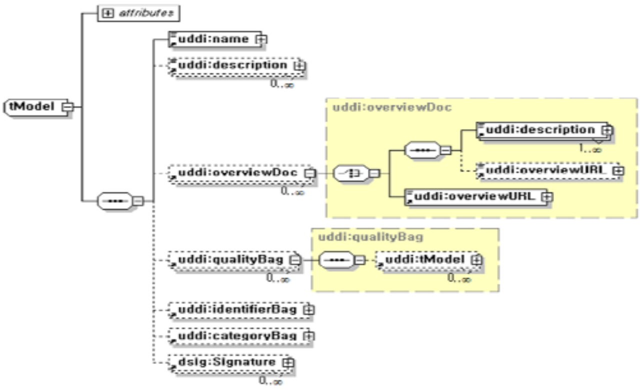

Figure 6. tModel Component and Schema including qualityBag

Second way is to make a specific tag, <*qualityBag*>, in tModel to store the reference to WSQDL. This requires additional processing modules, but enables the quality data to be used more widely. For example, this method allows the reference to WSQDL in <*qualityBag*> to be handled as the form of tModel, resulting that process related with tModel could have still flexibility. However, it requires updates of considerable part of the registry because the registry system should process two types of tModel for: WSDL and WSQDL. However, it is impossible to search a Web service effectively on the basis of quality data, because tModel has only reference data to WSQDL. Thus, it is desirable to implement architecture for referring Web service quality data by using the quality classification scheme. [Figure 6] shows the tModel component structure and XML schema including <*uddi:qualityBag*>.

In [Figure 6], the structure of <*overviewDoc*> is the same as previous tModel, but <*qualityBag*> is a new structure for referring any number of tModel.

Another way is to add quality context information to tModel for quality classification scheme. This allows the registry to have quality context in <*qualityBag*>, as a result, users can search the quality data of corresponding WSDL through tModel. As the previous search method of Web service registry by using <*categoryBag*>, a registry parses previously the quality data in <*qualityBag*> and stores the quality context so that users may just search a Web service satisfying some criteria by using the quality context or in quality classification. To represent quality context data consistently and to manage it requires further study.

[Figure 7] shows the <*qualityBag*> component structure and its XML schema including <*qualityContext*>, whose structure could include any type of character string. <*qualityBag*> stores any number of required <*qualityContext*> and represent any type of quality data. For example, as digital signature for message consistency and proof of message sender, a <*qualityContext*> as type of /eval/sec/Dsig/keySize/ could be made and we say that a system is safer when it has its value of 128 rather than 64 in the respect of digital signature safety. <*qualityContext*> representing Web service quality information should be registered on a registry and user can search the quality data according to the value of <*qualityContext*>.

The registry requires the additional APIs for processing the quality data in the relationship with WSQMS. Firstly, it is required for WSQMS to have APIs searching the new registered Web service. The APIs correspond to the functionality of searching Business Entity, Service, Binding, and tModel. APIs for representing the reference to the quality information sent from WSQMS are required.

If the reference to the quality data is stored in tModel, the additional APIs for processing tModel operation are required. Besides, it’s required the APIs for modifying and updating Web service quality information and synchronizing the Web service information between WSQMS and registries. [Table 1] summarizes the APIs for processing quality information by connecting WSQMS and registries.

```
<xsd:complexType name="tModel" final="restriction">
            <xsd:sequence>
                        <xsd:element ref="uddi:name"/>
                        <xsd:element ref="uddi:description" minOccurs="0"
maxOccurs="unbounded"/>
                        <xsd:element ref="uddi:overviewDoc" minOccurs="0"
maxOccurs="unbounded"/>
                        <xsd:element ref="uddi:qualityBag" minOccurs="0"
```

```
maxOccurs="unbounded"/>
                <xsd:element ref="uddi:identifierBag" minOccurs="0"/>
                <xsd:element ref="uddi:categoryBag" minOccurs="0"/>
                <xsd:element ref="dsig:Signature" minOccurs="0"
maxOccurs="unbounded"/>
          </xsd:sequence>
          <xsd:attribute name="tModelKey" type="uddi:tModelKey" use="optional"/>
          <xsd:attribute name="deleted" type="uddi:deleted" use="optional"
default="false"/>
</xsd:complexType>
<xsd:complexType name="qualityBag" final="restriction">
          <xsd:sequence>
                <xsd:element ref="uddi:qualityContext" minOccurs="0"
maxOccurs="unbounded"/>
          </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="qualityContext" final="restriction">
          <xsd:simpleContent>
                <xsd:extension base="uddi:validationTypeAnyURI4096">
                      <xsd:attribute name="useType" type="uddi:useType"
use="optional" default=""/>
                </xsd:extension>
          </xsd:simpleContent>
</xsd:complexType>
```

