Dynamic Web Service Selection and its Composition

Dr. Rajesh Kumar
Professor, Dept. of Computer Science
The East West College of Engineering
Magdhi Road, Bangalore ,India
Email: kumraj76@gmail.com

***Abstract*— In the past a few years, the Web has undergone a tremendous change towards a highly user-centric environment. Millions of users can participate and collaborate for their own interests and benefits. Service Oriented Computing and web services have created great potential opportunities for the users to build their own applications. Then, it is a pressing issue that, the users can compose services without too complex tasks and efforts. In this paper, we introduce a user-oriented approach which aims to simplify service composition. The purpose of this work is to develop an approach for dynamic and transparent service selection and to evaluate the proposed architecture in terms of what selection techniques should be applied.**

***Keywords*— Web Service, Composition, Selection WSDL**

1. INTRODUCTION

In recent years, a growing number of Web Services have emerged as the Internet develops at a fast rate. Individual Web services usually cannot satisfy some customers' requirements. So we always need to integrate existing services to create new value-added composed services. A composite Web service is a series of tasks that are tied together for accomplishing some specific works. Services able to provide the required functionalities are then associated with the individual tasks of the composite services and invoked during each execution of the composite services.

Web Services can encapsulate a specific task or can be designed as a composition of other services, representing a complex aggregation. Service providers describe their Web Services and advertise them in a universal registry called UDDI [5]. This enables service requestors to search the registry and find services. UDDI allows for the creation of registries that are accessible over the web. Once the list of services, offering the same functionality, is available and the criteria are specified, a set of steps (instructions, rules) for Web Service selection should be followed during the decision making process, in order to determine the component to handle the client's request. Various sets of instructions represent different selection techniques.

As Service Oriented Architecture[1] and Web Services promise some open standards (e.g., WSDL and SOAP) to leverage software components to services, users can then flexibly create their own applications by means of service composition. With the proliferation of services on the internet, there obviously exists a huge potential space of user-oriented service composition and delivery. Therefore, it is a pressing issue that the users can compose services in a fast, flexible and simple manner.

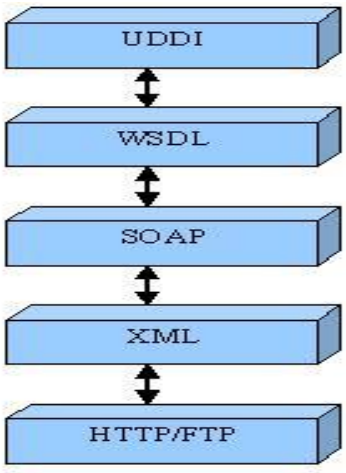

Fig1.Web Services Structure

On the other hand, researchers have made a lot of contributions, such as ontology engineering, artificial

intelligence, to facilitate truly dynamic service discovery and orchestration [8]. However, to best of our knowledge, little has been done to enable the users to directly participate and even dominate the service composition themselves. Our work aims to provide a user-oriented approach to facilitating the users to participate in the service composition in an easy-of user manner. From the user's perspective, the following key issues need to be considered:

There have been thousands of web services available over the internet. It is then difficult and time-consuming for users to find the suitable services. Especially, in the composition process, users would like to discover all composition possibilities instead of discovering elementary services iteratively. Then it is required an automatic way to analyze the potential space of composition possibilities, as well as suggest the users the suitable services. The users would like to have just-in-time effects of service composition. They can then make decision whether current composition meets their requirements. In this way, it allows the users to participate to build their desired composite services more flexibly. The remaining of the paper is organized as follows: Section 2 will introduce to the motivation behind the web service selection and related key technical issues. Section 3 will introduce to the proposed approach where we will introduce to the user oriented web service selection for composition Section 4 discusses about the conclusion.

2. WEB SERVICE SELECTION

Web Service Selection is a crucial step for the composition of the services. Researchers have proposed various approaches for dynamic web service selection.

2.1 Multi-Agent based Architecture

Maximilien and Singh [6] proposed a multi-agent based architecture to select the best service according to the consumers' preferences. 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 and the conceptual model they use to interact with the services is detailed elsewhere [7]. 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. The agencies contain data about the interactions between the clients and the services which is used during the Web Services selection process.

