User Oriented Web Services Discovery based on QoS Parameters in Multiple Registries

Vandan Tewari
Assistant Professor
CTA Dept., SGSITS, Indore

Nirmal Dagdee
Prof.& Head
Comp.Engg.Dept,. SD Bansal College of Tech,Indore

Aruna Tiwari
Assistant Professor
Comp. Engg. Dept.,IIT Indore

ABSTRACT
In the age of globalization, day by day business to business and business to consumer operations are finding huge importance in internet computation around the world. Web services are one means by which we can fulfil all these demands in an easy and efficient way.

Web Services are based on Service Oriented Architecture which enables application-to-application communication over the internet and easy accessibility to heterogeneous applications and devices. As web services proliferate and become more sophisticated and interdependent, the issues regarding their publication and discovery become of utmost importance. Discovering web services using search techniques offered by existing UDDI APIs may not result in the search results that are appropriate to service requestor's needs.

In this paper we have proposed a User Oriented Web Services Discovery framework that takes into account not only the functional parameters based on keywords but also non functional parameters such as QoS Parameters. Further we also use data mining techniques and Service Rating based on QoS Parameters for efficient and effective web service discovery. As an additional facility we have used a specially designed engine which can help a service provider in dynamic deployment as well as publishing of Web Services along with the QoS based search.

General Terms
Web Service Discovery

Keywords
UDDI, WSDL, SOA, XML, QoS.

1. INTRODUCTION

A. Web Service
The W3C[1] defines a "Web service" as "a software system designed to support interoperable machine-to-machine interaction over a network". It has an interface described in a machine process able format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages (Simple Object Access Protocol), typically conveyed using HTTP(Hyper Text Transfer Protocol) with an XML serialization in conjunction with other Web-related standards[1].

B.WSDL
Web Service Description language is an XML based language that provides a mode for describing web service. WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL[2].

C. UDDI
The Universal Description, Discovery and Integration (UDDI) specifications define a registry service for Web services and for other electronic and non-electronic services. A UDDI registry service is a Web service that manages information about service providers, service implementations, and service metadata. UDDI provides the registry for Web services to function as a service manager enabling the service providers to populate the registry with service descriptions and service types and the service requestors to query the registry to find and locate the services. These registries can be either private services within an enterprises or a specific community, or they can be public registries such as jUDDI[3].

D. SOAP
Simple Object Access Protocol, or SOAP, is a standard for a lightweight XML-based messaging protocol. It enables an exchange of information between two or more peers and enables them to communicate with each other in a decentralized, distributed application environment. Like XML, SOAP also is independent of the application object model, language, and running platforms or devices. In the core of the Web services model, SOAP is used as the messaging protocol for transport with binding on top of various Internet protocols such as HTTP,SMTP, FTP, and so on. SOAP uses XML as the messages format, and it uses a set of encoding rules for representing data as messages [4].

2. RELATED WORK
Web Services are means for building distributed applications and are used to build service-based applications. SOA (Service Oriented Architecture) 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 is build as three entities: Web service provider, Web service consumer and Universal Description, Discovery and Integration (UDDI). The Web service provider publishes a description of the service in the UDDI registry. Users (service consumers) will search that directory to get their desired services. This directory is a UDDI registry which includes businessEntity, businessService, bindingTemplate, and tModel data structures as implementation. In UDDI, information about businesses and services is described in XML.

Discovering web services using search techniques offered by existing UDDI APIs may not result in the search results that are appropriate to service requestor's needs. When discovering web services, clients search for the services that not only meet their functional needs but also quality of service (QoS)[5]. Authors here stresses upon that it is very important to keep track of the QoS parameters as well while discovering a service. Extending search in existing search API [6] only exploits keyword based search techniques which may not be suitable for web services particularly when differentiating between those that share similar functionalities. Differentiating between web services that share similar functionalities is significantly achieved by examining non functional web service attributes such as response time, throughput, availability etc. In [7] Authors have discussed an approach where they propose to use a third party agent to certify the QoS parameters proposed by the service provider. In [8] author proposes an extension to web services framework that enables collection of functional and non-functional service characteristics at run time . E. El- masri [9] in his paper stresses upon the need of discovering best services for a client based on non functional parameters of a service .Xu et al. [10] in his research work proposes to use 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. Authors have also developed service matching, ranking and selection algorithm, but have not not provided any process of verification process for QoS in that proposed discovery model. The proposed web service selection and ranking mechanism uses the QoS broker based architecture in [11]. Here the QoS broker is responsible for selection and ranking of functionally similar web services. and ranks the web services based on consumer's QoS demands. QoS can be used to select and rank the Web services by adding to existing architecture of generic SOA [12,13]. In the proposed architecture, the Web service is selected by matching requested QoS values against the given Web service QoS values. In [13], the Web service is selected by taking the average rate of acceptance for QoS properties [13].

