Publishing and Discovering Business Services on Private Business Registry in B2B Environment

NIZAMUDDIN CHANNA*, AND INTZAR ALI LASHAR*

RECEIVED ON 31.08.2012 ACCEPTED ON 20.03.2013

ABSTRACT

The need of data and information is crucial to the human users. WWW (World Wide Web) plays a major role for fulfilling their needs, but sometimes all the purposes of human users are not achieved from the limited structure of the data available on the internet. However, the requirements of the accessibility of the data by users have also been changed and the same data is required in machine understandable form. IT (Information Technology) has become mandatory to convert the existing web from people-centric to software application-centric. The WS (Web Services) provide the suitable solution for application-to-application interoperability and integration over the Internet. WS is a promising technology that make available extensible and logical infrastructure for application to application communication constructed based on the current Web standards. The most accepted charm of the WS architecture is SOA (Service Oriented Architecture). The major roles involved in the WS infrastructure are service broker, service consumer/client and service provider. The technical pillars of WSs involve SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) and UDDI (Universal Description and Directory Service). By looking at the core pillars of the SOA triangle, we have investigated and explored all implementation aspects of the WSDL, SOAP and UDDI. We have further explored what UDDI is, structure of UDDI. How to create and publish WSDL into UDDI via SOAP, Operations of UDDI, how to access UDDI using java and SOAP clients. The extensive exploration of the work on UDDI implementation, help the researchers benefiting from our implementations into their core web service researches such as composition, discovery, choreography, orchestration Semantic WS.

Key Words: WSDL, Private Business Registry, Web Services, jUDDIv3, SOAP.

1. INTRODUCTION

In the beginning of WWW, it was only used for fetching the data and information to the users, while with the passage of time, there has been a great change in the needs of common users, as well as the usage of same data for business purposes. Now they require the same data and information in automated business processing systems. WS provide the type of platform for providing the combine operations of businesses; reduce the cost and time for maintenance and development of web applications. WSs have encouraged the reuse of code for WWW. WSs with a reusable uniform arrangement have provided businesses to distribute functionality with

*Assistant Professor, Institute of Business Administration, University of Sindh, Jamshoro.

random number of partners to communicate with each other without having to pre negotiation. For example, a company of selling books is interested in shipping services; those books are properly distributed to peers [1]. WS is a small software program, which has been recognized by a URI (Uniform Resource Interface), using Internet via its standard interface could access that software program. The declared operations of interface description are achieved by the nature of messages, which are exchanged at the stage of interaction with the service. For examples, to design a service for converting inches into centimeters or centimeters into inches could be declared with the operation, which converts inches with one input (Source) of type float and an output of type float (for result). WS also can call or invoke other WS [2]. All data and service metadata is transmitted using XML, so programming language and system dependencies are removed. WS are the most popular invocation of the SOA given in Fig. 1 SOA [3].

Scalable: The past solutions were not designed with the scale of the Web in mind. SOA should work across the world.

Loosely Coupled: SOA does require that the parties in a transaction are known well. The system only needs the information to invoke the service.

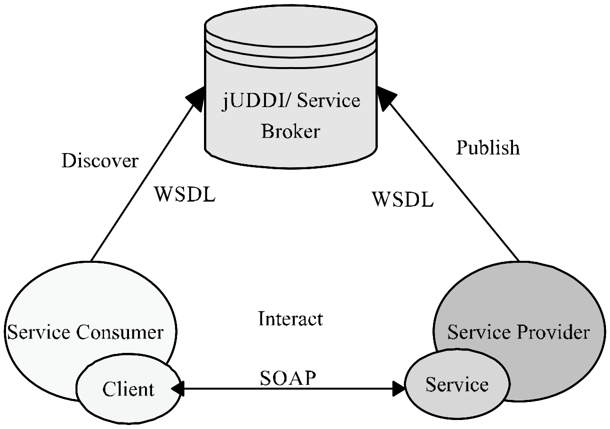


FIG. 1. SERVICE ORIENTED ARCHITECTURE

Interoperability: One party should be able to communicate with another party regardless of the machine they are running on.

Discovery: On party should be able to dynamically discover other partners to invoke.

Abstraction: A SOA abstracts the underlying technology.

Developers can concentrate on building services for business users rather than connecting systems and applications.

Standards: Standards are the basis of interoperable contract selection and execution.

The examples of WS [4] may include stock quotations, weather reports, credit checks, airline travel reservation processes, stock quotations, travel advisories, credit checks or airline travel reservation processes etc. All these WS straightforwardly incorporate with other WS in same or different companies for creating and completing business process [5]. Such mechanism of WS allowed various businesses to vigorously locate, bind and publish through the Internet range of WS [6].

