I.J. Information Technology and Computer Science, **2014, 10, 56-62**
Published Online September 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2014.10.08

Dynamic Discovery of Web Services using WSDL

Prof. **Sumathi**
Canara Engineering College/Computer Science and Engineering, Mangalore, 574219, India
Email: pawarsumathi@gmail.com

Dr. **Niranjan N. Chiplunkar**
NMAMIT/ Computer Science and Engineering, Nitte, Karkala, India
Email: nchiplunkar@nitte.edu.in

Dr. **Ashok Kumar A.**
NMAMIT/ Computer Science and Engineering, Nitte, Karkala, India
Email: ashok@nitte.edu.in

Abstract**—** Web service technology promises to provide a ynamic integration and interaction of heterogeneous systems, thereby to facilitate fast and efficient cooperation among the entities in cooperative environments. With the surge of Service Oriented Architecture (SOA) and Web Services, service discovery has become increasingly crucial. Web service discovery mechanism retrieves relevant web services from the internet to serve a wide range of users such as service consumers, service developers, service deployers, and service brokers. This paper demonstrates the model for discovery of web services using WSDL processing system and comparison between WSDL(Web Service Description Language) and UDDI(Universal Description Discovery and Integration) based discovery system. The solution leverages WSDL retrieval according to the requested term using horizontal and vertical search and processing WSDL to find suitable web services to satisfy the user request during run time.

Index Terms**—** WSDL, UDDI, SOA, OWL.

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.

[15]The combination of Service Oriented Architecture and Web services will be used to provide a rapid integration solution that will quickly and easily align Information Technology investments and corporate strategies by focusing on shared data and reusable services rather than proprietary integration products.

In a web services scenario a web application sends a request message to a service at some URI using the SOAP(Simple Object Access Protocol) over communication protocols like HTTP, SMTP or FTP. The service receives the request message, processes it and returns a response message. The URI of the service might be known or might be discovered using UDDI.

Existing Web Service discovery 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. But ontologies are static in nature.

Therefore in this paper we have given an approach of web service discovery using WSDL based method. 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. WSDL based method is further divided into 1)Text based 2)Structure based 3) Semantics based methods.

Text based method is the most straight forward way to conduct Web service discovery. The most widely used text-based method is keyword matching built in the UDDI public registry.

UDDI API also allows developers to specify keywords of particular interest and it then returns a list of web services whose service description contain those keywords. Beyond the literal keyword matching, research in XML schema matching has applied various string comparison algorithms (e.g. prefix, suffix,edit distance) to match different spellings. This method is particularly useful for scientific Web services where many special terms, jargons, and acronyms are widely used in their service descriptions. For example, a bioinformatics Web service might have an operation called 'DNACombo', which shall be relevant to a user search 'DNACombination'. The literal keyword method cannot tell the equivalence between Combo and Combination.

Using UDDI API a link to WSDL will be found and WSDL elements texts are extracted, analyzed, and expanded. But due to pitfalls given in the[2][3][4] we are retrieving WSDL using search engines and will be processing those WSDL files.

This paper demonstrates how it can be achieved by discovering the web services using WSDL based methods which have proper operation of web services according to the user requests during run time. The organization of this paper is, section II gives literature review, section III gives overview of the model, section IV gives dynamic search of web services with WSDL records, section V gives processing of WSDL elements to interpret the functionality, section VI gives generating WSDL from java, section VII gives finding web services using UDDI and section VIII and IX gives results and conclusion.

II. Literature Review

In Paper[1] 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 discover because of dynamic discovery.

Chen Wu and Elizabeth Chang [2] 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.

Literature review justifies the necessity of WSDL based discovery of web services by using WSDL based method.

Noh-sam Park [3] 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.

The authors [4] 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. Search engines partially match the search terms entered by the user with the web service name, location, business, or tModel defined in web service description file to get the results back.

Lijie Wang [5] shown that Large propotion of web services on the Internet could not provide enough descriptions in their WSDL files. Therefore it is necessary to enrich descriptions for public web services by extracting useful information and provide semantics by data mining technique.

