An Improved Discovery Engine for Efficient and Intelligent discovery of Web Service with publication facility

Vandan Tewari
Lecturer, SGSITS, Indore
vandantewari@gmail.com

Nirmal Dagdee
Principal, S D Bansal College of Engineering, Indore
nirmal_dagdee@rediffmail.com

Inderjeet Singh*
inderjeet_kalra10@yahoo.com

Nipur Garg *
garg_nipur15@yahoo.co.in

Preeti Soni*
preeti_soni25@yahoo.co.in

Abstract

The increasing demand of industry for enabling Business to Business and Application-to-Application communication has led to a growing requirement for Service Oriented Architecture. Web Services are based on Service Oriented Architecture which enable 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. This paper proposes design of A Discovery cum Publishing Engine for web service discovery with refined searching mechanism which uses service rating techniques for efficient and effective web service discovery within optimum response time. We have used Data Mining Techniques to narrow down the search space in UBRs. Further the proposed engine has an ability to publish or search web service across multiple UBRs. In addition to this an extended design of service registry is proposed to store service rating data along with the service information. The Engine publishes the web services in UBR by following a classification scheme and performs a validation test on discovered web services. Service reviews and rating have been utilized to help a user for selection of appropriate service.

1. Introduction

1.1 Service Oriented Architecture

A service-oriented architecture (SOA) is the underlying structure supporting communications between services. SOA[1] defines how two computing entities, such as programs, interact in such a way as to enable one entity to perform a unit of work on behalf of another entity. Service interactions are defined using a description language. Each interaction is self-contained and loosely coupled, so that each interaction is independent of any other interaction.

1.2 Web Service

A Web service is a paradigm designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

1.3UDDI Business Registries

Universal Description, Discovery and Integration (UDDI) [3] is a platform-independent, XML-based registry for businesses worldwide to list themselves on the Internet. UDDI was originally proposed as a core Web service standard. It is designed to be interrogated by SOAP[2] messages and to provide access to Web Services Description Language documents describing the protocol bindings and message formats required to interact with the web services listed in its directory.

1.4 Data mining

Data mining, the process of extracting patterns and knowledge hidden from large volumes of raw data, includes techniques such as classification, clustering etc.

* Student MCA final year SGSITS, Indore.

978-0-7695-3812-9/09 $26.00 © 2009 IEEE
DOI 10.1109/SERVICES-2.2009.21

1.4.1Classification

Classification is the task of assigning objects to one of the several predefined categories. Often training set is used to develop the specific parameters required by the technique.

1.4.2Clustering

Clustering also known as unsupervised classification is a process of finding groups of objects such that the objects in a group will be similar or related to one another and different from or unrelated to the objects in other groups. Objects can be clustered according to the distance calculation of defined criteria.

1.5 Service Rating

Service Rating involves techniques of capturing and incorporating user reviews and rating on service based on certain criteria.

The rest of this paper is organized as follows: Section two describes some of the related work. In Section three our proposal for discovery and publishing scheme is discussed. Section four covers the description of architecture. Algorithm for discovery and publishing is discussed in Section five of the paper while results and test cases are discussed in Section six. Finally conclusion and future work are discussed in Section seven.

2. Related Work

In research conducted by SIIA , computer world & Systinet[4] which had been published in October 2002, it was found that among 800 developers and IT managers which were interviewed ,over 68% planned to deploy web services in coming year [4].This data shows that web services publishing is the first step to make the deployed services available to other users. However the publishing of web services is complicated & lengthy procedure for a service provider. Web Service discovery is next step towards the usage of SOA for business applications over internet and is still an interesting area of research. Many researchers have proposed discovering Web services through a centralized UDDI registry [5][6].Although centralized registries can provide effective methods for the discovery of Web services, they suffer from problems associated with having centralized systems such as a single point of failure, and other bottlenecks. Other approaches like [7] [8] have focused on having multiple public/private registries grouped into registry federations. But similar to the centralized registry environment, it does not provide any means for advanced search techniques which are essential for locating appropriate business applications. In Previous work where web services were discovered through search engines [9], web search engines treats web services and general web documents in the similar way for search criteria which results in irrelevancy in fetched information about any type of web services. Further [10] [11] have proposed to use a crawler dedicated for web services. Though it provides link to web services, but its response time is quite observable since it scans the whole data of UBR, whose growing size may make the search inefficient. We have tried to mitigate these problems in our proposed design of Discovery cum Publishing engine which utilizes Data Mining techniques to make the search intelligent and also uses a service rating technique to narrow down the search space. Further there is an active research going on semantic search of web services which have been enriched using semantics which enhances the metadata for a service and in term can be used for discovery of web services[19][20].