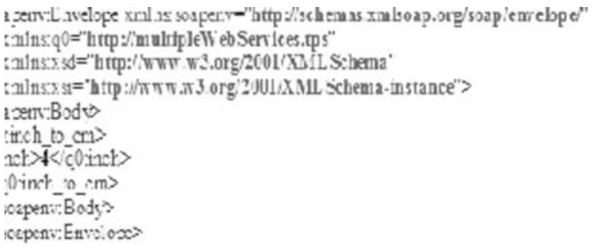
The rest of the paper is organized as follows: In Section 2 WS description has been discussed. Section 3 briefly describes the SOAP. Section 4 describe in detail universal description and discovery of WS. Section 5 provides the detail procedure of implementation of publishing and discovering WSs on private registry. Finally in Section 6 draws the conclusions and describes the future work.

2. WEB SERVICE DESCRIPTION LANGUAGE

WSDL is a based on XML Language, which defines [7] and describes how to access the WSs. The Instances and interfaces on the network of WS are described by

WSDL. WSDL is a extensible therefore, WS endpoints [8] could be described in message format for communication on network protocols. The major protocols such as WSDL, SOAP [9], XML [10] and HTTP [11] provide the means to invoke the functionality for one software components over the Internet. These web-based technologies could be used to combine various components of software and communicate with HTTP protocol over any network [12]. In order to combine software components data type information, binding information, address information and information on all available functions where, their calling parameters are also included are required. In WSDL data type information and binding information include the value specification for all XML [13] messages and specific transport protocol respectively. In WSDL for locating particular services address [14] information's are also needed. For combining software components a software developer might use the documentation to get this information for each WS. The main purpose of developing WSDL is extracting the information from used WS to reduce the cost of integration, reduce the time consuming and reduce the human error [15].

A WSDL file that is based in XML format provides the information of each software components. Software components could be integrated with any of the existing methods of a WS [16] by using WSDL. In an XML format one example of WSDL file for converting inches into miles is given in Fig. 2.



```
apenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:q0="http://multipleWebServices.tps"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
apenv:Body>
:inch_to_cm>
nch>4</q0:inch>
q0:inch_to_cm>
soapenv:Body>
soapenv:Envelope>
```

FIG. 2. EXAMPLE OF SOAP DOCUMENT IN XML FORMAT

The first section of the WSDL defines different namespaces including the URL. It is a place from where businesspersons, requester, consumer or software component can find the web services description. The second section of the WSDL documents uses the message for input and output purposes. The operations are defined in third section in case of getting output, which has been encapsulated as a port type. WSDL binding to its operations [17] the encoding rules for inputs and outputs as a port type are declared in fourth section. List of business entities and business services can be added by WSDL to describe web services. Where software program could interpret a list of WSDL description and find the essential requirements for integration [18]. The final portion of WDL locates and find out WS, which are offered for further utilization [19]. Invoking the operations of [20] WSDL services in Eclipse platform.

3. SIMPLE OBJECT ACCESS PROTOCOL

SOAP is XML based communication protocol [21] that support communication between various web-based applications. It is lightweight XML-based [22] protocol for messaging over HTTP (Hyper Text Transfer Protocol) and various other protocols for Internet. SOAP is a protocol for exchange the information in a distributed and decentralized atmosphere [19] SOAP has been designed to communicate through Internet and support the format for requesting and responding the messages. SOAP is a platform independent and XML based Protocol, which includes Envelop, Header, Body, and Encoding rules elements.

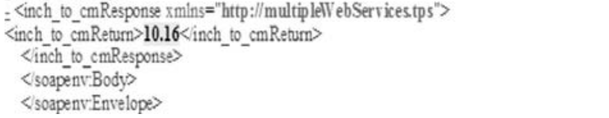
An Envelop is a compulsory part of SOAP, which recognizes the XML file as a SOAP message and defines an infrastructure for describing what and how to process it. However, another element of SOAP protocol, which contains header information, is optional is called Header element. The most important required part of SOAP, that contains request and response information is called Body element. The set of encoding rules, which defines data types and expressing instances of WS are described in Body of SOAP. The optional part of SOAP, which provides information about error during the execution of request is called Fault element. The SOAPUI-3.6.1 [23] protocol makes possible the usage of locating and binding web services by defining a path of message for finding messages. It is very easy to use SOAP for publishing and discovering Business Entity, tModels and Business Services to UDDI [24]. In WS environment user send request through SOAP to UDDI registry node and UDDI registry node contacting HTTP server and SOAP Processor send response to user according to his request is given in Fig. 3.

