ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 1, January 2013

An Enhanced Service-Oriented Relationship Modelling of Web-Service using Ontologies

T. Priyaradhikadevi[#1], Dr. R.M.S. Parvathi[$2], R. Sumathi[*3]

[#]*Research Scholar & Professor, Mailam Engineering College, Mailam, TN, India.*

[*]*PG Scholar, Mailam Engineering College, Mailam, TN, India.*

***Abstract*— This paper is an attempt to develop an Ontological bootstrapping which aims at automatically generating concepts and their relations in a given domain. Bootstrapping an ontology based on a set of predefined textual sources, such as Web services, must address the problem of multiple, largely unrelated concepts. In this paper, we present techniques for bootstrapping and populating specialized domain ontologies by organizing and mining a set of relevant overlapping services provided by the service provider. In order to enable widespread usability for the Semantic Web there is a need to bootstrap large, rich and up-to-date domain ontologies that organizes most relevant concepts, their relationships and instances. We construct the bootstrapping ontologies using the methods of Token Extraction, TF/IDF (Term Frequency / Inverse Document Frequency) and using Web Content extraction. Then we model the relationship between the services. We constructed the bootstrapping Ontologies in JAVA and show our results proving that these Ontologies helpful in Web service Discovery.**

***Keywords*— Ontologies, Service Discovery, Web Service, Ontology Evolution, Semantic Web.**

I. Introduction

Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine processable format (specifically WSDL – Web Service Description Language). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards [1]. In a simple way Web Service is defined as: a piece of business logic, located somewhere on the Internet, that is accessible through standard-based Internet protocols such as HTTP or SMTP [2].

The Web Services are interacting computer applications capable of running on different platforms, managed by different organizations. The Web Service technology includes three open standards named SOAP (Simple Object Access Protocol), WSDL and UDDI (Universal Description Discovery and Integration). To make the Web Service accessible, it must be based on at least two standards named SOAP and WSDL. By using Web services, business applications can publish its function or message to the rest of the world. Web services use XML to code and to decode data, and SOAP to transport it (using open protocols). Web service may not be discovered simply because hundreds of thousands of Web services are available. Locating the required Web service, therefore, is time consuming, inaccurate, and tiresome. Thus, an automated discovery mechanism is required, that can discover the required Web service. However, the design principle of the current Web service standards undermines the automation of discovery [3].

Current research areas of Web Service focus on Semantic and automated Web Service Discovery. We know that the process of obtaining a set of services which can possibly satisfy a request from the user is called service discovery. So, the web service descriptions should be matched with the user query request so that the request should be discovered. The simple form of web service discovery model is static model, where the descriptions of the service providers are published to a directory service. Whereas the contrast model is dynamic service discovery model where the does not exist the centralised directory service. By using this promising feature we adopt the model of dynamic in our system as it provides various desired features such as scalability, reliability and no central control etc. In the following chapters, we propose our architecture by comparing the various works done on the web service discovery model. Then we elaborate our technique with the implementation results. At last we prove that our system is more desired and helpful in Web Service Discovery model than the others.

II. Related Works

In existing methodologies, many heuristics were proposed for the automatic matching of schema and several theoretical models were proposed to represent various aspects of the matching process such as representation of mappings between Ontologies. However, all the methodologies described require comparison between existing Ontologies. Previous work on ontology bootstrapping focused on either a limited domain or expanding an existing ontology. UDDI registries have some major flaws. In particular, UDDI registries either are publicly available and contain many obsolete entries or require registration that limits access. In either case, a registry only stores a limited description of the available services.

N.F. Noy et.al [4] proposed Ontology Evolution: Not the Same as Schema Evolution. In this paper the authors [4] proposed as ontology development became a more ubiquitous and collaborative process, ontology versioning and evolution became an important area of ontology research. They also added that the many similarities between database-schema evolution and ontology evolution would allow users to build on the extensive research in schema evolution. However, there were also important differences between database schemas and ontologies. The differences stem from different usage paradigms, the presence of explicit semantics and different knowledge models. A lot of problems that existed only in theory in database research come to the forefront as practical problems in ontology evolution. These differences have important implications for the development of ontology-evolution frameworks: The traditional distinction between versioning and evolution is not applicable to ontologies. There were several dimensions along which compatibility between versions must be considered. The set of change operations for ontologies were different. The authors concluded with that they must develop automatic techniques for finding similarities and differences between versions.