Pat. P.W. Chan and Michael R. Lyu [6] indicated that It is challenging to integrate the semantics of servicesin automatic service composition as currently, semantics are captured through manual service composition. Semantics are important in service composition. To compose services semantically it must be known in advance functionality of the service and constraint it enforces. This function will be known by processing the web service's <definition>/ <service>/ <porttype>/ <operation> element of the WSDL which is explained in the section V.

Jan Hendrik Hausmann [7] described that OWL (Ontology Web language) is first step towards the creation of semantic web enabled Web Services. A problem is that 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. When large scale web services are available an innovative dynamic structured integration is required.

Fangfang Liu [8] proposed a approach to find the similarity of services is evaluated by the traditional measures such as Jaccard and Euclidian combined with wordnet to increase the precision. But in practice most text descriptions of published web services contain much useless information unrelated with the function of services, which hampers the application of this kind of approach.

Ning Gu[9] in his granted paper explained that SWORD(Software Ontology For Resource Description) is rule based expert system for web service composition. SWORD is set of tools for the composition of a class of web services including "information - providing " services. In SWORD, a service is represented by a rule to express that with given certain inputs, the service is capable of producing particular outputs. SWORD can compose services automatically, but it identifies services only syntatically - that is, by their inputs and outputs. Two services with the same inputs and outputs can have different functions.

Thomas Fischer [10] in his paper explained about recently WSMO-Lite(Web Service Modeling Ontology) has been proposed for describing Web Services semantically as the next evolutionary step after SAWSDL (Semantic Annotations for WSDL), filling SAWSDL annotations with concrete service semantic service descriptions. WSMO-Lite ontology is on one side lightweight and on the other side provides elements for modeling functionality of web services. But WSMO-Lite does not provide modeling of input and output parameters explicitly and relies on their derivation from free variables in the formulas for precondition and effect.

In paper [16] authors given a survey about service design attributes of reusability of the services. They given that service availability is also one of the factor for reusability of the service. Therefore in our paper we are discovering for service availability using different search methods.

In paper [17] there is a survey on semantic web which is used to extract semantic information from web documents. But it requires additional tools to add annotation to the existing documents as well as as it is known from literature survey if the underlying web information changes then it is necessary to required to change the ontologies. Therefore in this research we are not using any ontology based discovery method, instead we are processing tokenized WSDL elements and check for their semantics using MMA(Match Making Algorithm) for discovering suitable web services.

III. Model Of Web Services Discovery System

As it is explained in the literature survey additional efforts are required for discovery of web services and to get robust web service in low cost. Permanent shutdown of public UDDI has made UBR unavailable to start with.

Customized commercial search engines generally provide large number of online web services in the form of WSDL files. But this requires lot of preprocessing in WSDL file to analyze WSDL- portal like HTML pages and to infer relation between them and WSDL files in order to create WSDL focused discovery without using Google. Therefore it is necessary to collect and process WSDL, policy, metadata files to get suitable web service. Schema and policy documents are used as metadata documents of the web services.

An overview of an approach to discover suitable web service according to the user request is shown in the figure 1.

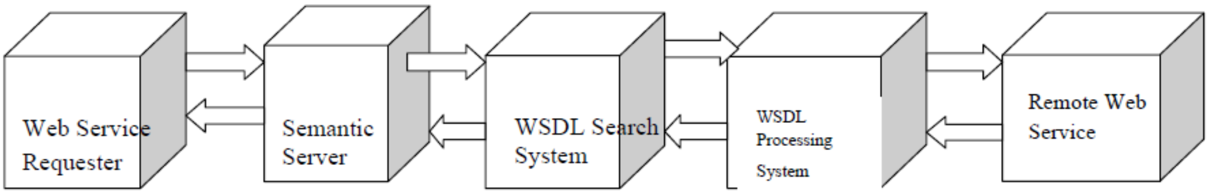

Fig. 1. Model of Web Services Discovery

User requests for a web service from Web service requestor system. Semantics of the request is collected by semantic server which enriches the request by additional suitable semantics and proper requesting function forwarded to the discovery system. Discovery system searches WSDL according to the user's requested term. WSDL processing system checks the input/output element of the <porttype>/<operation> element. If <input> as well as <output> element of <porttype>/<operation> is matching with user request then <port> element will be processed to extract <location> element.