Request for converting inches into centimeter through SOAP User interface in XML Format.

Response of converting inches into centimeter through SOAP User interface in XML Format.

4. UNIVERSAL DESCRIPTION DISCOVERY AND INTEGRATION

For WS management UDDI [25] is the genuine standard on the WWW. UDDI is XML based infrastructure for managing, describing and discovering business services, which make possible development of an incorporated e-commerce atmosphere [26]. The growth of a virtual marketplace is essential within the government Department of Atmosphere. The data and services within the government environment will allow the development into a distributed environment for controlling, planning and integrating [27]. There is no single contractor or organization which will be able to accomplish or envision such incorporated vision. All parties and variety of the most excellent of breed and from conducting experiment would evolve it from information of the existing services and resources. The tools and standards of UDDI that are developed by the business vendors could afford a non-proprietary marketplace where contractor and agencies could explain their operation related responsibilities and the various kinds of data and services that they afford. The UDDI registries would provide a subscribed and published mechanism to accumulate WS and agency descriptions [28]. These registries also point to complete procedural specification for defining the interfaces to WS. The organizations metadata that has been stored in XML schema (XML-S) could be discovered by standard searching tools. They will sustain scientific fingerprints for stipulation as changes in the categories of the WS interface published immediately on the WWW. However, UDDI registries is not only responsible to resolve the troubles of force standard web service interface or controlling data standards but it provides an identical structure for business for defining their responsibility in general government culture [29] UDDI Registries also support an identical techniques for such organizations to present, develop, discover and combines strategies and handle their entrance to web service distributed environment in the WWW [30]. For creating business applications the performance, availability and non-functional features



```
-    <soapenv:Envelope      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soapenv:Body>
- <inch_to_cmResponse xmlns="http://multipleWebServices.tps">
<inch_to_cmReturn>10.16</inch_to_cmReturn>
  </inch_to_cmResponse>
  </soapenv:Body>
  </soapenv:Envelope>
```

FIG. 3. EXAMPLE OF SIMPLE OBJECT ACCESS PROTOCOL

of business services would be considered by the WSDL files. Developing the WS application evaluate the qualities of a various application software components, which are supplied by various companies. The fundamental infrastructure of WS technologies has been extended by UDDI for creating a registry of business entity and business services on the WWW [31]. The extension of UDDI obtains business services into the area of business companies conducting business over the World Wide Web with each other. Fig. 4 shows the implementation of Juddi UDDIBrowser Portlet, where Business Entity, Binding templates and Business Service are published by SOAPUI 3.6.1 in UDDI Registry.

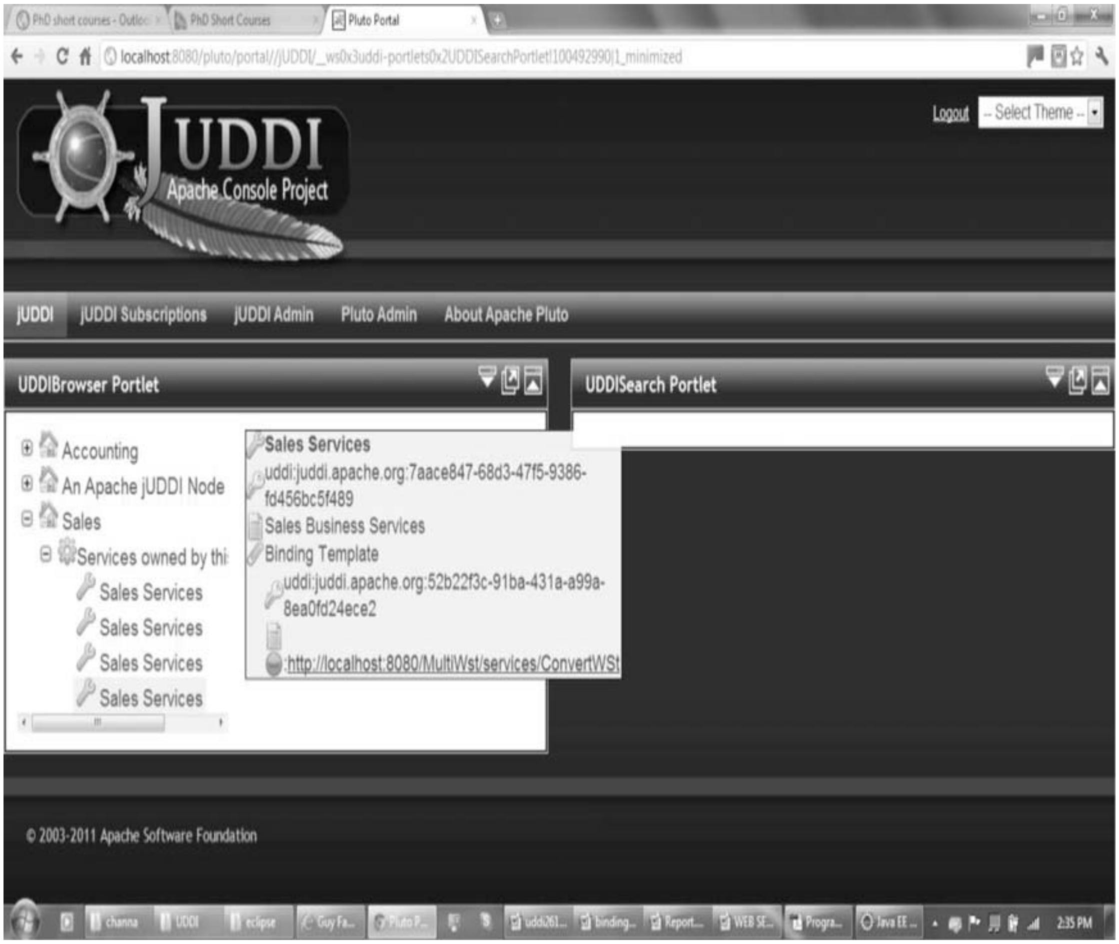
The specifications of UDDI facilitate business companies to dynamically, easily and quickly transact and find one another. UDDI facilitates a business company to:

