ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Web Service Selection: A Model Driven Approach

Anshu Parashar
Associate Professor Department of Computer Science & Engineering Haryana College of Technology & Management
Kaithal, India
parashar_anshul@yahoo.com

Jyoti Jakhar*
M.Tech. Student Department of Computer Science & Engineering Haryana College of Technology & Management
Kaithal, India
jyotijakhar023@gmail.com

Abstract: Web service is a software component invoked over the web by XML and SOAP protocol used for exchange information between ends. Web Services are based on distributed technology and provide standard means of interoperating between different software applications across and within organizational boundaries with the use of XML. The purpose of web service selection is to select optimal web service for a particular task.When we work with web for a single problem, we can get the number of solutions in the form of web service. These web services are hosted by different web servers. To get an efficient and secure web service user has to track all web servers, but it is not feasible. We are providing a mediator between the hosted web servers and users. The purpose of this work is to develop an approach for dynamic, transparent and user comfortable web service selection.

Keywords: Web service selection, selection models, selection mechanism

I. INTRODUCTION

Web services are loosely coupled software components, published, located, and invoked across the web. The growing number of web services available within an organization and on the Web raises a new and challenging search problem: locating desired web services. Web Services (WSs) technology is emerging as a powerful vehicle for organizations that need to integrate their applications within and across organizational boundaries.[1] Web Services is an integrated solution for realizing the vision of the next generation of the web.[2] As the use of Web services is growing, there is an increasing demand for fault tolerance. It is a well-known fact that fault tolerance can be achieved via spatial or temporal redundancy, including replication of hardware, software and time.[3] Web Services are based on distributed technology and provide standard means of interoperating between different software applications across and within organizational boundaries with the use of XML. The basic Web Service platform is combination of HTTP and XML. The HTTP protocol is the most used Internet protocol. XML provides a language which can be used between different platforms and programming languages.[4] Each SOAP message is presented as an envelope with two sections - a header and a body. The header contains information about the message itself, and the body consists of data that has to be transferred to the recipient. These standards make the Web services independent of any programming language, hardware and software platform. Web services use XML to code and to decode data, and SOAP. Web Services are web applications whose interfaces are exposed over protocols like HTML and XML. Web Services are described by Web Service Definition Language (WSDL) in XML format. WSDL is a major language that provides a model and an XML format to describe the syntax about Web services. It acts as a vocabulary, associated with UDDI.[5] In figure 1 we can see the architecture of the web service along with the description at each level. Web service consists of HTTP, XML, SOAP, WSDL and UDDI as shown in figure 1.

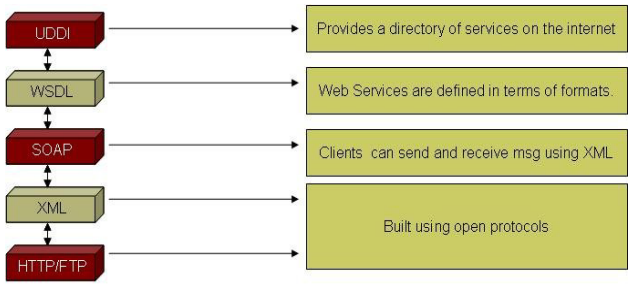

Figure 1. Web Service Architecture

Web service selection is done in various steps given below:

A. The user's intentional goal
B. The discovery of available services
C. The service composition
D. The service selection [6]

II. RELATED WORK

Researchers have proposed various models for dynamic web service selection. Maximilien and Singh propose a multi-agent based architecture to select the best service according to the consumers' preferences.[7] Maximilien and Singh describe a system in which proxy agents gather information on services, and also interact with other proxy agents to maximize their information. The proxy agents lie between the service consumer and the service providers. The agents contact a service broker, which contains information about all known services, as well as ratings about its observed QoS. From there, the information is combined with its own historical usage, and the combined knowledge is used to select a service, though the authors do not detail how.[7] The agencies contain data about the interactions between the clients and the services which is used during the Web Services selection process. The model is shown in

figure 2. In this figure we can note that proxy agent is connected with agencies which helps in the trust based web service selection.

