International Journal of Contemporary Technology and Management
Volume VI, Issue IX, September - 2017 ISSN (Online) 2278-8034

Web Service Discovery and Composition Based System for QoS Agent: A Survey Paper

Himshikha jamadari[#1] , Sachin Mahajan[#2]
Department of Computer Science and Engineering,
Jawahar Institute of Technology, Borawan
himshikhajamdari020@gmail.com[#1]
sachinmahajan248@gmail.com[#2]

Abstract - **Web services are the internet enabled applications for performing business needs, considered as the platform-independent and loosely coupled. Web service compositions build new services by organizing a set of existing services by providing reusability and interoperability. In today's Web, Web services are created and updated on the fly. It's previously beyond the human skill to analysis them and generates the composition plan manually. A Web service is a self-describing software component which is universally accessed by means of standard protocols. A web service registry UDDI provides interoperable, standards based approach for methodically documenting and publishing web services. In this paper, we survey on the web services of dynamic discovery and composition. A web service uses number of agent depends what they are specific for web discovery and composition. This paper provides an in-depth analysis on the existing approaches available for the discovery and composition of semantic web services.**

Keywords: Web Service Discovery, Composition, UDDI, Agent, HTTP, SOAP, WSDL

I. INTRODUCTION

Web services can be published and can be accessed with internet and business intranets for developing scattered applications. It can be defined as the model for software system [1] and designed to maintain interoperable communication. The capacity to select and compose the inter-organizational and mixed services at runtime on the web is the important issue to be considered in web service applications. With the help of a single web service, we cannot satisfy both the functional and non-functional requirements of the user, so we need to mix a set of composite already existing web services to satisfy the user needs [2].

A. Web Services

Web services are client and server applications that communicate over the World Wide Web's (WWW) HyperText Transfer Protocol (HTTP). The use of Web Services on the World Wide Web is expanding rapidly as the need for application-to-application communication and interoperability grows. These Web services provide a standard means of communication among different software applications, running on a variety of platforms and/or frameworks. The architecture presented in this document is intended to promote interoperability and extensibility among these various applications, platforms and frameworks in a manner that remains consistent with the architecture of the Web [3]. A core definition of web services is, "A web service is any piece of software that makes it available over the internet and uses a standardized XML messaging system. XML is used to encode all communications to a web service. For example, a client invokes a web service by sending an XML message, and then waits for a corresponding XML response. As all communication is in XML, web services are not tied to any one operating system or programming language--Java can talk with Perl; Windows applications can talk with UNIX applications [4]." The basic web services platform is XML + HTTP. All the standard web services work using the following components:

- ✓ SOAP (Simple Object Access Protocol)
- ✓ UDDI (Universal Description, Discovery and Integration)
- ✓ WSDL (Web Services Description Language)

The basic Web services architecture defines an interaction between software agents as an exchange of messages between service requesters and service providers. Requesters are software agents that request the execution of a service. Providers are software agents that provide a service. Agents can be both service requesters and providers. Providers are responsible for publishing a description of the service(s) they provide. Requesters must be able to find the description(s) of the services. The basic Web service architecture models the interactions between three roles: the service provider, service discovery agency, and service requestor. The interactions involve publish, find, and bind operations. These roles and operations act upon the web service artifacts: the web service software module and its description [5].