The <location> element gives URL of the physical location of the web service which is required by the discovery system. Once the location is found then non functional description such as reliability, security, policy will be analyzed using metadata of web services using WS-MetadataExchange. Satisfaction of these factors proceeds to further steps. Suitable SOAP message will be constructed and this message will be redirected to the same physical address via SOAP intermediary nodes.

Routing and addressing information will be added to SOAP header by the intermediary SOAP nodes and SOAP message will be redirected through several intermediary nodes according to address location of the web service. The SOAP message will be received by the remote service, extracts body of the SOAP message, understands and executes the request and response message transferred back via intermediary nodes to discovery system. Discovery system collects the results in to its dictionary which is required for further analysis. Semantic server will pose additional queries to requestor after getting results from discovery system. According to user response if user gets satisfied then the result will be forwarded to requester.

IV. Dynamic WSDL Search System

To retrieve WSDL according to the list of functional words of user request, the requested term is inputted to the WSDL search system which is developed in this research. Search results with more than zero value of confidence and support factors are used to find weather functional words are available or not? If requested terms are available, then those service's WSDL records are retrieved to select suitable web services.

To get information about availability of requested terms as web services, it is required to search for requested terms using search engines and should apply following equation (1) and equation (2) to find support and confidence of the requested terms in the search results.

$$\text{support } s(t1) = P(t1) = \|So_{t1}\| / \|S\| \quad (1)$$

$$\text{confidence } c(t1) = \|So_{t1}\| / \|So\| \quad (2)$$

………

………

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

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

Where t1,t2..tn are the requested terms, in this example it is temperature, language translator.

$\|Sot1\|$ 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. Support greater than zero shows the existence of the requested term in the output element. $\|So\|$ is total number of outputs in the particular web service, 100% confidence indicates web service is meant for requested output and confidence greater than zero indicates availability of requested term in the web services.

The search results which have high support is considered and those service's input element is extracted. If requester's input element for e.g. city_name doesn't matches with the extracted input element(zipcode) of the GetTemperature service then it is required to search for the service which gives zipcode with city as input. If it is found then find the support and confidence of that search.

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

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

Here ‖Stin,Stout‖ is total number of services that satisfies input, output parameters of requested terms in the input or output or operation elements of web services. In equation(3) ‖S‖ is total number of web services in the search results that contain requested terms in any elements of web services. In equation (4) ‖Stin‖ is total number web services which matches input term(city) 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 get 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 of requested term.

A. 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.

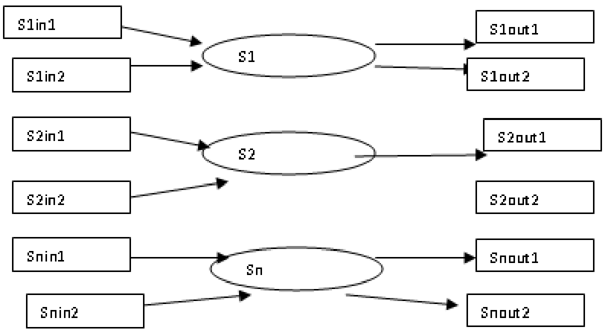

Fig. 2. vertical search

B. Horizontal Search

Consider the Request R1 for the service, 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..R1outn 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.

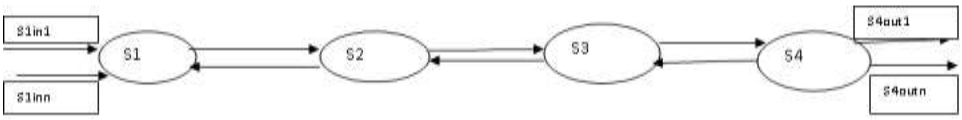

Fig. 3. Horizontal Search

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(5) and formulae(6).

$$s = P(Sout_{n-i}, Sin_{n-i+1}) = \| Sout_{n-i}, Sin_{n-i+1} \| \,/\, \| S \| \quad (5)$$

$$c = P(Sout_{n-1}, Sin_{n}) = \| Sout_{n-i}, Sin_{n-i+1} \| \,/\, \| Sout_{n-i} \| \quad (6)$$

Where i ranges from 1 to n-1 and n indicates the service number which satisfied the user requested function in output term of the web services.

