Web Services Quality Factors Version 1.0

Committee Specification 01

22 July 2011

Specification URIs

This version:

http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/cs01/WS-Quality-Factors-v1.0-cs01.doc (Authoritative)
http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/cs01/WS-Quality-Factors-v1.0-cs01.html
http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/cs01/WS-Quality-Factors-v1.0-cs01.pdf

Previous version:

http://docs.oasis-open.org/wsqm/wsqf/v1.0/WS-Quality-Factors-v1.0-cd02.doc (Authoritative)
http://docs.oasis-open.org/wsqm/wsqf/v1.0/WS-Quality-Factors-v1.0-cd02.html
http://docs.oasis-open.org/wsqm/wsqf/v1.0/WS-Quality-Factors-v1.0-cd02.pdf

Latest version:

http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/WS-Quality-Factors-v1.0.doc (Authoritative)
http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/WS-Quality-Factors-v1.0.html
http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/WS-Quality-Factors-v1.0.pdf

Technical Committee:

OASIS Web Services Quality Model TC

Chair:

Eunju Kim (outframe@nia.or.kr), National Information Society Agency

Editors:

Eunju Kim (outframe@nia.or.kr), National Information Society Agency
Yongkon Lee (yklee777@kpu.ac.kr), Individual
Yeongho Kim (kim05@disc.co.kr), Daewoo Information Systems
Hyungkeun Park (phk@kr.ibm.com), IBM
Jongwoo Kim (kjw@hanyang.ac.kr), Individual
Byoungsun Moon (bsmoon@gmail.com), Individual
Junghee Yun (yunjh@nia.or.kr), National Information Society Agency
Guil Kang (guilkang@nia.or.kr), National Information Society Agency

Abstract:

The purpose of this document is to provide a standard for quality factors of web services in their development, usage and management. Web services usually have distinguished characteristics. They are service-oriented, network-based, variously bind-able, loosely-coupled, platform independent, and standard-protocol based. As a result, a web service system requires its own quality factors unlike installation-based software. For instance, as the quality of web services can be altered in real-time according to changes by the service provider, considering real-time properties of web services is very meaningful in describing the web services quality. This document presents the quality factors of web services with definition, classification, and sub-factors case by case. For each quality factor, related specifications are cited with a brief

explanation. This specification can be generally extended to the definition of quality of SOA and to provide the foundation for quality in the SOA system.

Status:

This document was last revised or approved by the OASIS Web Services Quality Model TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at http://www.oasis-open.org/committees/wsqm/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/wsqm/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[WS-Quality-Factors-v1.0]

Web Services Quality Factors Version 1.0. 22 July 2011. OASIS Committee Specification 01. http://docs.oasis-open.org/wsqm/WS-Quality-Factors/v1.0/cs01/WS-Quality-Factors-v1.0-cs01.html.

Notices

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

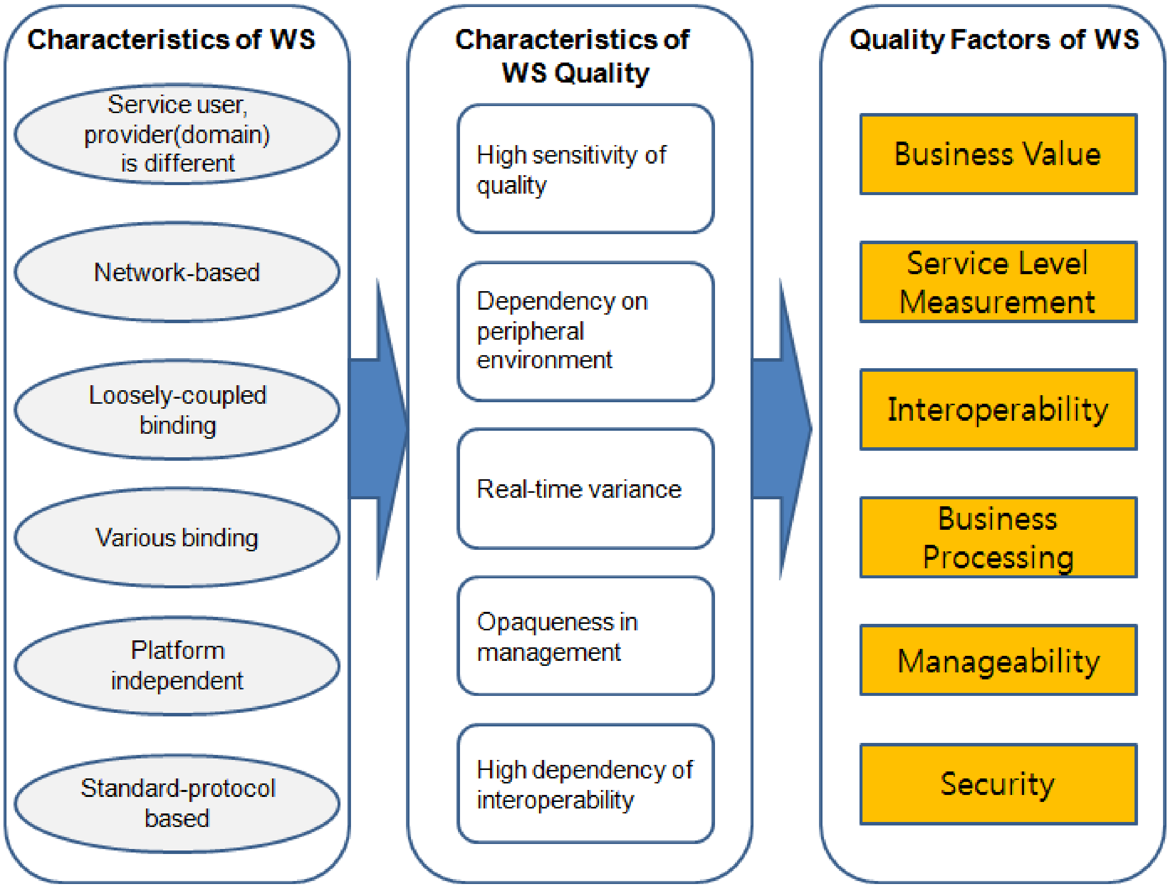
Table of Contents

1 Introduction

The importance of web services has been raised as an enabler of Service Oriented Architecture (SOA). As a result, most software communities who are related in the planning, development and management of SOA have significantly interested in Web Services quality. This document specifies web services quality factors conceptually along with definition and explanation of sub-factors. This chapter presents the basic characteristics of web services and quality factors induced from them.

1.1 Characteristics of WS Quality

Web services have distinguished characteristics different from installation-based software because of their service-oriented nature. The provider and consumer of services could belong to different ownership domains so that there are many cases that a service cannot meet the consumer's service requirements in respect of service quality and content. Web services are usually invoked through networks, so the network performance critically affects the overall web service quality. As a web service client binds to a web service server with loosely-coupled manner and various binding mechanisms, both client and server could be bound easily and flexibly. On the contrary, the client and the server cannot guarantee for proper operating performance. They may be operated platform-independently, so it requires more efforts for guaranteeing interoperability between them. Even though web services are based on standard protocols of communication, misconception of the protocols can produce critical results in non-interoperable services.