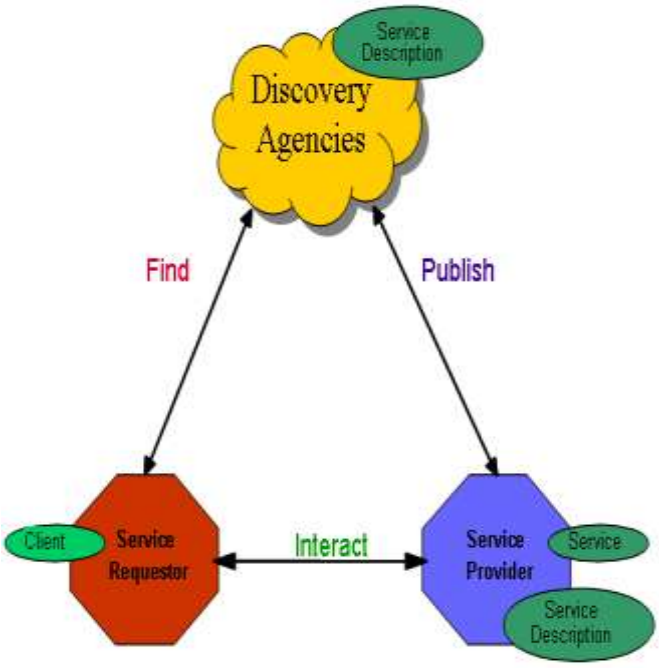

Figure 1: Web Service Architecture [6]

The figure above illustrates the basic Web services architecture, in which a service requestor and service provider interact, based on the service's description information published by the provider and discovered by the requester through some form of discovery agency. Service requesters and providers interact by exchanging messages, which may be aggregated to form MEPs [6].

B. Web Service Discovery

In general, the purpose of service discovery is threefold: finding a service that can possibly satisfy user requirements, choosing between several services, and composing services to form a single service. It is well known that all of these tasks are carried out with the help of service descriptions well before invoking them. These service descriptions may change over time, may be distributed over locations and may be resided in heterogeneous environments. Thus, service discovery architecture should at least support publication, finding and matching of up-to date Web service descriptions [7]. Web Service discovery is the process of finding an appropriate service provider for a service requestor through a service matchmaker. Natural language description of desired Web Services and WSDL specifications of all available services published through UDDI are given to discover a service. Here words should be extracted from WSDL, which are pre-processed and assigned the weight. According to the weights, the similarity between the given description and a description of web-service operation can be measured. The weighted words are used to build a matrix containing the information on all Web Services which have common words in their description [8].

C. Web Service Composition

Web service composition originated from the necessity to achieve a predetermined goal that cannot be realized by a standalone service. Internally, in a composition, services can interact with each other to exchange parameters, for example a service's result could be another service's input parameter. Web service composition is an important step towards development of web service applications. If no single web service can satisfy the functionality required by users, there should be possibility to combine existing services together in order to fulfill the request. Several initiatives such as usage of UDDI (Universal Description, Discovery and Integration), WSDL (Web Services Description Languages), SOAP (Simple Object Access Protocol) have been conducted that will allow easy integration of heterogeneous system. Despite all these efforts, web service composition still a complex task and it is already beyond the human capability to deal with whole process manually. Moreover, in current scenario, there might be very few service providers for cloud computing and there might be a very few services provided by them. But if the service providers and the services will increase in future, it would be difficult to find the optimized services on own. Hence several methods and frameworks have been proposed to carry out the process of service composition that includes Agents and Hadoop in cloud computing [9, 10].

D. Web Service Benefits

Web services provide several technological and business benefits, a few of which include [11, 12]:

- ✓ Application and data integration
- ✓ Versatility
- ✓ Code re-use
- ✓ Cost savings

The inherent interoperability that comes with using vendor, platform, and language independent XML technologies and the ubiquitous HTTP as a transport mean that any application can communicate with any other application using Web services. The client only requires the WSDL definition to effectively exchange data with the service – and neither part needs to know how the other is implemented or in what format its underlying data is stored. These benefits allow organizations to integrate disparate applications and data formats with relative ease.

Web services are also versatile by design. They can be accessed by humans via a Web-based client interface, or they can be accessed by other applications and other Web services. A client can even combine data from multiple Web services to, for instance, present a user with an application to update

sales, shipping, and ERP systems from one unified interface – even if the systems themselves are incompatible. Because the systems exchange information via Web services, a change to the sales database, for example, will not affect the service itself.

