2009 International Symposium on Computing, Communication, and Control (ISCCC 2009)
Proc .of CSIT vol.1 (2011) © (2011) IACSIT Press, Singapore

Efficient Approach Towards an Agent-Based Dynamic Web Service Discovery Framework with QoS Support

Rajendran Thangavel [1] and Balasubramanie Palanisamy [2+]

1Assistant Professor cum Research Scholar, Department of Computer Science Engineering SNS College of Technology, Coimbatore, India

[2] Professor, Department of Computer Science Engineering Kongu Engineering College, Perundurai India

Abstract. Web services are about the integration of applications via the Web. Hereby, the programming effort should be minimized through the reuse of standardized components and interfaces. One of the fundamental pillars of the Web service vision is a brokerage system that enables services to be published to a searchable repository and later retrieved by potential users. One of the subtasks in a service-oriented architecture is service discovery. Service discovery, the identification of existing Web Services that can be used by new Web applications, is one of the most critical problems deterring Web Service (WS) technology. Current solution is based on UDDI catalogue browsing that supports only primitive matching mechanisms and provides no control over the quality of registered services Quality of Service (QoS) is becoming an important criterion for selection of the best available service. Currently the problem is twofold. The Universal Description, Discovery and Integration (UDDI) registries do not have the ability to publish the QoS information, and the authenticity of the advertised QoS information available elsewhere may be questionable. We aim to refine the discovery process through designing a new framework that enhances retrieval algorithms by combining syntactic and semantic matching of services with QoS. We propose a model of QoS-based Web services discovery that combines an augmented UDDI registry to publish the QoS information and a reputation manager to assign reputation scores to the services based on customer feedback of their performance. The Certifier verifies the QoS claims from the Web service suppliers. A discovery agent facilitates QoS-based service discovery using the reputation scores in a service matching, ranking and selection algorithm. The novelty of our model lies in its simplicity and in its coordination of the above mentioned components. The Proposed framework should give Web services consumers some confidence about the quality of services of the discovered Web services.

Keywords: *Web Services discovery, Quality of Services (QoS), Web Service Broker, UDDI, WSDL, SOAP, tModel*

1. Introduction

Service Oriented Architecture (SOA) is an approach to build distributed systems that deliver application functionality as services which are language and platform independent. A Web service is a technology that realizes the SOA. The current Web services architecture encompasses three roles: Web service provider, Web service consumer and Universal Description, Discovery and Integration (UDDI). Web service providers use the Web Services Description Language (WSDL) [4] to describe the services they provide and how to invoke them. The service providers then register their services in a public service registry using UDDI. Application programs discover services in the registry and obtain a URL for the WSDL file that describes the service. Then, the applications can invoke the services using the XML-based Simple Object Access Protocol (SOAP) [8] in either asynchronous messaging or Remote Procedure call (RPC) mode.

+ Corresponding author. Tel.: + (919894676746); fax: +(04222665138).
E-mail address: (rajendran_tm@yahoo.com, pbalu_20032001@yahoo.co.in).

UDDI registry includes businessEntity, businessService, bindingTemplate, and tModel data structure [5]. If multiple Web services provide the same functionality, then a Quality of Service (QoS) requirement can be used as a secondary criterion for service selection. QoS is a set of non-functional attributes like service response time, throughput, reliability, and availability [5,6]. The current UDDI registries only support Web services discovery based on the functional aspects of services [5]. There are two major problems in using QoS for service discovery. First is the specification and storage of the QoS information, and second is the specification of the customer's requirements and matching these against the information available.

We propose a Web services discovery model that contains an extended UDDI to accommodate the QoS information, The certifier verifies the claims of quality of service for a Web service before its registration [5], a reputation management system to build and maintain service reputations, and a discovery agent to facilitate the service discovery. We develop a service matching, ranking and selection algorithm based on a matching algorithm proposed by Maximilien and Singh [3]. Our algorithm finds a set of services that match the consumer's requirements, ranks these services using their QoS information and reputation scores, and finally returns the top M services (M indicates the maximum number of services to be returned) based on the consumer's preferences in the service discovery request.