Further if we want to offer more or better services to a service consumer we would need our discovery to be spread of over several multiple service registries for various businesses and services. Using multiple registries allows this request to be forwarded transparently to the proposed virtual registry federation and hence the search space is enlarged and the opportunity for the service consumers to discover more services that satisfy their requirements is increased.

In [14] Authors have proposed to do so by making a federation of trading services in CORBA. And then adopt the idea of peer-based federation and policy, and enhance with authentication and authorization control for access to UDDI entries within the nodes. However it works well as long as all the UDDI registries are in same domain of trust.

We however have proposed to enable agent based QoS based discovery which can use multiple registries and can expand its search space by dynamically extending domain of trust as it crawls to those UDDIs which are not in its domain of trust.

3. OUR PROPOSAL

In order to provide QoS based ranking of Web services, It is important to collect QoS information about Web services. The QoS based discovery Framework uses a Discovery Agent for crawling URL of Web services and Rating Engine for ranking services. This Rating Engine also classifies the Services based on QoS Parameters using a decision tree based algorithm. The proposed framework for the QoS driven discovery is shown in Fig 1

A. ***WSM (Web Service Manager):*** It is Middleware between Client and UBR. Service Provider Provides QoS Metrics to Middleware for Verifying with standard Threshold values**.**

B. ***Rating Engine:*** A Rating Engine module within the WSM framework is responsible for measuring QoS information for the collected web services, and information is stored in the QoS database as shown in above Figure.1

C. ***Discovery Agent:*** The Discovery Agent module within the WSM framework work as a crawler for discovering services from UBR on client request.

D. ***WSM (Web Service Manager):*** It is Middleware between Client and UBR. Service Provider also Provides QoS Metrics to Middleware for Verifying with standard Threshold values**.**

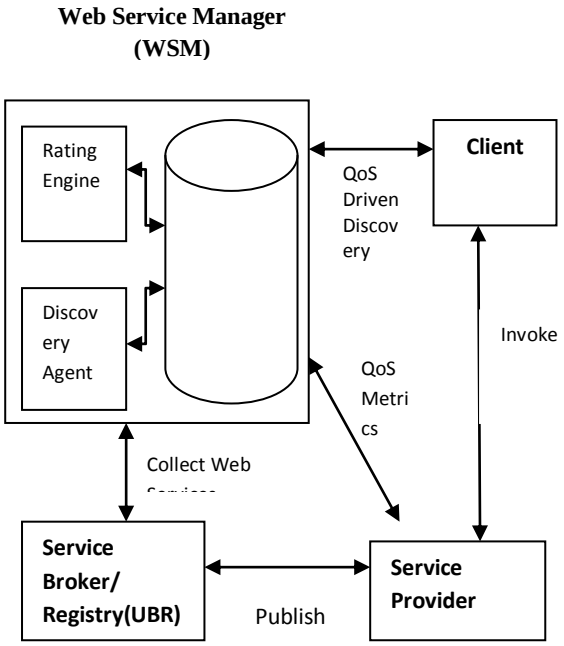

Figure 1: QoS based Discovery Framework

E. ***Rating Engine:*** A Rating Engine module within the WSM framework is responsible for measuring QoS information for the collected web services, and information is stored in the QoS database as shown in above Figure.1.

F. ***Discovery Agent****:* The Discovery Agent module within the WSM framework work as a crawler for discovering services from UBR on client request.