Code re-use is another positive side-effect of Web services' interoperability and flexibility. One service might be utilized by several clients, all of which employ the operations provided to fulfill different business objectives. Instead of having to create a custom service for each unique requirement, portions of a service are simply re-used as necessary

All these benefits add up to significant cost savings. Easy interoperability means the need to create highly customized applications for integrating data, which can be expensive, is removed. Existing investments in systems development and infrastructure can be utilized easily and combined to add additional value. Since Web services are based on open standards their cost is low and the associated learning curve is smaller than that of many proprietary solutions. Finally, Web services take advantage of ubiquitous protocols and the Web infrastructure that already exists in every organization, so they require little if any additional technology investment [13].

E. Agent Based System

The association of Agent and the Cloud is beneficial for both the parties: Cloud User and Cloud Providers. A simple agent program can be defined mathematically as an agent function which maps every possible percept sequence to a possible action the agent can perform or to a coefficient, function or constant that affects eventual actions.

The program agent, maps every possible percept to an action. Percept refers to the agent's perception inputs at any given instant [14]. A multi-agent system consists of number of agents, which interact with one another. Adopting multi-agent system in applications will bring the following advantages:

- ✓ The operation will be faster due to the parallel processing
- ✓ The demand for communication band width is lower because the information processing is carried out near the information source
- ✓ Errors of one Agent will not affect the whole system, so the system has higher reliability and
- ✓ System has higher response speed.

1. Popular Features of Agents

Agents typically include a set of features. The main features of agents contain the following [15]:

- ✓ **Autonomy:** The capacity to act autonomously to some degree on behalf of users or other programs also by modifying the way in which they achieve their objectives.
- ✓ **Pro-activity:** The capacity to pursue their own individual set goals, including by making decisions as result of internal decisions.
- ✓ **Re-activity:** The capacity to react to external events and stimuli and consequently adapt their behaviour and make decisions to carry out their tasks.
- ✓ **Communication and Cooperation:** The capacity to interact and communicate with other agents (in multiple agent systems), to exchange information, receive instructions and give responses and cooperate to fulfill their own goals.
- ✓ **Negotiation:** The capability to carry out organized conversations to achieve a degree of cooperation with other agents.
- ✓ **Learning:** Agents improve performance and decision making over time when interacting with the external environment

The rest of this paper is organized as follows: Section 2 Literature Survey of recent work, Section 3 includes the problem domain of agent based system. Finally, the paper is concluded in Section 4.

II. LITERATURE SURVEY

Several Web services may share similar functionalities, but possess different non-functional properties. Numerous researches have been done on web service discovery and composition by using various approaches. In our survey, we have considered the survey of those papers which are related to web service discovery and composition basically based on QoS of agent based broker.

M. Adel Serhani et al. [16] propose a two-phase verification technique that is performed by a third party broker. The first phase consists of syntactic and semantic verification of the service interface description including the QoS parameters description. The second phase consists of applying a measurement technique to compute the QoS metrics stated in the service interface and compares their values with the claimed one. This is used to verify the conformity of a web service from the QoS point of view (QoS testing). A methodological approach to generate QoS test cases, as input to QoS verification is used. Authors have implemented a prototype that includes the verification and certification components of the broker. Authors performed experiments to evaluate the importance of verification and certification features in the selection process using real web services.

Manoranjan Parhi et al. [17] deals with a hybrid multi-agent based web service discovery mechanism which involves an artificial intelligence approach to efficiently interpret the user requirements based on both functional and non-functional demands and fuzzy constraints. The proposed model utilizes

the services of intelligent software agents. Some of the agents like the reputation agent analyses the popularity of web services and assign ranks to the web services based on user feedback and statistical information. The behavior of individual user is being tracked, from which the intelligent agent interprets the fuzzy requirements of users through set of logical and analytical calculations whereas the composition agent provides flexibility to the user for custom composition of web service packages using different individual web services. In brief, we present an automated customer-centric web service discovery and composition approach which aims in an efficient web service discovery and composition followed by customer satisfaction using multi agents.

