31st May 2010. Vol.15. No.2.
© 2005-2010 JATIT. All rights reserved
www.jatit.org

AN EFFICIENT ARCHITECTURE FOR AGENT-BASED DYNAMIC WEB SERVICE DISCOVERY WITH QOS

[1]T.RAJENDRAN, [2] DR. P. BALASUBRAMANIE

[1]Associate Professor cum Research Scholar, Department of Computer Science Engineering,

SNS College of Technology, Coimbatore, India-641 035

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

ABSTRACT

The SOA enables the development of flexible large scale-applications in open environments by dynamically combining the web services. There exist many web services which exhibit similar functional characteristics. It is imperative to provide service consumers with facilities for selecting required web services according to their non-functional characteristics or QoS. An important issue arising from Web Service applications is how to conveniently, accurately and efficiently retrieve services from large-scale and expanding service repositories. The QoS based web service discovery play an essential role in SOA because most of the applications want to use services that accurately meet their requirements. This work proposes a web service discovery mechanism in which the functional and non-functional requirements are taken into account during service discovery. In this paper, we propose a novel approach for designing and developing a agent-based architecture and its QoS-based matching, ranking and selection algorithm for evaluating web services. The paper presents an optimal approach for discovering the most suitable web service according to the consumer's functional and quality requirements.

Keywords: *Agent, Quality of Service (QoS), Service Selection, Web Services, Web Service Discovery*

1 INTRODUCTION

Web Services are new forms of Internet software which can be invoked using standard Internet protocols. Web Services, as it is defined by the World Wide Web Consortium (W3C), is a software system designed to support interoperable machine-to-machine interaction over a network. Web services interact with each other, fulfilling tasks and requests that, in turn, carry out parts of complex transactions or workflows. If multiple Web services provide the same functionality, then Quality of Service (QoS) requirements 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 [1], [2]. The current Universal Description, Discovery and Integration (UDDI) registries only support Web services discovery based on the functional aspects of services [1]. The problem, for that reason, is firstly to accommodate the QoS information in the UDDI, and secondly to guarantee some extent of authenticity of the published QoS information. QoS information published by the service providers may not always be accurate and up-to-date.

There are two major problems in using QoS for web 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. Major efforts in this area include Web Services Level Agreements (WSLA) [3] by IBM, Web Services Policy Framework (WS Policy) [4], and the Ontology Web Language for Services (OWL-S) [5]. Most of these efforts represent a complex framework focusing not only on QoS specifications, but on a more complete set of aspects relating to Web services. Some researchers propose other simpler models and approaches [6]-[8] for dynamic Web services discovery. However, they all struggle with the same challenges related to QoS publishing and matching.

Currently, both Web Services 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 to accommodate the QoS information, and Web Service 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 [9]. Our algorithm finds a set of services that match the consumer's requirements, ranks these services using their QoS information and feedback rating, 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.

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 services discovery. 2) QoS aware web services publishing and discovery. In this paper, we propose a agent-based architecture for web services discovery and QoS characteristics. The role of the Web Service Agent (WSA) is to support QoS provisioning and assurance in delivering web services. It implements the concept of Quality Analysis, and QoS verifying and certifying process. The goal of this research is to investigate how dynamic Web service discovery can be realized to satisfy a customer's QoS requirements using a new architecture that can be accommodated within the existing basic Web service protocols.

The remainder of the paper is organized as follows. Section 2 outlines the related research conducted in the area of web services discovery, QoS and reputation. In Section 3, we describe our proposed Agent-Based architecture for web service discovery. Section 4 concludes the paper and presents possible future research in this direction.

2 RELATED WORK

Researchers have proposed various approaches for dynamic web service discovery. Maximilien and Singh [9] proposed a multi-agent based architecture to select the best service according to the consumers' preferences. Blum [10] proposes to extend the use of Technical models (tModels), within the UDDI to represent different categories of information such as version and QoS information. Ran [1] proposes an extended service discovery model containing the traditional components: service provider, service consumer and UDDI registry, along with a new component called a Certifier. Certifier verifies the 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. Although this model incorporates QoS into the UDDI, it does not integrate consumer feedback into the service discovery process. However, it lacks support for the dynamism of web services.

