Dynamic Search and Selection Of Web Services

Sumathi[1],Niranjan N. Chiplunkar[2]
[1]Research Scholar
[1,2]Computer Science and Engineering,NMAM Institute of Technology,Karkala, Udupi, India
[1]pawarsumathi@gmail.com,[2]nchiplunkar@nitte.edu.in

***Abstract*— Web service technology promises to provide a dynamicintegration and interaction of heterogeneous systems, thereby to facilitate fast and efficient cooperation among the entities in cooperative environments. It is difficult to predict how services will be selected in future applications. This uncertainty is one of the greatest challenges and this problem is attempted to solve in this research.This paper demonstrates selecting composite and complex web services dynamically according to the user requirement using different search methods and WSDL (Web Service Description Language) processing. Methodology used for searching the requested terms is simple search, horizontal search, and vertical search for web services. Search precession is computed as support and confidence of the requested terms in the search results. Filtering of search results is done by selecting highly weighting matched factors to select suitable web services according to the user requests during run time.**

Keywords—WSDL, MMA, SWRL, WSDL-S

I. Introduction

Web services are loosely coupled, self- contained software modules that can be accessed programmatically using existing Internet technology, found and assembled dynamically to serve a particular function, solve a specific problem or deliver a particular solution to a customer. Web service selection can be divided into two kinds, one is static selection and the other is dynamic selection[1]. Static selection is the web service selection that the designer selects the service according to the requirement of users and composes services according to a certain sequence in the design stage. Dynamic selection is the web service selection that the system automatically selects the service according to the requirement of users and produces a new web service at the run time.

Existing Web Service search methods classified into two categories. WSDL-based and Ontology- based. In ontology based method annotation have been made to reference to a domain ontology through the standard WSDL extension mechanism. Ontology based methods aims to provide 'semantically enriched' version of WSDL files in order to automate complicated tasks such as service composition.

Normally Web services are described by WSDL documents while semantic web services use Web Ontology Languages(OWL-S) as a description language. WSDL based discovery is most popular and supported by both industry and development tools [2]Service composition indicates the development of a complex application using several web services and it is related to the language for specifying business process behavior or WS-BPEL. Now the industry standard such as WSDL, UDDI, WS-BPEL can provide the powerful technology support for web service composition. But authors [6] in their paper point out that more than 53% of UDDI Business Registry (UBR) are invalid where as 92% of web services cached by search engines are valid and active. Therefore in this research we used search engine instead of UDDI to retrieve WSDL.

Search results may partially match the search terms entered by the user with the <definition>/ <service> or <definition>/<service>/<operation> elements of WSDL. This partial match is used to get information about the availability of the web services. If services are available, it is required to perform vertical and horizontal search of web services. The search results of high precession are used to get composible services and also to get complex web services according to the user requests. Then suitable web services will be selected from similar WSDL records in the search results. This will be done by selecting web services which score maximum weight in WSDL weight matrix. The sequence of web services selected by these processes gives functionally suitable web services which are used to satisfy user requests during run time.

II. RELATED WORK

Chen Wu and Elizabeth Chang [3] indicated that Public UDDI Business Registry - the primary service discovery mechanism over the Internet has been shut down permanently since January 12, 2006 due to several reasons. This has made the most important public Web service discovery mechanism missing from the Web Services Community. Therefore we are not using UDDI to search for WSDL.

Noh-sam Park [4] in his paper explained that consumers search web services with UDDI and manually access the web services that appear in the result. The pitfalls are that UDDI search results only provide specifications for registered web services and cannot express what consumers really want. Also it is impossible to know the states of registered web services.

In the existing paper[5] authors given an approach for selection of web services. In their paper they considered only input/output parameters of WSDL to match with user request, where as in this research we are comparing input, output, pre-condition and effect elements against user request, as well as a dynamic search of web services and selection as given in the section III.

In Paper[10] Service mining is used to discover and compose web services which make use of the service usage logs. But our research does not depend on any template information prior to compose because of dynamic discovery. This justifies the necessity of searching of WSDL using

ISBN No. 978-1-4799-3914-5/14/$31.00 ©2014 IEEE

search engines instead of UDDI(Universal Directory Description Interface).

