Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CONTENT-BASED SEMANTIC WEB SERVICE DISCOVERY, AN EMPIRICAL ANALYSIS WITH IMPLEMENTATION

[1]A.GNANASEKAR, [2]R.M.SURESH
[1]Assistant Professor, R.M.D Engineering College, Department of Computer Science and Engineering, Chennai, INDIA
[2]Principal, Sri Muthukumaran Institute of Technology, Department of computer Science and Engineering, Chennai, INDIA
E-mail: [1]gnanasekar14@hotmail.com , [2]rmsuresh@hotmail.com

ABSTRACT

Web Service discovery and composition is one of the key objectives of Service Oriented Architecture (SOA) which provides a loosely-coupled environment and enables flexible assembly of existing web services. Service discovery plays a vital role in the overall process of the service discovery and composition. The Web Ontology Language for Services (OWL-S) has emerged as the de facto standard for implementing business process from existing semantic web services. The semantic discovery is a challenging problem and attracts much attention both from industry and academia. This work focuses on the semantic web service discovery using empirical analysis.

Keywords: *Semantic Web Services, Service Discovery, Ontology, Service Oriented Architecture, OWL-S.*

1. INTRODUCTION

Service Oriented Architecture (SOA) has evolved to become a promising technology for the integration of disparate software components using internet protocols. The standards such as Web Services Description Language (WSDL) [13], Universal Description Discovery and Integration (UDDI) [15] and Simple Object Access Protocol (SOAP) [14] are three core standards for web services description, registration and binding respectively. Nowadays, a large number of companies and organizations are implementing their applications over internet. Thus, the ability to efficiently, effectively and dynamically discover and compose inter-organizational and heterogeneous services on the Web is an important step towards the development of the web services applications. When an atomic web service cannot satisfy the user's requirements, there should be a possibility to combine existing services together in order to meet the user request. A composite service includes a set of atomic services together with the control and data flow among the services. The rest of the paper is organized as follows: In section 2 the related works and the motivation of this work is explained. In section 3, the content based service discovery by using the WordNet ontology is explained with example. Section 4 comprehensively and empirically analyzes the service discovery with a test collection and section 5 briefs the conclusions and the future work.

2. RELATED WORK

The field of web service composition is very active in which the service discovery plays a vital role. Initial work has been done in discovering web services directly by querying syntactic web services through their WSDL documentation. A web service is an interface that describes a set of operations that can be accessed over the network through standardized XML messaging [7]. A client program finds a Web service that can fulfill certain requirements, and acquire a detailed specification from WSDL [13]. The client sends a request to the server through a standard message protocol (SOAP) [14] and in return receives a response from the server. When a service is published, there is a lack of concrete application ontology which is used to annotate the associated WSDL documents. In general semantically-lexical database called WordNet [19] is used to approximately resolve this problem. Recently, WordNet has been widely used in the information retrieval community by calculating and comparing their semantic similarity between words. WordNet can also be seen as ontology for natural language processing [5], and also used in semantic web service discovery. Automated composition method that explicitly uses the functional semantics of services to resolve the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

time complexity [10]. A framework IRS-III (Internet Reasoning System) is used for creation and running of Semantic Web Services, which takes a semantic broker-based approach for the mediation between service requesters and service providers [11]. Web service composition system can help us to automate the business process, from specifying functionalities, to develop the executable workflows that capture non-functional requirements to deploy them on a runtime infrastructure [12]. Aviv Segev et al. [3] proposed a context-based semantic approach to the problem of matching and ranking of web services for possible service composition and provided a numeric estimation about the possible composition to the designer. An OWL-S service profile ontology based framework is used, for the retrieval of web services based on subsumption relation and structural case-based reasoning, which performs domain-dependant discovery [4]. Tamer Ahmed Farrag et al. [6] proposed a mapping algorithm that helps to facilitate the integration of the current conventional web services into the new environment of the Semantic Web. This has been achieved by extracting information from WSDL files and using it to create a new semantic description files using OWL-S. Hai Dong et al [20] proposed a conceptual framework for a semantic focused crawler, which combines the speciality of ontology-based metadata classification from the ontology, based focused crawlers and the speciality of metadata abstraction from the metadata abstraction crawlers, in order to achieve the goal of automatic service discovery, annotation, and classification in the Digital Ecosystems environment. Antonio Brogi [8], had emphasized the importance of behavioral information in service contracts, that inhibits the possibility of guaranteeing the service interactions and highlighted the limitations of currently available service registries which do not take into account the behavioral information. In the research to web services, a lot of initiatives have been conducted with the intention to provide platforms and languages for Web Service Composition (WSC) such as Business Process Execution Languages for Web Services (BPEL4WS) [1]. Presently some languages have the ability to support semantic representation of the Web services available on the internet such as the Web Ontology Language for Web Services OWL-S [2] and the Web Service Modeling Ontology WSMO [9].