The rest of this paper is organized as follows. Section 2 outlines the related research conducted in the area of Web services discovery, and QoS. Our proposed discovery framework is illustrated in Section 3. Finally the conclusion and outline of future work are discussed in Section 4.

2. Related Work

Several Web services may share similar functionalities, but possess different non-functional properties. When discovering Web services, it is essential to take into consideration functional and non-functional properties in order to render an effective and reliable selection process. A number of research efforts have studied either QoS-based service discovery or reputation management systems. We provide an overview of some of this work as a context for the research discussed in the remainder of the paper.

Invoking a low quality service in the system could affect the overall performance of the system, among the basic QoS factors are service performance (throughput, response time, latency, transaction time), viability, accessibility, reliability, scalability, exception handling, execution cost, reputation, regulatory, accuracy, integrity, interoperability, security (authentication, authorization, confidentiality, traceability, data encryption, non-repudiation), privacy, network-based factors (network delay, delay variation, packet loss), etc. [5]. Assuring the quality of the selected Web services was discussed in many proposals [5,11,12,17]. In [5] Ran extend the traditional service discovery model with a new role called a Certifier, in addition to the existing three roles of Service Provider, Service Consumer and UDDI Registry. The Certifier verifies the advertised QoS of a Web service before its registration. The consumer can also verify the advertised QoS with the Certifier before binding to a Web service. This approach prevents publishing invalid QoS claims during the registration phase, and help consumers to verify the QoS claims. Although this model incorporates QoS into the UDDI, it does not provide a matching and ranking algorithm, nor does it integrate consumer feedback into service discovery process.

QoS can be used to select and rank the Web services by extending standard service oriented architecture (SOA) [15]. In this architecture, the Web service is selected by matching requested QoS property values against the potential Web service QoS property values [16].

3. The Proposed framework for Web services discovery with QoS

The framework involves four main participating roles. They are Web Service Consumer (client), Web Service Provider, Web Service Broker, in addition to a QoS enabled augmented UDDI registry[5]. Web Service Broker framework consists of four components which are Web Service Certifier [5], Discovery Agent [14], Reputation Manager [14], and Web Service Storage (WSS). Components of the framework are presented in figure 1.

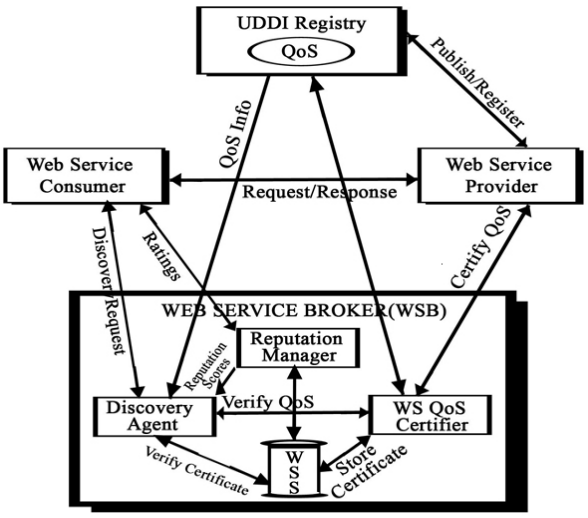

Fig.1: Framework for Agent-based Web Services Discovery with QoS

Finding the suitable service in the UDDI registry that satisfies the user needs or goals (Service Discovery) is the major problem. We extend the traditional Web service model consisting of a service provider, a service consumer and a UDDI to include a Web service QoS certifier, a discovery agent and a reputation manger, and use an augmented UDDI that contains QoS information to allow QoS-based service discovery (as shown in Figure 1). The proposed new registry differs from the current UDDI model by having information about the functional description of the Web service as well as its associated quality of service registered in the repository. The certifier verifies the claims of quality of service for a Web service before its registration. The discovery agent acts as a broker between a service consumer, a UDDI registry and a reputation manager and helps to discover Web services that satisfy the consumer's functional, QoS and reputation requirements. The reputation manager collects and processes service ratings from consumer, stores service reputation scores in a Web Service Storage (WSS) like a Rating Database, and provides the scores when requested by the discovery agent.

