



Extraction of web service build on Annotated Capability Specifications

Dr. N. Satish

Mahendra College of Engineering,
Salem, Tamilnadu
India

Abstract: - Web service extraction focuses on the detection of helpful and fascinating web services suitable for Web service composition. This work enhance Ontology Web Language –Semantics (OWL-S) service capability specification for automatic and successful extraction of Web services. The annotated ability condition are planned with the help of environment ontology and constructed based on the connection between atmosphere entities. The Web service extraction structure utilizes annotated capability specification as a input characteristic for categorization, indexing and position. We improve capability specification using ontological annotation to get better Web service extraction and automatic detection of services successfully. Ontological annotation recommends set of linked atmosphere entities to improve the capabilities and semantics of the services. The web service categorization is done using Support Vector Machine (SVM) algorithm. Indexing is achieved using capability profile keywords in web service atmosphere entities. The ranking of Web service is performed powerfully using semantic organization of capability specification. Web services are delivered to the user according to the utmost position and the user can simply write the concerned Web services. The investigational results confirms that the proposed capability specification performs much better than the existing Web Services Description Language (WSDL) based text demonstration in terms of precision and execution time.

Keywords: - OWL-S, Capability Specifications, Composition, Web Service Extraction, Environment ontology, Classification, Service Discovery.

1 INTRODUCTION

Nowadays, the web service has gained momentum due to the fact that core businesses and outsourcing are published over the internet. Web service is a Service Oriented Architecture (SOA) paradigm supports interoperable interaction between machines through standard protocols like Hyper Text Transmission Protocol (HTTP)/Simple Object Access Protocol (SOAP) over the internet. It defines set of standards (WSDL [1], SOAP [2] and Universal Description Discovery and Integration (UDDI) [3]) to support service description, discovery and invocation in a uniform interchangeable format between heterogeneous applications. It emerges as a major technology for deploying a loosely coupled internet based applications and enables automated integration of distributed and heterogeneous software systems. It defines and identifies global elements of the web services in order to ensure interoperability between web services.

Web service standards provide syntactic interoperability but still automatic interoperability is not fully supported. Web services fail to provide unambiguous and intelligent processing of service in order to discover and compose services efficiently. The semantic web represents and exchanges information in a meaningful way to facilitate automated processing of web descriptions [4]. Ontology is a widely accepted standard for knowledge representation and identified as enabling technology for the Semantic Web [5]. The ontology supports semantically enhanced information processing and interoperability. To enrich semantics of web services, ontology based web service standards are introduced (OWL-S [6], Web Services Semantics (WSDL-S) [7], Web Service Modeling Ontology (WSMO) [8]) to enhance a machine processable and understandable

semantics. The semantic web service technology enables unambiguous description of web services for automated and machine processable form to provide better service discovery, composition and execution [9].

Semantic Web services express the necessary information for web service discovery and extraction of appropriate services in order to fulfill the interestingness and usefulness of end users. OWL-S is an ontological approach for describing web services. It provides a formal mechanism for describing the semantics of web services. The OWL-S service capabilities are modeled as four attributes inputs, outputs, preconditions, and results of the Web services. To discover, compose and execute the Web service automatically and semantically, OWL-S provides three main components to publish and discover Web services which are service profile, service model and service grounding. OWL-S supports declarative advertisements of service properties and capabilities that can be used for automatic service discovery. Capabilities and futures of Web services are described in OWL-S service profile. The central function of the OWL-S grounding is to show how inputs and outputs are realized as messages in communicable format [10]. But, the OWL-S service capabilities are not precise and expressive to elaborate service specifications.

To enhance the capability of OWL-S in terms of expressiveness and machine intelligent processing, environment ontology approach is feasible for evolving Web services capability profiling [11]. It is not just having an interface based descriptor. Here the capability specification is implemented in the form of specification profile with ontological annotations. The specification profile contains a state transition of each web service and its effect on the environment. Thus, all the capabilities are based on environmental ontology, whose characteristics and

interconnections are observable and applicable during service discovery and composition. Annotations are derived based on environment entities and its relations to expose the real capabilities of Web services. These annotated capability specification is shared among the service provider and client.