1. Describe the Business Entity and Business Services.

2. Locate other business companies that offer various business services.

3. Combine their business services with other business services.

For example, if a developer has developed a WS that relies on a checking credit function with the purpose of



Sales Services
uddi:juddi.apache.org:7aace847-68d3-47f5-9386-fd456bc5f489
Sales Business Services
Binding Template
uddi:juddi.apache.org:52b22f3c-91ba-431a-a99a-8ea0fd24ece2
:http://localhost:8080/MultiWst/services/ConvertWSt

FIG. 4. JUDDIV3 : UDDIBROWSER PORTLET

validating his consumers, for finding that function he could use a UDDI registry. The requester send request through SOAP to the UDDI registry for checking credit function and may also offer various other conditions like a performance condition, security requirements, cost limits etc. Than UDDI business registry would recommend single or various different companies for providing such a function that permitting him to select the favorite supplier. The UDDI specification allowing the use and creating business registry that include the information in relation to business entity, tModel and business services they provide. The information in UDDI is structured as under:

4.1 Business Entity

The information about a company is represented in business entity. Each business entity includes the company name, distinctive identifier, basic contact information, brief description of the company, list of identifiers and categories that describe the company.

4.2 Business Services

Business services are related with the Business entity. The list of business services is available in the business entity. The entry of each business services includes a list of categories that describe the business services, a description of WS and list of pointers to indicate information associated with those services.

4.3 Binding Template

The technical description of WS is called binding templates. The information that is required for invoking or binding to a particular WS is available in the structure of binding template. The structures of binding templates support for finding remotely hosted services or help for finding technical entry point. These structures also provide support for describing distinctive technical attributes of a particular execution. Binding templates also support for technology, setting files and specific parameters of particular applications. The information, which is available in binding template, contains load balancing and routing facilities, the WS URL and reference corresponding to tModel structure [32]. There may be various binding template to a single business services. Binding templates represents the concrete accomplishments of the WS. However, there are multiple binding templates for a single business service therefore, WS may be identifying various executions for the same WS. Where, each WS bound to a various network address or varieties of protocols.

However the main objective of UDDI's is to facilitate discovery and description of business services information for this most interesting and important technical data is provided by binding templates. UDDI binding template has specify that how exactly access a specific business services might be HTTP, FTP, Fax, electronic mail or telephone. The every structure of binding template has a unique parent of logical business service and a unique logical business entity.

4.3.1 tModel

The technical model of web services in UDDI registry is called tModel. tModel is a mechanism of describing and storing the variety of services, businesses and template structures in the UDDI registry. Within UDDI any abstract concept could be registered as a tModel. After that a given business service would be implemented with the help of associating tModel port type with the binding template of that business WS.

Service type definitions are usually referred by tModels. tModels characterize constructs or unique concepts. To describe agreement with a concept, a specification or a shared design, tModels are used. In the UDDI registry tModels have a variety of utilizations. In relation of mapping WSDL with describing business WS there are two uses of tModels.