Using OWL-S, web services are extended with an unambiguous description by relating the input and output parameters of the service to common concepts defined in *Web Ontology* [17], which serve as the key mechanism to globally define and reference concepts. In this paper, architecture is proposed for the content based semantic web service discovery.

2.1 Motivation

Most of the works in this area had used a user query in the form of input and/or output for the service discovery, but not the textual content. This limitation in the service discovery process has given the motivation to propose a content based service discovery method. In order to facilitate service discovery based on the user given content, a new method is proposed in this paper which reads the textual content and after a refinement process it discovers more relevant web services from the registry.

3. SEMANTIC WEB SERVICE DISCOVERY

As shown in Figure 1, the user given content will passes through the following refinement steps to extract the annotations (nouns) that are used to discover the services.

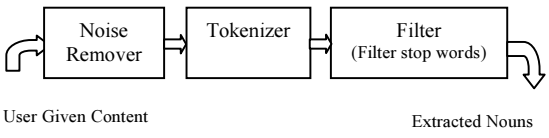

Figure 1: Refining Process Modules

- Removing noise from the query
- Tokenizing the query
- Filtering "stop" words

The user query is content or a sentence which may contains several special characters like comma (,), dot (.), plus (+) etc. These special characters are indeed, not useful for the service discovery. So these types of words are considered as a noise and these are to be removed from the content to enable the service matchmaking. The noise remover module removes the noise, if any, present in the user given content. Once the noise is removed from the content, the remaining words are to be tokenized. The tokenizer will analyze the noise free content and separate each word as a token. The result of the tokenizer is a set of words which may also contain "stop words". Stop words are nothing but verbs in natural language, such as "want", "which", "is", "what" etc. These stop words are also indeed not useful for the service discovery. So these types of stop words are to be removed from the tokenized content. The noise free tokenized

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

content is fed to the filter module to filter the "stop words". The noise free and "stop word" free content is used for the service discovery. For example, suppose a user want to travel abroad and assume that the user has given the content "*I want to travel*" as the input. The noise remover module in the proposed framework will first check the existence of noises and are will be removed from the content. Since the given sentence is noise free and this content will then pass through the tokenizer module. Tokenizer will make the noise free content as a group of tokens. These tokens are then passed through the filterization module to find and remove the "stop words". The words such as "*I*", "*want*" "*to*" are all the stop words and are removed from the content. Finally, the word "*travel*" will remain in the original content. The word travel is given as the input and the services which are used for the travel booking will be discovered by the system.

4. EMPIRICAL ANALTSIS

An experiment has been conducted with the dataset for some selected text queries. A dedicated personal computer running Windows XP with 2GB RAM is used for the experiment. The proposed matching algorithm is implemented in Java. Apache web server is used to access all ontologies used by the matchmaker and thus to avoid the incorrect path problems to access ontologies used to describe service IO concepts. OWL-S API [18] is used to parse the service profile ontology. OWL-S API is the most comprehensive library for working with OWL-S documents. For reasoning operations Pellet reasoner [16] has been used. Pellet is the mapping engine that matches the service advertisements with the service requests. To prove the validity of the matching algorithm, the same algorithm is applied to the dataset which was obtained from the OWL-S service retrieval test collection, OWL-S TC (version 4.0). This dataset consists of 1083 semantic web services from nine different domains namely, education, medical care, food, travel, communication, economy, weapons, geography and simulation. OWL-TC provides a set of 42 test queries covering all the nine domains.

The user given text contents, the extracted nouns and the number of services returned by the experiment is given in Table 1 and Table 2.