In Web service extraction bottom-up approach is adapted that need not to give the accurate requirements of requester. It may retrieve a large number of web services using the semantic relationship among them [12]. This approach is used to get the combinatorial detonation and assessment of interestingness and effectiveness of web services. Web services are recognized based on methods like direct, indirect, promotion and inhibition. Finally it retrieves a possible number of web services from OWL-S capability specification related to the approximate user requirements. Due to the tremendous increase of web services, the search becomes a time consuming process and retrieves a vast amount of irrelevant web services. This motivates the need for the efficient web service extraction framework. Finding and invoking the portable composition for web services lead to challenging activity because of the huge amount of web services availability and short content of WSDL description. The semantic web service description may have more than one interface relationships among other web services causes complex association. Therefore, web service requester did not attain the exact useful services. Various complex relationships may lead to an incompatible for delivering and identifying efficient web services. These problems can be addressed by the proposed extraction framework supported by capability profile specifications based on environment ontology.

The major contributions are:

- A extraction framework is proposed that utilize OWL-S capability specification annotated based on environment ontology to describe the web services more elaborately and extraction the number of possible web services more effectively.
- The proposed Web service extraction framework consumes annotated capability specification as a key feature of classification, indexing and ranking to make the discovery more efficient.
- To classify the Web services and reduce the searching time, SVM classifier is applied to annotated capability profile. To simplify indexing and ranking process, relationships among various web services are derived based on the capability profile keywords.

The rest of the paper is organized as follows. Section 2 discusses related work of the web service extraction concepts. Section 3 examines an overview of the proposed capability profile framework. Section 4 organizes the experimental setup, the performance evaluation and results. Finally, section 5 concludes the paper along with future enhancement.

2 RELATED WORK

The web service specifications are described semantically. These specifications examine the semantic relation between the web services. The extraction and composition technique are developed manually or semi automatically. The

traditional method of service discovery is mainly based on matching of keywords, WSDL or other service description information. The major advantage is that the realization logic is simple, but it can't identify the semantic information perfectly, the recall and the precision are low. The discovery of service is a standardized function of any service registry conforming to UDDI. Furthermore, the service discovery is a very important task for automated business service composition and integration [13]. UDDI has the limitation that the service discovery supports only functional requirements. Service community extraction for the composite web service discovery was proposed to effectively discover and dynamically compose the web services [14]. Context based approach is used to classify free text available in WSDL descriptions for service matching and composition [15], but it does not support ontological approach.

Web services classification is performed based on the category and interface descriptions given in the WSDL document. Association rules [16] [17] have been applied for building web services classifier for automatically classifying web services. This determines the category of a web service and set of predefined categories. The main goals of this work are 1) To build a classification system using association rule which is applied on the category of Web services, operations and its textual documentation namely argument definitions and comments written by developers. 2) To analyze the importance of a term to a particular category varies with its frequency and appearance in other documents. 3) To analyze whether this system gives better accuracy other than the primitive methods used for classification or not. But this classification mechanism does not support semantic web service extraction and classification. A Naïve Bayes classifier called METEOR-S, which extracts the names of all operations and arguments declared in WSDL documents of pre-categorized web services [18].

The extraction based on WordNet based Web Service Similarity Extraction (WSSM) uses the clustering approach to find the efficient services [19]. The clustering is constructed using the word matrix but it may not support the verb in the semantics. Woogle is a search engine based on similarity search for web services [20]. Consequently, it leads to better search engine result but it does not provide support for semantic web services. The spectrum clustering [21] may lead a simple search process, although it does not guarantee the good quality service selection. The semantic web service composition in IRS-III approach contains the semi automated tool to guide the user step by step process. Therefore, this approach is considered as a semi automated process and user need to do a manual composition [22]. Association rule extraction contains knowledge repository and rules. These rules can help to know which service are most related, best performance, quality and usage among the available services in compositions [23].

Semantic web services are facing some problems and limitations because of lack of automation and different languages [24]. Therefore, these problems limit their improvement and progression. There is no agreement on which language has to be used while describing the semantic information. It mainly uses OWL-S and WSMO. The WSMO [8] provides a conceptual model for the semantic markup of Web Services together. Web Service Modeling

Language (WSML) provides the formal logic-based grounding of information [25] but, it lacks of formal semantics of service capabilities.