D. Kim et. Al [5] proposed Practical Ontology Systems for Enterprise Application. The authors [5] added that One of the main challenges in building enterprise applications has been to balance between built-in functionality and domain/scenario-specific customization. The lack of formal ways to extract, distill and standardize the embedded domain knowledge had been a barrier to effective and efficient customization. Ontology may provide, as many would hope, the much needed methodology and standard to achieve the objective of building flexible enterprise solutions. This article examined the uses, issues and challenges of using ontology in enterprise applications. The authors believed that they were seriously lacking in modeling methodology, domain user tools, and lifecycle management methodology for the creation and maintenance of ontology on a large deployable scale. So they presented the issues based on an ongoing project to build product ontology for a public procurement system. Through real life scenarios, they were hoping to convey important research directions to better enable ontology

G. Zhang et. Al [6] proposed Bootstrapping Ontology Learning for Information Retrieval Using Formal Concept Analysis and Information Anchors. In this paper the authors attempted to present an innovative approach to information retrieval for domain-specific digital library collections. They used a combination of Formal Concept Analysis (FCA) and a notion of information anchors to facilitate information delivery to the end user. This approach (1) used ranked objects in attribute concepts to facilitate topical queries for experts and expertise profiles; (2) formulated (keyword by keyword) context for concept lattice construction via a set of heuristics, including those based on information anchors for selecting descriptive phrases, (3) bootstraps the learning of domain-specific concept hierarchies using FCA, and (4) incorporated the learnt concept hierarchies and WordNet for content-based document classification. To demonstrate the feasibility and utility of this approach, the authors implemented a prototype online information retrieval systemmemsworldonline.case.edu (MWOL) for the emerging engineering discipline of MEMS (microelectromechanical systems) incorporating these ideas. MWOL has been actively used by a non-trivial group of MEMS practitioners; all user queries are processed in a fraction of a second as a result of inverse indexing strategy using Berkeley DB. Voluntary user feedback using online forms has been encouraging. However, no other systems with similar features are available for a comparative study at this point.

S. Castano et. Al [7] proposed Ontology Dynamics with Multimedia Information: The BOEMIE Evolution Methodology. In this paper, the authors contributed to present the ontology evolution methodology developed in the context of the BOEMIE project. Ontology evolution in BOEMIE relies on the results obtained through reasoning for the interpretation of multimedia resources in order to evolve (enhance) the ontology, through population of the ontology with new instances, or through enrichment of the ontology with new concepts and new semantic relations.

L. Ding et. Al. [8] introduced Swoogle: A Search and Metadata Engine for the Semantic Web. Swoogle is a crawler-based indexing and retrieval system for the Semantic Web documents – i.e., RDF or OWL documents. It analyzed the documents it discovered to compute useful metadata properties and relationships between them. The documents were also indexed by using an information retrieval system which can use either character N-Gram or URIs as terms to find documents matching a user's query or to compute the similarity among a set of documents. One of the interesting properties computed for each Semantic Web document is its rank – a measure of the document's importance on the Semantic Web. 1.

The problems pertaining to Web service discovery have long been taking attention of both academia and industry. The architecture proposed by us works on the semantic level and allows interlinking of resources through the use of RDF data model.

III. Proposed Technique

The ontology bootstrapping process is based on analysing a Web service using three different methods, where each method represents a different perspective of viewing the Web service. As a result, the process provided a more accurate definition of the ontology and yields better results. In particular, the Term Frequency/ Inverse Document Frequency (TF/IDF) method analyses the Web service from an internal point of view, i.e., what concept in the text best described the WSDL document content. The Web Context Extraction method describes the WSDL document from an external point of view, i.e., what most common concept represents the answers to the Web search queries based on the WSDL content. Finally, the Free Text Description Verification method is used to resolve inconsistencies with the current ontology.

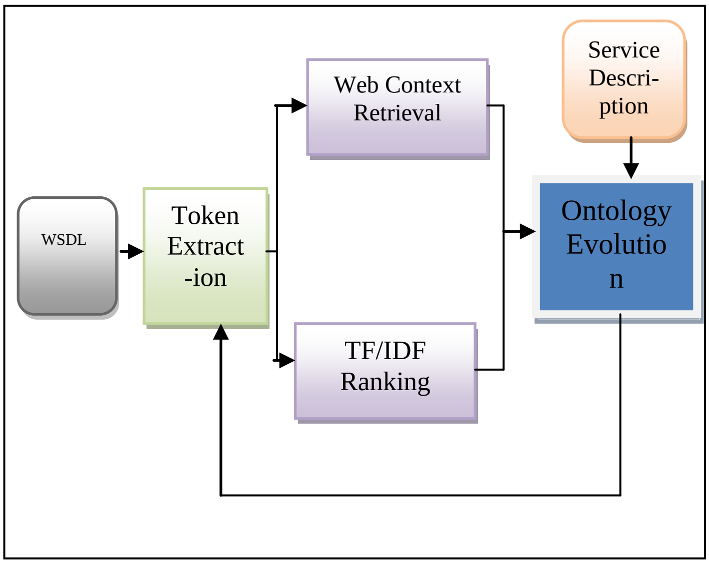

Figure 1: Our Proposed architecture of construction ontologies

The web service ontology bootstrapping process proposed in this paper is based on the advantage that a web service can be separated into two types of descriptions: 1) The Web Service Description Language (WSDL) describing "how" the service should be used and 2) A textual description of the web service in free text describing "what" the service does. This advantage allows bootstrapping the ontology based on WSDL and verifying the process based on the web service free text descriptor.

