# A QOS AWARE QUANTITATIVE WEB SERVICE SELECTION MODEL

R. Dinesh Kumar[#1] and Dr.G. Zayaraz[*2]

[#1]PG Scholar, [*2] Associate Professor, Department of Computer Science and Engineering,
Pondicherry Engineering College, Puducherry, India.

**Abstract— Web service is a core technology for sharing information resources and integrating processes in companies or organizations. 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 selection is also an indispensable process for web service composition as to select best web service to a client's requirement. In this paper, we have proposed a web service selection model using Analytic Hierarchy Process which is used to select best web service based on QoS Constraints. The QoS manager actuates as an agent for service providers and clients to perform publish and find web service operations. The QoS attributes of a web service such as response time, throughput, reliability, availability and cost will be optimized and ranked by the Analytic Hierarchy Process. The highest ranked service will be selected and provided to the requested for further processing.**

**Keywords—** ***Web Service Selection, Quality of Service, QoS, Analytic Hierarchy Process, AHP.***

## I. 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 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) [2] specification. 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 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.

Today, the principal use of the World Wide Web is for interactive access to documents and applications. In almost all cases, such access is by human users, typically working through Web browsers, audio players, or other interactive front-end systems. The Web can grow significantly in power and scope if it is extended to support communication between Applications from one program to another.

A web service is any service that is available over the Internet, uses a standardized XML messaging system, and is not tied to any one operating system or programming language. (See Figure 1.)

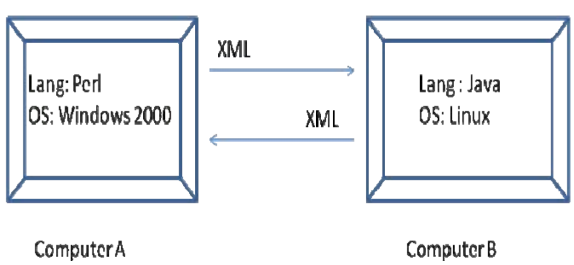


Fig 1: A Basic Web Service

There are several alternatives for XML messaging. For example, you could use XML Remote Procedure Calls (XML-RPC) or SOAP, both of which are described later in this chapter. Alternatively, you could just use HTTP GET/POST and pass arbitrary XML documents. Any of these options can work. Although they are not required, a web service may also have two additional (and desirable) properties:

**A web service should be self-describing:** If you publish a new web service, you should also publish a public interface to the service. At a minimum, your service should include human-readable documentation so that other developers can more easily integrate your service. If you have created a SOAP service, you should also ideally include a public interface written in a common XML grammar. The XML grammar can be used to identify all public methods, method arguments, and return values.

**A web service should be discoverable:** If you create a web service, there should be a relatively simple mechanism for you to publish this fact. Likewise, there should be some simple mechanism whereby interested parties can find the service and locate its public interface. The exact mechanism could be via a completely decentralized system or a more logically centralized registry system.

The strength of various QoS aware Web service mechanisms can be evaluated based on the following seven parameters. They are:

C1. Is the requester allowed to specify the desired QoS properties for selection?

C2. Are the requester’s preferences (weight) for QoS properties considered for selection?

C3. Is the selection mechanism that optimizes (filters) the candidate Web services (functionally similar Web services) based on requirements?

C4. Does the selection mechanism explore multiple Web services in many cases (situations)?

C5. Does the selection mechanism find the best Web services correctly in all circumstances?

C6. Is the selection mechanism defined for multiple QoS properties?

C7. Does the selection mechanism allow the requester to specify desired QoS property values in the requirements?

The evaluation parameters are defined to analyze the strength and weakness of specific QoS aware Web service selection mechanisms. The parameters consider the nature and accuracy of selection result, facility provided to supply QoS requirements involving multiple QoS properties, requester’s QoS preferences and optimization of selection mechanism in terms of reduction in candidate Web services. Table I presents the table showing summary of evaluation (strength and weakness) of various QoS aware Web service selection techniques.

## II. RELATED WORKS

A study for classification scheme for UDDI 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 [7] 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[8] 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.