3 Web Service Extraction Framework

3.1 Capability Specification for Web Services using atmosphere Ontology

Construction and detailed algorithm description is presented in [10]. The environment ontology based capability specification is similar to WSMO. Because of loosely coupled web service environment entities are shared among multiple web services and it supported to describe an expressive capability specification in service interface [26][27].The environment entity of a web service is considering a state transmission among various domain entities surrounding that web service. Environment ontology satisfies the difficulty of different concept levels. The environment ontology construction looks like tree based structure. It contains the domain Tree like Hierarchical State Machine (THSM) and contains the relationship between the various THSM [28]. Then it obtains the specific domain Hierarchical State Machine (HSM) which is derived from domain THSM and uses the relationship with other THSM domains. The capability specification of the web service is represented using the state transitions among the surrounding entities.

3.2 Classification approach

Web services may belong to various categories and business domains. For efficient web service identification and retrieval, classification provides the foundation on which rest of the process has to be built. There is a large number of algorithms exist to classify web services such as string matching, clustering and keyword identification. This research applies SVM to classify the web service in terms of keywords that are presented in capability profile specifications. The annotated capability profile contains more detailed information about web services and it may lead to better classification of web services. Classification may consist of predefined categories and domains. Web services are assigned to the specific categories based on the top-most keywords identified. This categorization is useful for future identification and retrieval of web services. SVM is a supervised machine learning algorithm and it should contain the predefined set of training data for categorization. Testing web services (i.e.) uncategorized services are categorized based on predefined training data [29].

First, extract the capability profile keyword from web service interfaces using recognition methods. The Term Frequency (TF) is calculated from extracted keywords in the whole web service interfaces and capability specification of each web service. The Invert Document Frequency (IDF) is calculated upon comparing the capability profile keyword in a single web service and the whole testing web services. Finally, the capability profile keyword weight is calculated and categorized based on the weight. The training web services have some predefined capability profile keywords while categorization takes place.

Input:

Predefined training web services with capability profile denoted as **Twscp** and the training capability profile keyword is **Tcpk**.

Parameters:

1. Training web services with capability profile **Twscp** and training capability profile keyword **Tcpk**.
2. Testing web services with capability profile description **WScp** and weighted capability profile keywords **W (CPk)**.

Output:

Categorized web services based on capability profile

$$Cws = \{WScp1, WScp2... WScpn\};$$

Step 1: Consider a web service description OWL-S with capability profile (WScp)

$$WScp = \{d1, d1, d3.....dn\};$$

Step 2: Extract a capability profile keywords (CPk) from WScp.

$$CPk = \{k1, k2, k3 ...kn\};$$

Step 3: Calculate frequency of CPk in a WScp

$$Freq (CPk, WScp) = \frac{(CPk, d)}{|WScp|}$$

Step 4: Calculate the TF of CPk

$$TF_{cpk} = \frac{Freq (CPk, WScp)}{|WScp|}$$

Step 5: Calculate the invert capability profile of keywords frequency

$$IDF (CPk) = \log \frac{|d|}{|dn \in WScp|}$$

Step 6: Calculate the TF/IDF for CPk

$$W(CPk) = TF_{cpk} \times IDF^2 (CPk)$$

Step 7: According to different weights (W) the capability profile keywords are compared to the training capability profile keywords Tcpk.

$$WScp = compare (W (CPk), Tcpk);$$

Step 8: According to the SVM the predefined training capability keyword and testing capability profile is considered for classification. Instead of considering hyperplane in SVM, the capability profile keyword is used. If keyword is related to the capability profile keyword, web service assigned to that category.

Step 9: Categorized web services based on capability profile.

$$Cws = \{WScp1, WScp2... WScpn\};$$

3.3 Indexing web Services

The web services index is constructed based on annotated capability profile extracted from environment ontology. Web service index improves retrieval efficiency and composition [32]. The index may contain the top-most occurring identified keywords from capability profile. Each and every web service has been constructed in the service index. An OWL supports richer semantics and facilitates more flexible automation of services in terms of use. The indexing includes capability profile keyword and web services name.

Input: Number of web service domains with capability profile specification.

$$Cws = \{WScp1, WScp2 \dots WScpn\}.$$

Output: Capability profile index Iws.
Consider a capability profile keyword k.

```

Foreach k ( Cws do
  If k( WScp then
    Add (k, d);
    Iws ( add (k, d);
  Else
    Continue loop;
  End if
  Iws = {Iws1, Iws2... Iwsn};
End for
    
```

The keyword matching algorithm plays a vital role to match the indexes. The specific keyword may match more than one web services in the indexing semantic description of capability profile. Therefore, it will provide a better web service for the upcoming ranking process.