<Figure 1-1> Extracting Quality Factors of WS

Due to the characteristics described above, web services show distinct quality characteristics from those of general software. Firstly, the usage of web services is highly sensitive to their quality, especially in regard to performance and business. A web service consumer is willing to change a service while using it if it cannot satisfy his requirements on performance or business. Most of web service consumers have an interest in the quality of web services and thus would correspond to the quality problem immediately. Secondly, as web services are operated in the close relation with the other systems, the web service quality depends on the peripheral technical environment: network, security system, and software resource system, and business effectiveness. For example, the quality of transport media influences deeply on the web service quality. Consequently, even though a web service shows very rapid response time on the server side, we cannot expect rapid response time if the bandwidth of transport network is narrow. In the same way, although a web service has been implemented efficiently, it is difficult to expect good performance of the web service when a service provider has low processing capability. Thirdly, a web service client and a server are bound loosely and variously. The web service client can change the web service server dynamically, so the client can experience considerable variation of web service quality. The client can change web services in real-time when the quality is not satisfied. Fourthly, a web service consumer is usually not able or restricted to manage and control a web service, because in many cases a consumer’s domain is different from a service provider’s. Accordingly, the web service consumer requires guaranteeing the higher level of web service quality. Finally, more effort to assure interoperability of web services is required, because a web service client and a server system could be deployed on heterogeneous platforms and web service developers could misunderstand related standards.

To summarize above, the characteristics of web services lead to the distinguished characteristics of web service qualities unlike those of installation-based software. Therefore, it is required to induce quality items in alignment with consideration of these characteristics of web service quality during overall web service lifecycle.

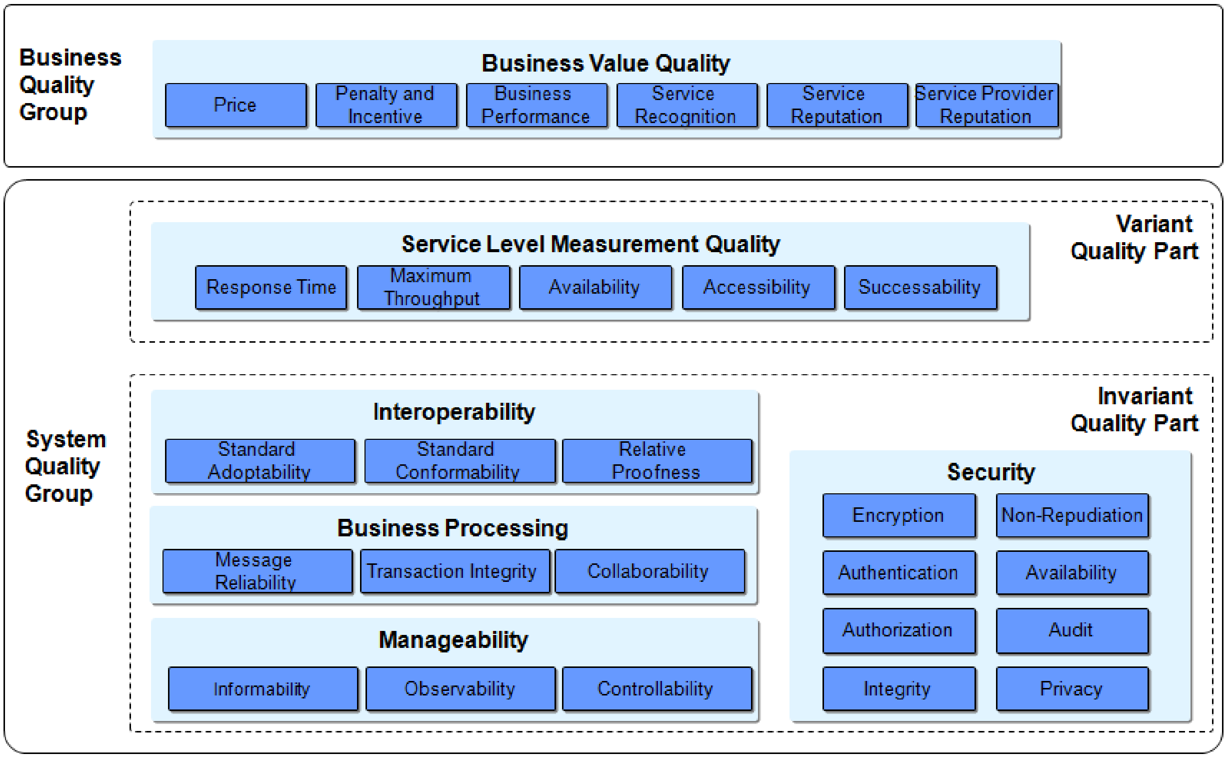
1.2 WS Quality Factors

A web service quality factor refers to a group of items which represent web service’s functional and non-functional properties (or values) to share the concept of web services quality among web service stakeholders. Based on the characteristics of web service quality described previously, the web service quality factors are composed of business value quality, service level measurement quality, interoperability quality, business processing quality, manageability quality and security quality.

Based on whether quality factors are related with business perspective or system perspective, they can be categorized into two groups: the business quality group and the system quality group (Refer to <Figure 1-2>). Business quality group includes only the business value quality factor. System quality group is comprised of the variant quality part and the invariant quality part. The variant quality part includes quality factors whose values can be dynamically varied in run-time while a service is being used. On the while, the invariant quality part refers to quality factors whose values are determined as soon as the service development is completed. The invariant quality part includes interoperability quality, business processing quality, manageability quality and security quality.

Business value quality refers to a business perspective to help to make the right selection of a service by evaluating the business value of web services. For evaluating business value, it includes the sub-factors: price, penalty and incentive, business performance, service recognition, service reputation and service provider reputation. Service level measurement quality measures the performance of web services in numeric value: response time, maximum throughput, availability, accessibility and successability.

Interoperability quality is a quality factor to evaluate whether a web service system conforms to standard adoptability, standard conformability and relative proofness. Web services may be used in mission-critical work between business partners, and in that case reliability and stability of web services are very important quality items. The business processing quality factor evaluates these items, including messaging reliability, transaction integrity and collaborability. Manageability quality is about to whether web services are manageable or managed items, including informability, observability and controllability, web services are also vulnerable to security attack and fraud of their frequent exposure to open networks. Security quality guarantees the safety of web services for use. That is, it is a collection of quality items to evaluate the functionality and the metric performance of a security system. It includes the sub-factors: encryption, authentication, authorization, integrity, non-repudiation, availability, audit and privacy.



<Figure 1-2> Structure of Web Services quality factor

1.3 Audience

The intended audiences of this document include non-exhaustively:

- Quality associates of web services and SOA: quality managers, quality assurers, quality authenticators, quality information providers, etc.
- Architects and developers designing, identifying or developing a system based on web services or SOA concept.
- Standard architects and analysts developing specifications of web services or SOA.
- Decision makers seeking a "consistent and common" understanding of web services or SOA.