3. Our Proposal

A discovery cum publishing engine has been proposed with capabilities of using service reviews for publishing as well as discovery. We have used Travel web services to demonstrate the proposed design. For demonstrating the advantage of using service rating, a simple example of location based web services is used. The proposed design tries to ease the task of service provider as well as service consumer regarding publishing and discovery.

3.1Extended Service Oriented Architecture

An extended SOA architecture has been proposed so as to have a logical layer of discovery cum publishing engine that reside between the UBRs and client or providers.

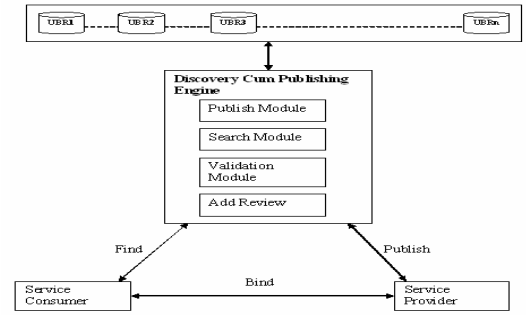

Fig 3.1 Extended Service Oriented Architecture

3.2Publishing web services through engine

To make a web service available to a user, it necessarily needs to be published in a UBR. The proposed design of engine eases the task of publishing a web service. The engine has very simple interface which requires only simple information to be furnished by the service provider. In the proposed system ***Publish Manager*** performs the task of publishing web service along with classification scheme and has been discussed later in Section four .

3.3 The proposed approach for Service Discovery

As the web services proliferate, size and magnitude of UDDI Business Registry is likely to increase. Further UBR registration is voluntary for service providers and therefore at any time, any service in UBR can easily become passive. This increases the passive searching data in UBR that increases passivity in search results which increases the search response time. To overcome this deficit we propose to validate the wsdl [12] for an active web service and automatically refresh the UBR to reduce irrelevant population by removing passive entries. This task is performed by the Validation Module and Deletion Module. Their functioning in detail will be discussed in Section four of the paper. Further the search mechanism is now refined by using data mining techniques such as classification and clustering. Let us here consider a scenario where a person visits a new city. Assuming that he doesn't have any prior reservations except hotel services, he now needs certain services such as restaurant services, taxi services, etc. These services obviously should be closest to his residence of stay or location. These are examples of services which have a parameter called location of most significance. In such a case we have chosen a parameter called location to classify the services. In ***Publish Manager's*** functionality it has been shown that a service is published under a classification scheme i.e. this Manager predicts the class (Category) under which it should be published while taking service information from provider such as service category, service description along with location & metadata. Further Data Mining techniques have been used to make search intelligent which decreases the runtime by restricting the search to a smaller subset or cluster and hence speeds up the search. Clustering techniques have been utilized to identify similar services and helps engine to suggest second most suitable services to client. The proposed strategy to cluster is based on creating cluster of similar types of service classes. A distance based clustering technique has been used to show the results. In the proposed design ***Cluster Module*** performs all these above mentioned tasks.

3.4Application of service rating

Service rating technique allows service consumers to review and rate services based on a set of criteria. Service rating provides an overall evaluation for a service while service review provides detailed evaluations in text format for used services. The reviews and rating serve as references for other service consumers to inquire. Before delving into service rating technique in detail, let us take an everyday life example to understand its utility in service selection process. When people are planning to purchase a product or service like planning to go to a restaurant, reviews and rating from other customers generally play a very important roll in their decision. Experience of previous user plays a vital role in selection of services by other users. We are proposing an extended service registry design with a template to capture service reviews and rating along with service name. The schema design of this template table is as follows.

Sname	Person Name	e-id	Review	Rating

The proposed engine presenting the active web service links to the client for a query request add this rating and reviews to help user to choose an appropriate service. For service rating we have

considered an equal share of user reviews as well as frequency of usage of a service. Further for generalizing the service rating algorithm, QWS parameters such as service availability and response time can be considered.

4. Proposed Architecture of the engine

The proposed architecture consists of set of modules that are:

4.1 Dynamic IP Module

This module will update the IP Table of Engine dynamically by the following strategy. It starts crawling on initial seeds and from each initial seed it finds out the IP Addresses of the service providers. From each provider it fetches the IP Addresses of UBRs in which they have published their other web services. These fetched IP Addresses have been compared with initial seeds. If any new IP Address is identified, it will be stored in its local IP Table, rest will be overlooked. In this way redundancy of IP addresses can be avoided and next time engine will crawl on the new IP Addresses fetched along with the previous IP Address List.

