ijecs open access www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume1 Issue 3 Dec 2012 Page No. 156-163

Web service QoS Based Broker Architecture and Heuristic Algorithm applied for an Apartment Management system

Prof.SushmaA Shirke ,Prof.Sharayu Lokhande ,Prof.Rushali Patil

AIT,Pune

Abstract

Web Services are based on distributed technology and provide standard means of interoperating between different software applications across and within organizational boundaries. Web Services technologies allow interaction between applications. Sometimes a single service given alone does not meet user's needs. In this case, it is necessary to compose several services in order to achieve the user's goal. Quality of Service (QoS) support in web services plays a great role for the success of this emerging technology. In this paper, we presentQoS broker- based architecture for web services. The main goal of the architecture is to support the client in selecting web services based on his/her required QoS It also presents an efficient method to finding the suitable web service based on the consumer's requirements along with QoS.

We intend to use this architecture for several services like Loan system,location based search etc.

1.0 Introduction:

A Web service is an API that describes a collection of operations that are accessible through Internet based on standardized XML messaging. Web services can be published, found and used across the Internet using SOAP, WSDL and UDDI standards. Web Services are encapsulating application functionality and information resources. The main emphasis of Web services is to save development time and cost through the reuse of components.

Nowadays, both Web Service providers and clients are concerned with the QoS guaranteed by web services. From the client point of view, web service based QoS discovery is a multi-criteria decision mechanism that requires knowledge about the service and its QoS description. However, most of clients are not experienced enough to acquire the best selection of web service based on its described QoS characteristics. They simply trust the QoS information published by the provider; however most of web services providers do not guarantee and assure the level of QoS offered by their web services. Based on the above we propose a Web Services discovery architecture that contains an extended UDDI toaccommodate the QoSinformation, and WS-QoS Broker to facilitate the Web Service discovery.

Measuring the degrees to which the web services can deliver the functionality through a combination of QoS parameters becomes significant, particularly in distinguishing services competing in the same domain. The QoS parameters can be used to characterize the web services' overall behavior. Service providers QoS claims may not be trustworthy. Hence some method is needed to automate the process of measuring QoS for registered web services. QoS delivered to a client may be affected by many factors, including the performance of the web service itself, the hosting platform and the underlying network. A set of verification procedures is essential for providers to remain competitive and for clients to make the right selection and trust the published QoS metrics. For the success of any QoS based web services architecture, it should support a set of features: 1) QoS Verification and Certification to guide web service selection 2) QoS aware web services publishing and discovery. In this paper, we propose a broker based architecture for web service selection and QoS management. The role of the WS-QoS broker is to support QoS provisioning and assurance in delivering web services.

In this paper, we have defined various Service-offers and proposed a tree structure to represent the CSP's requirements limited in the QoS and Service-offers

Case study 1: Travel Reservation Application –

The travelers normally prefer reservation for his/her distant travel location through travel agent. The main objective of a traveler is to get the best combination of services like, quality, price and valid offers which satisfy his/her needs. On the other hand, travel agent tries to satisfy the customer's needs and mint money by charging extra fees like service charge for each trip. In this Travel Reservation Application, travel agent is a service Intermediator, who has to find the best services for the individual tour package based on the traveler's demands on the service quality and offers. The travel agent uses the Web service system to find and integrate different services that are provided by the different travel service providers. The travel agent service can publish the specific service information into service registry, but QoS of the service is permitted to create a document either in digital or hard copies, in whole or part of a traveler.

To republish, or to post on servers, agent requires prior specific permission or a traveler has to pay specified trip fee, depending on the QoS of constituent services. With respect to the above mentioned example, the customer's Travel Reservation application consists following activities:

(a) Book an Air ticket from Bangalore to Mauritius. (b) Book a single AC Room in Mauritius star Hotel for 02 days. (c) Book a Taxi in Mauritius from airport to Hotel. (d) Book a ticket for Mauritius city tour. (e) Booking for the dinner at Mauritius Beach Hotel.