G. ***QoS Parameters:*** QoS Parameters [5] help to determine which of the available web services best and meets client requirements. Because of their significance, we selected the following QoS Parameters –

- ***Response Time (RT):*** The time taken to send request

and receive a response (unit :milliseconds).

- ***Throughput (TP):*** The maximum requests that are handled at a given unit in time unit: requests/min).

- ***Availability (AV):*** A ratio of the time period when a web service is available .(unit:%/3-day period).

***H. The Decision tree based classifier*:** We have used a decision tree based classifier[15] which uses the above mentioned QoS Parameters to classify services in five different classes as per the value of overall QoS parameter designated 'Q' where the value of Q is as defined as:
{Q | 0<=Q <=15}

Here any QoS parameter can get at the maximum value 5.0 or minimum of 0.0 and hence the total maximum value of first three parameters is 15.0 while the minimum is 0.0. The first splitting attribute chosen for the decision tree is response time as it is a continuous attribute and we have done a discretization of the same for a two way split and for the test data its GINI index was maximum. At each splitting the value of Q gets added upon .

To calculate effect of overall QoS parameters:

if Response Time < 5.0 ms then Q =5.0 else Q = 3.0.(The value is calculated as per actual percentage mapped upon 5.0)
Then we have the throughput as splitting attribute which again is a continuous attribute but has been descretized for a two way split where :
If Throughput > 30 then Q = 5.0 else Q = Value
The third splitting takes place on Availability where if Availability > 95% then Q= 5.0 else Q = Value (The value is calculated as per actual percentage mapped upon 5.0)The final classes are based on overall value of Q as per the given algorithm :

If Q > = 13 then service belongs to Class A
Else If Q <=12 and Q > = 10 then Class B
Else If Q <=09 and Q > = 07 then Class C
Else If Q <=06 and Q > = 04 then Class D
Else Class selected is default.
Endif

As an example if a services has response time of 6.0 ms , its will be Q = 4.2 and if its throughput is say 30 then its overall Q = 4.2+5.0=9.2 and suppose its availability is 80% so Q =9.2+4.4=13.6 and hence it will be classified in Class A. The Overall architecture for discovering a service using QoS parameters is shown in Fig 2.