Table 1: Experimental Result For The User Given Text Contents

Data Set	Text Content		Extracted Nouns		No. of Services Discovered
	Input	Output	Input	Output	
1	I Want to purchase a Book	Want to Know the Price	Book	Price	20
2	I Want to purchase a Car	Want to Know the Price	Car	Price	18
3	I Know how many Miles	Need to know the Kilometers	Miles	Kilometers	1
4	Hospital names	Want to know the investigating details	Hospital	Investigating	4
5	I want to go to a city in a country	Want to know the Hotel	City, Country	Hotel	6
6	I will give the Zip code	I want to know the Latitude, Longitude	Zip code	Latitude, Longitude	1
7	I have a degree and I apply to the government	I want to know the scholarship details	Degree, Government	Scholarship	8
8	I came to know about a novel	I want to know the author name	Novel	Author	14
9	I know the name of a Grocery store	Lists the available food	Grocery store	Food	4
10	A person want to purchase a book using credit card account	Want to know the price.	Person, Book, Credit card account	Price	23

Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Experiment And Analysis Number Of Services Discovery By Service Discovery Algorithm.

Test Case	Query	Discovered Services	Relevant services Retrieved	Relevant Services not Retrieved	Irrelevant services Retrieved	Precision (%)	Recall (%)
1	book_price_service.owls	17	10	5	7	58.82	66.67
2	1personbicyclecar_price_service.owls	14	9	4	5	64.29	69.23
3	bookpersoncreditcardaccount__service.owls	82	68	14	14	82.93	82.93
4	bookpersoncreditcardaccount_price_service.owls	16	10	7	6	62.50	58.82
5	car_price_service.owls	8	3	26	5	37.50	10.34
6	citycountry_hotel_service.owls	9	5	4	4	55.56	55.56
7	country_skilledoccupation_service.owls	10	7	6	3	70.00	53.85
8	dvdplayermp3player_price_service.owls	13	8	4	5	61.54	66.67
9	EBookOrder1.owls	6	4	3	2	66.67	57.14
10	fall_down_pill.owls	1	1	0	0	100.00	100.00
11	getLatitudeAboveSeaLevelOfLocation.owls	2	2	2	0	100.00	50.00
12	getDistanceBetweenCitiesWorldwide.owls	1	1	0	0	100.00	100.00
13	getLocationOfCityState.owls	1	1	1	0	100.00	50.00
14	getLocationOfUSZipcode.owls	2	2	1	0	100.00	66.67
15	getZipcodeForUSCity.owls	1	1	0	0	100.00	100.00
16	getMapOfUSAddress.owls	1	1	0	0	100.00	100.00
17	getSunsetSunriseTimeOfLocation.owls	1	1	0	0	100.00	100.00
18	getLocationOfCityState.owls	2	2	1	0	100.00	66.67
19	getMapOfUSAddress.owls	3	3	4	0	100.00	42.86
20	geographical-region_map_service.owls	7	2	4	5	28.57	33.33
21	geopolitical-entity_weatherprocess_service.owls	2	2	1	0	100.00	66.67
22	governmentdegree_scholarship_service.owls	9	4	4	5	44.44	50.00
23	governmentmissile_funding_service.owls	7	1	15	6	14.29	6.25
24	grocerystore_food_service.owls	3	1	8	2	33.33	11.11
25	hospital_investigating_service.owls	3	2	3	1	66.67	40.00
26	lock_door.owls	49	42	6	7	85.71	87.50
27	maxprice_cola_service.owls	6	3	9	3	50.00	25.00
28	mileToKilometerConverter.owls	1	1	0	0	100.00	100.00
29	novel_author_service.owls	13	8	7	5	61.54	53.33
30	open_door.owls	39	39	9	0	100.00	81.25
31	preparedfood_price_service.owls	9	7	4	2	77.78	63.64
32	publication-number_publication_service.owls	5	2	2	3	40.00	50.00
33	recommendedprice_coffeewhiskey_service.owls	3	1	3	2	33.33	25.00
34	researcher-in-academia_address_service.owls	3	2	3	1	66.67	40.00
35	shoppingmall_cameraprice_service.owls	2	1	1	1	50.00	50.00
36	surfing_destination_service.owls	4	3	0	1	75.00	100.00
37	surfinghiking_destination_service.owls	9	3	5	6	33.33	37.50
38	surfingorganization_destination_service.owls	7	6	3	1	85.71	66.67
39	title_comedyfilm_service.owls	5	4	3	1	80.00	57.14
40	title_videomedia_service.owls	8	5	2	3	62.50	71.43
41	university_lecturer-in-academia_service.owls	7	4	1	3	57.14	80.00
42	userscience-fiction-novel_price_service.owls	16	11	4	5	68.75	73.33