The paper [11] the discovery of the atomic services that take part in the composition, are significantly facilitated by the incorporation of semantic information. OWL-S web service descriptions are transformed into a planning problem described in a standardized fashion, while semantic information is used for the enhancement of the composition process. A problem is that OWL-S concentrate on the description of static information. This contradicts the demand of a flexible description of innovative web service in the dynamic nature of eBusiness. When large scale web services are available an innovative dynamic structured integration is required. Therefore in this research we are not using ontologies for getting semantic information. Instead of that we are matching WSDL elements semantically with user requests using MMA(Match Making Algorithm).

III. Dynamic Search of the Web Services

The proposed research dynamically searches the required web services and selects the services according to the user requests. User requested functions may not be available as single web service or it may be available as composite web service.

[16] As a first step availability of requested terms as a single web service or as composite web service is need to be checked. To get information about availability of requested terms as web services, we need to search for requested terms using search engines and should apply following formulae (1) and formulae (2) to find search precession using support and confidence of the requested terms in the search results.

Search Precession of requested term in the search result and output element of web service is computed as

support $s(t1) = P(t1) = \|S_{t1}\| / \|S\|$ -----(1)

confidence $c(t1) = \|So_{t1}\| / \|S_{t1}\|$ ----------(2)

………

………

support $s(tn) = P(tn) = \|S_{tn}\| / \|S\|$

confidence $c(tn) = \|So_{tn}\| / \|St1\|$

Where t1,t2..tn are the requested terms.

$\|S_{t1}\|$ indicates number of services contains requested term, $\|So_{t1}\|$ indicates number of services contains requested term in the output element and $\|S\|$ indicates total number of web services which contain requested term in any element of WSDL in the particular search result. In formulae(1) support greater than zero indicates availability of requested term in any element of web services. In formulae(2) confidence greater than zero indicates requested term is available in the output element of the web services

The search results which have high confidence are considered and these service's input elements are extracted. If the requester's input element doesn't matches with the extracted input element of the service then it is required to search for the service which gives that input as output of particular web service which results in horizontal search. If requested term is result of execution of many services concurrently then it results in vertical search.

A. *Simple Search*

Another type of search is simple search that searches for requested input/output terms and support/confidence of this search is given below.

$s = P(tin, tout) = \|S_{tin}, St_{out}\| \ / \ \|S\|$ ---(3)

$c = P(tin, tout) = \|S_{tin}, S_{tout}\| \ / \ \|S_{tin}\|$ ---(4)

Here $\|S_{tin}, S_{tout}\|$ is total number of services that satisfies input, output parameters of requested terms in the input or output of operation elements of web services. In formulae(3) $\|S\|$ is total number of web services in the search results that contain requested terms in any elements of web services. In formulae (4) $\|S_{tin}\|$ is total number web services which matches input term of the request with only input element of the web services. Here 100% confidence indicates web service meant for the requested service. Zero support and zero confidence indicate unavailability of web services in the search result.

Now the search results which satisfy formulae (3) and formulae (4) in the particular search in different services with more than threshold value of support and confidence indicates the necessity of composition. We need to process those web service's WSDL records. In these WSDL records, the WSDLs which gives the required output element is compared for its input element with requested input parameter. If input element of this web service does not match with requested input, then it is concluded that more than one web services need to be composed for this request and shows the necessity of horizontal or vertical search of web services for those input/output elements.

B. *Vertical Search*

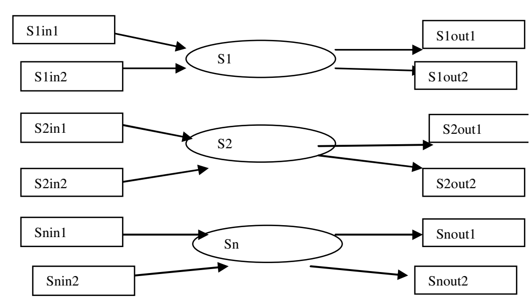

Figure 1 Vertical Search

If all input parameters of a request is collectively available in the set of inputs of different services and all output parameters of the same request are collectively available in the set of output of different services then those services need to be considered as list of sub-requests which collectively satisfies the user request. Therefore these services need to be searched and this type of searching is called as vertical search.