1.4 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in **[RFC2119]**.

1.5 Normative References

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

1.6 Non-Normative References

None

2 Business Value Quality

2.1 Definition

When a service party decides to use a web service for business, it should consider surely the value of the service on the business. In some cases, the web service may give positive value such as profit, convenience, collaboration to the service party. But in the other cases, it may impose more burdens on the party than the value it delivers. For example, it may require an extra cost for keeping the service in stable condition or cause economic loss due to service provider's failure to meet promised quality. As a result, web service consumers tend to be very sensitive generally to the value of web services on a business.

The business value of web services means the economic worth delivered by applying web services on a business. The business value depends on the price of a service, a penalty/compensation policy, service recognition, service reputation and service provider reputation. In addition to those sub-factors, business benefit, profit, and ROI (return on investment) caused by web services could be added in the business value quality. But, it's very difficult to evaluate these values by the effect caused by web services alone because the values depend heavily on each individual business context. Thus, we exclude the business benefit, profit, and ROI from the business value quality.

The price and the penalty/compensation sub-factors represent the monetary value, which could be determined by a service provider or the contract between a service provider and a consumer. The service recognition, the service reputation and the service provider reputation are related with the trust of web services, so they could be evaluated as a part of business value.

Business value quality provides a business perspective to help to make a right selection of service by evaluating the business value of web services. Consumers refer to the value of this factor to reach a decision to select the most appropriate web service for a given business.

2.2 Sub Quality Factors

2.2.1 Price

The price sub-factor is a monetary value of service that a consumer pays for services to a provider while or after using web services. The price of a web service can be determined by a service provider. A service consumer considers the price of a web service in respect of the functions, contents, and the quality of the web service in order to make the decision whether he uses the web service. For general software, users pay overall price for a software package. On the contrary, a consumer has to pay the fee of a web service continuously based on the amount of time usage or use data, so he has continuously interest in the service price. Therefore, the price of a web service affects in the usage of the web service. A service provider should decide the appropriate price by considering the service quality and value. In relation to the price factor, a billing method is also important quality factor for measuring business value of a web service. For example, a reasonable and systematic billing system can improve the trust of a web service. A convenient billing system, a discount policy, and mileage points enhance consumer's loyalty.

2.2.2 Penalty and Incentive

Penalty or compensation is the financial compensation for business losses due to nonfulfillment of a contract or failure to meet promised quality. Penalty can be charged to a service provider or a service consumer based on a contract. When a service provider fails to keep service quality levels specified in the contract, the service provider needs to compensate for the loss of a service consumer. The compensation rules need to be specified in the contract. Penalty can be calculated based on service downtime, maximum or average response times, or security requirements of a service, and so on. The performance monitoring of the web service is necessary to determine whether compensation is required or not, and how much compensation is required. Penalty can be charged to a service consumer when the consumer breaches the contract unilaterally, which brings financial loss to a service provider.

On the contrary, incentives as positive rewards can be specified in a contract. For example, an incentive can be paid when a service provider has provided higher quality than the quality level specified in the contract. In addition, an incentive can be paid to a service consumer when the servicer consumer uses a service more than a certain usage level during a given time period.

2.2.3 Business Performance

In the case that a service provider provides commodities and services in the real world as well as information by web services, the performance of business activity provided for them affects the business value of the web services directly. For example, consider a delivery service with which web services are provided for an order and a payment process. In this case, all the processes including the order, confirmation, delivery, notification and payment which are all connected within the information flow and business activity flow. Overall service quality is related with the capability of the business body as well as the quality of web services.

Business performance is defined as the capability of a business party performing business activities for services. The business performance can be measured by the time it takes to complete a business service or the throughput. The time to complete is composed of the duration for performing business activities and a latency for ready or a condition. The throughput is the amount of outcomes for a service per a unit time

2.2.4 Service Recognition

Service recognition quality is defined by how many potential consumers perceive the existence of a service. That is, it is related to the popularity of the service. A highly recognized service means that it has more potential for many people to use the web service. Service recognition can be measured by various methods. For example, it can be estimated by the number of clicks on a service description in a service registry or the number of page views on a service web page.

Service recognition is not derived from the service consumer’s experience of service usage. Also, the service recognition level can be improved through promotion or advertising of a web service. However, it does not guarantee the superiority of the other quality factors of the web service such as response time, availability, and reliability.

2.2.5 Service Reputation

Service reputation is a social evaluation of service consumers toward a web service. It refers to consumers’ opinions on the quality of web services. Service reputation can be evaluated by performing a survey or vote on service quality and consumer satisfaction. In addition, service reputation can be estimated from the replies, comments or reviews of service consumers. Service reputation is very influential for potential service consumers to select web services.

While service recognition reflects expectation of the service value before use, service reputation mainly refers to the experienced service quality after use.

2.2.6 Service Provider Reputation

Service provider reputation is the opinion of the group of service parties toward a service provider on certain criteria. Service provider reputation is an asset that gives the service provider a competitive advantage because a service provider with a good reputation will be regarded as a reliable, credible, trustworthy and responsible one for service consumers. It can be sustained through consistent quality management activities on services as a whole. Service provider reputation can be influenced by customer’s previous experience on other services of the provider as well as advertisement or public relations.

Service consumers pay attention to service provider reputation as well as a service itself. Service provider reputation can be estimated by brand value, financial soundness, the quality of customer service, technical support and sustainability of a service provider. .

3 Service Level Measurement Quality

3.1 Definition

As a service could be provided by third parties and invoked dynamically via network, service performance might be varied by the network speed or the number of connected users at a given time. Service Level Measurement quality is a set of quantitative attributes which describe the runtime service responsiveness in a view of consumers. This quality factor represents how quickly and soundly web services can respond which can be measured numerically on system.

Service Level Measurement Quality consists of five sub-quality factors; response time, maximum throughput, availability, accessibility, and successability.

3.2 Sub Quality Factors

3.2.1 Response Time

Response time refers to duration from the time of sending a request to the time of receiving a response. The response time can be varied by the point of measurement and affected by three types of latency: client latency, network latency and server latency as depicted in <Figure 3-1>.

Client latency refers to the delay time caused by a client system in the whole processing time for a service request. It is a sum of the time taken between 'a client application requests a service' event and 'the request is sent by a client' event (t1~t2), and the time taken between the 'response arrives to the client' event and 'the application system receives the response' event (t7~t8).

Network latency refers to the time taken on a network for transmitting request message and response message. It is a sum of the time taken between 'a client sends a request' event and 'the web services server receives the request' event (t2~t3), and the time taken between 'the server sends a response' event and 'the client receives the response' event (t6~t7).

Server latency is a delay time caused by a server system in the whole processing time for a service request. It is a sum of the time taken between 'the server sends the request' event and 'web services receives the request' event (t3~t4), 'the time taken for processing the service' event (t4~t5), and the time taken between 'the response is sent by the web services' event and 'the server receives the response' event (t5~t6).