The aim of ***Rohallah Benaboud et al. [18]*** is the development of an agent based approach for Web services discovery and selection in witch, OWL-S is used to describe Web services, QoS and service customer request. We develop an efficient semantic service matching which takes into account concepts properties to match concepts in Web service and service customer request descriptions. Our approach is based on an architecture composed of four layers: Web service and Request description layer, Functional match layer, QoS computing layer and Reputation computing layer.

Due to the high proliferation of web services, selecting the best services from functional equivalent service providers have become a real challenge, where the quality of the services plays a crucial role. But quality is uncertain, therefore, several researchers have applied Fuzzy logic to address the imprecision of the quality of service (QoS) constraints. Furthermore, the service market is highly dynamic and competitive, where web services are constantly entering and exiting this market, and they are continually improving themselves due to the competition. Current fuzzy-based techniques are expert and/or consensus based, and therefore too fragile, expensive, non-scalable and non self-adaptive. In this paper ***Romina Torres et al. [19]*** introduce a new methodology to support requesters in selecting Web services by automatically connecting imprecisely defined QoS constraints with overly precise service QoS offerings over the time. Authors address the dynamism of the market by using each time a modified fuzzy c-means module that allows providers to automatically organize themselves around the QoS levels. The advantage of our approach is that consumers can specify their QoS constraints without really knowing what the current best quality ranges are. Authors illustrate our approach with a case of study

Web Service has been playing a significant role in application development and integration. It facilitates pre-built applications/application components enclosed as Web Services to interact with each other via consistent interfaces and form larger application systems. ***Manoranjan Parhi et al. [20]*** deals with a proper technique to Web Services identification, which is an important step in designing and developing effective Web Services. Discovery of a suitable Web Service for a particular task has become a challenging issue due to the increasing number of Web Services offering similar functionalities. The functional properties describe what the service can do and the nonfunctional properties depict how the service can do it. There is a need to select the most appropriate Web Service based on the customer's QoS requirements and preferences. In this paper we propose the Multi-Agent based architecture for both services registration and service discovery. The proposed architecture utilizes the services of response agent, certification agent and query agent. Thus, a novel approach for designing and developing Multi-Agent System architecture is used for evaluating Web Services. The experiments evaluated the importance of service registration and service discovery process using real Web Services.

III. PROBLEM DOMAIN

A problem domain is the area of expertise or application that needs to be examined to solve a problem. A problem domain is simply looking at only the topics you are interested in, and excluding everything else.

The World Wide Web has evolved from being a pure information repository to a more functional and service oriented platform using technologies such as Web Services. With the tremendous success of the Web service technology in both business and research area, quality of service (QoS) problems play an significant role for Web service providers. The need for QoS in Web services is driven by two demands. Clients aim to experience a good service performance, e.g. low waiting time, high reliability, and availability to successfully use services whenever the need arises. On the other hand, especially in e-business, service providers need to formulate QoS-aware offers in order to gain the highest possible profit from their business. In most of the cases whenever we build the application, web services fails due to their access nature. In this scenario, two conditions are problematic of web service i.e. user dependency and environmental dependency of the system. For this reason performance of the QoS agent based performance is degraded. Therefore we need to choose an optimum solution which gives the robust performance for the web based service for discovery and composition. Need to build new training model on the basis of web service instances.

IV. CONCLUSION

With the integration of Web Services as a business solution in many enterprise applications, the QoS of Web Services is becoming the main concern of both service providers and clients. The integration of the various QoS properties is essential for the success of the web service technology. Due to the increasing popularity of Web services technology and

the potential of dynamic service selection and integration, multiple service providers are now providing similar services. QoS is a decisive factor to distinguish functionally similar Web services. The major problem with the current web service selection is the absence of a mechanism that considers QoS properties for the web service selection. Therefore, in this survey we have presented various approaches of web services discovery and composition, each approach has its own advantages and its limitations.