The customer's QoS and business offer requirements are as follows – 1) The price should be minimum. 2)The most esteemed service offers with good discount. When travel agents get such requirements from a traveller, the agent has to find the service that satisfies all requirements of a traveller. Usually a Travel agent is interested in reliable travel service provider to improve the reliability of a travel composite service **[1] [3].**

The Composite or Complex service provider (CSP) defines the requirements to travel agents on the multiple QoS properties and Service-offers involving AND/OR operators, it is very tedious work to find the best Web services for the individual task of the recapitulation. This paper addresses few issues related to service selection in recapitulation. This paper is organized as; Section 2 specifies literature study of the topic , Section 3 defines QoS Broker Architecture, Section 4 defines the QoS Selection Algorithm and Section 5 depicts the conclusion and future work.

2.0 Related Work :

Web services paradigm is a recent concept of emerging web applications. It connects a set of technologies, protocols, and languages to allow automatic communication between web applications through the Internet. A Web Service is an application that exposes its functionality through an interface description and makes it publicly available for use by other programs. As web services are a new emerging technology, most existing work focuses more on their development and their interfacing practices. QoS support in web services, and in particular QoS management, is still an immature research area. Efforts are still carried for enumerating the requirements and defining the approaches. In addition, standard web services protocols such as WSDL and UDDI were designed mainly for their functional features with only minor consideration for QoS support and verification. Until recently, considerable efforts have been conducted to work on QoS for web services.

Web service technology uses an interface description to expose its functionality and makes it publicly available for use by other programs. Standard web services protocols such as WSDL and UDDI are designed mainly for their functional features. Such protocols do not provide QoS support and verification. Several web services may have similar functionalities but with different QoS property values. When discovering web services, it is necessary to consider both functional and non-functional properties. But the UDDI registry does not include QoS information. To solve this problem, some work has been implemented for enhancing UBR's inquiry operations by embedding QoS information within the message. An example is the UDDIe**[2],** which provides an API that can associate QoS information through a set of user defined properties. The search queries are executed based on these properties.

Tian et al **[4]** explained the WS-QoS architecture that enables QoS-aware service specifications as well as the broker based web service selection model that enables an efficient QoS-aware service selection. Eyhab and Qusay**[5]** introduced a mechanism that extends the Web Services Repository Builder (WSRB) of Web Services. It also introduced the Web Service Relevancy Function (WsRF) used for measuring the relevancy ranking of a particular Web service based on client's preferences and QoS metrics. Xu et al **[6]** presented a web service discovery model that contains an extended UDDI to accommodate the QoS information, a reputation management system to build and maintain service reputations and a discovery agent to facilitate service discovery. A service matching, ranking and selection algorithm is also developed. Demian et al **[7]** explored different types of requester's QoS requirements and a tree model for requester's QoS requirements. It also proposed a QoS broker based web service architecture which facilitates the requester to select a suitable web service based on QoS requirements and preferences. The Web service selection and ranking mechanism uses the QoS broker based architecture **[8].** The QoS broker is responsible for the selection and ranking of functionally similar Web services.

Most of the above works do not consider performance evaluation of web services and scalability issue while the number of clients is continuously increasing and their requirements are always changing. In the next section, we describe the design of the proposed QoS broker-based architecture; we describe in details the QoS verification and certification functions.

3. QoS broker based architecture: components and interfaces

3.1. Architecture description

The architecture extends the standard Service Oriented Architecture (SOA) **[9] [10]** with QoS support for web services. It includes QoS description during the service publication, and performs dynamic QoS- aware invocations. In addition, it verifies, certifies, confirms, and monitors QoS dynamically via a web service-based broker. The architecture involves four main participating roles the web service broker, the web service provider, the client, in addition to a QoS- enabled UDDIe registry **[11]**. Components of the architecture are presented in figure 1. A sequence of interactions between these components is presented in figure 2.