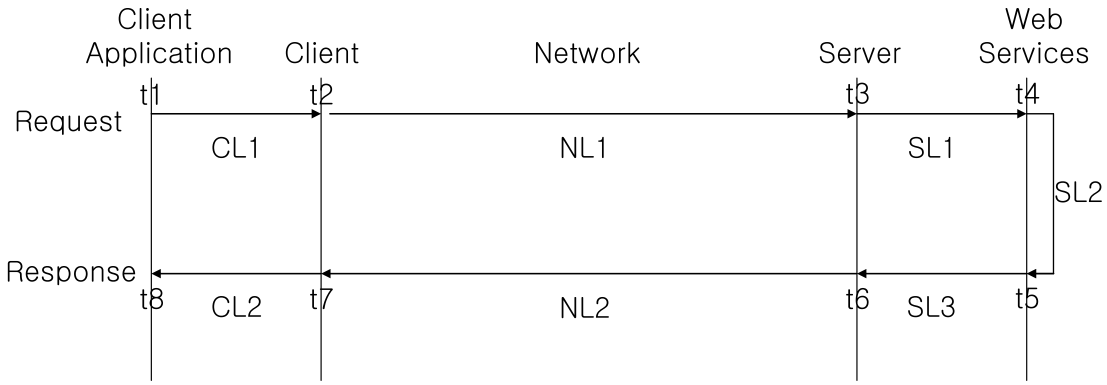
Three types of latency and response time can be calculated by the following formulas.

$$ClientLatency = CL1 + CL2$$

$$NetworkLatency = NL1 + NL2$$

$$ServerLatency = SL1 + SL2 + SL3$$

$$\mathrm{Re}\,sponseTime = ClientLatency + NetworkLatency + ServerLatency$$



* CL : Client Latency, NL : Network Latency, SL : Server Latency, ti : Measurement Time

<Figure 3-1> Response Time and Latency

3.2.2 Maximum Throughput

Maximum throughput refers to the maximum amount of services that the service provider can process in a given time period. It is the maximum number of responses which can be processed in a unit time. The following formula expresses the maximum throughput.

$$MaximumThroughput = \max\left(\frac{Numberof\ RequestsProcessedbyServiceProviderInMeasuredTime}{MeasuredTime}\right)$$

3.2.3 Availability

Availability is a measurement which represents the degree of which web services are available in operational status. This refers to a ratio of time in which the web services server is up and running. As the DownTime represents the time when a web services server is not available to use and UpTime represents the time when the server is available, Availability refers to ratio of UpTime to measured time. In order to calculate Availability, it is conveniently rather using DownTime than UpTime and it can be expressed as the following formula.

$$Availability = 1 - \frac{DownTime}{MeasuredTime}$$

3.2.4 Accessibility

Accessibility represents the probability of which web services platform is accessible while the system is available. This is a ratio of receiving Ack message from the platform when requesting services. That is, it is expressed as the ratio of the number of returned Ack message to the number of request messages in a given time. To increase accessibility, a system needs to be built in expansible architecture.

$$Accessibility = \frac{Numberof\ AckMessage}{NumberofRequested\ Message}$$

3.2.5 Successability

Successability is a probability of returning responses after web services are successfully processed. In other words, it refers to a ratio of the number of response messages to the number of request messages after successfully processing services in a given time. ‘Being successful’ means the case that a response message defined in WSDL is returned. In this time, it is assumed that a request message is an error free message.

$$\mathit{Successability} = \frac{\mathit{Numberof\ ResponseMessage}}{\mathit{Numberof\ Requested\ Message}}$$

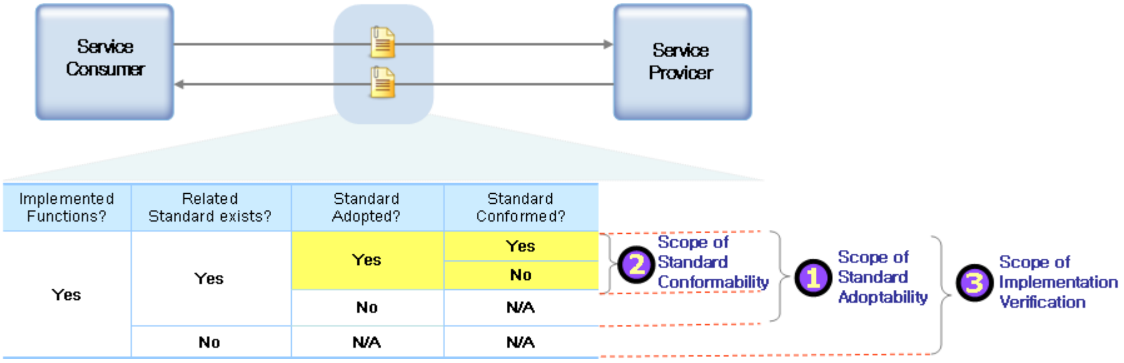
4 Interoperability Quality

4.1 Definition

For executing web services, there should be no semantic and technical problems in processing a message transmitted between a service provider and a service consumer. No semantic problem refers to the process when a receiver understands a message in the exact meaning as the sender intended. A prerequisite condition for the mutual understanding of semantics in a message, the name of service, the name and type of parameters, the type of return values are consistent between a service provider and a service consumer. This prerequisite condition may be satisfied if the service consumer implemented its system exactly according to the information of a service description (i.e. WSDL). But, this requires an agreement between all service parties for using service contents such as the name of items in e-documents and codes without any semantic problems. No technical problem refers to the conditions when all components for messaging including transport, security, reliability, encoding, and message structure coincide during implementation. Two systems of communication parties are said to be interoperable when they exchange and use information as if both could operate appropriately on the same platform. Implementing the messaging technology adopts related standards for assuring interoperability. If there is no standard available, one of service associates could adjust its technical implementation to the other’s or both can agree to match their implementation specifications bilaterally. However, if a standard exists, the service associates can achieve interoperability by adjusting their implementation to the standard specification. Even though a service associate follows a standard, interoperability problems could arise in the case that an implementer misunderstands the standard or implements a module with a different intention. In some cases, the implementer may add new functions not described in the standard arbitrary. However this difference of a platform or network device could damage the interoperability between the two parties.

Considering the above cases, we can categorize the problems of interoperability into 3 groups: ①a system which has been implemented by not adopting a standard, ②a system which has been implemented according to a standard, with some functions implemented without regard to the standard, ③ a system which has been implemented according to standards properly, but which has a problem of interoperability due to difference of platform of network. All of these issues must be considered in evaluating interoperability of a service system

Interoperability quality includes standard adoptability, standard conformability and relative proofness as shown in <Figure 4-1>. Standard Adoptability of web services evaluates how many functions of a web service are implemented by adopting related standards. The function of a web service means necessary requirements such as user authentication, data encryption, service delivery, transaction processing, etc. They could also include the original features of the business such as codes, document formats, business terms etc. Standard conformability of a web service evaluates whether the standards adopted to implement the functions of the web service conforms completely and correctively to the specification of the standards. Relative proofness evaluates whether a client and a service can communicate successfully on specific platforms.