1. The technical specifications of business services like wire protocols; service types and bindings are represented with tModels.

2. The category system put in action for categorizing WS and technical specification tModels are also used.

In the UDDI registry when a particular WS in published as a tModel then tModel assigned a distinctive key which key is called tModel key. UDDI entities use tModel key to reference the tModel. In the specification of tModel one element is overviewURL. The function of overviewURL element is providing the address of WSDL document.

5. IMPLEMENTATION

The intended purpose of the research paper is to explore the implementation aspects of the deployment of WS into UDDI by using the implementation of UDDIv3 Specification by jUDDI. This section of implementation of UDDI is further discussed below in the following sections i.e. Section-A which describes how to find and use security token, Section-B describe how to publish and delete WS into the UDDI and Section-C describes how to find or inquire the published services.

Since for the implementation and deployment of web services following the SOA architecture we need the UDDI framework which we have selected jUDDI, the WSDL descriptions we have achieved by creating our WS using Apache Axis Framework. The client side access needs the interfacing using SOAP, we have used the SOAP 3.6 client Software for accessing the jUDDI specification as well as the invocation of the actual Web Services deployed into the UDDI Registry.

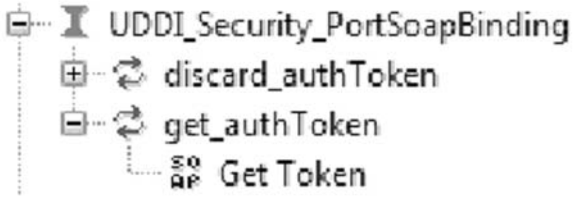
5.1 Security Services

The Security API provides the methods for business entity, business services, tModels, bindings and related business and business services using UDDI_Secuirty_PortSoapBinding. This API of UDDI security includes get_authToken and discard_authtoken to negotiate a UDDI registry for finding and publishing particular businesses and services information's. Fig. 5 shows the security API calls that jUDDIv3 registry support.

The brief description of the elements of UDDI_Security is given as:

Discard_authToken is used bring up to date a node that is required and would be considered unacceptable if used after receiving this message.

Get_authToken is used to send request to a UDDI node for authentication token. Authentication Token is a system based on user Id and Password credential for providing authenticated security.



UDDI_Security_PortSoapBinding
discard_authToken
get_authToken
Get Token

FIG. 5. UDDI_SECURITY_PORTSOAPBINDING

5.2 Publishing and Deleting Businesses and Services

The Publishing API provides the methods for adding and deleting Business entities, Business Services, binding, tModels and Publisher Assertions using the UDDI Publication API (Application Programming Interface). This API of UDDI publication included the information about white pages, yellow pages and green pages. Publication API includes operations for creating, updating, reading and deleting the information in jUDDIV3. For modify or register these operations a caller can use any number of business or services. Fig. 6 shows the publication API calls that jUDDIV3 support.

The brief description of the elements of UDDI_Security is given as:

Add_PublisherAssertions is used to add single or multiple publisher Assertions and show the relationship of one Business entity with another business entity.

Delete_binding is used to delete single or multiple instances data of binding template from the jUDDI registry.

Delete_business is used to delete single or multiple registration of business entity form a jUDDI registry.

Delete_publisherAssertions is used to removed single or multiple elements of publisher assertions from a collection of publisher's assertions.

Delete_service is used to delete single or multiple elements of business service from the jUDDI registry.

Delete_tModel is used to remove logically single or multiple structures of tModel.

Get_assertionsStatusReport is used to provide the status of existing outstanding of publisher assertions who is involved in any registration of business which is managed by a publisher.

Get_publisherAssertions is used to get the complete list of a publisher assertions associated with a publisher.

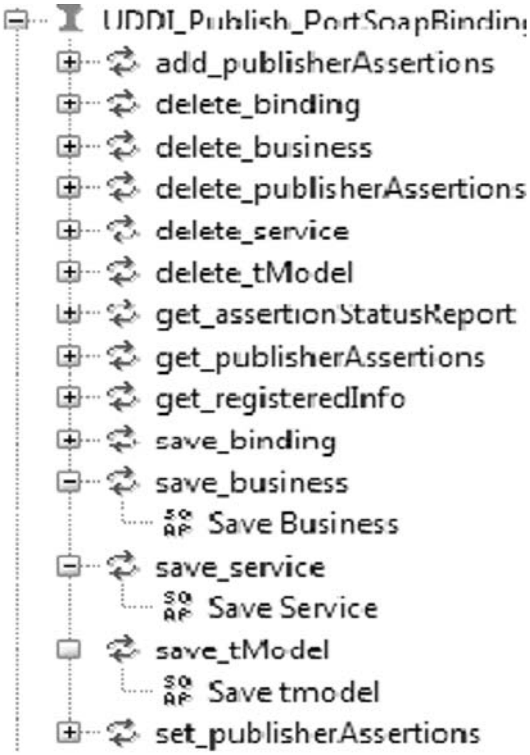
Get_registeredInfo is used to retrieve an abbreviated list of all tModel and business entity data for a publisher.