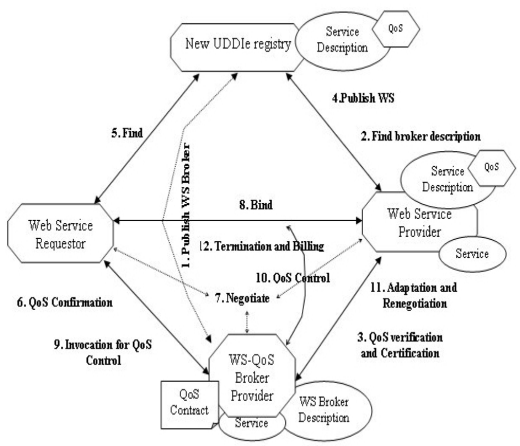

Figure 1. WS-QoS broker architecture

Figure 1 presents an architecture based broker with features that overcome limitations, of existing approaches, described above. Its important features include the support of service selection based on client requirement, QoS verification and certification. QoS verification is the process of validating the correctness of information described in the service interface as well as the described QoS parameters. The QoS verification is performed using an approach that generates test cases to measure QoS parameters. The verification will be used as input for the certification process that will be issued when the verification succeed. The broker arbitrates the negotiation process between clients and their providers until they reach an agreement. During web service invocation, the broker measures dynamically QoS attributes and uses their values to monitor the provision of the selected QoS level; then, it notifies the interested entities of any violation. The broker updates, regularly, its database whenever significant changes happen. In the architecture, the certification process goes beyond certifying just the QoS provider's claims. Additional tests can be performed to make sure that these QoS claims are fulfilled. The broker publishes its interface description in the UDDIe registry (operation 1 in Figure 1). A web services provider looks for the broker's WSDL document in the UDDIe registry (operation 2). Then, it requests the broker to certify the web services and their supported QoS (operation 3). The certification is performed before issuing a certificate, the provider publishes his/her QoS-aware web services in the

UDDIe registry (operation 4). Clients can check the UDDIe registry for QoS-enabled web services satisfying their needs (operation 5). Before starting in the negotiation process with the provider, clients have the possibility to confirm that the published classes of QoS have been previously certified by the broker (operation 6). The broker arbitrates the QoS negotiation between the client and the provider (operation 7).

Figure 2. Sequence Diagram of WS-Qos broker architecture

If an agreement is reached, the client binds to the web service using the agreed class of QoS (operation 8). During invocation, the client can ask the broker to monitor and control the delivered QoS (operation 9 and 10). If the QoS degrades, the broker notifies the provider who initiates QoS adaptation in order to maintain the agreed QoS (operation 11). The QoS renegotiation is initiated if the adaptation operations fail to maintain the agreed QoS (operation 11). The processes terminate by releasing resources and issuing the corresponding bill (operation 12).

3.1.1. Web services broker. The web services broker assists clients in selecting web services based on a set of QoS parameters. The broker is a web services performing a collection of QoSfunctionalities. It is the entity that performs the verification and certification tasks. It is also involved in other operations, such as QoS negotiation, monitoring, and adaptation.

3.1.2. Web services provider (server). The provider is the entity that develops the web service and describes its functionalities in addition to the QoS it provides.

3.1.3. Web services client. The client application operates as a service consumer of the advertised web services

3.1.4. UDDI enabled QoS registry. UDDIe is a registry that supports QoS aware web services publication and discovery **[11].** It supports the notion of "blue pages", to record user defined properties associated with a service, and to enable discovery of services based on these.