Support and confidence of vertical search is computed in the formulae 5 and formulae 6.

Supp= || Sxt_in1 $\subseteq$ S1out U Sxt_in2 $\subseteq$ S2out U …Sxt_inn $\subseteq$ Snout || / ||S1..Sn|| -------(5)

Conf = || Sxt_in1 $\subseteq$ S1out U Sxt_in2 $\subseteq$ S2out U …Sxt_inn $\subseteq$ Snout || / ||Sout1..Soutn|| ----(6)

If the extracted input elements from required web services exists in the subset of output elements of different existing web services then find the support and confidence of existence.

Here Sxt_in1.. Sxt_inn are extracted input parameters of required and available collection of web services and S1out, S2out….Snout are output elements of retrieved collection of web services that should be executed parallel.

C. Horizontal Search

Consider the Request R1, if its input parameters Rin1, Rin2, Rinn are matched to the input parameters S1in1, S1in2.. S1inn of the service S1 and Request's output parameter R1out1, R1ou2..R1out$_n$ matches to the S4out1, S4out2…S4outn of the S4. Then we need to find S2 and S3. This type of search is called horizontal search. Algorithm for horizontal search is given here.

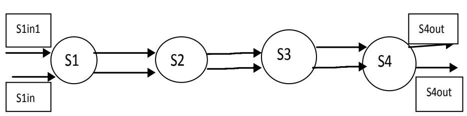

Figure 2 Horizontal Search

The results of horizontal and vertical search may give different web services which need to be executed in the sequential order. The set of web services which are resulted from different *horizontal* and *vertical* search are checked for their support and confidence of input/output terms with respect to their prior and next services as given in *formulae(7)* and *formulae(8).*

$S = P(S_{xtin}, S) = Sup= \| S_{xtin} \| / \| S \|$-----------(7)

$C= P(S_{xtin}, S_{out}) = \| S_{xtin} \| / \| S_{out} \|$-------------(8)

In formulae (7) $\| S_{xtin}, S_{out} \| / \| S \|$ indicates number of web services which matches extracted input element of the web services to the output element of particular web service. $\| S \|$ indicates the total number of web services in the particular search which contain S_{xtin} terms in any elements web services in the search results. In formulae (8) $\|$Sout$\|$ indicates the total number of web services which satisfies the extracted terms in the output elements of web services.

If confidence and support are higher in particular search method than other methods then those web services WSDL records are used to compute WSDL weight matrix.

C. Selection of suitable web services

Horizontal and vertical search may results into search of many similar web services. These similar web services need to be processed with respect to their WSDL records to get most suitable web service which matches to requested terms. Suitability of services will be found by assigning suitable weights according to the matching factors of requested parameters with functional WSDL parameters using MMA algorithm. During matching of WSDL elements with requested parameters, Exact match of functional parameters with user requested functions are be considered with highest weight, Plugin with next considerable weight as shown here.

Exact> Plugin> Subsume > Sibbling > Fail

Among similar services, a maximum matching service (which has scored more weight) is considered as suitable web service. If more than one services matched with user requested parameters then their quality of service parameters are considered for selection of suitable web service which is not discussed here.

IV. EXPERIMENTS

In this section, we evaluated the effectiveness of our approach by conducting extensive experiments towards different types of search. All searches are performed on a PC with 2.2GHz Intel Pentium Duo2 CPU, 2048M of RAM, Microsoft Windows XP Operating System, NetBean IDE with WS-BPEL for invoking different services. The WSDL used by WS-BPEL are selected and filtered according to the requirement which is already discussed in the section III, IV, and V.

According to the search results <output> element of "GetTemperature" service matches to the requested term. But <input> element does not matches to the requested input term city_name. Because <input> of the "GetTemperature" operation is zipcode. Therefore <input> elements shows the necessity of getting zipcode of city which is resulted into horizontal search. Horizontal search results into the service which has operation "CityStateToZipCode" with <output> element zipcode for inputted city. This shows the necessity of selection of "CityStateToZipCode" operation prior to "GetTemperature" operation.