4.2 Publish Manager

The ***Publish Manager*** consists of set of modules that provides facility to service provider to publish their web services in to UBR with the help of simple interface. The provider furnishes few details like IP Address of UBR in which it wants to publish its Web Services, Service name, Organization name, Access Point URL and Location that acts as an attribute for applying classification in UBR. Also it uses this parameter to store details of web service information into selected UBR in a particular class to which that service belongs.

4.3 Validate WS

This module validates web services by identifying whether it is active or passive web service. The module has functionality of parsing the WSDL document of web services and it returns the name of the method if service is active else it will return null which indicates that the service is passive or not available for binding.

4.4 Delete Module

This module takes service key as input parameter, on the basis of which it will store details of the passive web services into cache and delete them from the respective UBR.

4.5 Create Cluster

This module accepts the location/class parameter that acts as a centroid for the distance based clustering algorithm and on the basis of minimum threshold value it will find out classes nearer to the centroid and put them in to same cluster and then return the elements of a cluster.

4.6 Search WS in Cluster Module

This module searches the web services in all the classes of the cluster from UBRs.

4.7 Add Review Module

This module accepts the reviews given by the user and stores it in to cache so that upcoming user can choose the best possible services.

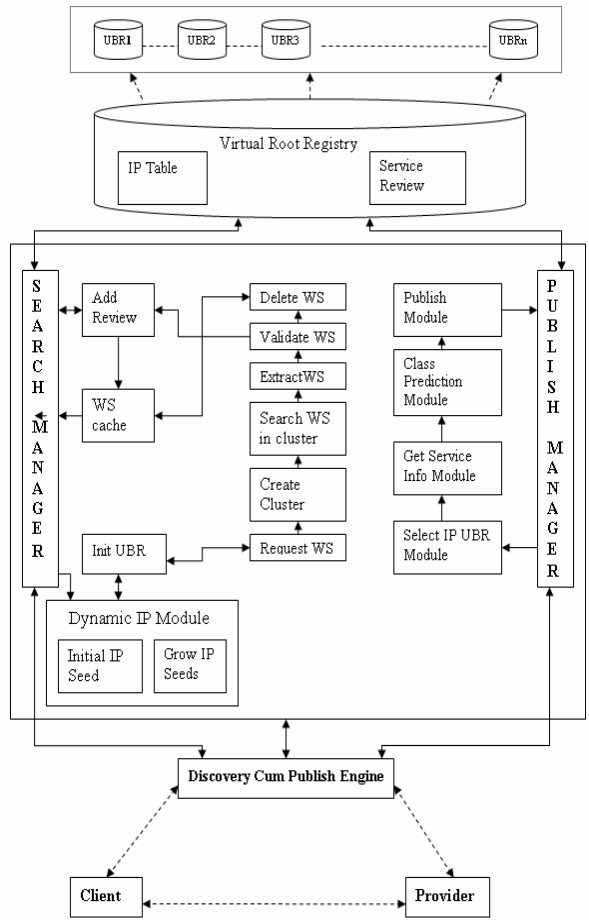

Fig 4.1 Proposed Architecture of engine.

5. Algorithms used for demonstrating the proposed work

5.1 Publishing Algorithm

Step 1: Start
Step 2: Select UBRs IP Address where publishing is required.
Step 3: Accept details of web services along with its location that acts as a predefined class from the service provider
Step 4: Classify the web service based on its location which acts as a class with verification on fields.
Step 5: Store the details of web service information into selected UBR in a particular class to which that service belongs.
Step 6: End

5.2 Searching Algorithm

Step 1: Start
Step 2: Enter keyword for which services to be searched (for ex. Travel).
Step 3: Select ***Location*** of service (Let it be denoted by a class and serve as centroid for cluster module).
Step 4: Initialize IPTable for initial seeds of UBR
Do
Step 5: Visit each seed and find out IPAddresses of service providers to fetch the IP Address of UBRs in which they have published other web services.
Step 6: Compare these entries with initial IP seeds if any new IP found populate the engine IP table with newly found IPs.
Step 7: Call Cluster module to create cluster based on location attribute entered that acts as a centroid.
Step 8: If (Location is not chosen)
Treat all classes in a single cluster.
Else
Treat location as Centroid
If (Class distance < min threshold)
Put the class in same cluster.
Step 9: For each class in cluster, fetch all services belonging to the same class.
Step 10: Parse WSDL document against access point URL for each discovered web services i.e. validate web service.

Step 11: If Web service is Active
Store it locally
Else
Fetch service Key against that access point URL from UBRs and pass it to delete module that store it locally for future use and delete the web service from respective UBR.
Until all IP Seeds are visited from UBR crawl queue.