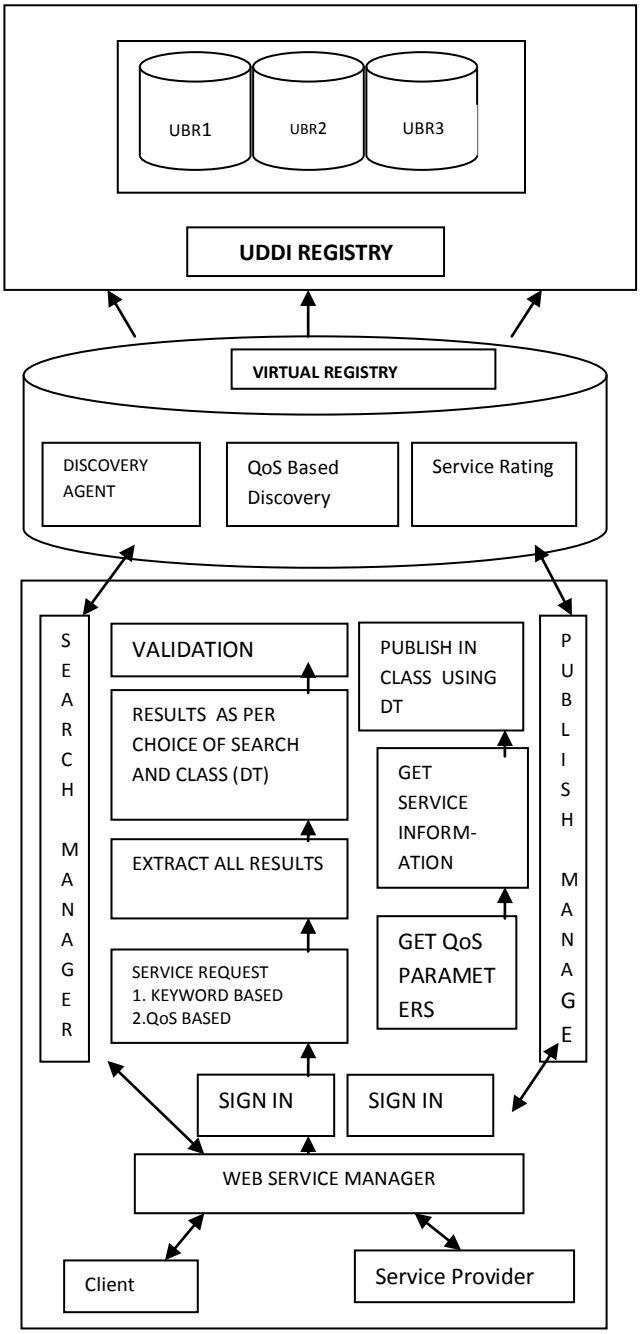

Figure 2: Discovery cum publishing Engine Architecture

When the publisher publishes any service into UBR , the publisher publishes or claims some initial values for the QOS parameters .These values are then compared with already stored threshold values in Web Service Manager's QoS data base. If the values claimed by the provider are appropriate for a particular type of service then that service will be published in a class of services based on class predicted by decision tree.[15].

While discovering a best available web service, client can enter some keywords or service name with some desired QoS parameters. If the request of the client is purely technical and can be satisfied by parsing WSDL file then it can be directly invoked. However If the client enters search string, then from search string, Discovery Agent module will first Discovers service type related to those keywords and then according to

standard threshold values must select most appropriate service which satisfies the client's requested QoS parameters. However for both of these discoveries we have used an improved web service discovery engine [16] which can crawl to multiple registries and discover the best services. The final offering of services will be arranged on the basis of rank. Ranking for services has been done on the basis of QoS parameters, number of hits as well as user's feedback.

*1) **Dynamic Deployment of Services:***

For the ease of consumers we are providing an interface through which a service provider can deploy his services on WWW. For this process consumer has to provide Web service name and service java class file.

```
Step 1: Sign in/Sign up Deployer
Step 2: If you want to Deploy Services
          Enter the name of the Service and select
          WebService/ Class file.
        Else If you want to Undeploy Services
          Select the name of the service which you
          want to Undeploy.
Step 3: STOP
```

*2) **Dynamic Publishing Of Services:-***

For this process consumer has to perform following steps:

```
Step 1: Sign in/Sign up Publisher
Step 2: Ask for routine Publishing.
Step 3: If yes
            Enter required information of QoS
            Parameters for a Service.
            If yes
                //Publisher fills the information
                  And publishes a Service in a
                  particular class based on
                  decision tree based classifier//
                //These QOS Parameters can be
                  verified by a third party as
                  proposed in[7]//.
            Else
                Go to the Previous Page
        Else
            Go to the Home Page
        End If
Step 4: STOP
```

*3) **Algorithm to Search a Web Services:***

```
Step 1: Start

Step 2: Select discovery on functional or
non-functional parameters (QoS Parameters).

Step 3: Input keywords or QoS parameters.

Step 4:If Search on Qos parameters selected, then
Check for the class of QoS as predicted by
the decision tree.
If result found ,then
(a) List highest ranked services based
on number of hits, QoS parameters,
and consumer's Feedback.
(b) After using service take consumer's
feedback.
Else
Continue or go to home page.
End If.
Step5: If Search on functional Parameters
selected, then
(a) Search services based on keyword.
List higher ranked services based on
consumer's feedback & number of hits .
(b) After using service take consumer's
feedback.
End If
Step 6:If required Service not found for given
keywords.
Continue or go to home page.
End If.
Step 7: Stop.
```

4. TESTING

For testing proposed design and implementation of framework, we have deployed simple products web services on multiple UBRs. For testing number of categories have been fixed to 7. We have published various services under various categories through publishing module as per design of the system. The domain for testing has been taken as e-shopping. We had used 14 UBRs and 191 different services in 7 categories and the tabular result for relevancy is shown in table1.