3.1. UDDI Registry and Web service certifier of QoS

In the proposed model, a Web service provider needs to supply information about the company, the functional aspects of the provided service as requested be the current UDDI registry, as well as to supply quality of service information related to the proposed Web service. The claimed quality of service needs to be certified and registered in the repository [5]. Once the verification is passed successfully, the certification process is initiated. The certification process consists of issuing a certificate to the service provider. These certificate states that the offered QoS are conform to their descriptions. The Web service provider first needs to communicate its QoS claim to the Web service QoS certifier. The certifier checks the claims and either certifies or down grade the claim. The outcome is sent back to the provider with certification identification information. A certificate is sent to the Web services provider and a copy is stored in the broker's database (WSS) identified by a certification Id for future use. A certificate includes information such as certificate number (certification Id), certificate issue date, and number of years in business, and services location. The certifier provides a set of Web services for any interested parties to access its repository about QoS claims for verification purposes. After the QoS certification been issued by the certifier, the supplier then registers with the UDDI registry with both functional description of the service and its associated certified quality of service information. The UDDI registry communicates with the certifier to check the existence of the certification. After successful checking, the registry then registers the service in its repository [5].

3.2. Reputation Manager

The reputation manager collects feedback regarding the QoS of the Web services from the service consumers, calculates reputation scores, and updates these scores in the Rating DB. For this work, we assume that all ratings are available, objective and valid. Service consumers provide a rating indicating the level of satisfaction with a service after each interaction with the service. A rating is simply an integer

ranging from 1 to 10, where 10 means extreme satisfaction and 1 means extreme dissatisfaction [14]. Our service rating storage system is similar to the one proposed by Wishart et al. [7]. A local database contains the reputation information which consists of service ID, consumer ID, rating value and a timestamp. The service key in the UDDI registry of the service is used as the service ID, and the IP address of the consumer is used as the consumer ID. Only the most recent rating by a customer for a service is stored in the table. New ratings from the same customers for the same service replace older rating [14].

3.3. Discovery Agent

A discovery agent receives service requests containing specifications for functional, QoS, and reputation requirements from the service consumer, finds the services that meet the specified criteria, and then returns a list of services to the consumer [14]. A consumer can specify only QoS requirements in the request, or both QoS and reputation requirements using separate weights for each to indicate their relative importance, where the weights for QoS and reputation requirements must sum to 1. Higher weights represent greater importance. After the agent receives the discovery request, it contacts the UDDI registry to find services that match the customer's functional requirements, and retrieves their QoS information from the corresponding tModels. The agent then uses the service matching, ranking and selection algorithm described in the next section to select the top M services (M is specified by the customer in the discovery request) to return to the customer. If no service is found, the discovery agent returns an empty result to the customer [14].

3.4. Service Matching, Ranking and Selection Algorithm

Figure 2 shows a simplified version of our service selection algorithm where the leftmost numbers denote the line numbers. When the discovery agent receives a discovery request, it executes fMatch (line 2) which returns a list of services LS1 that meet the functional requirements. If QoS requirements are specified, qosMatch (line 4) is executed next on the set of services LS1 and it returns a subset of services LS2 that meet the QoS requirements. selectServices (line 6) always returns a list of M services to the customer where M denotes the maximum number of services to be returned as specified in the discovery request. If QoS requirements are not specified, selectServices returns M randomly selected services from LS1. If only one service satisfies the selection criteria, it returns this service to the customer [14].

```
/*Web services matching, ranking and selection algorithm */
1  findServices (functionRequirements, qosRequirements, repuRequirements, maxNumServices) {
   // find services that meet the functional requirements
2 fMatches = fMatch (functionRequirements);
3 if QoS requirements specified {       // match services with QoS information
4  qMatches = qosMatch (fMatches, qosRequirements); }
5 else {      // select max number of services to be returned
6 return selectServices (fMatches, maxNumServices, "random"); }
7 if reputation requirements specified {      // matches with QoS and reputation information
8  matches = reputationRank (qMatches, qosRequirements, repuRequirements);
      // select max number of services to be returned
9 return selectServices (matches, maxNumServices, "byQoS"); }
10 else {       // matches with QoS information
11  matches = qosRank (qMatches, qosRequirements);     // select max number of services to be returned
12  return selectServices (matches, maxNumServices, "byOverall"); } }
```