3.4 Web Service Ranking

As more than one web service leads inefficient web service identification. Hence, it needs to be ranked for finding the best service. The proposed ranking approach uses association relationship among capability profile of web services. The highest ranking based order web services are retrieved from the web service ranking table.

Ranking using capability profile semantic association

Relationships exist among the domain entities are complex. This relationship is expressed as semantic associations. The semantic relationship may be found using capability profile association among web services [33].

Ontology represents the associations that span across multiple domains. These are paths connect at least two entities and may involve multiple intermediate entities and relations. Ranking of the capability profile semantic associations facilitate the selection of most relevant relationships among entities. This may have a large collection of association with other web services capability profile. Consider domain ontology below.

$$D1 = \{ws1, ws2 \dots wsn\}; \text{ and } D2 = \{Ws1, Ws2 \dots Wsn\};$$

$$RELOfCP = \{(ws1, Ws2) \dots (wsn, Wsn)\};$$

The single web service capability profile may have more than one association relationship over other capability profiles. Thus, it needs to find the relationship with every web service.

Foreach WS (Cws do

For WScpk (Cws then

```

Find relations (WScpk, Cws); // finding the relationships
Calculate (CPrel); // calculating the number of association
Rank (CPrel); // assigning ranks
Display Rank = {CPrel1, CPrel2 ...};
    
```

End for

End for

After finding the relationship with every capability profiles, calculate the association status for which one having more. Then rank web services based on the highest number of association. Finally, display the ranked web services. The retrieved web service's details and interface relationships are extracted from the web service repository. Web service description is represented using OWL-S. The results are displayed as a highest ranking order and user picks any one from given services.

4 Performance Analysis

This section describes the experimental setup and provides empirical analysis on the capability profile framework. The proposed capability profile web service framework is compared with existing ontology based discovery in terms of execution time, precision and recall.

4.1 Experimental Setup

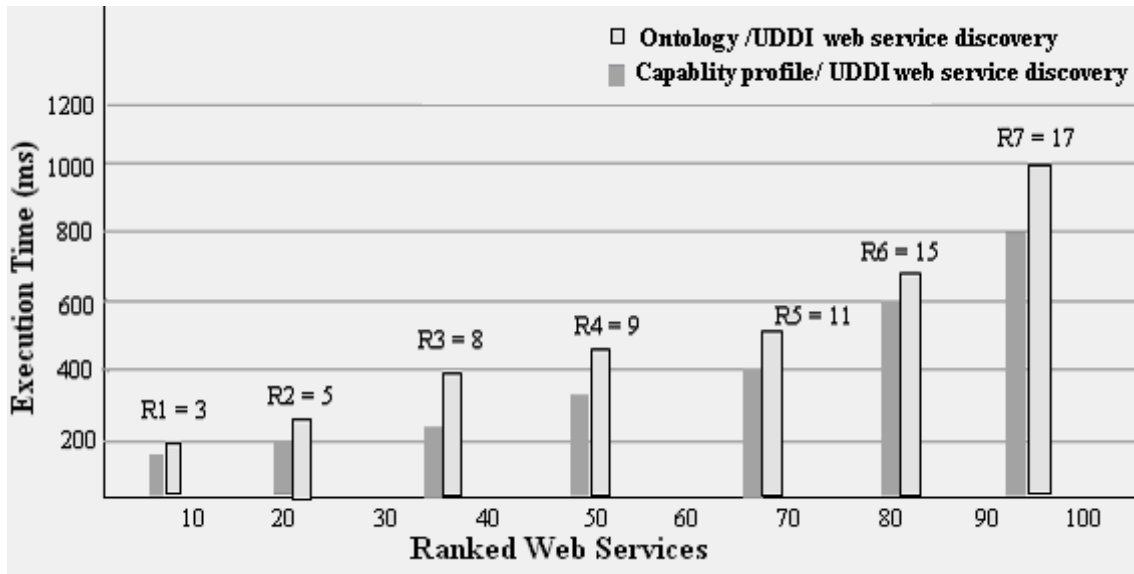
The web services are deployed in open source Java Enterprise Edition / Netbean /Glassfish/MYSQL environment for the experimental evaluation. Experimental setup contains 600 web services classified into 30 categories. For each Web service, the repository provides annotated capability profile based on environment ontology. The proposed framework retrieves keywords from the annotated capability profile. SVM classifies web services based on the keywords. Indexing and ranking is performed to order and rank services for efficient discovery and composition.