Save_binding is used for update and save all elements of binding template.

Save_business is used to update or save information concerning complete structure of business entity. Save Service is used to updates or adds single or multiple elements of business service.

Save_tModel is used to update or adds single or multiple registered tModel elements.

Set_PublisherAssertions is used to change the information of assertion related with a publisher.



```
UDDI_Publish_PortSoapBindin
    add_publisherAssertions
    delete_binding
    delete_business
    delete_publisherAssertions
    delete_service
    delete_tModel
    get_assertionStatusReport
    get_publisherAssertions
    get_registeredInfo
    save_binding
    save_business
        Save Business
    save_service
        Save Service
    save_tModel
        Save tmodel
    set_publisherAssertions
```

FIG. 6. UDDI_PUBLICATION_PORTSOAPBINDING

5.3 Discovering Business and Business Services

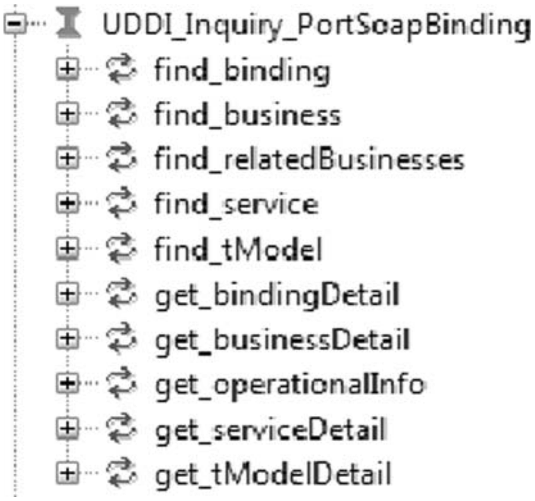
The inquiry API provides the methods for Discovering Business Entity, Business services, tModels, related Business, binding, service detail, business detail, binding detail, tModel detail and operational information using UDDI_Inquiry_PortSoapBinding. This API of UDDI inquiry includes operations that make possible to browse a UDDI registry and negotiate a UDDI registry regarding to finding particular businesses and services information's. Fig. 7 shows the inquiry API calls that jUDDIv3 registry support.

The brief description of the elements of UDDI_Inquiry is given as:

Find_binding is used to discover bindings inside or crossways single or multiple regirstered business services.

Find_business is used to discover information about single or multiple businesses.

Find_related Businesses is used to discover detail about the registrations of business entity, which is a related to a particular business entity, where business key is accepted in an UDDI inquiry API.



UDDI_Inquiry_PortSoapBinding
find_binding
find_business
find_relatedBusinesses
find_service
find_tModel
get_bindingDetail
get_businessDetail
get_operationalInfo
get_serviceDetail
get_tModelDetail

FIG. 7. UDDI_INQUIRY_PORTSOAP_BINDING

Find_service is used to discover particular services, which are registered inside the business entities.

Find_tModel is used to discover structure information about single or multiple tModels.

Get_bingingDetail is used to obtain information about binding template, which is appropriate requesting service.

Get_businessDetail is used to obtain the information business entity for single or multiple organizations or business.

Get_operationalInfo is used to obtain the extended information of business entity.

Get_serviceDetail is used to obtain complete details of data for a particular list of registered business service. Get_tModel is used to obtain complete details of data for a particular list of registered tModel. The information, which is provided in UDDI Registry, has been divided in three categories.

5.3.1 White Pages

Basic information such as Contact information, Business names, Business Addresses, Data universal numbering system and Web site names about a company are search from UDDI in white pages.

5.3.2 Yellow Pages

The information about business types, business location, industry types, geographical location, Business Id according to industrial categorization and taxonomies are retrieved from yellow pages.

5.3.3 Green Pages

Green pages provide Technical UDDI. Information about Business Services, as well as how to describe, interacts and executes WS. The extensive exploration of the work on UDDI implementation, help the researchers for benefiting our implementations into their core WS researches such as composition, discovery, orchestration, choreography [33], and Semantic WS [34].