Hunaity and Rashid [11] refines the web service discovery process through designing a new framework that enhances retrieval algorithms by combining syntactic and semantic matching of services. It proposes a new framework for smarter WS discovery that provides clients with QoS information which will enhance the selection process and reduce the failure chances by getting endorsements or recommendations from other services or special agents about each service. The proposed model consists of the basic web service model components (Service Provider, Service Consumer, and UDDI Registry) with one addition, which is the capability to store QoS information using tModel data structure. The model is enhanced with a three agents (Discovery Agent, Service Mediator and Reputation Manager). The service provider will describe the entire functional and non functional attribute in the UDDI directly, or through the service mediator agent. The service mediator agent will handle all communication with registries, bindings, negotiations, voting, requests, and responses for that service. The service consumer can search for a specific service directly in the UDDI or it can interact with its specific discovery agent. This framework does not provide any verification or certification process for QoS.

Majithia et al [12] propose a framework for reputation-based semantic service discovery. Ratings of services in different contexts are collected from service consumers by a reputation management system. Reference [13] shows a framework for agent-based web services discovery with QoS to select the suitable web service that satisfies the client's preferences and QoS constraints. It contains an extended UDDI to accommodate the QoS information. IBM proposes Web Service Level Agreements (WSLA), which is an XML specification of SLAs for web services focusing on QoS Constraints [14]. Many of these approaches do not provide guarantees as to the accuracy of the QoS values over time or having up-to-date QoS information. Farkas and Charaf [15] proposed software architecture to provide QoS-enabled web services by adding a QoS broker between clients and service providers to discover the QoS aware services in UDDI. However, no detailed information about QoS broker, such as how it is designed and the functionality of it is presented.

UDDI extension to support QoS- enriched service publication and discovery has generated several research efforts. ShaikhAli's approach [16] is based on the extension of the UDDI business service structure, but potential QoS changes are not considered. Chen et al [17] propose a registry that receives reports made by consumers to generate QoS summaries for invoked web services. Kalepu et al [18] evaluate the reputation of a service as a function of three factors: ratings made by users, service quality compliance, and the changes of service quality conformance over time. However, these solutions do not take into account the trustworthiness of QoS reports produced by users, which is important to assure the accuracy of the QoS-based web service selection and ranking results. Liu et al [19] proposes an approach for rates services computationally in terms of their quality performance from QoS information provided by monitoring services and users. The authors also employ a simple approach of reputation management by identifying every requester to avoid report flooding. In [20], an extended Web service architecture to support QoS management. The architecture is currently being integrated with Business Process Management (BPM) Technology. The major contributions are: Extending the WS policy framework to specify QoS policies for web services, extending the UDDI information model and API set to refine service discovery and using tModels to define QoS related concepts.

Tian et al [21] shows a 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. Reference [22] introduces 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 [23] provides 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, but they did not provide any certification or verification process in that model. Reference [24] explores 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 [26]. The QoS broker is responsible for the selection and ranking of functionally similar Web services. The Web service selection mechanism [26] ranks the Web services based on prospective levels of satisfaction of requester's QoS constraints and preferences. Serhani [27] proposes web service architecture employs an extended UDDI registry to support service selection based on QoS, but only the certification approach is used to verify QoS and no information is provided about the QoS specification.

Chen et al [28] presents a description and an implementation of broker-based architecture for controlling QoS of web services. The broker acts as an intermediary third party to make web services selection and QoS negotiation on behalf of the client. Delegation of selection and negotiation raises trustworthiness issues mainly for clients. Performance of the broker is not considered in this approach. Moreover, performance of the broker can be a key to the success of any proposed architecture; if the user does not get a response to his/her request with an acceptable response time, he/she will switch to another provider. Hu et al [31] shows the Web service is selected by matching requested QoS property values against the potential Web service QoS property values. In this literature, the Web service is selected by taking the requester's average preference for QoS properties. Many of these approaches do not provide guarantees as to the accuracy of the QoS values over time or having up-to-date QoS information. In the next section, we describe the design of the proposed Web Service agent-based architecture which overcomes many of the limitations imposed by current discovery model.

3 AGENT-BASED ARCHITECTURE FOR WEB SERVICE DISCOVERY WITH QOS

The purpose of web service discovery 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. We propose a technique for dynamic discovery of Web Services which will also handle the problem of redundant Web Services.