Performance – Execution Time

In this experiment execution time is calculated for the requested service. Here, consider 50 queries each with two inputs and one output of required web services. Execution time of a web service does not depend on the number of web services published in the repository. Hence, the execution time depends only on finding and linking required web services from the ranked web services collection. This implementation proves that the time required to find a service can be achieved by the relative constant time. Therefore, time to attain web services depends on time to link the web services from ranked web services.

$$WS \text{ ExeTime} = (WSOP + WSIP) * (Time_{Find} + Time_{Link})$$

Graph 1



Graph 1 shows that the execution time of web services increasing linearly according to the various requester web services query matches (R1= 3, R2 = 5, R3 = 8, R4 = 9, R5 = 11, R6 = 15, R7 = 17) in ranked web services. Consequently, the execution time increased if the number of requested web services match increases in ranked web services collection. Since, the proposed OWL-S annotated capability profile based framework enhances the execution time compared to ontology based discovery. Thus, the proposed capability profile web service description and discovery is much better than the existing ontology based web services discovery.

To evaluate the effectiveness of the capability profile based web service extraction framework, for a given web service request, compare the precision and recall of finding requested Web services. The graph 2 explores the precision of annotated capability profile web service versus WSDL extraction framework based on the sample collection of web services. The precision and recall results are presented on a graph method.

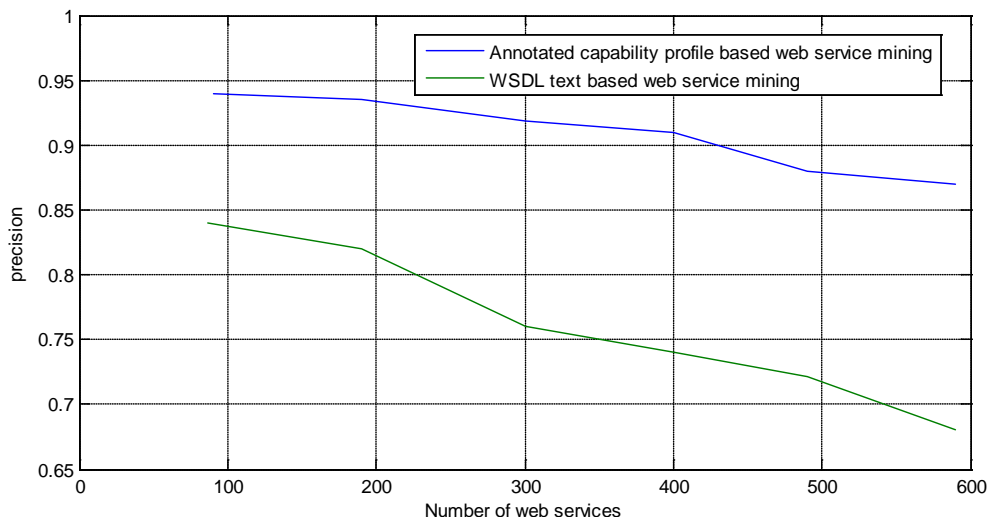
Precision

This is a fraction of the returned results that are relevant. Precision performance metric is commonly used to measure accuracy of information recovery. It measures the fraction of the correct web services among retrieved recommended web services.

$$\text{Precision} = \frac{\text{Number of Related Web services Extracted}}{\text{Number of Extracted Web Services}}$$

Graph 2 shows the precision of the proposed framework and existing WSDL based extraction framework for 50 queries executed. The precision of the result reveals that the proposed extraction framework is more accurate in retrieving Web services.