Service Consumer
Service Provicer
Scope of Standard Conformability
Scope of Standard Adoptability
Scope of Implementation Verification
2
1
3

Implemented Functions?	Related Standard exists?	Standard Adopted?	Standard Conformed?
Yes	Yes	Yes	Yes
			No
		No	N/A
	No	N/A	N/A

<Figure 4-1> Interoperability of web services

4.2 Sub Quality Factors

4.2.1 Standard Adoptability

In order to guarantee interoperability of web services, functions of a web service should be implemented by adopting related standards. Standard Adoptability is measured by the ratio of functions which are implemented by adopting related standards. Standard adoption function f is defined on the set of functions X={x_1, …., x_n} of a web service S and returns one of binary values, 0 and 1. *f* is formulated as follows:

$$f(x_i) = \begin{cases} 1 & \text{if function } x_i \text{ is implemented by adopting of related standards} \\ 0 & \text{otherwise} \end{cases}$$

Based on the standard adoption function *f*, *standard adoptability* of web service *S* is defined as follows:

$$StandardAdoptability(S) = \frac{\sum_{i=1}^{n} f(x_i)}{n}$$

4.2.2 Standard Conformability

Assuming that a web service S has *n* functions, and only *m* functions of them are implemented by adopting related standards. *Standard conformability function g* is defined by the set of standard adopted functions X_a={ x_1, …., x_m} of web service *S* and returns one of binary values, 0 and 1. *g* is defined as follows:

$$g(x_i) = \begin{cases} 1 & \text{if function } x_i \text{ comforms all adopted standards} \\ 0 & \text{otherwise} \end{cases}$$

Based on the standard conformability function, *standard conformability* of web service *S* is defined as follows:

$$StandardConformability(S) = \frac{\sum_{i=1}^{m} g(x_i)}{m}$$

4.2.3 Relative Proofness

Relative proofness indicates that web services are successful in exchanging and using the information between two special system platforms. In real environments, services based on a technology platform cannot be fully interoperable with other services on a different technology platform even if a standard conformability test is passed. Whether the functions of web services are implemented by standards, vendor specification or non-standards, relative proofness of service interoperability is tested and verified in a real service platform that satisfies specific environments. VPI (Verified Platform Information) represents the basic information of an opponent’s platform and additional descriptions of the verification when web services are tested and verified in real service environments of the opponent’s platform.

$$Relative\ Proofness = \{VPI_1, VPI_2, ..., VPI_n\}, where\ n\ is\ the\ number\ of\ \ platforms\ verified$$

4.3 Relationships to other Standards

- WS-I Basic Profile 1.1

This is a profile for interoperability of SOAP, WSDL, UDDI and is administered by WS-I.

URL: http://www.ws-i.org/Profiles/BasicProfile-1.1.html

- WS-I Basic Profile 1.2

This is a profile for interoperability of SOAP, WSDL, UDDI and is administered by WS-I.

URL: http://ws-i.org/profiles/BasicProfile-2.0-WGD.html

- WS-I Basic Security Profile Version 1.0

This is a profile for interoperability of web services security and is administered by WS-I.

URL: http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0-2007-03-30.html

- WS-I Basic Security Profile Version 1.1

This is a profile for interoperability of web services security and is administered by WS-I.

URL: http://www.ws-i.org/Profiles/BasicSecurityProfile-1.1.html

- WS-I Reliable Secure Profile 1.2

This is a profile for interoperability of reliable message and secured transmission and is administered by WS-I. (WS-ReliableMessaging 1.1, WS-SecureConversation 1.3)

URL: http://www.ws-i.org/Profiles/ReliableSecureProfile-1.0.html

- WS-I Simple SOAP Binding Profile Version 1.2

This is a profile for interoperability of SOAP Binding and is administered by WS-I.

URL: http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html

5 Business Processing Quality

5.1 Definition

As the applying areas of web services are growing on a wide scale, the cases that use them in communication between service units (i.e., enterprise, department, agency, program, division) are increasing rapidly. Applying web services in business means that the service unit executing them has to take responsibility of the execution result.

For applying web services in business, the intention of service providers and consumers has to be reflected correctly in business results. A service unit can assure correctness and reliability in the business context for business processing. In order to achieve this, a web service platform for business should possess functions of reliable messaging, transaction processing, and collaborability. These functions could be optionally used according to the requirement of a service unit. Accordingly, business processing quality is defined as the capability of a web service platform for assuring correctness and reliability in business processing.

5.2 Sub Quality Factors

5.2.1 Messaging Reliability

Most networks and their communication channels are not fairly reliable in real world. They are exposed to unexpected circumstance variances, internal system errors and inexperienced users. As a result, the messages for service requests and response could be lost and duplicated and their sequence confused. These cases would cause serious results in business, which could give a fatal blow to a service unit.

Messaging reliability refers to the capability for messaging functionality which ensures the intention of messaging for service units. The level of messaging reliability depends on the requirement of service units. For example, a service unit could require a very restrict level of reliability in which a message is transferred once and only once at any case (i.e. money transfer). In some cases, a service unit requires at least one message transferred (i.e. request message for search). The sequence of messages could be disregarded (i.e. request for an invoice). On the other hand, the sequence of messages could have major effects to a business (i.e. request for stock trading).

The level of messaging reliability could be determined by the agreement of parties participating in a business relationship. The business parties MUST provide the reliability functions which guarantee the level of reliability more than the level agreed. The messaging reliability includes 4 basic factors, which can be combined. Certain combinations are of particular interest due to their widespread application: exactly once and ordered (also referred to as exactly once ordered).

- Transmitting at least once (guaranteed delivery)

  Every message MUST be transmitted at least once or an error MUST be raised on at least one endpoint. Some messages SHOULD be delivered more than once until an Acknowledgement is received from the receiver. Each message MUST be transmitted at least once, otherwise both receiver and sender MUST issue an error message. In addition, the sender MUST keep retransmitting the message until Ack is received from the receiver.

- Transmitting at most once (guaranteed duplicate elimination)

  Each Message MUST be transmitted at most once without duplication. Otherwise an error will be raised on at least one endpoint. The receiver MUST block the duplicated message. Each message MUST be transmitted at most once. The receiver MUST block the duplicated message. Each message MUST be transmitted precisely once, otherwise both receiver and sender MUST issue an error message. In addition, the sender MUST keep retransmitting the message until Ack is received from the receiver. The receiver MUST block the duplicated message.

- Transmitting precisely once (guaranteed delivery and duplicate elimination)

Each message MUST be transmitted precisely once, otherwise both receiver and sender MUST issue an error message. In addition, the sender MUST keep retransmitting the message until an Acknowledgement is received from the receiver. The receiver MUST block the duplicated message. This delivery assurance is the logical "and" for the two prior delivery assurances.

- Transmitting sequentially(guaranteed delivery order)

Messages in sequence MUST be transmitted from where they are created to the receivers who the messages are intended to be orderly delivered to. To do this, the sender MUST sequentially send the message with message sequence information embedded in the messages. And the receiver would have to be able to rearrange the messages according to sequence information embedded in the messages.