We have implemented our system in Java/J2EE as front end and MySQL as backend. We implemented and tested with a system configuration on Intel Dual Core processor, Windows XP, Apache Tomcat Web Server. We have used the following modules in our implementation part. The details of each modules for this project are as follows:

Data Extraction: In this module we develop the data extraction process using Whois. Whois is a Web service that allows domain details to be identified by based on the domain name .It maintains a web services related with operations and services.

Token Extraction: In this module we develop the token extraction process using WSDL (Web Service Description Language). WSDL document with the token list bolded. The extracted token list serves as a baseline. These tokens are extracted from the WSDL document of a Web service Whois. The service is used as an initial step in our example in building the ontology. Additional services will be used later to illustrate the process of expanding the ontology.

Term Frequency/IDF Analysis: Term Frequency/Inverse Document Frequency analysis is made in this module. TF/IDF is applied here to the WSDL descriptors. By building an independent corpus for each document, irrelevant terms are more distinct and can be thrown away with a higher confidence. To formally define TF/IDF, we start by defining frequency as the number of occurrences of the token within the document descriptor.

Web context extraction: In this module, we develop the web context extraction process. Where, the Web pages clustering algorithm is based on the concise all pairs profiling (CAPP) clustering method. This method approximates profiling of large classifications. It compares all classes' pair wise and then minimizes the total number of features required to guarantee that each pair of classes is contrasted by at least one feature.

Ontology Evolution: Ontology evolution is the last module where, the descriptor is further validated using the textual service descriptor. The analysis is based on the advantage that a Web service can be separated into two descriptions: the WSDL description and a textual description of the Web service in free text. The WSDL descriptor is analyzed to extract the context descriptors and possible concepts as described.

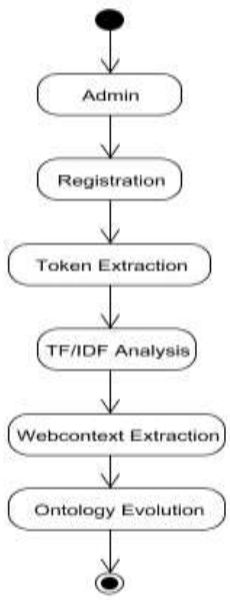

Figure 2: Proposed Flow of the project

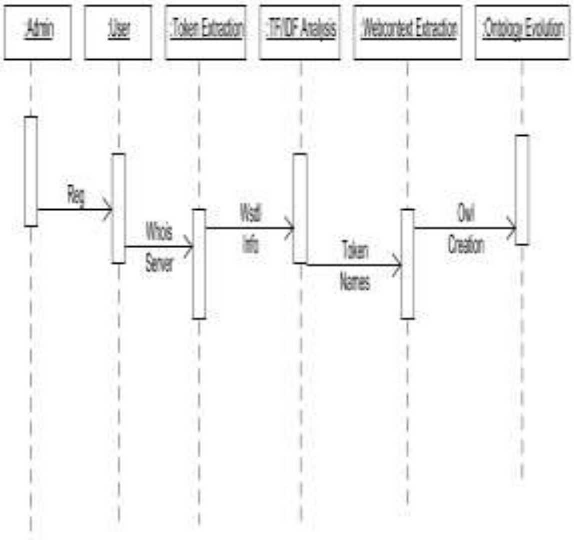

Figure 3: Sequence diagram of the proposed system

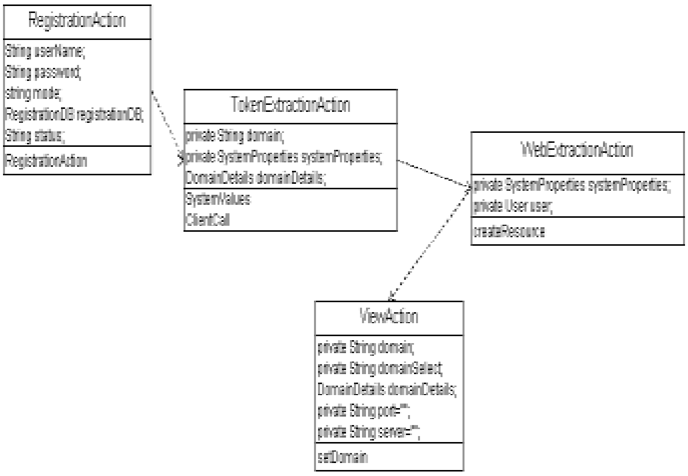

Figure 4: Class diagram of the proposed system

Figure 5 Experimental Result

Fig. 6: Term Frequency extraction part

Figure 7: Creation of OWL file.