Step 12: Add Service Reviews to each service of active service list which has been stored locally from virtual UBR on which engine resides.

Step 13: Display the list of web service to the end-user in the form of hyperlink along with reviews to show binding instances of services.
Step 14: If User binds the service
Execute the service and display the output of service to end user. And ask whether he wants to write review or rate a service.
Else
Accept details of user along with comment and rating to the service and store these details to extended service registry structure.
Step 15: END

6. Results and Test cases

For testing proposed design and working of engine, we have deployed simple travel web services on multiple UBRs. Number of classes has been fixed to the nineteen Locations of Indore city that are well distributed geographically across the city. We have published 42 services under various classes [Location] through publishing module of designed system by following interface

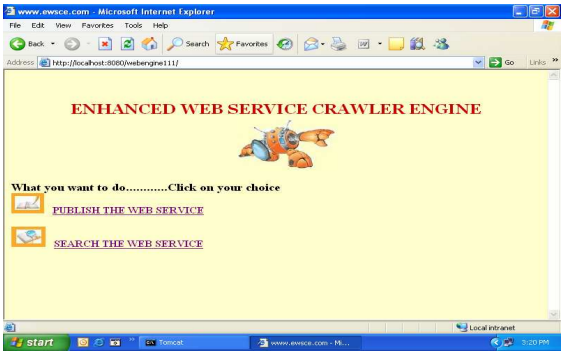

fig 6.1 Engine's first interface

Fig 6.2 Form for publishing web services

List of number of published web services at each location is as follows.

Table 6.1 List of Class along with published web services

Location /Class	Published WS	Location	Published WS
AnnapurnaRoad	2	Pardesi Pura	2
HukumchandMarg	2	RajendraNagar	2
Indore Ho	2	Rajwada	2
Khajrana	3	Shivji Nagar	1
Khatipura	2	Southtokoganj	3
LaxmibaiNagar	1	VallabhNagar	3
Malwa Mill	1	Vijay Nagar	3
M G Road	4	Y N Road	2
Navlakha	2	YashwantNRoad	2
Old Palasia	3	**Total**	**42**

Clustering of these classes has been done based on distance parameter between these locations. If user searches for Travel web services at a location like Annapurna Road as shown in fig 6.3.

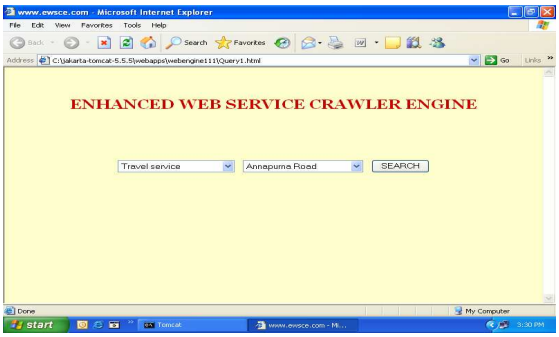

Fig 6.3 Request for travel web service search.

Engine returns six search results out of which two travel services are located exactly at Annapurna road, next two at Hukumchand Marg and next two are at Rajendra Nagar since Hukumchand Marg and Rajendra Nagar class are in same cluster when centroid is Annapurna road.

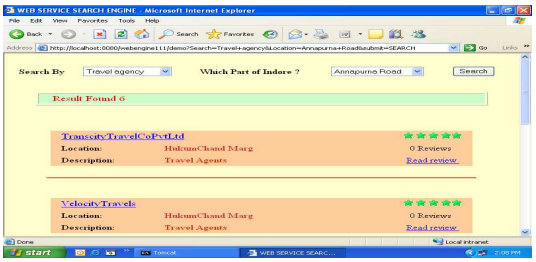

Fig 6.4 List of search result for Annapurna Road

Fig 6.5 continued fig 6.4

Instead of this if user does not chose a location i.e. does not mention class even though he is interested in travel web services located at

Annapurna Road, engine returns 42 results for this request as shown in fig 6.6.

Fig 6.6 search results when class is not chosen

These intriguing results on small data set shows that amount of data need to be scanned reduce significantly if we use a classification scheme. By employing clustering, search results will be supplemented with next best result for chosen location. This ensures that search result will not be null even though if none of the travel service is available at Annapurna Road by manually scanning the 42 search results. The technique which we have implemented automatically floats the travel web services at desired location on top of the search results.