5.2.2 Transaction Integrity

Transactions running across multiple services over multiple domains need to maintain business integrity. Traditionally, a transaction is a business processing unit (a unit of work) that involves one or more services and is either completed in this entirety or is not done at all.

Transaction integrity refers to whether a service has functionality for processing transactions or a transaction integrity platform environment can be applied. The transaction model of web services can be divided into either a short-term transaction (atomic transaction) or a long-term transaction (business activity).

- Short-term transaction (atomic transaction, all-or-nothing property)

Short-term transaction is a transaction which requires a service locked for a short period of time such as purchasing a book online. The major function of the transaction is to reset to default when a request of the transaction is not processed or a request of a transaction is processed so that all the changes resulting from the transaction are applied. This transaction is also called an Atomic transaction and MUST satisfy the following 4 ACID (Atomicity, Consistency, Isolation, and Durability) requisites.

- Atomicity

The transaction completes successfully (commits) or if it fails (aborts), all of its effects are undone (roll-back)

- Consistency

Transactions produce consistent results and preserve application specific invariants

- Isolation

The results of a task are not shared with other transactions unless it is successfully completed.

- Durability

Once a transaction is successfully completed, its results SHOULD be permanently applied to a system.

- Long-term transaction (long-running transaction)

Long-term transaction refers to a transaction which requires a longer processing time or its resources cannot be locked exclusively during processing. It is also referred to business activity. Because a long-term transaction consists of some short-term transactions or independent web services, Commit or Roll-back mechanisms of short-term transactions cannot be used. Therefore, long-term transaction quality is evaluated by the following criteria, not by ACID attributes.

- Consistency

Long-term transaction SHOULD be able to change consistently the status of participating systems.

- Compensatory

Long-term transaction MUST support an independent and alternative flow to compensate for failed transactions. Because long-term transactions consist of some short-term transactions or independent web services, an alternative flow of processing is needed without individual processes being reset to default.

5.2.3 Collaborability

The application of web service collaboration is prevalent to implement business processes. A business process can be defined as the execution of activities according to a defined set of rules in order to achieve a common goal between participants. Collaborability is the capability of a service platform to define, control and manage service flow between participants. There are two types of collaborability: orchestration which is executed or coordinated by single conductor and choreography which is executed or coordinated by multiple participants.

- Orchestration

  Orchestration is a technique used to compose hierarchical and self-contained service-oriented business processes that are executed and coordinated by a single agent acting in a "conductor" role [OASIS, Reference Architecture for Service Oriented Architecture]. In other words, orchestration is the technique to define and execute a flow or procedure of services to achieve business processing. An orchestration is typically implemented using a scripting approach to compose service-oriented business processes. This typically involves use of a standards-based orchestration scripting language. An example of such a language is the Web Services Business Process Execution Language (WS-BPEL) [WS-BPEL].

- Choreography

  Choreography is a technique used to characterize and to compose service-oriented business collaborations based on ordered message exchanges between participants in order to achieve a common business goal. [OASIS, Reference Architecture for Service Oriented Architecture] In other words, choreography defines the sequence and dependencies of interactions between multiple participants to implement a business process composing multiple web services. Choreography differs from orchestration primarily in that each party in a business collaboration describes its part in the service interaction in terms of public message exchanges that occur between the multiple parties as standard atomic or composite services, rather than as specific service-oriented business processes that a single conductor/coordinator (e.g., orchestration engine) executes [OASIS, Reference Architecture for Service Oriented Architecture]. To be specific, choreography describes the sequence of interactions for web service messages. WSDL describes the static interface and choreography defines the dynamic behavior external interface. It is the Peer-to-Peer collaboration model of exchanging messages among related partners as a part of a bigger business transaction with many participants.

5.3 Relationship to other Standards

- WS-Reliability 1.1

  WS-Reliability is a SOAP-based protocol for exchanging SOAP messages with guaranteed delivery, no duplicates, and guaranteed message ordering.

  URL: http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf

- WS-ReliableMessaging 1.2

  WS-Reliable Messaging is a protocol that allows messages to be transferred reliably between nodes implementing this protocol in the presence of software component, system, or network failures.

  URL: http://docs.oasis-open.org/ws-rx/wsrm/200702/wsrm-1.2-spec-os.html

- WS-Context 1.0

  WS-Context provides a definition, structuring mechanism, and service definitions for organizing and sharing context across multiple execution endpoints

  URL: http://docs.oasis-open.org/ws-caf/ws-context/v1.0/wsctx.html

- WS-Coordination 1.2

  WS-Coordination specifies an extensible framework for providing protocols that coordinate the actions of distributed applications.

  URL: http://docs.oasis-open.org/ws-tx/wstx-wscoor-1.2-spec.html

- WS-AtomicTransaction 1.2

WS-AtomicTransaction specifies the definition of the Atomic Transaction coordination type that is to be used with the extensible coordination framework described in WS-Coordination.

URL: http://docs.oasis-open.org/ws-tx/wstx-wsat-1.2-spec-os/wstx-wsat-1.2-spec-os.html

- WS-BusinessActivity 1.2

WS-BusinessActivity specifies the definition of two Business Activity coordination types: AtomicOutcome or MixedOutcome, that are to be used with the extensible coordination framework described in the WS-Coordination specification.

URL: http://docs.oasis-open.org/ws-tx/wstx-wsba-1.2-spec-os/wstx-wsba-1.2-spec-os.html

- WSBPEL 2.0

WSBPEL describes a business process activity using web services and defines the way they are linked with each other. Using Orchestration, business process collaboration is composed. WS-BPEL 2.0 is approved as OASIS standard.

URL: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

- WS-CDL 1.0

Web Services Choreography Description Language is a language to describe XML based web services collaboration as choreography. It is a standard for the decentralized business process.

URL: http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/

6 Manageability Quality

6.1 Definition

The more web services gain weight in business, the more the web service management scheme is needed for maintaining web service quality. Web service can be managed not only locally by a web service manager or provider, but also remotely by a consumer and a third party system. Web service management may be a prerequisite for the foundation of trust between a service consumer and a provider.

Manageability is defined as an ability which keeps a web service and its resources being manageable. At this point, the web service resource includes the software and hardware components used by the web service and a platform on which the web service operates. Manageability can be achieved by implementing manageability capabilities, which are exposed as an access point, for each web service. A manageability implementation means an implementation of a manageability endpoint and all of its manageability capabilities. The manageability capability helps targeting a web service, provides a function to monitor operational status, and controls operations along with web service protocol. As the manageability capability enables a service consumer to use web services with reliability and stability, it may be an important criterion for one to select a web service. The manageability is classified into 3 sub-factors: informability, observability and controllability.

6.2 Sub Quality Factors

6.2.1 Informability

The management of a web service requires the primitive information to be settled in the implementation phase in order to cope with troubles and to support a web service operation. Informability is a sub-quality factor to measure whether the primitive information can provide enough to manage a web service. The primitive information for managing a web service is divided into the manageability access information, assessment of web service management capability and the web service primitive information.