Table 1: Search results

Category	Number of services	QoS based discovery No.of services in the search result	Keyword based discovery No.of services in the search result
grocery	34	12	28
shoes	17	06	12
electronics	25	11	20
Books	36	10	23
computers	33	19	21
sports	25	12	19
others	21	9	13

5. CONCLUSIONS

In this paper we have presented effective and fruitful discovery of web services based on QoS parameters through a specialised search Engine. This proposed solution provides an effective web service discovery model based on ranking of services ,taking both users as well as providers into consideration and distinguishes the appropriateness of functionally similar Web services to a user. Further we have used a decision tree based classifier to classify the services based on their QoS parameters. In the proposed framework, new selection algorithms have been proposed and also has the facility so as to not to consider the services with lowest or unacceptable feedback . Further since there are no changes to be done in Existing UDDI Data structures or its existing API ,this solution is simple to implement. The proposed layer simply eases of the searching as well as publishing process in a UBR. Further a method for applying Decision tree based classification on the collected QoS values further enhances the quality of results provided to a client and affects the search when we have thousands of services with the same functionality are present in UBRs

6. REFERENCES

[1] Web Services

http://www.w3schools.com/webservices

http://en.wikipedia.org/wiki/Web_service

[2] WSDL

http://www.w3.org/TR/wsdl

[3] UDDI, 2003] UDDI Version 3.0.1, available at

http://uddi.org/pubs/uddi-v3.0.1-20031014.htm

http://www.theserverside.com/tt/articles/article.tss?1=Systinet UDDI

[4] SOAP http://www.w3.org/TR/soap12-part0/

[5] Eyhab Al-Masri, Qusay H. Mahmoud. "QoS-based Discovery and Ranking of Web Services" IEEE2007.http://ieeexplore.ieee.org/xpls/abs_all.jsp?

arnumber=4317873&tag=1

[6] Nirmal Dagdee, Urjita Thakar, Vandan Tewari, Appropriate Service Discovery from UDDI through application of Data Mining Techniques, International Conference on Emerging trends in Computing (ICETiC'09), January 08-10, 2009.

[7] Vandan Tiwari, Dr. Nirmal Dagdee , Dr. Aruna Tiwari, Divesh Dixit, "Extended SOA To Enable Web Service Discovery on Non Functional Parameters Accepted for publication in IJECCE, April 2012 issue.

[8] Zafar U. Singhera, "Extended Web Services Framework

to meet Non-Functional Requirements" 2004 Symposium on Applications and the Internet-Workshops (SAINT 2004 Workshops) http://doi.ieeecomputersociety.org/10.1109/SAINTW.2004.1268656

[9] Eyhab AI-Masri and Qusay H. Mahmoud, "Discovering the best Web Service", WWW 2007, Canada, 2007.

[10] Ziqiang Xu, Patrick Martin, Wendy Powley and Farhana

Z.2007,"Reputation Enhanced QoS-based Web services Discovery", IEEE International Conference on Web Services (ICWS 2007), IEEE 2007.

[11] Demian.A.D'Mello and V.S.Ananthanarayana, "A QoS Model and Selection Mechanism for QoS-Aware Web Services", Proceedingsof the International Conference on Data Management (ICDM2008), 2008.

[12] M. A. Serhani, R. Dssouli, A. Hafid, and H. Sahraoui, A QoS broker based architecture for efficient web service selection",Proceedings of the IEEE International Conference on Web Services(ICWS'05), IEEE 2005.

[13] J. Hu,, C. Guo, H. Wang, and P. Zou, "Quality Driven Web Services Selection", Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE'05), IEEE 2005of the 1st International Web Services Quality Workshop – WQW 2003 at WISE 2003, Rome, Italy, pp.1-10, 2003

[14] P. Rompothong and T. Senivongse. ,"A query federation

Of UDDI registries". In Proc.. of the 1st international Symposium on Information and Communication Technologies.ISICT '03 Trinity College Dublin, 2003. Pages 561 – 566

[15] Tan, Steinbach, Kumar, " Introduction to Data Mining" www.users.cs.umn.edu/~kumar

.../chap4_basic_**classification**.pdf

[16] Tewari, V.; Dagdee, N.; Singh, I.; Garg, N.; Soni, P.; , "An Improved Discovery Engine for Efficient and Intelligent Discovery of Web Service with Publication Facility, "Services - II, 2009. SERVICES-2 '09. IEEE World Conference on Web services.vol, no., pp.63-70, 21-25 Sept. 2009