4. SERVICE SELECTION ALGORITHMS FOR SEQUENTIAL FLOW STRUCTURE

4.1 Algorithms for the Combinatorial Model

For a composite service that has N service classes (S1, S2... SN) in a process flow plan and with m QoS constraints, we map the service selection problem to a 0-1 multi-dimension multi-choice knapsack problem (MMKP)

The QoS service selection problem is to select one service candidate from each service class to construct a composite service that meets a user's QoS constraints and maximizes the total utility. The QoS service selection problem is mapped to MMKP as follows.

—Each service class is mapped to an object group in MMKP.

—Each atomic service in a service class is mapped to an object in a group in MMKP.

—The QoS attributes of each candidate are mapped to the resources required by the object in MMKP.

—The utility a candidate produces is mapped to the profit of the object.

—A user's QoS constraints are considered as the resources available in the knapsack.

MMKP problem can be solved by using W HEU Algorithm very efficiently.

WS HEU Algorithm. The computation time for BBLP grows exponentially with the size of the problem. This may not be acceptable for QosBrokers that need to make runtime decisions. Heuristic algorithms may be useful to find feasible solutions in polynomial time. We use a heuristic algorithm WS HEU to find solutions for MMKP. The algorithm has three main steps.

(1) Find an initial feasible solution. For each service class Si, WSHEU selects a service ρi that has $\min_j\{\max_\alpha\{q^\alpha_{ij}/Q^\alpha_c\}\}$ in the class. It then checks the feasibility of the initial solution. If the solution is infeasible, the algorithm iteratively improves the solution by replacing the service ρ_i with the largest saving of aggregated QoS among all service classes. The

service replacement continues until a feasible solution is found (or else the algorithm fails).

(2) Improve the solution by feasible upgrades. Among all classes, WS HEU finds service ρ_i to replace ρ_i for S_i to get a higher utility without violating the constraint requirements. The service replacement criterion is based on paper work of Toyoda, If no such service can be found, WS HEU picks the one that maximizes Δp_{ij}.

(3) Improve the solution by infeasible upgrades followed by downgrades. Performing only feasible upgrades may reach a local optimal in the search space. To achieve the global optimal, WS HEU further improves the solution by using F5 to select the service that maximizes Δp_{ij}. This replacement makes the solution infeasible. So one or more downgrades are followed by selecting the service which minimizes Δp_{ij}. This method of upgrades followed by downgrades may increase the total utility of the solution.

WS_HEU is extended from algorithm HEU **[12]** which uses an initial feasible solution by always searching for the lowest utility item in each class. The search, however, is time-consuming. WS_HEU prunes out more infeasible items from each class and finds a feasible solution in a shorter time. In our simulation study, WS HEU finds a feasible solution at the first try in most cases (more than 98% of them), while HEU conducts further revision 70% of the time. WS HEU saves around 50% of the computation time compared to HEU. For a composite service that has N service classes, each with l candidates and m QoS constraints, the time complexity of WS HEU is $O(N^2(l - 1)^2 m)$ (same as HEU **[12]**), which is a polynomial function.