IV. EVALUATION

The figure 7 depicted below shows the precision results of the three methods (i.e., Bootstrapping, WSDL TF/IDF, and the WSDL Context) The X-axis represents the number of analyzed web services, ranging from 1 to 392, while the Y -axis represents the precision of concept generation. It is clear that the Bootstrapping method achieves the highest precision, starting from 88.89 percent when 10 services are analyzed and converging (stabilizing) at 95 percent when the number of services is more than 250. The Context method achieves an almost similar precision of 88.76 percent when 10 services are analyzed but only 88.70 percent when the number of services reaches 392. In most cases, the precision results of the Context method are lower by about 10 percent than those of the Bootstrapping method.

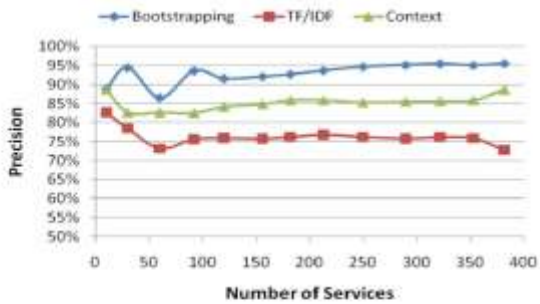

Figure 8: Evaluation Graph

ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 1, January 2013

V. Conclusions

In this project we proposed an approach for bootstrapping an ontology based on Web service descriptions. The approach is based on analysing Web services from multiple perspectives and integrating the results. Our approach takes advantage of the fact that Web services usually consist of both WSDL and free text descriptors. From the experiments conducted we prove that the proposed model is very efficient in providing semantic information in Web Service Discovery. Thus we have achieved an enhanced service-oriented relationship modelling of web service using ontologies.

VI. References

[1] Web Service definition Retrieved from http://www.w3.org/TR/ws-gloss/ Chappell D. and Jewell T. Java Web Services. O'Reilly, March 2002.

[2] Alonso G., Casati F., Kuno H., and Machiraju V. Web Services: Concepts, Architectures and Applications. Springer, 2004.

[3] Sun Microsystems, Inc. *Jini Architectural Overview, Technical Report*, 1999. http://www.sun.com/software/jini/whitepapers/architecture.pdf.

[4] N.F. Noy and M. Klein, "Ontology Evolution: Not the Same as Schema Evolution," Knowledge and Information Systems, vol. 6, no. 4, pp. 428-440, 2004.

[5] D. Kim, S. Lee, J. Shim, J. Chun, Z. Lee, and H. Park, "Practical Ontology Systems for Enterprise Application," Proc. 10th Asian Computing Science Conf. (ASIAN '05), 2005.

[6] G. Zhang, A. Troy, and K. Bourgoin, "Bootstrapping Ontology Learning for Information Retrieval Using Formal Concept Analysis and Information Anchors," Proc. 14th Int'l Conf. Conceptual Structures (ICCS '06), 2006.

[7] S. Castano, S. Espinosa, A. Ferrara, V. Karkaletsis, A. Kaya, S. Melzer, R. Moller, S. Montanelli, and G. Petasis, "Ontology Dynamics with Multimedia Information: The BOEMIE Evolution Methodology," Proc. Int'l Workshop Ontology Dynamics (IWOD '07), held with the Fourth European Semantic Web Conf. (ESWC '07), 2007.

[8] L. Ding, T. Finin, A. Joshi, R. Pan, R. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs, "Swoogle: A Search and Metadata Engine for the Semantic Web," Proc. 13th ACM Conf. Information and Knowledge Management (CIKM '04), 2004.

[9] A. Patil, S. Oundhakar, A. Sheth, and K. Verma, "METEOR-S Web Service Annotation Framework," Proc. 13th Int'l World Wide Web Conf. (WWW '04), 2004.

[10] Y. Chabeb, S. Tata, and D. Belad, "Toward an Integrated Ontology for Web Services," Proc. Fourth Int'l Conf. Internet and Web Applications and Services (ICIW '09), 2009.

[11] Z. Duo, J. Li, and X. Bin, "Web Service Annotation Using Ontology Mapping," Proc. IEEE Int'l Workshop Service-Oriented System Eng. (SOSE '05), 2005.

[12] N. Oldham, C. Thomas, A.P. Sheth, and K. Verma, "METEOR-S Web Service Annotation Framework with Machine Learning Classification," Proc. First Int'l Workshop Semantic Web Services and Web Process Composition (SWSWPC '04), 2004.

[13] A. Heß, E. Johnston, and N. Kushmerick, "ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services," Proc. Third Int'l Semantic Web Conf. (ISWC '04), 2004.