## III. PROPOSED SYSTEM

The Analytic Hierarchy Process (AHP) is a structured technique for dealing with complex decisions. Rather than prescribing a "correct" decision, the AHP helps the decision makers find the one that best suits their needs and their understanding of the problem. The AHP provides a comprehensive and rational framework for structuring a decision problem, for representing and quantifying its elements, for relating those elements to overall goals, and for evaluating alternative solutions. It is used around the world in a wide variety of decision situations, in fields such as government, business, industry, healthcare, and education.

AHP first decompose decision problem into a hierarchy of more easily comprehended sub-problems, each of which can be analyzed independently. The elements of the hierarchy can relate to any aspect of the decision problem—tangible or intangible, carefully measured or roughly estimated, well- or poorly-understood—anything at all that applies to the decision at hand.

The structure of hierarchy can be drawn as the following

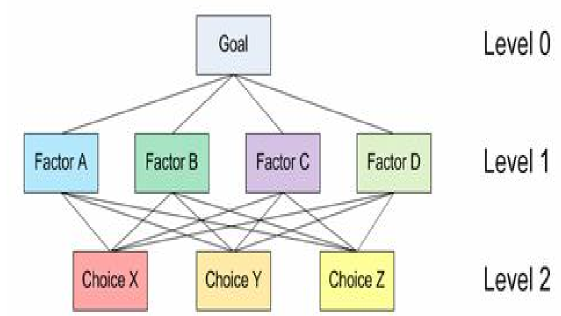


Figure2: Structure of Hierarchy

Level 0 is the goal of the analysis. Level 1 is multi criteria that consist of several factors. The last level (level 2 in figure above) is the alternative choices. The lines between levels indicate relationship between factors, choices and goal. In level 1, one comparison matrix corresponds to pair-wise comparisons between 4 factors with respect to the goal. Thus, the comparison matrix of level 1 has size of 4 by 4. Because each choice is connected to each factor, 3 choices and 4 factors are there, and then in general you will have 4 comparison matrices at level 2. Each of these matrices has size 3 by 3. However, in this particular example, some weight of level 2 matrices are too small to contribute to overall decision, thus we can ignore them.

The proposed architecture is shown in the figure 1. In the figure, the provider first deploys the service and registers with the registry. The QoS parameters of the service are stored in the QoS DB. The services are then ranked and selected based on the analytical hierarchy process (AHP). Here 4 criteria are used for the AHP, Performance, latency, availability and cost of the service. These criteria are at the level 1 of the process and at the next level the services with respect to each of these criteria are calculated. From this, the services are ranked.

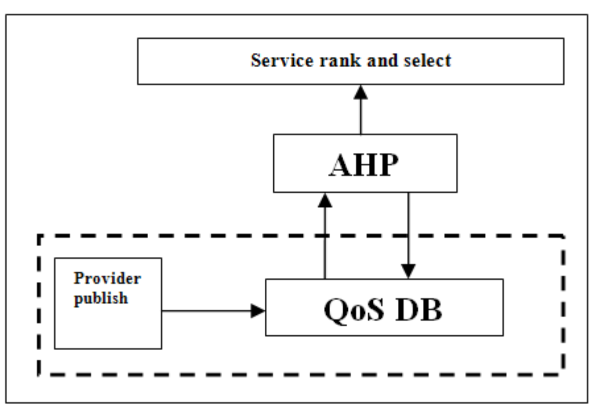


Figure 3: Design of proposed work

Table 1: QoS Metrics

| *A. QoS Parameter* | *B. Descriptio n* | *C. Notation* |
|---|---|---|
| *D. Response time* | *E. Elapse time between a request and response* | *F.* $T = T_1 - T_0$ |
| *G. Throughput* | *H. Maximum no of requests are successfull y executed by a given time* | *I.*<br>*J. (X)=* $\frac{Max.complete\ request}{Unit\ Time}$ |
| *K. Availability* | *L. Probabilit y that a request Successfull y access the service* | *M. A=* $\frac{Ts}{Ta}$ |
| *N. Reliability* | *O. Probabilit y that a request successfull y invoked the service* | *P. R=* $\frac{Ns}{Nn}$ |
| *Q. Cost* | *R. Economic condition of using a service* | *S. Monetary value of the service* |