4.1 Degree Of Match

In the matching algorithm, the degree of match between two inputs or two outputs depends on the match between the concepts that represent them.

Journal of Theoretical and Applied Information Technology
30th June 2014. Vol. 64 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The match is based on the relation between these concepts in their OWL ontologies. The matched services basically have associated with one of the three degree of matches: exact, plug-in, subsume. It may be assumed that, Cinput(R) and Coutput(R) respectively are the input and output concepts of a request R as well as Cinput(S) and Coutput(S) are the input and output concepts of a service S. The proposed matchmaking algorithms uses only 'exact' degree of match. The exact degree of match between the request R and services S is that, if Cinput(S) ≡ Cinput(R) Λ Coutput(S) ≡ Coutput(R) then service S exactly matches the request R.

4.2 Precision And Recall

The performance of discovery algorithm is measured using the information retrieval metrics called the precision and recall of the services returned by the algorithm. Precision is the ability to retrieve the most precise services. Higher the precision means better the relevance and more precise results, but may imply fewer results returned. The precision of n results of a query q is computed as follow.

$$\frac{A}{A+B} \times 100 \qquad (1)$$

Where, A is number of relevant services retrieved and B is number of irrelevant services retrieved. Thus, precision is the ratio of the total number of relevant services retrieved to the total number of irrelevant and relevant services retrieved. The precision of the services returned by the algorithm is shown in Figure 2.

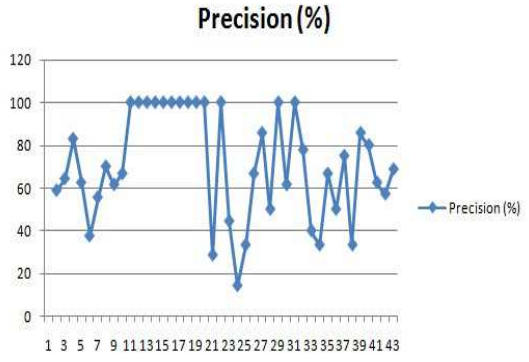

Figure 2: Precision of the discovered services

Recall means the ability to retrieve as many services as possible that match or are related to a query. The recall of n results of a query q is computed as follows.

$$\frac{A}{A+C} \times 100 \qquad (2)$$

Where, A is number of relevant services retrieved and C is number of relevant services not retrieved. Hence, recall is the ratio of the number of relevant services retrieved to the total number of relevant services in the repository. The recall of the services returned by the algorithm is shown in Figure 2.