Graph-2 Precision



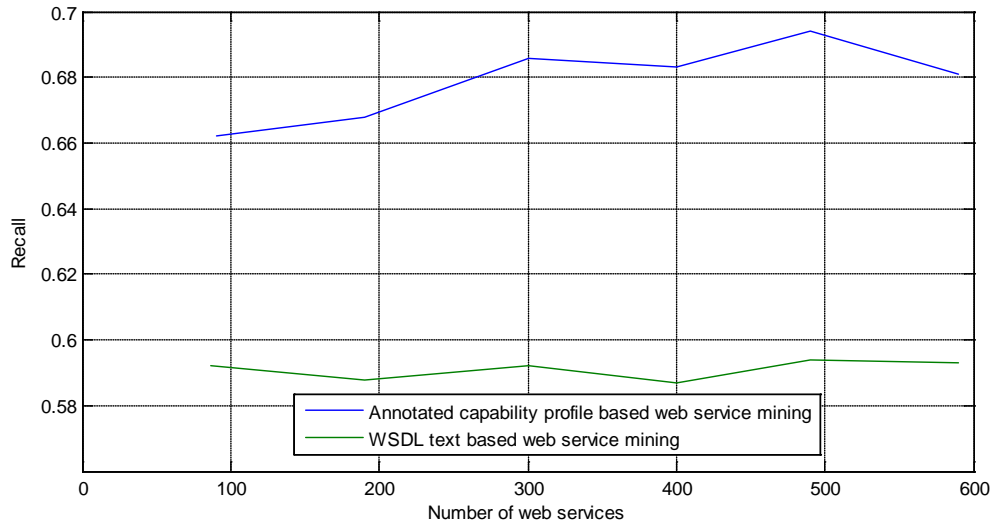
Recall

This is a fraction of the relevant services in the collection that were returned by the system

$$\text{Recall} = \frac{\text{Number of Related Web services Extracted}}{\text{Number of Related Web services}}$$

The graph-3 presents the recall of the relevance of web services retrieval. The extraction framework for a given number of services presented on X-axis and recall of the given framework is presented on the Y-axis. The recall is compared with a capability based extraction framework and WSDL text based extraction framework. The annotated capability profile based web service extraction framework recall is more accurate than WSDL based framework.

Graph 3 -Recall



The proposed annotated capability profile framework avoids the data redundancy and reduces the total service execution time. As a result, comparing with web service extraction methods, the capability profile web service discovery strategy is put forward that the precision and recall ratio have been improved. The quality of the capability profile based description has improved the efficiency of discovery and composition of web services. This indicates that the web services based on WSDL is not enough for extraction web service. The results summarize the significance of adding annotated capability profile descriptions to the web services when supplying the service for public use.

5 CONCLUSION

This work introduces extraction framework based on annotated capability profile description for the rich and enhanced meaning representation of web services. This work contributes to an effective discovery and composition of web services. Capability specification is enriched using ontological annotation to improve the effectiveness of Web service extraction. The annotated capability specifications are designed using environment ontology which is constructed based on environment entities. The constructed Web service extraction framework uses annotated capability specification for classification, indexing and ranking. The experimental results show an improved execution time, precision and recall. The indexing and ranking with the uses of capability profile performs much better than existing WSDL based extraction.

REFERENCES:

[1] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "WSDL Web Services Description

Language," W3C Candidate Recommendation, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, 2001.

- [2] SOAP-Simple Object Access Protocol, www.w3.org/TR/2000/NOTE-SOAP-20000508/ -2000.
- [3] UDDIV3, <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3>, 2004.
- [4] T. Berners-Lee, J. Hendler, and O. Lassila "The Semantic Web", Scientific American, May 2001.
- [5] N. Guarino, "Formal Ontology in Information Systems", Proceedings of FOIS 1998.
- [6] OWL-S- Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.1/overview/>, 2004.
- [7] WSDL-S, W3C, <http://www.w3.org/Submission/WSDL-S/>, 2006.
- [8] J. Bruijn et al., "Web Service Modeling Ontology (WSMO)," <http://www.wsmo.org>, 2006.
- [9] S.A. McIlraith, T.C. Son, and H. Zeng, "Semantic Web Services," IEEE Intelligent Systems, vol. 16, no. 2, 2001, pp. 46-53.
- [10] D. Martin, M. Burstein, D. Mcdermott, S.McIlraith, M. Paolucci, K.Sycara, D.L. Mcguinness, E. Sirin, and N. Srinivasan, "Bringing Semantics to Web Services with OWL-S," World Wide Web, Vol. 10, No. 3, 2007, pp. 243-277.
- [11] Puwei Wang, Zhi Jin, Lin Liu, Guangjun Cai "Building toward Capability Specifications of Web Services Based on an Environment Ontology", IEEE Transactions on Knowledge and Data Engineering, Volume 20, No. 4, 2008, pp. 547-561.
- [12] George Zheng and Athman Bouguettaya, "Service Extraction on the Web", IEEE Transactions on Service Computing, Vol 2, No 1, 2009, pp. 65-78.
- [13] Qianhui Althea Liang, Jen-Yao Chung, Steven Miller, Yang Ouyang, "Service Pattern Discovery of Web Service Extraction in Web Service Registry-Repository", Proceeding of the IEEE International Conference on e-Business Engineering, 2006.