In equation (5) $\| Sout_{n-i}, Sin_{n-i+1} \|$ indicates number of web services which matches output element of n-ith web services to the input element of n-i+1st web service. ‖S‖ indicates the total number of web services in the particular search which contain $Sout_{n-i}$ terms in any elements web services in the search results. In equation (6) $\| Sout_{n-i} \|$ indicates the total number of web services which satisfies the requested term in the output elements of n-ith web services.

Finally these list of services with high support(s) and confidence(c) are used to satisfy the user request.

V. WSDL PROCESSING SYSTEM

WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME.

The WSDL document of a service usually consists of ports, portTypes, operations, input/output messages and other definitions to express its function. The names of its components generally are concatenation of words which declares the function of the service such as " getAccountDetails " of PortType element. It is function of the WSDL processing system to decompose the names into individual terms according to some rules. Then tokens "get Account Details" is obtained as functional description of the service.

```
<porttype name=”MyBank”>
<operation name=”getAccountDetails”
```

```
parameterOrder="String_1" >
<input message= "tns:MyBank_
getAccountDetails"/>
<output message="tns:MyBank_
getAccountDetailsResponse "/>
</operation>
</portType>
```

Function of web service will be found in the operation element of WSDL file by processing <definition>/<service>/<porttype> element of WSDL. This operation should be matched with user requested functionality by using MMA(Match Making Algorithm). Matching of requested term will be weighted in increasing order of different matches of MMA as given here.

Exact> Plugin> Subsume > Sibbling > Fail

Among similar services, a maximum matching service (which has more weight) is considered as suitable web service.

To make a Web service useful, a service consumer who has discovered a set of useful services must be able to determine invocation details of the services. This will be described by WSDL. WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). The <definition>/<service><location> element of WSDL is given below.

```
<port name="BanksIFPort"
binding="tns:MyBankbinding">
<soap:address xmlns:wsdl="
http://schemes.xmlnsoap.org/wsdl" >
<location="http://www.Mybank.getAccounts /">
</port>
```

This location is actual service location where the service methods exists and should be always available. Thus the required web services will be discovered.

VI. Generating WSDL From Java

There are many tools to generate a WSDL document, given a Java remote interface. Current tools will, however, generate only WSDL that describes an RPC style invocation for SOAP-based Web services. The **xrpcc** utility can be used to generate a WSDL document with SOAP-HTTP binding, given a Java remote interface, using the following

```
command:
xrpcc -classpath %classpath% -server -keep -d
<destination directory> <configuration xml file>
```

VII. Finding Service Using UDDI

The ability to publish services in a UDDI registry requires the application used to publish the service interface definitions to understand WSDL. WSDL4J is one mechanism that allows an application to read and interpret WSDL.

```
// Read the WSDL service interface document
Definition definition = WSDLReader.readWSDL(null,
wsdlURL);
// Create a new tModel to be used to map the WSDL
service interface
Tmodel tModel = new tModel();
OverviewDoc overviewDoc = new OverviewDoc();
OverviewURL overviewURL = new
OverviewURL(wsdlURL);
tModel.setOverviewDoc(overviewDoc);
```

This is the last step in parsing WSDL to create the appropriate UDDI entities. UDDI4J contains APIs that allows to publish, find, and bind to a Web service. Because UDDI4J is open source, it comes with source code, JavaDoc, and several sample applications. It contains multiple APIs but the one most frequently used is the UDDIProxy class. Let us look at how UDDIProxy class interacts with a registry:

UDDIProxy proxy = new UDDIProxy();

A user who willing to find all businesses that meet a specified criterion, such as companies that start with the name "Flute," would use the find_business method of the proxy similar to this:

BusinessList bl = proxy.find_business ("Flute", null, 0);

The UDDI4J is one way for querying the registry. The Java API for XML registries (JAXR) is another.