REFERENCES

[1] Dhavachelvan, P., Uma, G.V., Venkatachalapathy, V.S.K., 2006. A new approach in development of distributed framework for automated software testing using agents, In: International Journal on Knowledge-Based Systems, Elsevier, Vol. 19, No. 4, pp. 235– 247, August 2006

[2] Papazoglou, Michael P., Dubray, Jean-jacques, "A survey of web service technologies", 2004

[3] DRAFT: Architecture Principles of the World-Wide Web; Ian Jacobs, 30 August 2002, available online at: http://www.w3.org/TR/2002/WD-webarch-20020830/

[4] Ran, Shuping, "A model for web services discovery with QoS", ACM Sigecom exchanges 4.1 (2003): 1-10.

[5] "Chapter 18: Introduction to Web Services", The Java EE 6 Tutorial, available online at: http://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html

[6] Michael Champion and Chris Ferris, "Web Services Architecture", W3C Working Draft 14 November 2002,

[7] Brahmananda Sapkota and Dumitru Roman, "Distributed Web Service Discovery Architecture, IEEE International Conference on Internet and Web Applications and Services/Advanced (AICT-ICIW), 2006.

[8] Shirin Akther Khanam and Hee Yong Youn, "A Web Service Discovery Scheme Based on Structural and Semantic Similarity", Journal of Information Science and Engineering 32, pp. 153-176 (2016)

[9] Rajesh, Web Service Composition using Multi-Agent System, International Journal of Research in Engineering & Advanced Technology; 2013 Vol. 1, Issue: 1.

[10] Jinghai Rao, Xiaomeng Su, A survey of Automated Web Service Composition Methods, International Workshop on Semantic Web Services and Web Process Composition; volume 3387.

[11] Cavanaugh, Erin, "Web services: Benefits, challenges, and a unique, visual development solution", white paper, Feb 10 (2006).

[12] Florije Ismaili and Florije Ismaili, "Web Services Research Challenges, Limitations and Opportunities", WSEAS Transactions on Information Science & Applications, Issue 10, Volume 5, October 2008

[13] Fakorede Oluwatoyin Johnson, "An Investigation into the Implementation Issues and Challenges of Service Oriented Architecture", September 2007

[14] J. Octavio Gutierrez- Garcia, Kwang Mong Sim, Agent based Cloud Service Composition, Elsevier Applied Intelligence; Volume 22, No. 2, 2013

[15] Francis Michael R.T and Siva Pragasam. P, "Implementation Of Agent Based Web Service Composition Using Dependency Relation Algorithm", International Journal Of Technology Enhancements and Emerging Engineering Research, Volume 4, Issue 4, 2016.

[16] Serhani, Mohamed Adel, et al. "A QoS broker based architecture for efficient web services selection", Proceedings IEEE International Conference on Web Services (ICWS), IEEE, 2005.

[17] Manoranjan Parhi and Binod Kumar Pattanayak, "A Multi-Agent-Based QoS-Driven Web Service Discovery and Composition Framework", ARPN Journal of Engineering and Applied Sciences, Volume 9, Number 4, April 2014.

[18] Rohallah Benaboud, Ramdane Maamri and Zaidi Sahnoun, "Agents and Owl-S Based Semantic Web Service Discovery With User Preference Support", International Journal of Web & Semantic Technology (IJWesT) Vol.4, No.2, April 2013

[19] Torres, Romina, and Rodrigo Salas, "Self-adaptive fuzzy QoS-driven web service discovery", 2011 IEEE International Conference on Services Computing (SCC), 2011

[20] Rajendran, T., and P. Balasubramanie, "An efficient multi-agent-based architecture for web service registration and discovery with QoS", European Journal of Scientific Research 60.3 (2011): 421-432.