The architecture consists of the basic web service model components like the web service provider, web service consumer and the UDDI registry. In addition, UDDI registry has the capability to store QoS information using tModel data structure and a Web Service Agent (WSA). Components of the architecture are presented in Fig 1. We propose a web services discovery architecture which contain an extended UDDI to accommodate the QoS parameters [23]. The WSA assists clients in selecting web services based on a set of QoS parameters. The WSA has four components: Service Publisher, Verifier and Certifier [32], Retrieval Agent, Quality Analyzer and Web Service Storage (WSS) [25]. Agent services may be used to facilitate service registry access. The agent performs the interaction with the UDDI.

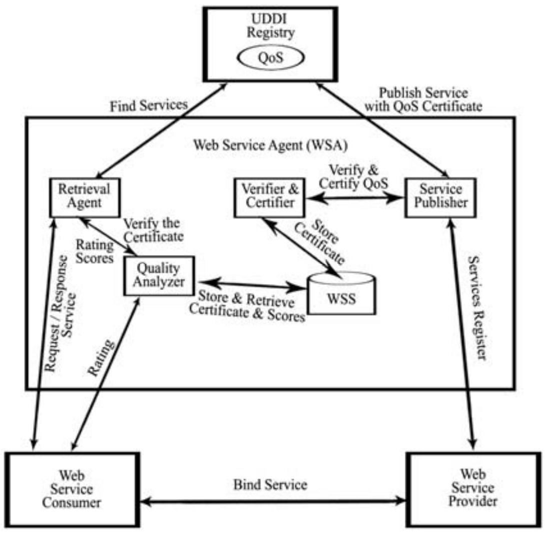

Fig. 1. Architecture for WSA

The WSA assists clients in selecting web services based on a set of QoS parameters. The broker is a web services performing a collection of QoS functionalities. It is the entity that performs the verification and certification tasks. It is also involved in other operations, such as registering and selecting services with QoS functions.

To overcome many of the limitations imposed by current discovery model, we introduce the Web Service Agent (WSA) architecture shown in Fig.1. The service publisher component facilitates the registration, updating and deletion of web service related information. It gets the business specific and performance specific QoS property values of web services from the service providers. The service provider publishes its service functionality to the UDDI registry through the service publisher after certification and verification. For every service or group of services there exists a service publisher that handles all communication with registries, bindings, negotiations, requests and responses for that service. The service consumer can search the UDDI registry for a specific service through the retrieval agent. The main functionality of the retrieval agent component is to select the most suitable web service satisfying requester's QoS constraints and preferences, along with service functionality. The service requester can verify the advertised QoS with the retrieval agent before binding to a web service. The WSA performs the verification and certification tasks. QoS verification is the process of validating the correctness of information described in the service interface as well as the described QoS parameters. The result of verification will be used as input for the certification process that will be issued when the verification succeed. The QoS property values obtained from the service providers are verified and certified by the Verifier and Certifier component before registering them into the UDDI registry. The Verifier and Certifier [32] component is implemented within the WSA and is responsible for certifying web services and their provided QoS. A certificate is sent to the web services provider and a copy is stored in the WSS for future use. A sequence of interaction between these components is presented in Fig.2.

A typical usage scenario (Fig.2) is described here by considering an example in which a consumer uses a web service of a provider in its application.

1. Initially Web Service Agent (WSA) publishes the interface to the UDDI registry.
2. Web service provider finds the agent interface in UDDI registry.
3. The service provider registers the web service with the service publisher which is available in the WSA and provides functional and non-functional information about the offered services.
4. The Verifier and Certifier component of the WSA verifies the QoS information and issues a certificate.
5. A copy of the QoS certificate is stored in WSS and a copy is sent to the service provider.
6. The service publisher then publishes the web service in the UDDI registry along with the QoS certificate.

7. The consumer application requests service discovery and provides functional and QoS requirements.
8. The retrieval agent finds the service in the UDDI registry according to the required service functionality and QoS requirements of the application.
9. Retrieval agent can communicate with quality analyzer to verify the provided QoS certificate and rating scores with the one stored in the WSS.
10. The retrieval agent then reports the discovered service back to the application.
11. The web service consumer then binds the web service from the service provider.
12. Consumer sends the feedback to the quality analyzer after invoked the service.
13. Quality analyzer calculates the rating scores for consumer feedback then stores it into database (WSS) for service discovery process.