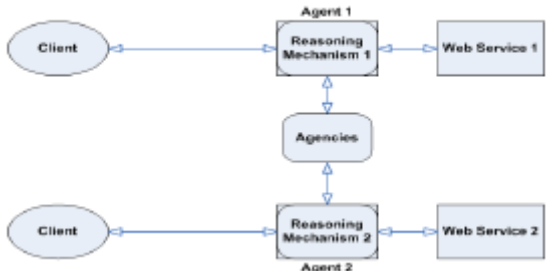

Fig2.1. Model proposed by Maximilien and Singh [6]

In his work, trust and reputation are taken into account during the decision process. Their approach divides the QoS attributes into objective and subjective.

2.2 QoS based service selection

Liu, Ngu, and Zeng [8] considered the former features in their proposed approach as well but their major selection criteria is based on the QoS based service selection. Their approach is shown in figure 2.2. 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 [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. In order to distinguish one service from another using the specified criteria, this unit requires a set of instructions that help evaluate each component and choose the most appropriate one respectively. A set of instructions can be seen as a selection technique. The major components of a reasoning mechanism are criteria, model, and selection technique. The model collects information about the participants of the client-server interaction as well as represents it as aggregated measures. Different selection techniques can implement various business logics in order to make a decision.

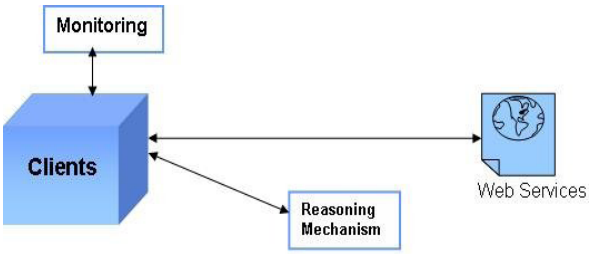

Fig2.2. Model proposed by Liu, Ngu, and Zeng [8]

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, such as normalization. Clients can then access the registry, and are given a service based on the parameters that the client prefers. The bottleneck of the approach is the dependency on the consumers to give regular feedback about their past experience with the Web Services. An overview of their approach is shown in figure 2.2 .The Success of this model is based on the clients or the end users and their will to provide the necessary feedback on QoS.

2.3 Repository based Web Service Selection

Abhishek Pandey, S.K.Jena[4], proposed a Repository based Web Service Selection. Their approach is shown in figure 2.3. In the proposed approach Web Service repository, will act as an independent unit possessing a definite functionality. This repository will be used to redirect the client's request. This will also provide a level of security since it will not be allowed to invoke directly by the clients. This technique will prevent unauthorized access to the real services. This provision will also help to hide the systems complexity from the clients.

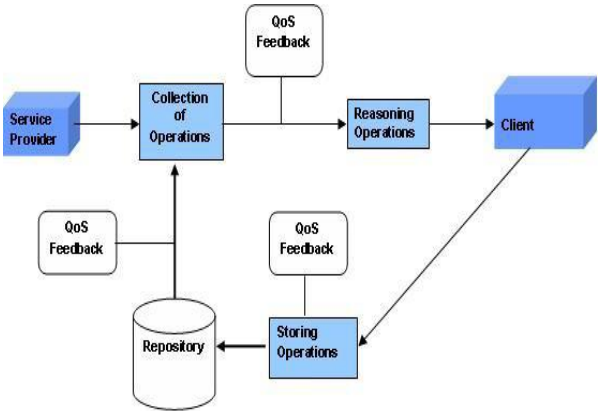

Fig2.3. Repository based Web Service Selection [4]

The collecting, storing and reasoning mechanism interacts with the web services to find the most appropriate of the services and the results are stored in the repository for future reference.

The summary of the reviewed Literature is when dealing with dynamic components like Web Services; it is hard to observe all of their possible features. The researchers focus mostly on some generic criteria, since they can be applied to any service. Availability, reliability, and response time are the most popular ones, as they provide an overview of the services and at the same time they can be evaluated relatively easily.

3. PROPOSED APPROACH

3.1Problem Definition

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 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 [2], failure rate, trust and reputation, response time, price, and network load and domain specific features [8].The reasoning mechanism is responsible for the selection of a Web Service at a particular moment of time. The previous works as already discussed in section 2, reflects that the clients are not involved in the selection process. The purpose of this work is to develop an approach for dynamic and transparent service selection and to evaluate the proposed architecture in terms of what selection techniques should be applied. In this proposed model the clients are also involved in the selection process 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.

3.2 Web Services Selection Model

We propose a technique for user-oriented selection of Web Services which will also handle the problem of redundant Web Services. In this work, we introduce a

model with a Web Service process generator, as shown in figure 3.1 will act as an independent unit possessing a definite functionality. As there have been a number of available web services together with their descriptions (WSDL files) over internet, the potential space of compassable services will be huge enough. Besides the literal descriptions residing in WSDL, we also notice the fact that users can annotate services [3] themselves with tags. These tags imply more plentiful semantic information such as usage, feedback and context, and can help us to rank and improve the compensability beyond syntactic information.