Figure 7. tModel component and schema including qualityContext

Category	API	description
Search for New Web service	find_new_business	Search new business entity
	find_new_service	Search new business service
	find_new_binding	Search new binding
	find_new_*tModel*	Search new *tModel*
Store Web service quality information	save_qualityData save_qualityContext	Describe the reference to quality data in *tModel* Store the quality context
Update Web service quality information	update_qualityData update_qualityContext	Update the quality data Update the quality context data
Search for Web service quality information	get_qualityData get_qualityContext	Search quality data Search quality context
Delete Web service quality information	delete_qualityData delete_qualityContext	Delete quality data from *tModel* Delete quality context from tMode
Synchronize Web service information	synch_*WSDL*	Synchronization Web service information on *WSQMS* with that on UDDI periodically
Update time of quality information	get_lastUpdateTimeQD get_lastUpdateTimeQC	The last update time of *WSQDL* The last update time of quality context

Table 1. Registry APIs for connecting WSQMS

6. Conclusions

By integrating Web service registries with WSQMS, we expect at least three following benefits. First, Web service providers reduce their effort to test and measure Web service performance and monitor the performance continuously. The users can get WSDL and WSQDL data with just one access to a registry or a WSQMS and determine whether they use the Web service. The Web service providers also get the quality data of their Web service just by registering a registry.

Secondly, Web service becomes more reliable. When a Web service is registered a registry, test and evaluation for it is performed nearly at the same time. As a result, the Web service should be more reliable and better performance. Web service provider will try to enhance the quality of his Web service.

Lastly, the integration contributes to activate Web service and its dissemination. It accelerates the lifecycle of Web service, implementation, test, and adjustment, as a result, the coverage of Web service will become more broaden in various areas.

References

[1] C. M. Mackenzie and et al, “ Reference Model for Service Oriented Architecture 1.0,” OASIS Committee draft, 2006.8

[2] D. Min and Eunju Kim, “ A Study for Web service quality Model and Test guideline,” NCA IV-RER-04052, 2004.12

[3] UDDI TC, “ Universal Description, Discovery and Integration v3.0.2 (UDDI),” http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm, OASIS Standard, 2005.2

[4] Youngkon Lee and Eunju Kim, “ A Study for Web Service Quality Description Language (*WSQDL*),” NCA IV-RER-04052, 2006.11

[5] KDCIA, “ Development for Korea UDDI for Web service registration and Management,” NCA IV-RER-04058, 2004.11

[6] W. Kim, “ A Study for National UDDI for UDDI Interoperability,” NCA, 2005.8

[7] kCALS, “ Guideline for e-Catalog Standardization,” ECIF, 2003.10

[8] Jeongkuk Choi, “ An implementation report for Web service Quality Management System,” NIA, 2007.1

[9] I. Sedukhin, “ Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.0,” http://docs.oasis-open.org/wsdm/2004/12/wsdm-mows-1.0.pdf, OASIS Standard, 2005.3

Authors

1997.8: KAIST Ph.D. in Information and Communication Tech.
1998.9-2005.2: Manager of e-Biz System development in POSDATA R&D center
2005.3- now: assistant professor in Korea Polytechnic University
OASIS individual member(TAG/TAMIE/WSQM),
ISO TC 154/SC 32 member
Interesting area: SOA, Web service, QoS, S/W development