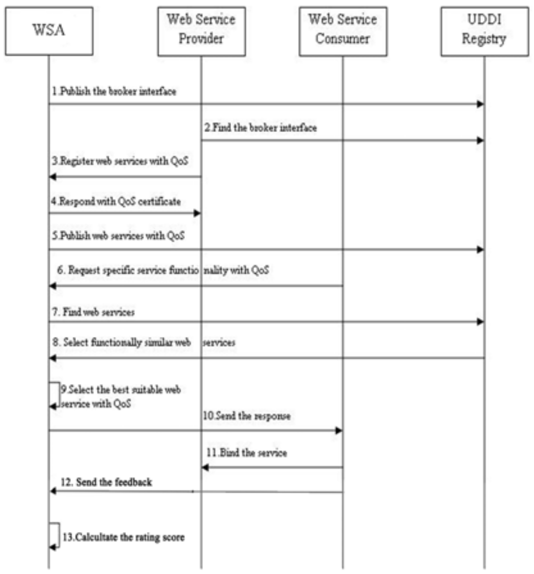
WSA
Web Service Provider
Web Service Consumer
UDDI Registry
1.Publish the broker interface
2.Find the broker interface
3.Register web services with QoS
4.Respond with QoS certificate
5.Publish web services with QoS
6. Request specific service functio nality with QoS
7. Find web services
8. Select functionally similar web services
9.Select the best suitable web service with QoS
10.Send the response
11.Bind the service
12. Send the feedback
13.Calcultate the rating score

Fig. 2. Architectural Component Interactions

3.1 UDDI with QoS Information

Although Web service technology allows the development and execution of distributed applications, it still lacks facilities to deal with QoS. Consumers may require services with particular nonfunctional characteristics and expect quality level guarantees. If multiple Web services provide the same functionality, then a QoS requirement can be used as a secondary criterion for service selection. The QoS information is represented in UDDI registry by a tModel, which allows specification, standardization and reuse of QoS related concepts. This extension allows the use of agents to facilitate service discovery according to functional and non-functional requirements, and monitors to verify QoS attributes. QoS represents the non-functional aspects of the service being provided to the web service users [30]. The following QoS parameters are considered:

- **Price**: The cost involved in requesting the service which can be estimated by operation or volume of data
- **Response Time**: Time taken by a service to respond to the client request
- **Availability**: Percentage of time that the service is operating
- **Throughput**: The maximum requests that can be handled at a given unit in time.

A tModel consists of a key, a name, an optional description and a Uniform Resource Locator (URL), that point to the location where the details about the actual concept can be found. When a service is published in the UDDI registry, a tModel is created to represent the QoS information of the service. It is registered with the UDDI registry and referenced in the bindingTemplate that represents the deployment information of the web service. In the tModel, each QoS metrics is represented as a KeyedReference, which contains the name of a QoS attribute as keyName and keyValue, which contains the value. A service provider should regularly update the QoS information of the services, it publishes, to ensure that the information is accurate and up-to-date. To update the QoS information of a service, the service provider searches the UDDI registry through the service publisher to find the corresponding tModel. It then updates the QoS information in the tModel and saves it back using the same tModel key that was assigned to the tModel when it was created.

The units of QoS attributes are not represented in the *tModel*. We assume default units are used for the values of QoS attributes in the *tModel*. For example, the default unit used for price is CAN$ per transaction, for response time is second, for availability is percentage, and for throughput is transaction per second. For example, a company publishes its Stock Quote service in a UDDI registry with the QoS information.

```
<tModel tModelKey = "anycompany.com: Stock
          QuoteService:PrimaryBinding:
QoSInformation">
 <name>QoS Information for Stock Quote
Service</name>
```