Fig.2: Service matching, ranking and selection Algorithm

4. Conclusion

In this paper we have presented a new approach for Web Service discovery process. Due to the increasing popularity of Web services technology and the potential of dynamic service discovery and

integration, multiple service providers are now providing similar services. QoS is a decisive factor to distinguish functionally similar Web services. We proposed a simple yet novel approach to provide QoS-based service discovery. The Certifier verifies the QoS claims from the Web service provider. A discovery agent helps finding services that meet the functional and QoS requirements specified by the consumers. With the assumption that the consumers provide non-malicious and mostly accurate QoS ratings to the reputation manager, these matched services are then ranked based on both their reputation scores generated by the reputation manager and their non-functional QoS attributes values. The top ranked services are returned to the service consumers. To achieve our goal we need to architect our new model and start developing an agent-based framework to support QoS aware discovery of Web services. In the implementation we will use wsdl4j library to parse WSDL files and JWordNetlibrary to enable search of synonymous terms. The paper presents an algorithm for effective service matching, ranking and selection. A mass of services is needed where we can test the performance of our system. For future work, we plan to extend QoS parameters to include information such as reliability, fault rates and Security.

5. References

[1] DAML-S/OWL-S, Available at: http://www.daml.org/services/owl-s/, 2006.

[2] D.Gouscos, M.Kalikakis and P.Georgiadis, “An Approach to Modeling Web Service QoS and Provision Price”. In Proc. of the 1st Int. Web Services Quality Workshop - WQW 2003, Rome, Italy, 2003, pp.1-10.

[3] E.Maximilien and M.Singh, “Toward Autonomic Web Services Trust and Selection”. In Proc. of the 2nd Intl. conf. on Service Oriented Computing, New York City, USA, 2004, pp.212-221.

[4] R. Chinnici et al., “Web Services Description Language (WSDL) Version 1.2”, World Wide Web Consortium, www.w3.org/TR/wsdl12/, 2002.

[5] S.Ran, “A Model for Web Services Discovery with QoS”. SIGEcom Exchanges, Vol. 4(1), 2004. pp.1–10.

[6] W3C, “QoS for Web Services: Requirements and Possible Approaches”, Available: http://www.w3c.or.kr/kr-office/TR/2003/NOTE-ws-qos-20031125/, 2003.

[7] R.Wishart, R.Robinson, J.Indulska, and A.Josang, “Superstring Rep: Reputation-enhanced Service Discovery”. In Proc. of the 28th Australasian conf. on Computer Science, Vol. 38, 2005, pp.49-57.

[8] D.Box et al., “Simple Object Access Protocol(SOAP) 1.1”, W3C Note 08, World Wide Web Consortium, www.w3.org/TR/SOAP/, May 2000,

[9] Daniel A. Menascé, “QoS Issues in Web Services”, IEEE Internet Computing, December 2002.

[10] Liangzhao, “Quality Driven Web Services Composition”, the Twelfth International World Wide Web Conference, Budapest, Hungary, May 2003.

[11] A. Mani and A. Nagarajan, “Understanding quality of service for Web services”, IBM paper, January 2002.

[12] Ying Li, Xiaochen DING, Ying CHEN, Dong LIU, Thomas LI, “The Framework Supporting QoS-enabled Web Services”, IEEE International Conference on Web Services, Las Vegas, Nevada, USA, June 2003.

[13] UDDI Technical White Paper, Retrieved from http://uddi.org/pubs/udditech-wp.pdf.

[14] Ziqiang Xu, Patrick Martin, W. Powely and F. Zulkernine, “Reputation-Enhanced QoS-based Web service discovery”, IEEE International Conference on Web Services, 2007.

[15] M.Adel Serhani, Rachida Dssouli, Abdelhakim Hafidand Houari Sahraoui, “A QoS broker based architecture for efficient Web services selection”, Proceedings of the IEEE International Conference on Web Services (ICWS'05), IEEE 2005.

[16] Demian Antony D’Mello, V.S. Ananthanarayana and T.Shanthi, “A QoS broker based architecture for Dynamic Web services selection”, Proceedings of the Second Asia International Conference on Modeling & Simulation, 2008.