6. CONCLUSIONS

The use of WS has become a mandatory part of any globalized Enterprise information system, and it is necessary for them to publish their WS into some central or P2P (Peer to Peer) located repositories so that the intended users can find their services easily. The publication and discovery of the WS into the registries is an interesting topic in the filed of WS. The core purpose of this work is to create a informative knowledgebase for the research and industry which may help them getting the help and way for working on the deployment and access of their services efficiently and effectively. This research will elegantly support Business Companies that dynamically, quickly and easily offer business services on private business registry in B2B (Business to Business) environment.

The future intentions includes the integration of more than one UDDI implementations in order to provide the P2P discovery approaches for the better and improved B2B applications in the field of WS using SOA architecture. However, we also intend to work on extension of UDDI and publish OWL-S in extended version of UDDI. Extension of UDDI gracefully supports the automatically publishing, discovering, choreography, orchestration and composing the semantic web services in B2B environment.

ACKNOWLEDGMENTS

The First author would like to appreciatively and genuinely thanks to Prof. Jose Eluiz Fiadeiro, Department of Computer Science, University of Leicester, UK, for his guidance and most importantly, his patience during my post doctoral Studies. Authors also pay great thanks to HEC (Higher Education Commission), Pakistan, whose financial support (First Author) made us to conceive and complete this research endeavour.

REFERENCES

[1] Channa, N., Li, S., Shi, W., and Peng, G., "A CAN-Based P2P Infrastructure for Semantic Web Services Publishing and Discovery", Proceedings of 1st IEEE & IFIP International Conference in Central Asia on Internet The Next Generation of Mobile, Wireless and Optical Communications Networks Information and Communications Technologies and Application to E- and M-Commerce, Kyrgyz Republic, September 26-28, 2005.

[2] D'Mello, D., and Ananthanarayana, V., "Quality Driven Web Service Selection and Ranking" Proceedings of 5th International Conference on Information Technology Las Vegas, NV, April 7-9, 2008.

[3] Wang Fei, Z., Cairong, Y., and Xin, D., "Web Services and Grid Technology in Digital Forestry" Proceedings of 2nd International Workshop on Education Technology and Computer Science, Wuhan China, March 6-7, 2010.

[4] Tang, M., and Ai, L., "A Hybrid Genetic Algorithm for the Optimal Constrained Web Service Selection Problem in Web Service Composition" Proccedings of IEEE Congress on Evolutionary Computation, Barcelona, July 18-23, 2010.

[5] Aalst, W., Hofstede, A., and Weske, M., "Business Process Management: A Survey", Proceedings of International Conference, Eindhoven, Volume 2678 of Lecture Notes in Computer Science, Springer-Verlag, The Netherlands, June 26-27, 2003.

[6] Liu, L., and Cai, H., "WSPOM: An Ontology Model for Web Service Planning", Proceedings of 2nd International Conference, Qinhuangdao, China, October 28-31, 2011.

[7] Chinnini, R., Moreau, J., et al., “Web Services Description Language (WSDL) Version 2.0 Part-1: Core Language”, W3C Candidate Recommendation, 2006.

[8] Barros, A., Dumas, M., and Oaks, P., "Standards for Web Service Choreography and Orchestration: Status and Perspectives", Proceedings of International Workshops on Business Process Management, Nancy, France, September 5, 2005.

[9] Waleed, G., and Ahmad, R., "Security Protection Using Simple Object Access Protocol (SOAP) Messages Techniques", Proceedings of International Conference on Electronic Design, Penang, Decmber 1-3, 2008.

[10] Meier, W., "eXist: An Open Source Native XML Database," Web, Web-Services, and Database Systems\-NODe 2002 Web-and Database-Related Workshops, Volume 2593, 2003.

[11] Curbera, F., Nagy, W., and Weerawarana, "Web Services: Why and How." Workshop on Object-Oriented Web Services, Tampa, Florida, USA, October, 2001

[12] Tang C., and Dwarkadas, S.,"Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks", Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 175-186. ACM Press, 2003

[13] Böttcher, S., Hartel, R., and Messinger, C., "Searchable Compression of Office Documents by XML Schema Subtraction", Proceedings of the 7th International Conference on XML Database and XML Technologies, Springer-Verlag Berlin, Heidelberg, 2010.

[14] Matjaz, B., Sasa, A., Brumen, B., and Rozman, I., "WSDL and UDDI Extensions for Version Support in Web Services", Journal of Systems and Software, Volume 82 No. 8, pp. 1326-1343, Elsevier Science Inc. New York, NY, USA, August, 2009.