31st May 2010. Vol.15. No.2.
© 2005-2010 JATIT. All rights reserved
www.jatit.org

```
 <overviewDoc>
   <overviewURL>
     http://<URL describing schema of QoS attributes>
   </overviewURL>
 </overviewDoc>
 <categoryBag>
   <keyedReference
        tModelKey="uddi:uddi.org:QoS:Price"
        keyName="Price "
        keyValue=" 0.01" />
   <keyedReference
   tModelKey="uddi:uddi.org:QoS:ResponseTime"
        keyName="ResponseTime"
        keyValue="0.05" />
   <keyedReference
       tModelKey="uddi:uddi.org:QoS:Availability"
        keyName="Availability"
        keyValue="99.5" />
   <keyedReference
       tModelKey="uddi:uddi.org:QoS:Throughput"
        keyName=" Throughput"
        keyValue="300" />
 </categoryBag>
</tModel>
```

Fig. 3. tModel with QoS Information

Above given is an example of the QoS Information *tModel*, which contains a *categoryBag*, which is a list of name-value pairs specifying QoS metrics. This *tModel* contains a *categoryBag* that specifies four QoS metrics of Price, Response Time, Availability and Throughput. The *tModelKey* in each *keyedReference* is used as a namespace which provides a uniform naming scheme. The company creates and registers a *tModel* that contains the QoS information for this service before it publishes the service with the UDDI registry.

3.2 Service Publisher

The service publisher component communicates with the service provider and the UDDI registry. The service provider registers the business and web service related information with the service publisher. It also gets the specific QoS property values of web services from providers. Once the QoS property values and other information are obtained from the provider it is presented to the Verifier and Certifier component. The QoS information is verified and certified before publishing it in the UDDI registry.

3.3 Verifier and Certifier

This is the key component of the WSA that performs the verification of the QoS information supplied by the service provider and issues a certificate to the service provider through the service publisher. This QoS certificate assures that the QoS offered by the provider confirm to their descriptions. The service provider initiates the verification process through the service publisher by supplying the QoS property values. The verifier is provided with the WSDL document and additional information about resources available at the provider’s platform. The verifier performs the testing of the service URI, the XML schema definition, the service binding information and the availability of all operations described in the service interface. Verifier also performs the verification of the QoS information introduced in the service interface.

The QoS verification is conducted through a set of test cases generated by the verifier. For each test, additional information like server capacity, network bandwidth about the provider and its web service are needed. The four QoS parameters [30] (Response Time, Availability, Throughput, and Price) are also verified. The verification process is done in three levels: General web services information verification, WSDL content verification and QoS verification. A web service is said to be compliant with a given level when it passes the corresponding tests described in the verification document. Based on this, web services can be classified into three classes. Class A includes web services for which all verification tests have succeeded. Class B includes web services for which more than 80% of the verification tests have succeeded. Class C contains the services for which most of the verification scenarios have failed.

Once the verification process is completed successfully, the certification process is initiated. The certifier issues a certificate to the service provider through the service publisher which indicates that the offered QoS confirm to their descriptions. The main responsibility of the certifier is to certify the web services and their provided QoS. A copy of the certificate sent to the service provider, which is also stored in the WSS for future use. The certificate includes information such as certificate number, certificate issue date, number of years in business and service location. In case, if the certificate cannot be issued, feedback will be sent to the provider. After the QoS certification process, the service publisher can register with the UDDI registry, the functional description of the service and the certified QoS information.

3.4 Retrieval Agent

The retrieval agent component is concerned with selecting the most suitable web service satisfying the consumer’s QoS constraints and the specific service functionality. It receives messages from the web service consumer, specifying the service functionality along with the QoS constraints. Based on the received requirements specification, it discovers functionally similar web services from the UDDI registry. The retrieval agent can check the validity of the QoS information in the UDDI registry by comparing the QoS certificate provided by the Verifier and Certifier with the one stored in the WSS.

3.5 Quality Analyzer

After a web service task is finished, a client may summarize the QoS experience and send them to the quality analyzer in WSA. The quality analyzer collects feedback regarding the QoS of the Web services from the service consumers, calculates feedback rating scores, and updates these scores in the WSS. The quality analyzer uses this information during the service discovery phase. 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 score is simply an integer ranging from 1 to 10, where 10 means extreme satisfaction and 1 means extreme dissatisfaction. Our service rating storage system is similar to the one proposed by Wishart *et al.* [29]. A local database contains the rating 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 database. New ratings from the same customers for the same service replace older ratings.