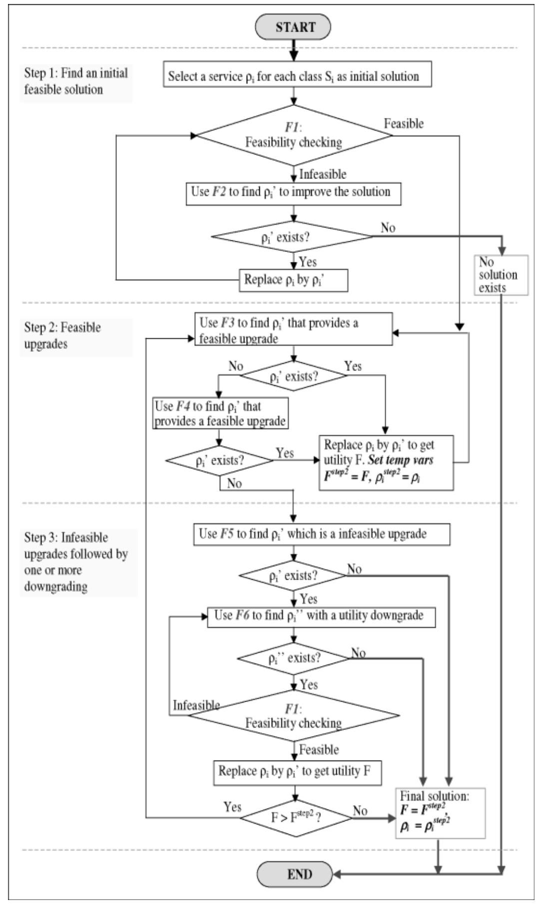

Figure 2. Algorithm structure for WS_HEU

5. CONCLUSION

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. We propose an approach that reduces the complexity of matching user requests according to the specified functional and QoS requirements. We implement a new WS-QoS broker based architecture that solves the problems associated with web service selection. The broker performs the process of publishing and selection of web services. Our suggested theoretical architecture will be based and implemented on QoS properties. An amount of services isneeded to test the performance of the system. This will enable a more flexible, and trustable architecture.

In our future work we intend to use this web services broken architecture in real time systems like apartment management system where the customer can enter his requirements as well as particular field of quality service which he doesn't want to get ignored.The most optimum results based on the requirements will be displayed in.With the popularity of web services increasing day by day we are sure that this system will soon be a part of almost every system around us.

6. REFERENCES

[1] Ashok kumar.P.S, G.Mahadevan, C.Gopal Krishna, Recapitulation of Web services based on Tree structure, International Journal of Computer Application, 2012,Vol. ACCTHPCA,Issue-1, pp. 32- 36

[2] Martin-Diaz, O., Ruiz-Cortes, A., Corchuelo, R., and Toro, M., 2003, "A Framework for Classifying and Comparing Web Services Procurement Platforms", Proc.of 1 st Int'l Web Services Quality Workshop, Italy, pp. 37-46.

[3] Rajendran, T. and Balasubramanie, P., 2009, "An Agent- Based Dynamic Web Service Discovery Framework with QoS Support", International J. of Engg. Research &Indu.Appls. (IJERIA), Vol.2(5),pp 1-13.

[4] Tian, M., Gramm, A., Ritter, H. and Schiller, J., 2004, "Efficient Selection and Monitoring of QoS aware Web Services with the WS-QoS Framework". Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'04) Exchanges, vol. 4, no. 1, pp. 1–10

[5] Eyhab Al- Masri, and Qusay H. Mahmoud, 2007, "QoS- based Discovery and Ranking of Web services", Proceedings of IEEE International Conference.

[6] ZiqiangXu, Patrick Martin, Wendy Powley and FarhanaZulkernine, 2007, "Reputation Enhanced QoS-based Web services Discovery", IEEE International Conference on Web Services (ICWS 2007).

[7] Demian Antony D' Mello, V.S.Ananthanarayana and Santhi.T, 2008, "A QoS Broker Based Architecture for Web Service Selection", Proceedings of IEEE International Conference .

[8] Demian. A. D'Mello and Ananthanarayana, V.S., 2008, "A QoS Model and Selection Mechanism for QoS-Aware Web Services", Proceedings of the International Conference on Data Management (ICDM 2008).

[9] http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[10] D. Gisofli, "Web services architect: An introduction to dynamic e-business", IBM paper, April 2001.

[11] Ali Shaikhl Ali, Omer F. Rana, Rashid Al-Ali, David W. Walker, "UDDIe: An Extended Registry for Web Services", Workshop on Service Oriented Computing: Models, Architectures and Applications, IEEE Computer Society Press, Florida, USA, January 2003.

[12] Khan,S.,LI,K.F.,Manning,E.G.,And Akbar,M.2002.Solvingtheknapsackproblemforadaptive multimedia systems. StudiaInformaticaUniversalis 2, 1, 157–178.