Figure 3 shows the result of search precession in terms of confidence and support of horizontal search with a composible web service for an appropriate user request (relevant). The precession of search in searching the web services towards requested terms in horizontal search is given in formulae(7) and formulae(8).

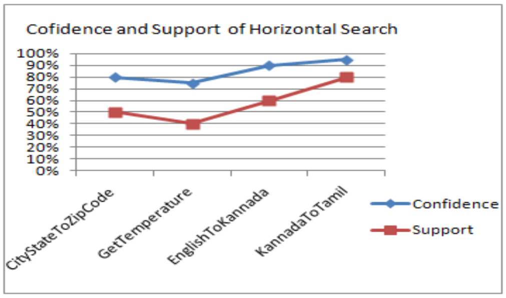

Figure 3 Confidence and support of horizontal search for temperature in native language request

In this graph we got 80% confidence and 50% of support for search of cityStateToZipcode which is composible with GetTemperature web service. Also we got 75% confidence and 40% support when we search for this composible web service GetTemperature.

Support of different requests calculated with simple search, horizontal search and vertical search is given in formulae (3), (5) and (7) and result is given in the figure 4.

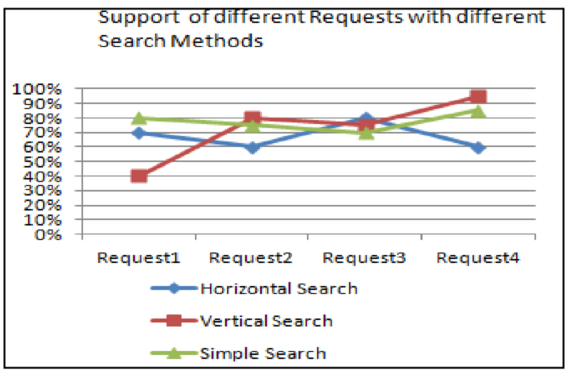

Figure 4 the Support of different searches for different search methods proposed in this research.

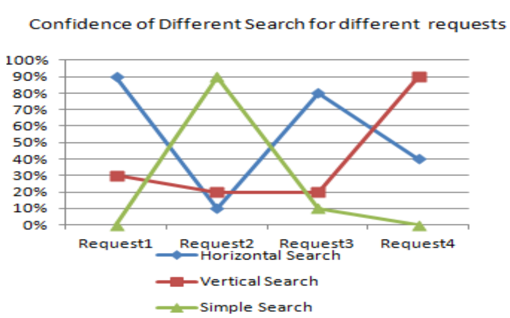

Figure 5 Confidence of different search for different requests

As shown in the figure 5 we got 90% maximum confidence of search for requested term in the <definition>/<service>/<operation>/<output> element of WSDL for the request2. It is observed that confidence of the simple search for request2 is more than all other types of searches for same requests. Confidence of vertical search is more for requests3 than all other searches. Also we can see that confidence of request2 in horizontal search is more than all other searches.

According to the support and confidence, the search results which gives more support and confidence are considered for selection of web services.

VII. ANALYSIS

In this research it is necessary to compute the search precession because we are considered the results of only high search precession, as well as we computed the WSDL elements matching factors weight to get suitability of web services with user requests. This filters our search space to get quality and security aspects of only highly functionally matching selected web services.

Authors David J. M. Cavalcanti and Fábio N. Souza in their paper [15] proposed a solution that allows changing the ranking strategy at runtime based on historical data of quality attributes of web services. The quality attributes are collected at runtime for all the web services that are to be selected which results in more time than our proposed model.

In paper[14] Jiaxing Shang and Lianchen Liu proposed a model for Web Service Composition based on Complex Networks, the atomic Web service is simply represented by a service node and two data nodes. However, real world Web services are usually more complicated. For example, the status of a Web service after its execution may change. Furthermore, this framework models the web services network as a static network, whose edges and nodes should not change. But real world Web services may change from time to time. Therefore our research provides a proper environment for dynamic search of the web services during each request of the services.

In this research model we are not using any static template for service composition, therefore search precession might change for same requests during different search iteration. Therefore from the result shown in figure 3, 4 and figure 5 we selected best result among different results.