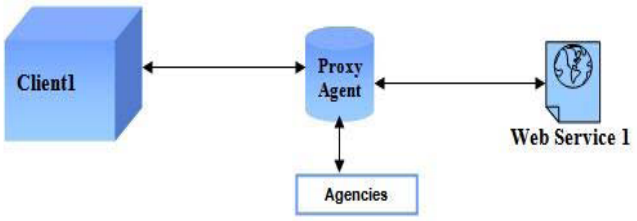

Figure 2 Model proposed by Maximilien and Singh

Y. Liu, S. Ngu, and L. Zeng consider these features in their proposed approach as well but their major selection criteria is based on the QoS based service selection.[8] They have considered three quality criteria namely execution time, execution duration and reputation for the selection. In addition, execution price, duration, transactions support, compensation and penalty rate are the other criteria. The authors of suggest an open and fair framework that evaluates the QoS of the available Web Services by using clients' feedback and monitoring. The model is shown in the figure 3 where reasoning mechanism is attached with the client means selection is done at client side.

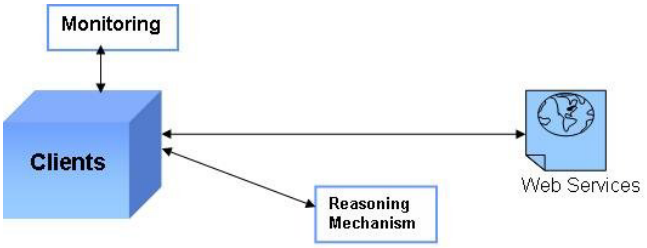

Figure 3 Model proposed by Liu, Ngu, and Zeng

The authors of [8] suggest an open, fair, and dynamic framework that evaluates the QoS of the available Web Services by using clients' feedback and monitoring. The reasoning mechanism is responsible for the selection of a Web Service at a particular moment of time. The reasoning mechanism in the approach proposed by Liu, Ngu, and Zeng[8] computes the QoS of the Web Services, ranks them, and selects the most appropriate one. To perform the selection, the QoS registry in their system takes in data collected from the clients, stores it in a matrix of web service data in which each row represents a web service and each column a QoS parameter, and then performs a number of computations on the data. The bottleneck of the approach is the dependency on the consumers to give regular feedback about their past experience with the Web Services.

P. Chan, M. Lyu, and M. Malek develop a **reliability model** of the proposed Web service paradigm using Petri-Net and Markov chains. The model is shown in Figure . The reliability model is analyzed and verified through using the tool SHARPE tool Petri-Net is built for evaluating the performance of the system and the Markov chains model is developed for analyzing the system availability. In Figure 4 we model a system which is composed of two Web services.

When the messages arrive at the first Web service, a message queue is formed and the Web service will handle the request in the message queue. In the model, the messages are queued up in the *queue* state. Next, the token will be passed to the service and arrive at *finished job* state.

Then, the token is passed to the *quene2* which is for the second Web service. The availability of the service is directly affected by the availability of the Web server. When the server is down, the two Web services will not be available. In our system, when the server fails, the backup server will be invoked.[9]