VIII. Results

Comparison of web service discovery time between WSDL based discovery and UDDI Based Discovery is shown below in the graph.

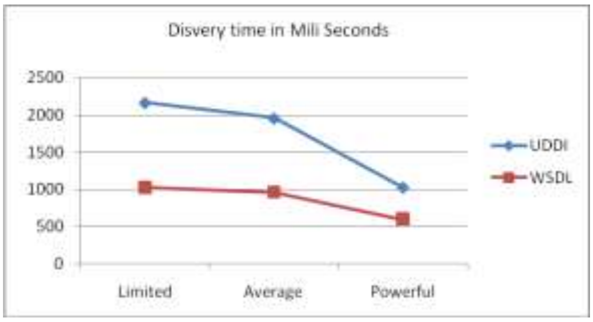

Fig. 4. Comparison between WSDL & UDDI based discovery

Average of support and confidence of different search matches for a particular user request is given below in the graph. As given in the graph below horizontal search satisfies the particular user request with more support and confidence than all other searches.

As given in the section VII, the tModel of UDDI contains an overviewURL element that points to the location of the service interface definition (WSDL). Therefore WSDL has to be retrieved from that location to get service signature and other details.

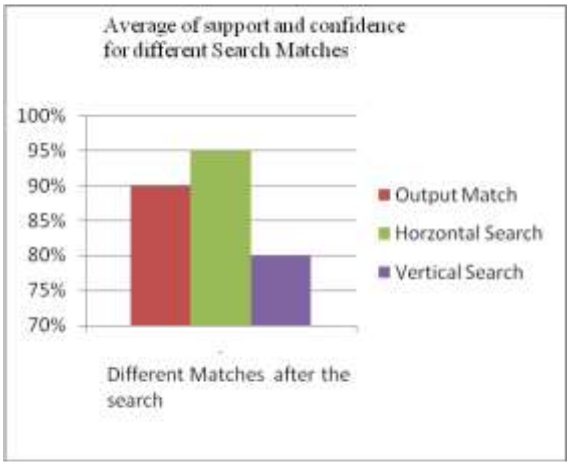

Fig. 5. Average of support & confidence for different search Matches of a request

But to find appropriate business service from UDDI and to retrieve WSDL, takes more time than directly retrieving WSDL using search engine.

IX. Conclusion and 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 results 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. When a large scale web services are available, an innovative dynamic structured integration is required. This can be achieved by processing WSDL to develop a suitable discovery mechanism which requires information retrieval and data mining technique.

It is necessary to provide robust web services to the user in less cost and high speed. As a future work it is planned to develop a knowledgebase of WSDL by using data mining technique. As a result the time of discovering the appropriate service will be reduced after the request is received.

Acknowledgement

I Mrs. Sumathi would like to give my sincere thanks to my guide Dr. Niranjan N. Chiplunkar, Principal, NMAMIT, Nitte, Karkala, India, for his motivation, guidance and giving me an opportunity to continue my research under the absents of my first guide Dr. Ashok Kumar(passed away). I also feel my heartfelt thanks to Dr. Ashok Kumar who was professor of Computer Science and Engineering, NMAMIT, Nitte, for his great support and motivation.

References

[1] Wil van der Aalst, Senior Member, IEEE, “Service Mining: Using Process Mining to Discover, Check, and Improve Service Behavior”, IEEE transactions on services computing – 2013.

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

[3] Noh-sam Park, Gil-haeng Lee, Electronics and Telecommunications Research Institute, KOREA "Agent-Based Web Services Middleware" IEEE GLOBECOM 2003.

[4] Khalid Elgazzar, Ahmed E. Hassan, Patrick Martin , School of Computing, Queen's University, Canada " Clustering WSDL Documents to Bootstrap the Discovery of Web Services" 2010 IEEE international conference on Web Services.

[5] Lijie Wang, Fei Lie, Liangjie Zhang, Ge Li, Bing Xie , Software institute, School of Electronic Engineering and Computer Science, Peking University, Beijing " Enriching Descriptions for public Web services using Information Captured from Related Web pages on the Internet" 2010 Fifth IEEE International Symposium on Service Oriented System Engineering.