The process generator will perform three functions namely, collecting and reasoning and user composition. The collecting operation retrieves all necessary data from providers for the reasoning operation. The reasoning operation manages to select the best services provided for the user composition operation. Besides the literal descriptions residing in WSDL, we also provide annotated services with tags. These tags imply more plentiful semantic information such as usage, feedback and context, and can help us to rank and improve the composability beyond syntactic information.

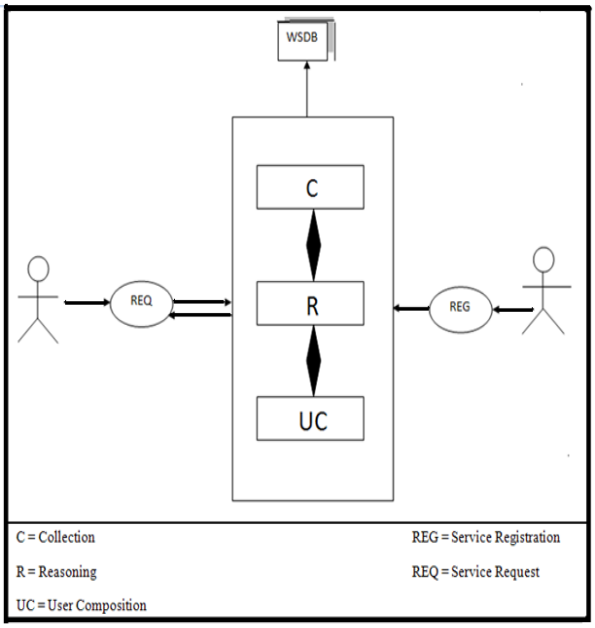

Fig3.1. User-Oriented Web Service Selection and Composition

Based on the annotated services with tags the user selects the web services for composition. The collecting, storing and reasoning mechanism interacts with the web services to find the most appropriate of the services and the results are stored in the repository for future reference. Web Services here interacts with the reasoning mechanism to find out the appropriate services. Once the service is selected, the request is forwarded to it. Finally, when the result is generated, it is passed to the repository which sends it back to the client.

Algorithm for Web Services Selection: This algorithm shows the necessary steps to choose a service and get the maximum quality results. For finding a service for a specified task, perform a search on service descriptions. Arrange all discovered services by their signature parameter (service tags) and discard all other services. Get the desired Service Parameters. Collect the services result and order by their utility. If no results are found, let the client reconsider the constraints.

4. CONCLUSION

Our proposed approach aims to facilitate the users to compose services in an easy-of-use manner. To alleviate the users from the time-consuming service discovery, we leverage the available WSDL files and user-annotated tags to retrieve the potential space of composable services. It then assists and guides users to compose services with detection of correctness and compatibility. A user friendly model enables the users to interactively compose services. It also supports the just-in-time effect incarnation, so that users can evaluate the composition result.

Approach has the following advantages in comparison with previous approaches:

- It hides the system's complexity from the clients.
- It provides a transparent service selection from the client's point of view.

REFERENCES

[1] Mustafa, Faisal and McCluskey, T.L., "*Dynamic web services composition: current issues"*, 2009 International Conference on Computer Engineering and Technology.

[2] Daniel A.Menasce. *"QoS Issues in Web Services.IEEE Internet: Computing,*" 6(6),2002.

[3] Xuanzhe Liu, Gang Huang, Hong Mei, "*A User-Oriented Approach to Automated Service Composition*," 2008 IEEE International Conference on Web Services.

[4] Abhishek Pandey, S.K.Jena, "*Dynamic Approach for Web Services Selection*," IMECS 2009, March 18 - 20, 2009, Hong Kong.

[5] UDDI Technical White Paper, UDDI Browser, www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, September 6, 2000.
[6] E. Maximilien and M. Singh. *Towards Autonomic Web Services, Trust and Selection*. ICSOC'04 pages 212–221, November 2004.
[7] Maximilien, E.M. and Singh, M.P. "*Reputation and endorsement for web services*". SIGecom Exch. Volume 3.1, pages 24-31. 2001.
[8] Y. Liu, S. Ngu, and L. Zeng."*QoS Computation and Policing in DynamicWeb Service Selection"* WWW2004, (8 pages), May 2004