- Manageability access information

  The manageability capabilities are exposed as a web service access endpoint thus a manageability consumer SHOULD get the access point for managing a web service and manageability access information. There are two ways to achieve this. One is to implement an additional web service whose functionality is to return a manageability endpoint for managing a web service. The other is to implement each web service equipped with an operation which returns its own manageability endpoint. The former has a disadvantage that a consumer has to know previously the reference of a web service which informs the access point of manageability capabilities. In the latter, a web service developer is burdened to implement an additional operation to inform the manageability endpoint. By utilizing both, a manageability consumer SHOULD get the manageability endpoint precisely through simple web service interface.

- Web service primitive information

  The manageability capability can provide primitive information of a web service to be managed. The primitive information includes web service properties (e.g. protocol version number, encryption algorithm, messaging pattern, etc), description of web services and their resources, characteristics of a manageability implementation, relation information between web services and resource. The manageability endpoint SHOULD also have an identity capability to discriminate whether two web services are the same by referring to their identity information.

6.2.2 Observability

Observability measures how effectively a manageability implementation can provide status information of a web service. The status of a web service system can be revealed by gathering the values of performance metrics. Therefore, observability can be evaluated by how effectively the metrics are taken when gathering information. The effectiveness of gathering metrics is represented in three aspects:

- How much are metrics provided?
- How exactly metrics are provided?
- Are the metrics provided in real-time?

A manageability consumer can get the status information by two methods: monitoring and notification. In the former, he can request the information anytime to a manageability endpoint actively, then a corresponding manageability capability returns a metric value based on monitoring results. In the latter, a manageability consumer subscribes previously significant events or issues to be notified. Then, when a subscribed event or trouble occurs, he will be notified. The target information to be observed includes all the operational status information such as performance metrics, a utilization ratio of memory, and a history of messages.

6.2.3 Controllability

There may be a case that a web service and its resources have to be regulated for keeping a stable status or coping with performance degradation. For example, a manageability consumer may increase the size of a web service message queue when there are too many messages received at the same time. If there is an error found in the messaging process, a web service platform SHOULD be stopped to cope with the problem. Thus, Controllability measures whether a manageability implementation can provide enough control functions to keep a web service in controllable status. The control functions are classified into operation control functions and configuration control functions.

- Operation control functions

  These are to change an operational status of a web service by executing commands such as start, stop, fork, and exit to the web service or related resources.

- Configuration control functions

  These are to modify the value of configuration parameters. For example, according to the change of a circumstance, a manageability consumer wants to adjust the value of a web service configuration such as a queue size, an encoding method and an encryption algorithm.

  For measuring controllability, the scope, stability, voluntary, and easiness of control functions in a manageability implementation MUST be evaluated.

6.3 Relationship to other Standards

- W3C Web Services Architecture

  It defines the structure of web services including the state model of web services and the structure for management. The standardization was completed in January, 2004. It consists of Web Services Architecture for defining web services structure, Web Services Usage Scenarios, and Web Services Management: Services Life Cycle for defining the state model of web services.

  URL: http://www.w3.org/TR/ws-arch/

- OASIS Web Services Notification (WSN) v1.3

  WSN is the OASIS standard to define the mechanism of asynchronous message exchange using an event. It was completed in October, 2006. It consists of WS-BaseNotification to define the message exchange mechanism of basic event method, WS-BrokeredNotification to define the asynchronous message exchange mechanism using the broker like MOM and WS-Topics for exchanging the event information.

  URL: http://www.oasis-open.org/specs/#wsnv1.3

- OASIS Web Services Resource Framework (WSRF) v1.2

  WSRF proposes the common framework for managing various resources that exist on networks. It was completed on April, 2006. It consists of WS-Resource to define resources, WS-ResourceProperties to define exchange method of the resources, WS-ResourceLifetime to define lifecycle of the resources, WS-ServiceGroup for define the way for managing the numerous resources as a group and WS-BaseFaults to define basic malfunctions that can occur during the attributes management process. Although WS-Resource Metadata Descriptor is not approved as a standard, it is very important for managing web services and is being used in TC such as WSDM.

URL: http://www.oasis-open.org/specs/#wsrfv1.2

- OASIS Web Services Distributes Management (WSDM) v1.1

WSDM proposes the framework for managing various resources on networks. It consists of MUWS (Management Using Web Services) and MOWS (Management of Web Services).

URL: http://www.oasis-open.org/specs/#wsdmv1.1

7 Security Quality

7.1 Definition

Security quality is the degree of ability that can protect web services from various threats on confidentiality, integrity and availability. Typical threats on web services environment are unauthorized access, exposure, forgery and destruction of web services. These security threats can destroy web services environment by identity theft, forgery of financial data and blockage of services. Therefore, security quality is becoming significantly critical and essential for web service.

This security quality should be considered on two technical perspectives.

- Transport Level Security

  Transport level security is to provide a secure data transfer on the transport layer. This is regardless of the characteristics and complexities of web services because its implementation is based on transport layer protocols and web services are applied on the application layer. Because security on the parts of a message is not supported, it has a limitation that the security cannot be guaranteed during intermediary processing.

- Message Level Security

  Message level security is a method which provides security service using XML based message to provide confidentiality and integrity of SOAP messages. Message level security uses End-To-End model, thus it provides persistent security.

The sub quality factors of the security quality include encryption, authentication, authorization, integrity, availability, audit, non-repudiation and privacy. Each quality factor is related to confidentiality, Integrity and availability.

7.2 Sub Quality Factor

7.2.1 Encryption

Encryption is data protection and disposure control on web services with cryptographic functionality to prevent unauthorized user access of confidential or sensitive information. Frequently ‘encryption’ is regarded as ‘confidentiality’ in a narrow sense.

- Transport Level Encryption

  Transport level encryption supports only the encryption of the entire message, using the cryptographic features provided by SSL, TLS or IPSec protocol. It ensures confidentiality of data when sending and receiving the data on the transport layer. However, if it has any intermediary processing, the data should be decrypted and revealed to the intermediator.

- Message Level Encryption

  To ensure confidentiality of web services’ message, message level encryption is provided by either the XML-Encryption or the cryptographic functionality (e.g. PGP for the attachment) in the WS-Security. Especially, the XML-Encryption can encrypt the part of the message with the WS-SecurityPolicy along with message protection policy, thus confidentiality of the service on transport level can be improved.

Firstly, the encryption quality can be measured by whether the encryption feature is applied or not. If the encryption feature is applied, the strength of the encryption function (e.g. AES-128-CBC is a stronger encryption algorism than 3DES-CBC.), the size of encryption key, or the life cycle of key will affect the encryption quality.

7.2.2 Authentication

Authentication is the identification of services' consumer/ provider and the verification of the credential that can be assured by the identification and trusted for the transmission.