[15] Channa, N., Li, S., Shaikh, A., and Fu, X., "Constraint Satisfaction in Dynamic Web Service Composition", Proceedings of 16th International Workshop on Database and Expert Systems Applications, Copenhagen, Denmark, August 22-26, 2005.

[16] Skene, J., Raimondi, F., and Emmerich, W., "Service-Level Agreements for Electronic Services", IEEE Transactions on Software Engineering, Volume 36, No. 2, March-April, 2010.

[17] Hwang, S., Liao, W., and Lee, C., "Web Services Selection in Support of Reliable Web Service Choreography", Proceedings of IEEE International Conference on Web Services Miami July 5-10, 2010.

[18] Hwang, S., Liao, W., and Lee, C. "Verifying Web Services in a Choreography Environment", IEEE Proceeding of International Conference on Service-Oriented Computing and Applications, Irvine, CA Decmber 12-14, 2011.

[19] Jianjun Yu, J., Chen, R., Su, H., and Guo, S., "Web Services Publishing and Discovery on Peer-to-Peer Overlay", Proceedings of IEEE Asia-Pacific Conference on Services Computing, Guangzhou, Guangdong, December 26, 2006.

[20] Wang, L., "Design and Implementation of Web Service Plug-in on Eclipse Platform", Proceedings of International Conference on Computer, Mechatronics, Control and Electronic Engineering, Changchun, October 25, 2010.

[21] Fu, X., and Ming, G., Li, S., and Channa, N., "Methodology for Semantic Representing of Product Data in XML", Proceedings of Advance Workshop on Content Computing, Zhen Jiang, Jiang Su, China, November 15-17, 2004..

[22] Zhen, H.,Chang, H., and Sthanikam, B., "Efficient Support of XQuery Update Facility in XML Enabled RDBMS", Proceedings of 28th International Conference on Data Engineering, Washington, DC, July 2, 2012.

[23] Huhns, M., and Singh, M., "Service-Oriented Computing: Key Concepts and Principles", Journals & Magazines, Internet Computing, IEEE, Volume 9, No. 1, pp. 75-81, January-February, 2005.

[24] Tamilarasi, K., and Ramakrishnan, M., "Design and Development of an Enhanced UDDI for Efficient Discovery of Web Services", Proceedings of 3rd International Conference on Advances in Communication, Network, and Computing, Chennai, India, February 24-25, 2012.

[25] Paolucci, M., Kawamura, R., and Sycara, K., "Importing the Semantic Web in UDDI", Proceedings of Web Services E-Business and Semantic Web Workshop, Toronto, Canada, May 27-31, 2002.

[26] Colgrave, J., Akkiraju, R., and Goodwin, R.,"External Matching in UDDI", Proceedings of IEEE International Conference on Web Services, San Diego California USA, July 6-9, 2004.

[27] Stoica, I., Morris, R., Karger, D., Kaashoek, F., and Balakrishnan, H., "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM New York, NY, USA, October 4, 2001.

[28] Yasuda, K., "Adding Quality of Service to UDDI for Dynamic Composition”, Graduation Thesis, University of Aizu, March, 2008.

[29] Verma, K., Sheth, A., and Miller, J., "Meteor-s WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services", Journal of Information Technology and Management, Volume 6, No. 1, pp. 17-39, Number, 2005.

[30] Goodrich, M., Nelson, J., and Sun, J., "The Rainbow Skip Graph: A Fault-Tolerant Constant-Degree Distributed Data Structure", Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm, Miami Florida, January 22-24, 2006.

[31] Du, Z., Huai, J., and Liu, Y., "Ad-UDDI: An Active and Distributed Service Registry", Procedings of 6th VLDB Workshop on Technologies for E-Services, Trondheim, Norway, September 2-3, 2005.

[32] Maamar, Z., Yahyaoui, H., and Mahmoud, Q., "Dynamic Management of UDDI Registries in a Wireless Environment of Web Services: Concepts, Architecture, Operation, and Deployment", Journal of Intelligent Information Systems, Volume 28, No. 2, Kluwer Academic Publishers, April, 2007.

[33] Ke, Z., Zongyan, Q., and Hongli, Y., "Generating Scenarios from Web Service Choreography", Proceedings of IEEE Asia-Pacific Services Computing Conference, Jeju, Korea, December 12-15, 2011.

[34] McIlraith, S., Son, T., and Zeng, H., "Semantic Web Services", IEEE Journal & Magazine of Intelligent Systems, Volume 16, No. 2, pp. 46-53, 2001, 2001.