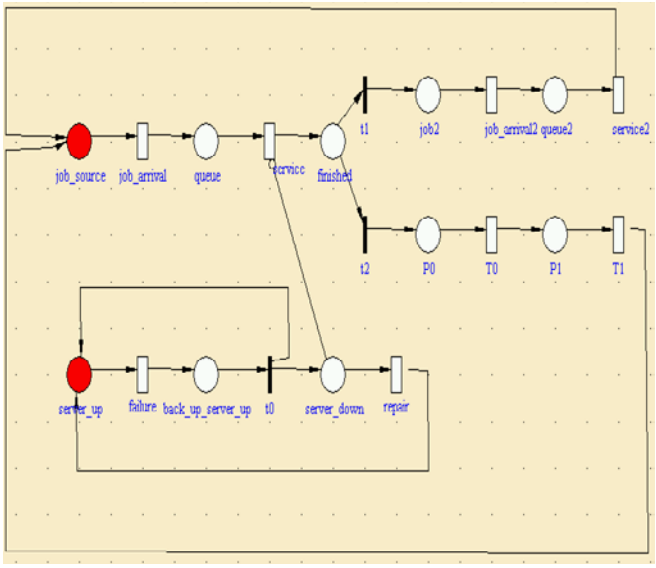

Figure 4 Model proposed by P. Chan, M. Lyu, and M. Malek

Raman Kazhamiakin, Paritosh Pandya, Marco Pistore proposed a **time model**. The behavior of the Web service is described by sequences of activities. The semantic of these activities and the execution time depend on their type. The assignment of variables may be considered as an instantaneous activity, while the service invocation operation may require an arbitrary amount of time. In order to model such behavior, we propose the Web Service Timed Transition System (WSTTS) model, which adopts the formalism of *timed automata* for capturing the aspects specific to the Web service domain. In this formalism, the fact that the operation takes certain amount of time is represented by time increment in the state, followed by the immediate execution of the operation. In order to guarantee that the transition will take place at the right moment of time, the states and transitions of timed automata are annotated with the invariants and guards of the special clock variables.[10]

III. PROBLEM DEFINITION

A. Web Service Selection:

The purpose of web service (WS) selection is to select optimal web service for a particular task. When dynamic discovery is used in Web Services, it is common that the result of the discovery contains more than one provider. Even for a composite Web Service consisting of many atomic Web Services, the selection issue still needs to be addressed when there are multiple providers available for an atomic service. In order to make a distinction between the services which provide the same functionality, selection criteria should be used. They help evaluate the Web Services within a group and choose the component that matches the needs and the preferences of the consumers, while taking into account the abilities of the providers. Web Services can be ranked by the Quality of Service (QoS) they offer. QoS is a means to enable selection and filter out unqualified providers. QoS can be seen as an aggregated measure of generic criteria such as availability, reliability, failure rate, trust and reputation, response time etc. The reasoning

mechanism is responsible for the selection of a Web Service at a particular moment of time. In previous work as already discussed either the reasoning mechanism is attached with the client only which makes the system complex for client and it is less efficient and less user friendly or the clients does not participate in the reasoning process and cannot interfere with it.

The purpose of this work is to develop a web Service selection model which is user friendly, involves client participation but hide the complexity from client making the process transparent.

IV. WEB SERVICE SELECTION MODEL

We propose a technique for dynamic selection of Web Services which will also handle the problem of redundant Web Services. In this work, we introduce a model with a Middle Layer, as shown in figure 5 will act as an independent unit possessing a definite functionality.

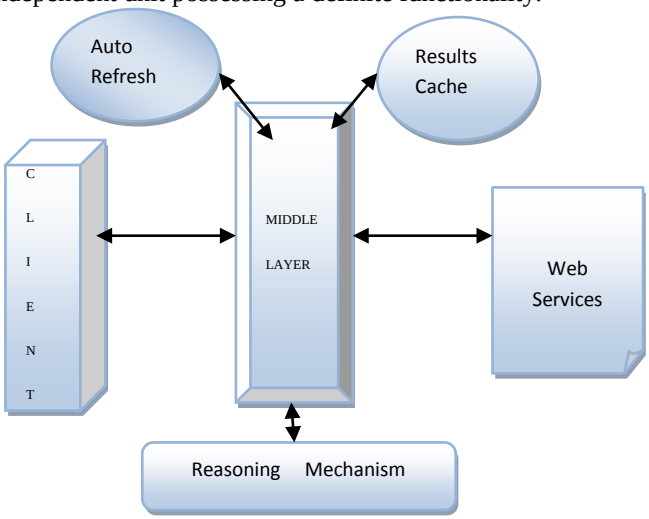

Figure 5 Middle Layer Model

In figure 5 we can see that client and web service are interacting with the middle layer. This hides the system complexity from the clients. Middle layer mainly perform three functions. Firstly it act as independent unit and whole selection process is done at middle layer. Client submits its input and middle layer process it by getting information from web services. This makes the process transparent to client. Three functions attached with middle layer are reasoning mechanism, auto refresh and result cache. As the selection process is done ate middle layer so reasoning mechanism is attached with middle layer. It can be any type vary from user to user according to client's needs. Mainly the reasoning mechanism is QoS based including both static and dynamic QoS parameters but it can also be trust and reputation based or of another type.