VIII. CONCLUSION & FUTURE WORK

Unlike ontology based methods of semantic web services it is better to process WSDL of the web services to get its semantics, but this may result into slow process. A problem in ontology based method is they concentrate on the description of static information. This contradicts the demand of a flexible description of innovative web service in the dynamic nature of eBusiness, and it is solved in this research.

Also to overcome the drawback of UDDI based discovery, search engines are used even though this requires additional techniques to filter and find out appropriate web service according to user request.

In this research we are only given composition plan about sequencing/parallelizing the execution of different atomic web services to get complex web services. As a future work the repository of composition plan will be maintained and this repository is mined to get information about frequently used web services. We are also planning for dynamic XML mining to mine the difference between two similar request's composition plans which is required to get dynamicity of different composition plans of similar requests.

REFERENCES

[1] C Jaya Prakash, P Maruthurkarasi (Alias) Rohini, R Balaji Ganesh, V Maheswari ,Hybrid Reliability Model to Enhance the Efficiency of Composite Web Services 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology ICECCN 2013.

[2] Abdellah Kouider, Mohammed Erradi SIME Research Hamid Azzoune LRIA , A Discovery Service for Automatic Composition of Web Services Oriented-Agent, Research Laboratory, USTHB, Algiers, Algeria. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE INTERNET COMPUTING 2013.

[3] Chen Wu and Elizabeth Chang , Curtin University of Technology, Australia " Searching services on the web : A public Web Services discovery approach" 2008 IEEE Int. conf. on Internet Base System.

[4] Noh-sam Park, Gil-haeng Lee , Electronics and Telecommunications Research Institute, KOREA "Agent-Based Web Services Middleware" IEEE GLOBECOM.

[5] Festim Halili, Eip Rufati, Ilia Ninka, State University of Tetovo,, Macedonia , Styles of Service Composition – Analysis and Comparison Methods, Fifth International Conference on Computational Intelligence, Communication Systems and Networks, IEEE 2013.

[6] Guisheng Fan, Huiqun Yu, Liqiong Chen, Dongmei Liu , Aspect Orientation based Test Case selection Strategy for Service Composition, , Department of Computer Science and Engineering East China University of Science and Technology, Shanghai 200237, China, International Symposium on Theoretical Aspects of Software Engineering IEEE, 2013.

[7] Fangfang Liu, Yuliang Shi, Jie Yu, Tianhong Wang, Jingzhe Wu " Measuring Similarity of Web Services Based on WSDL" , 2010 IEEE International Conference on Web Services.

[8] Thomas Fischer, Johaness Rubland, Friedrich Schiller University Jena "Towards Knowledge Discovery in the Semantic Web" , MIKWI 2010.

[9] Ping-Ning-Nang "Introduction to Data Mining" Text Book.

[10] Wil van der Aalst, Senior Member, IEEE, Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior. IEEE transactions on services computing

[11] Ourania Hatzi, Dimitris Vrakas, Mara Nikolaidou, Nick Bassiliades, An Integrated Approach to Automated Semantic Web Service Composition through Planning IEEE transactions on services computing -2012.

[12] F. Lecue and A. Leger, "A Formal Model for Semantic Web Service Composition," Proc. Leading the Web in Concurrent Eng.: Next Generation Concurrent Eng., pp. 385-398, 2006.

[13] M. C. Jaeger and G. MuThl, "QoS-based Selection of Services: The Implementation of a Genetic Algorithm," In Torsten Braun, Georg Carle, and Burkhard Stiller, editors, Kommunikation in Verteilten Systemen (KiVS 2007) Industriebetrage, Kurzbeitrage und Workshops, March 2007, pp. 359-350, Bern, Switzerland, VDE Verlag, Berlin und Offenbach.

[14] Jiaxing Shang, Lianchen Liu, Web Service Composition based on Complex Networks pp 208-213 International Conference on Service Science - 2013

[15] David J. M. Cavalcanti and Fábio N. Souza Adaptive and Dynamic Quality-Aware Service Selection pp 323-327,21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing – 2013

[16] Prof. Sumathi, Dr. Niranjan N. Chiplunkar, Dr. Ashok Kumar A. "Dynamic Discovery of Web Services", IJITCS, ISSN: 2074-9015 (Online) in press, Mecs Publisher , Volume 6 – 2014.