[6] Pat. P.W. Chan and Michael R. Lyu , Chinese University of Hong Kong , China ,"Dynamic Web Service Composition : A new approach in building reliable Web Service" 22nd IEEE Int. conf. on Advanced Information Networking and Application.

[7] Jan Hendrik Hausmann, Reiko Heckel, Marc Lohmann, University of paderborn " Model based Discovery of Web Services" IEEE International Conference on Web Services.

[8] 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.

[9] Ning Gu, Juntao Cui, Wei Ye, Haixun Wang, Jian Pei " A system Framework for Web Service Semantic and Automatic Orchestration" Granted by NSFC Grant No 60473124.

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

[11] Mohamed Gharzouli , Mahmoud Boufaida " A generic P2P Collaborative strategy for discovering and Composing Semantic Web Services" IEEE 2009 Fourth Int. Conf. on Internet and Web Applications and Services.

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

[13] 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.

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

[15] Quist Aphetsi Kester, Ghartey Nana Gyankumah, Ajibade

Ibrahim Kayode, “Using Web Services Standards for Dealing with Complexities of Multiple Incompatible Applications”, I.J. Information Technology and Computer Science,2012/11,34-41, http://www.mecs-press.org/
[16] T. Karthikeyan, J. Geetha, “A Study and Critical Survey on Service Reusability Metrics”, I.J. Information Technology and Computer Science, 2012, 5, 25-31, http://www.mecs-press.org.
[17] Vishal Jain, Dr.Mayank “Singh Ontology Based Information Retrieval in Semantic Web: A Survey, I.J. Information Technology and Computer Science”, 2013/10,62-69, http://www.mecs-press.org.

Authors Profiles

Dr. Niranjan Chiplunkar did his BE(E&C) from NIE Mysore in 1986, M.Tech(CSE) from MIT, Manipal in 1991 and Ph.D. in computer science and engineering from University of Mysore in 2002. His areas of interest include "CAD for VLSI", "Web Services", "Embedded Computing" and "Computer Networks". He is a member of IEEE, Computer Society of India and Indian Society for Technical Education. He is a fellow of Institute of Engineers(India). He has more than 27 years of teaching experience. He is currently the Principal and Professor in Computer Science and Engineering at NMAM Institute of Technology, Nitte, India. Prof. Chiplunkar has successfully completed two research projects grants and presented more than 50 technical papers in National and International Conferences and journals. He also written two text books and has been awarded with “Excellent Achievement Award” from Centre for International Cooperation on Computerization, Govt. of Japan in March 2002. During 2007, he has been given “Distinguished Alumnus” award by the Manipal University.

Dr. Ashok Kumar has done his B.E.(E&E) 1988,B E College, Bagalkot, India. M.S. (Software Systems) in Birla Institute of Science, Pilani, India and Ph.D. in 2011 – Computer Sc. & Engg, VTU,Belgaum,India. He was working as a professor in the dept of Computer Sci & Engg in NMAMIT, Nitte, India. He had 13 years of experience and his area of interest was VLSI, Open Source Softwares and Web Services. He published 12 in International and National level conferences and 2 in International Journals and also had written several books.

Prof. Sumathi did her M.Tech from NITK, Suratkal, Karnataka, India and doing her PhD under the guidance of Dr. Niranjan N. Chiplunkar in Computer Science and Engineering Department, NMAMIT, Nitte, Karkala, VTU, University, India. Presently she is working as Associate Professor in Computer Science and Engineering Department of Canara Engineering College, Mangalore. India. She has 13 years of teaching experience. Her research area includes Web Services, Cloud Computing and Data Mining. She is associated with ISTE and also published papers in National and International conferences as well as in journals.

How to cite this paper: Sumathi, Niranjan N. Chiplunkar, Ashok Kumar A.,"Dynamic Discovery of Web Services using WSDL", International Journal of Information Technology and Computer Science(IJITCS), vol.6, no.10, pp.56-62, 2014. DOI: 10.5815/ijitcs.2014.10.08