Other functions performed by middle layer are auto refresh and result cache. Any of the function can be choose by the client according to its needs. Auto refresh , refreshes the screen or we can say selection results after a particular period of time. This time period is adjusted by user. This function is used mainly in selection of very critical and highly changing web services e.g. stock rates etc. As the stock rates changes within the fraction of minute, this auto update function prevent user to work on expired data or we can say selection results. This increases the accuracy and quality of data.

On the other hand result cache option is choose by the user, efficiency of web service selection is increased. This option is used when the data or selection results are not much critical and highly changing. In this case efficiency is more required than the accuracy. This function is used in daily life web service selection like thought of the day, horoscopes etc. In this option the selection results are saved in cache for a fixed time period. During that period even if the user reloads the selection results, the results available ib cache are provided to user.

So the working of Middle Layer Model mainly depends upon the middle layer it takes query from the client process it with help of reasoning mechanism, make it more efficient by choosing any one option. Find out the results and forward it to the client.

A. Algorithm for Execution:

The algorithm given below gives the simple steps to use the Middle Layer Model for web service selection.

a. Forward the selection query to middle layer.
b. Middle layer searches the web service for query.
c. Reasoning mechanism rank the selected services.
d. One of the options are choose by user.
e. If auto refresh is selected results refreshed after a time period.
f. If results cache is selected results are stored in cache for some time period.
g. Else no change to results.
h. Exit.

V. CONCLUSION

We propose a model based approach for web service selection which has the following advantage:

A. Hides the system's complexity from clients.
B. It provides a transparent service selection from client's point of view.
C. It increases the selection efficiency by including result cache.
D. Increases client's satisfaction by proving the option for auto update.
E. Any type of reasoning mechanism can be applied.

In Middle Layer selection model some variations can be made by improving the reasoning mechanism and giving some more quality options to clients.

VI. REFERENCES

[1]. Xin Dong Alon Halevy Jayant Madhavan Ema Nemes Jun Zhang "Similarity Search for Web Services." Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004

[2]. Vassiliki Diamadopoulou, Christos Makris, Yannis Panagis, Evangelos Sakkopoulos "Techniques to support Web Service selection and consumption with QoS characteristics." Journal of Network and Computer Applications 31 (2008) 108–130

[3]. Pat. P.W. Chan, Michael R. Lyu, and Miroslaw Malek. Making Services Fault Tolerant. ISAS 2006, LNCS 4328, pp. 43–61, 2006. Springer-Verlag Berlin Heidelberg 2006

[4]. Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, Eve Maler, and Fran-cois Yergeau. Extensible markup language (xml) 1.0 (third edition)

[5]. Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web services description language (wsdl) 1.1. http://www.w3.org/TR/wsdl.

[6]. Wolf-Tilo Balke, Matthias Wagner "Towards Personalized Selection of Web services." citeseerx.ist.psu.edu 2003

[7]. E. Maximilien and M. Singh. " Towards Autonomic Web Services, Trust and Selection." ICSOC'04 pages 212–221, November 2004.

[8]. Y. Liu, S. Ngu, and L. Zeng."QoS Computation and Policing in DynamicWeb Service Selection" WWW2004, (8 pages), May 2004

[9]. P. Chan, M. Lyu, and M. Malek. Reliable web services: Methodology, experiment and modeling. In Proc. of IEEE International Conference on Web Services, Salt Lake City, Utah, USA, 9-13 Jul 2007.

[10]. Raman Kazhamiakin, Paritosh Pandya, Marco Pistore "Timed Modelling and Analysis in Web Service Compositions" IEEE Computer Society pp.840-846, First International Conference on Availability, Reliability and Security (ARES'06), 2006.