7. Conclusion and future work

On the advent of service-oriented computing, exploring appropriate business applications published as web services in UBRs is mainly done by scanning through UBRs. As the number of web services increases, the success of businesses will depend on publishing a service, service discovery and response time while searching multiple UBRs. In this paper, **A publishing cum discovery engine** has been presented for the purpose of effective and appropriate discovery of web services within acceptable response time. The presented searching scheme incorporates intelligence based refined searching mechanism which hire data mining techniques like classification and clustering to narrow down the search space of UBR hence reduces the search response time. Further the proposed engine assures service availability since it performs validation test on searched services corresponding to a query request hence removes passivity from search result. As a result engine saves cost in terms of processing power and time. Our system has an edge over the other existing system in number of ways, first of all it has employed clustering data mining technique which suggest second best appropriate service to client and assures that search result will never be null, secondly the virtue of search result for a web service query request has been improved since it is now endorsed with user reviews and ratings. The existing service registry has therefore been modified to include service reviews and rating details along with service information. For demonstrating the use of data mining techniques, a specific context has been chosen. But in future service rating can essentially include contribution of QWS parameters such as service availability and response time by observing number of binding instances and actual time taken to receive a response from service provider. In future, if some classification scheme may be proposed to be included in UDDI standards, over the existing classification on geographical locations, it may be possible to generalize this work to other domains too. Such is a work in WSMO [18] where semantics of services have been included on which classification can be proposed.

Acknowledgement

We sincerely thank all the learned reviewers who have largely contributed to this paper in terms of contents and future work.

8. References

[1] SOA http://searchsoa.techtarget.com/sDefinition/0,, sid26_gci 929153, 00.html

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

[3] UDDI Version 3.0.1, available at http://uddi.org/pubs/uddi-v3.0.1-20031014.htm

[4] http://www.theserverside.com/tt/articles/article. tss?l=SystinetUDDI

[5] M Paolucci, T.Kawamura, T.Payne and K Sycara. Semantic matching of web service capabilities. In *proceedings of ISWC*, pp1104-1111, 2002

[6] U Thaden, WSiberski and W.Nejdl.A semantic web based peer to peer Service registry network. In *Technical report, Learning Lab Lowry Saxony*, 2003

[7] K.Sivashanmugam, K.Verma and A Seth.Discovery

of web services in a federated environment. In *proceedings of ISWC*, pp270-278,2004
[8] C Zhou, L.Chia, B.Silverajan and B.Lee.UX-an architecture providing QoS-aware and federated support for UDDI.In *Proceedings of ICWS*, pp.171-176, 2003
[9] E.Al-Masri, Q.H.Mahmoud, Discovering Web Services in Search Engine, WWW 2007, May8-12, 2007, Banff, Alberta, Canada.
[10] E.Al-Masri and Q.H. Mahmoud, Crawling Multiple UDDI Business Registries, Proc. *16th Int'l World Wide Web Conf.*, ACM.
[11] E. Al-Masri, and Q. H. Mahmoud, WSCE: A crawler engine for large-scale discovery of web services, ICWS pp.1104-1111, 2007.
[12] Web Services Description Language (WSDL) 2.0, http://www.w3.org/TR/wsdl20/
[13] XML http://www.w3schools.com/xml/default.asp
[14] K. Sivashanmugam, K. Verma, and A. Sheth, Discovery ofWeb Services in a Federated Registry Environment, Proceedings of IEEE ICWS, pp. 270-278, 2004.
[15] Eyhab Al-Masri and Qusay H. Mahmoud, Investigating Web Services on the world wide web. In *proceeding of www2008*, pg.no.795-804. 2008
[16] Yan Li, Yao Liu, Liangjie Zhang, Ge Li, Bing Xie, Jiasu Sun, An Exploratory study of Web Services on the internet,. In ICWS 2007(IEEE).
[17] Developing java web services by Ramesh Nagappan, Robert Skoczylas, Rima Patel Sriganesh published by Wiley India (P) ltd pg. no. 201-269,493-561.
[18] Cristina Feier1, Dumitru Roman1, Axel Polleres, John Domingue, Michael Stollberg, and Dieter Fensel," Towards Intelligent Web Services: The Web Service Modeling Ontology (WSMO)" *, members.deri.at*/~michaels/publications/ Wsmo-icic.pdf
[19] Aabhas V. Paliwal, Nabil R. Adam, Hui Xiong," Web Service Discovery via Semantic Association Ranking and Hyper clique Pattern Discovery"**,** Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06) 0-7695-2747-7/06 $20.00 © 2006
[20] Le-Hung Vu, Manfred Hauswirth, Fabio Porto, and Karl Aberer," A Search Engine for QoS-enabled Discovery of Semantic Web Services**",** International Journal of Business Process Integration and Management (IJBPIM), Volume 1, No. 4, pages 244-255, 2006.