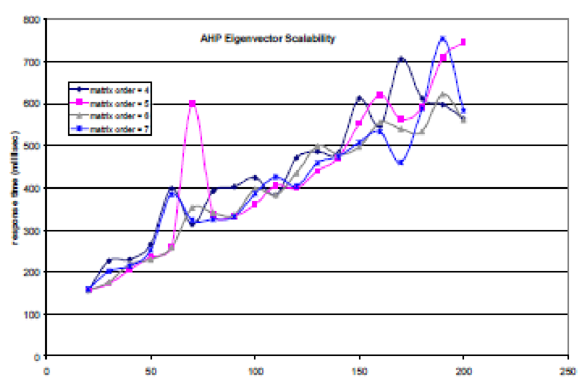


Figure 4: AHP EigenVector Scalability Test Result

In order to evaluate the scalability of the AHP solution, we have designed an experiment, where the response time is measured given the increased number of reviewers during the AHP comparison phases. It examines the sensitivity of the AHP eigenvector response time when the system load (i.e. the number of the users – reviewers) increases in at least one order of magnitude. Thus, four test runs have been conducted based on four different matrix orders (4 – 7). This yields four curves in Figure 5. Within each test run, a certain number (20 – 200) of reviewers concurrently submit their comparison matrices to the AHP engine. The engine calculates the derived principal eigenvector and its associated Consistence Rate (CR). The reviewer aggregation processes is omitted so that we can focus solely on the eigenvector test result. The eigenvector and the CR are sent back to reviewers via HTTP response afterwards. When all reviewers receive the result, the Test Run manager then collects time duration of each reviewer and calculates the average response time (in milliseconds) shown as the Y-axis in Figure 5. The results suggest that when the system loads increase, the response time does increase but at a rather lower rate (roughly 1/3 of the rate of loads increase). For example, when 200 reviewers simultaneously submit their comparison matrices of order of 7, the average response time for each reviewer is about 580 milliseconds (around more than half second). When 20 reviewers submit at the same time, the result is 160 milliseconds. Hence the difference is 420 milliseconds (less than half second), which is hardly noticed by most users. However, the number of users has increased 10 times.

## IV. CONCLUSIONS

This paper focuses on the problem that how to select the optimal service among many Web services which all meet the functional needs, establishes an index system for Web services products selection. Based on this, using AHP method, through statistical analysis of the data collected from questionnaire survey, we get the weights of comparatively importance of the indices in different levels, and then propose Web services selection model. This model can be a reference to Web services managers when they selecting products, and also contributes to in-depth research on the adoption of Web services based information system.

## REFERENCES

[1] Youngkon Lee, “Web Services Registry implementation for processing quality of Service,”ICALPWIT,2008.
[2] C. M. Mackenzie and et al, “ Reference Model for Service Oriented Architecture 1.0,” OASIS Committee draft, 2006.8
[3] D. Min and Eunju Kim, “ A Study for Web service quality Model and Test guideline,” NCA IVRER- 04052, 2004.12
[4] UDDI TC, “Universal Description, Discovery and Integration v3.0.2 (UDDI)” http://www.oasisopen. org/committees/uddi-spec/doc/spec/v3/uddiv3.0.2- 20041019.htm, OASIS Standard, 2005.2
[5] Youngkon Lee and Eunju Kim, “A Study for Web Service Quality Description Language (WSQDL)” NCA IV-RER-04052, 2006.11
[6] KDCIA, “ Development for Korea UDDI for Web service registration and Management,” NCA IVRER- 04058, 2004.11
[7] W. Kim, “A Study for National UDDI for UDDI Interoperability,” NCA, 2005.
[8] kCALS, “ Guideline for e-Catalog Standardization,” ECIF, 2003.10
[9] Jeongkuk Choi, “An implementation report for Web service Quality Management System,” NIA,2007.