3.6 QoS Matching, Ranking and Selection Algorithm

A web service consumer sends a service discovery request to the retrieval agent, which then contacts the UDDI registry to find services that meet the customer’s functional and QoS requirements. A service is said to be a “match” if it satisfies the customer’s functional requirements and its QoS information. If no matched service is found by the matching process, the retrieval agent returns an empty result to the customer. If multiple services match the functional and QoS requirements, the retrieval agent calculates a QoS score for each matched service based on the dominant QoS attribute specified by the customer, or on the default dominant attribute, average response time. The best service is assigned a score of 1, and the other services are assigned scores based on the value of the dominant QoS attribute. The top M services (M being the maximum number of services to be returned as specified by the customer) with the highest QoS scores are returned to the customer. If M is not specified, one service is randomly selected from those services whose QoS score is greater than LowLimit.

```
/* Web services matching, ranking and selection
algorithm */

1 findServices (functionRequirements,
qosRequirements, feedbackRequirements,
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 feedback requirements specified {
   // matches with QoS and feedback information
8  matches = feedbackRank (qMatches,
qosRequirements, feedbackRequirements);
      // 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. 4. Service matching, ranking and selection Algorithm

Fig 4 shows a simplified version of our service selection algorithm where the leftmost numbers denote the line numbers. When the retrieval 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.

4 CONCLUSION

In this paper, we presented a agent-based architecture for web services discovery. The goal of the agent is to support web services discovery with QoS registration, verification, certification, and confirmation. The agent performs the process of publishing and selection of web services. We described the key features of the agent that are not supported by existing approaches dealing with QoS for web services. We propose an approach for dynamic service discovery and, which 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.
- It assures a level of security, since the clients do not have direct access to the Web Services.

The service provider does not have to design and develop her/his own agent but just invoke one from the published agents. The client will also find a good support during its web services discovery using the agent services. Our suggested theoretical architecture will be based and implemented on QoS properties. An amount of services is needed to test the performance of the system. This will enable a more flexible, and trustable architecture. Results of this work will be reported in a future paper. Future work involves enhancing the capabilities of the proposed architecture to handle other QoS attributes and adapting the architecture to support mobile Web services.

REFERENCES

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

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

[3] IBM Corporation,"Web Service Level Agreement (WSLA) Language Specification" Ver. 1.0. Retrieved from http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf, 2003.

[4] W3C, "WS-Policy Framework ver.1.2", Available at: http://www.w3.org/Submission/WS-Policy/, 2006.

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

[6] E.M.Maximilien, & M.P.Singh, "Reputation and Endorsement for Web Services". *ACM SIGecom Exchanges*, Vol. 3(1), pp.24–31, 2002.

[7] E.M.Maximilien, and M.P.Singh, "Self-Adjusting Trust and Selection for Web Services. In extended Proc. *Of 2nd IEEE Intl. conf. on Autonomic Computing (ICAC)*, pp.385-386, 2005.

[8] L.Vu, M.Hauswirth, and K.Aberer, "QoSbased service selection and ranking with trust and reputation management". In *Proc. of the Intl. conf. on Cooperative Information Systems (CoopIS)*, Agia Napa, Cyprus, 2005.

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

[10] A.Blum, "UDDI as an Extended Web Services Registry: Versioning, quality of service, and more". *White paper, SOA World magazine*, Vol. 4(6), 2004.

[11] Mossab Ahmmad Rashid, Hunaity, "Towards an Efficient Quality Based Web Service Discovery Framework". *IEEE Congress on Services*, 2008.