- [14] Xizhe Zhang, Ying Yin, Mingwei Zhang, Bin Zhang, "A Composite Web Services Discovery Technique Based on Community Extraction", IEEE Asian-Pacific Service Computing Conference, December 2009, pp.445-450.
- [15] Aviv Segev, Eran Toch, "Context-Based Matching and Ranking of Web Services for Composition", IEEE Transactions on Services Computing, Vol 2, issue 3, 2009, pp. 210-222.
- [16] Shailja Sharma and Shalini Batra, "Applying Association Rules For Web Services Categorization" International Journal of Computer and Electrical Engineering, Vol. 2, No. 3, 2010.
- [17] Ali Shaikh Ali, Omer F. Rana and Ian J. Taylor, "Web Services Composition for Distributed Data Extraction", Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'05), 2005.
- [18] Nicole Oklham, Christopher Thomas, Amit P,sheth and kunal verma, "METEOR-S web service annotation framework with machine learning classification", In Proceedings of First international Workshop on Semantic Web Services and Web Process Composition (SWSWPC), 2004.
- [19] Xianyang Qu, Hailong son, Xiang Li, Xudong Lu, Wei Lin, "WSSM: A WordNet based Web Service Similarity Extraction", Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009.
- [20] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, "Similarity Search for Web Services", Proceeding VLDB '04 Proceedings of the Thirtieth international conference on Very large data bases - Volume 30, 2004, pp. 372-383.
- [21] Xizhe Zhang, Ying Yin, Mingwei Zhang, Bin Zhang "Web Service Community Discovery Based on Spectrum Clustering" Proceedings of the International Conference on Computational Intelligence and Security, 2009.
- [22] Hakimpour, F., Sell, D, Cabral, L., Domingue, J., Motta, E., "Semantic Web Service Composition in IRS III: The Structured Approach", Proceedings of the International IEEE Conference on ECommerce Technology, 2005.
- [23] Shahab Bayati, Ali Farahmand Nejad, Sadegh Kharazmi, Ardeshir Bahreininejad, "Using Association Rule Extraction to Improve Semantic Web Services Composition Performance" Proceedings of the 2nd International Conference on Computer, Control and Communication, 2009.
- [24] Miguel Angel Corella, Jose Maria Fuents, Pablo Castells, "Semi-automatic web service generation and classification", 2nd International Workshop on Semantic and Dynamic Web Processes (SDWP), 2005.
- [25] Uwe Keller,Rubben Lara,Holger Lausen,Axel Polleres,Livia Predoiu,Ioan Toma "Semantic Web Service Discovery" In Proceedings of the 8th International Semantic Web Conference, USA, 2009.
- [26] P. Wang, Z. Jin, and L. Liu, "An Approach for Specifying Capability of Web Services based on Environment Ontology," Proceedings of IEEE International Conference on Web Services, 2006.
- [27] P. Wang, Z. Jin, and L. Liu, "On Constructing Environment Ontology for Semantic Web Services," Proceedins of 1st International Conference on Knowledge Science, Engg. and Management, 2006.
- [28] R. Alur, R. Kannan, S. Yannakakis, and M. Yannakakis, "Communicating Hierarchical State Machines," Proceedings of 26th International Colloquium on Automata, Languages and Programming, 1999.
- [29] Joachims, T, "Text categorization with Support Vector Machines: Learning with many relevant features in Machine Learning", Proceedings of Tenth European Conference on Machine Learning, 1998, pp. 137--142.
- [30] Aabhas V. Paliwal, Nabil R. Adam, Hui Xiong, Christof Bornhövd, "Web Service Discovery via Semantic Association Ranking and Hyperclique Pattern Discovery", Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2006.
- [31] B. Medjahed, A. Bouguettaya, and A.K. Elmagarmid, "Composing Web Services on the Semantic Web", The International Journal on Very Large Data Bases, Vol. 12, Issue 4, 2003.
- [32] Marco Aiello, Christian Platzer, Florian Rosenberg, Huy Tran, Martin Vasko, Schahram Dustdar "Web Service Indexing for Efficient Retrieval and Composition" Proceedings of the 8th IEEE International Conference on E-Commerce Technology, 2006.
- [33] Wenge Rong, Kecheng Liu and Lin Liang "Personalized Web Service Ranking via User Group combining Association Rule", Proceedings of the IEEE International Conference on WebServices,2009.