- Transport Level Authentication

  Transport level authentication is the same method as traditional web environment. Normally the authentication method is ID/Password, X.509 based certificate, Kerberos and so on. And transport level authentication is an authentication under point-to-point model. Thus, if it has no intermediary processing, it can be trusted and provide many vendors with good interoperability. But, if intermediary processing exists, the consumer's credential cannot be trusted and the provider cannot know who the first origin is, because the identification is propagated and changed.

- Message Level Authentication

  Message level authentication is an XML message based authentication using standard of W3C such as WS-Security. Message level authentication method is sometimes similar to transport level authentication. It can use ID/Password, X.509 based certificate token, or Kerberos token just inserted in XML. And for more secure requirements and standard based interoperability, a more special authentication method is used with XML token such as SAML, WS-Federation, Liberty, and so on. But, because the interconversion is not supported among XML tokens, the provider must consider STS (Security Token Service) in WS-Trust specification under the mixed XML token environment.

The authentication quality can be measured by whether the authentication feature is applied or not. And it can be the strength of the implementation of authentication method such as password policy, multi-factor authentication or the possibility of bypass in an authentication mechanism.

7.2.3 Authorization

Authorization is the control over access on service/message for each actor's right. It is used to support Confidentiality and Integrity. It uses various policies, access control models and security levels as means of support of Authorization.

- Transport Level Authorization

  Transport level authorization refers to the access control on the resources of users of the transport channel. It is implemented on application, middleware – web application server, directory, or security device using various security models such as RBAC (Rule based access control) and so on. In this case, the Transport level authorization has the same limitation as the point-to-point model. In essence the transport level authorization is decentralized control. Thus, Origin authorization policy cannot persist under transport level authorization with intermediary processing.

- Message Level Authorization

  Message level authorization is the XML messaged based authorization using standard of W3C such as WS-Security. It is represented with XACML as service authorization policy and WS-SecurityPolicy as message protected policy. Moreover, it can be encapsulated with the access right defined as XACML or SAML to carry XACML. Using this function, the access right over the actual resources is controlled.

The authorization quality can be measured by whether the authorization feature is applied or not. It can be the possible bypass in an authorization mechanism.

7.2.4 Integrity

Integrity is to protect from unauthorized service/message modify, delete and create. It uses access control and briefing message.

- Transport Level Data Integrity

Transport level data integrity refers to a feature such as the packet comparison and message digest provided by IPSEC or TLS to provide the data integrity when sending and receiving data between transport channels.

- Message Level Data Integrity

Message level data integrity refers to the data integrity of SOAP message level. It can be guaranteed by XML-Signature in WS-Security. Also, XKMS to manage the digital signature for the data integrity can be used.

The Integrity quality can be measured by whether the integrity feature is applied or not.

7.2.5 Non-Repudiation

Non-repudiation is to prevent receivers and senders from denying that they send and receive messages. It uses digital-signature for non-repudiation.

- Transport Level Non-Repudiation

Under the transport layer, non-repudiation cannot be built using digital-signature. Usually it is included in application or business logic. Hence, we do not mention non-repudiation of transport level here.

- Message Level Non-Repudiation

Message level non-repudiation- can be built using XML-Signature in WS-Security. XKMS is used to manage the digital signature for non-repudiation on the transport level. The non-repudiation quality of the message level can be measured by whether the non-repudiation feature is applied or not.

7.2.6 Availability

Availability is to allow only authorized consumers to access services whenever they need. The techniques such as IDS, IPS or Anti-DoS(Denial of Services) can be implemented to ensure availability.

- Transport Level Availability

Transport level availability refers to service continuity on the transport layer to prevent exhausting web resources by excessive or malicious requests which can make services unavailable. It is established by surveilling the packets with IDS, IPS or Anti-DoS equipment on the transport layer.

- Message Level Availability

Message level availability refers to service continuity on the message level to protect malicious XML messages which can make services unavailable. It is accomplished by filtering and verifying messages on the application level or XML firewall.

The availability quality can be measured by whether the availability feature is applied or not.

7.2.7 Audit

Audit is the capability to trace and verify activities and events on web services providing and consuming. During the security audit process, security vulnerability or security attack can be identified from the traced information.

- Transport Level Audit

Transport level audit is to trace and verify send/receive information on transport layer.

- Message Level Audit

Message level audit is to trace and verify request/response message on the application layer.

The audit quality can be measured by whether the audit feature is applied or not.

7.2.8 Privacy

Privacy is the protection of sensitive information of web services consumers and providers. To support privacy, it is necessary to guarantee above, the sub quality factors of security quality. Additionally privacy policy is required in compliance with law and regulations related with privacy.

The privacy quality can be measured by whether the protection of privacy information is implemented and the privacy policy is defined appropriately.

7.3 Relationship to other Standards

- W3C XML Signature

  XML digital signature standard

  URL: http://www.w3.org/TR/xmldsig-core/

- W3C XML Encryption

  XML encryption standard

  URL: http://www.w3.org/TR/xmlenc-core/

- W3C XKMS (XML Key Management Specification)

  Key management service standard which makes the integration between PKI and XML application easy

  URL: http://www.w3.org/2001/XKMS/

- OASIS WS-Security (Web Services Security: SOAP Message Security)

  The standard to provide the authentication, integrity, non-repudiation, and confidentiality of SOAP

  URL: http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

- MS, VeriSign, IBM WS-SecurityPolicy (Web Services Policy)

  The standard to provide the security policy applied on WS-Security.

  URL: http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.html

- OASIS SAML (Security Assertion Markup Language)

  The standard to reliably exchange the authentication and approval information based on XML

  URL: http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf

- OASIS XACML (eXtensible Access Control Markup Language)

  The access control standard that consists of XML based policy language and access control decision, request/response language

  URL: http://www.oasis-open.org/committees/download.php/2406/oasis-xacml-1.0.pdf

- MS, VeriSign, BEA, IBM WS-Trust (Web Services Trust Language)

  The standard about issuing and exchange the security token and configuring trust relationship in various trust domains

  URL: http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.html

- MS, VeriSign, BEA, IBM WS-Federation (Web Services Federation Language)

  The mechanism definition to intervene user identification, attributes, and authentication among web services applications what belong to different security domains.

  URL: http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html

8 Conformance

A product, document or service conforms to this specification if it adopts or provides all the quality factors and sub quality factors in this specification. If there are some quality factors or sub quality factors with no value or not adopted in a product, then they should be supplied with no value or the value indicating “not applicable” such as N/A.

The name and meaning of each quality factor or sub quality factor should not be changed. It is possible to include additional quality factors or sub quality factors which are not given in this specification. For the adopted or provided quality factors, a product, document or service should satisfy all of the MUST or REQUIRED level requirements defined in the part of the quality factors in this specification.

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

- Eunju Kim, National Information Society Agency
- Yongkon Lee, Individual
- Yeongho Kim, Daewoo Information Systems
- Hyungkeun Park, IBM
- Jongwoo Kim, Individual
- Byoungsun Moon, Individual
- Junghee Yun, National Information Society Agency
- Guil Kang, National Information Society Agency

External Contributors:

- Dugki Min, Konkuk University
- Mujeong An, LG CNS
- Sojung Kim, National University of Singapore