[12] S.Majitha, A.Shaikhali, O.Rana, and D.Walker, "Reputation based semantic service Discovery", *In Proc. Of the 13 th IEEE Intl Workshops on Enabling Technologies Infrastructures for collaborative Enterprises (WETICE)*, Modena, Italy, pp.297-302, 2004.

[13] T.Rajendran, and P.Balasubramanie, "An Efficient Framework for Agent-Based Quality Driven Web Services Discovery", *IEEE International conference on Intelligent Agent and Multi Agent Systems (IAMA2009)*, Chennai, 2009.

[14] A.Keller and H.Ludwing. "The WSLA framework: Specifying and Monitoring Service Level Agreements for Web Services", *IBM Research Report*, 2002.

[15] P. Farkas and H. Charaf, "Web Services Planning Concepts", *1st International Workshop on C# and .NET Technologies on*

Algorithms, Computer Graphics, Visualization, Distributed and WEB Computing, Feb. 2003

[16] A.ShaikAli, O.F.Rana, R.Al-Ali and D.W.Walker, "UDDIe: An extended registry for web services". *In Proc. Of the Symposium on Applications and the Internet workshops, IEEE CS*, pp 85-89, 2003.

[17] Z.Chen, C.Liang-Tien, B.Silverajan and L.Bu-Sung, "UX-An architecture providing QoS-aware and federated support for UDDI". *In proc. of the Int'l Conf. on web services, CSREA Press*, pp 171-176, 2003.

[18] S.Kalepu, S.Krishnaswamy and S.W.Loke, "Reputation = f (User Ranking, Compliance, Verity)", *Proceedings of ICWS'04*, 2004.

[19] Y.Liu, A.Ngu and L.Zheng, "QoS Computation and Policing in Dynamic Web Service Selection", *Proceedings of WWW 2004 Conf*, 2004.

[20] Diego Zuquim Guimaraes Garcia and Maria Beatriz Felgar de Toledo, "A web service Architecture providing QoS Management", *Web Congress, LA-Web '06, Fourth Latin American*, pp -189-198, 2006.

[21] M.Tian, A.Gramm, H.Ritter and J.Schiller, "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, 2004.

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

[23] Ziqiang Xu, Patrick Martin, Wendy Powley and Farhana Zulkernine, "Reputation Enhanced QoS-based Web services Discovery", *IEEE International Conference on Web Services (ICWS 2007)*, 2007.

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

[25] T.Rajendran and P.Balasubramanie, "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, 2009.

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

[27] M.A.Serhani, R.Dssouli, A.Hafid and H.Sahraoui, "A QoS broker based architecture for efficient Web services selection". *In Proc. of the IEEE Int'l Conf. on Web Services, IEEE CS*, pages 113–120, 2005.

[28] Hongan Chen, Tao Yu, Kwei-Jay Lin, "QCWS: an implementation of QoS-capable multimedia web services", *IEEE Fifth International Symposium on Multimedia Software Engineering*, 2003.

[29] R.Wishart, R.Robinson, J.Indulska and A.Josang, "SuperstringRep: Reputation-enhanced Service Discovery". In *Proc. of the 28th Australasian conf. on Computer Science*, Vol. 38, pp.49-57, 2005.

[30] T.Rajendran, and P.Balasubramanie, "Analysis on the Study of QoS-Aware Web Services Discovery", *Journal of Computing*, Vol. 1(1), pp 119-130, 2009.

[31] J. Hu, C. Guo, H. Wang, and P. Zou, 2005, "Quality Driven Web Services Selection" *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'05)*, 2005.

[32] T.Rajendran, P.Balasubramanie, and Resmi Cherian, "An Efficient WS-QoS Broker Based Architecture for Web Services Selection", *International Journal of Computer Applications*, Vol. 1(9), pp 110-115, 2010.

31st May 2010. Vol.15. No.2.
© 2005-2010 JATIT. All rights reserved
www.jatit.org

AUTHOR PROFILES:

T.Rajendran is a PhD Scholar in the Department of Computer Science and Engineering at Kongu Engineering College, affiliated to Anna University, Coimbatore, Tamilnadu, India. Currently he is an Associate Professor in the Department of Computer Science and Engineering in SNS College of Technology, Coimbatore. His research interest includes Web Services, Network Security and Web Technology. He has obtained ME degree in Computer Science and Engineering. He is a life member of ISTE. He has published more than 30 articles in International/ National Journals/Conferences.

Dr.P.Balasubramanie has been awarded Junior Research Fellowship (JRF) by CSIR in the year 1990. He completed his PhD degree in 1996 at Anna University, Chennai. Currently he is a professor in the Department of Computer Science and Engineering in Kongu Engineering College, Perundurai, and Tamilnadu, India. He has guided 7 PhD scholars and guiding 20 scholars Under Anna University. He has published more than 63 articles in International/ National Journals/Conferences. He has authored 6 books with the reputed publishers.