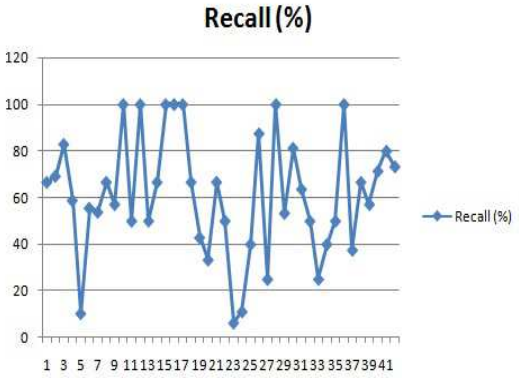

Figure 3: Recall of the discovered services

5. CONCLUSION

This paper presents a framework for serviced discovery and composition of semantic web services. The service discovery phase finds relevant web services from the registry based on the content given by the user. The nouns, which are used to find the web services, are fed into the WordNet to find the possible meanings. The semantic meanings return by the WordNet ontology is used to search the registry. The web service composition method supports both synchronous and asynchronous web services and can be able to dynamically compose several web services. Since more than one service exists with similar functionalities, the proposed system will not select best out of this multiple services. The user has to select the best service based on the non-functional QoS parameters. One of the limitations of the proposed method is the users are restricted to enter the input and output contents separately. For instance, a user wants to know the price of a book, he/she need to enter only the text like "I want to know the Price of the Book". Here, book is the required input and the price is the user expected output. But in the proposed method the user should enter this text into two separate entries as shown in the dataset 1 in Table 1. In future, these limitations would be avoided, so that the user can enter the content as a single entry and the system will automatically select a best choice.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D.Smith, S.Thatte, I.Trickovic, and S.Weerawarana , "Business Process Execution Language for Web Services Version 1.1",- *http://msdn.microsoft.com/en-us/library/ee251594(v=bts.10).aspx*, May 2003.

[2] David Martin et al., "OWL-S Semantic Markup for Web Services,"- *http://www.w3.org/Submission/OWL-S/*, November 2004.

[3] Aviv Segev and Eran Toch, "Context – based Matching and Ranking of web services for composition", *IEEE Transaction on Services Computing*, Vol. 2, No. 3, 2009, pp. 210-222.

[4] Georgios Meditskos and Nick Bassiliades, "Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 2, 2010, pp. 278-290.

[5] Kaijun Ren, Jinjun Chen, Nong Xiao, Junqiang Song, Jinyang Li and P.R.China, "Building Quick Service Query List using WordNet for automated Service Composition", *IEEE Asia-Pacific Services Computing Conference(IEEE APSCC'08)*, 2008, pp. 297-302.

[6] Tamer Ahmed Farrag, Ahmed Ibrahim saleh and Hesham Arafat Ali, "Toward SWSs Discovery: Mapping from WSDL to OWL-S Based on ontology search and standardization Engine", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 5, 2013, pp. 1135-1147.

[7] Jacek Kopecky, Tomas Vitvar, Carine Bournez and Joel Farrell, "SAWSDL: Semantic Annotations for WSDL and XML Schema", *IEEE Internet Computing*, Vol. 11, No. 6, 2007, pp. 60-67.

[8] Antonio Brogi, "On the potential advantages of exploiting behavioural information for contract-based service discovery and composition", *The Journal of Logic and Algebraic Programming*, Vol. 80, No. 1, 2011, pp. 3-12.

[9] Dumitru Roman, Uwe Keller, Holger Lausen, Jos de Bruijn, Ruben Lara, Michael Stollberg, Axel Polleres, Dieter Fensel, and Christoph Bussler, "Web Service Modeling Ontology", *Applied Ontology Journal*, Vol. 1, No. 1, 2005, pp. 77-106.

[10] Dong-Hoon Shina, Kyong-Ho Leea and Tatsuya Sudab, "Automated generation of composite web services based on functional semantics", *Journal of Web Semantics : Science, Services and Agents on the World Wide Web*, Vol.7, No. 4, 2009, pp.332–343.

[11] John Domingue, Liliana Cabral, Stefania Galizia, Vlad Tanasescu, Alessio Gugliotta, Barry Norton and Carlos Pedrinaci, "IRS-III: A broker-based approach to semantic Web services", *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 6, No. 2, 2008, pp. 109–132.

[12] H Vikas Agarwal, Girish Chafle, Koustuv Dasgupta, Neeran Karnik, Arun Kumar, Sumit Mittal and Biplav Srivastava, "Synthy: A system for end to end composition of web services", *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 3, No. 4, 2005, pp. 311-339.

[13] Erik Christensen, Francisco Curbera, Greg Meredith and Sanjiva Weerawarana, "Web Services Description Language (WSDL) 1.1", - *http://www.w3.org/TR/wsdl*, 2001.

[14] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreah,Henrik Frystyk Nielson,Anish Karmarkar and Yves Lafon ,"SOAP version 1.2 part 1: Messaging Framework (Second Edition)", *http://www.w3.org/TR/soap12 -part1*, 2007.

[15] *http://uddi.org/pubs/uddi_ v3.htm*

[16] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur and Yarden Katz, "Pellet: A Practical OWL DL Reasoner", *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, No. 2, 2007, pp. 51-53.

[17] Sean Bechhofer, Frank Van Harmellen, Jim Hendler, lan Horrocks, Deborch L. McGuinness, Peter F. Patel and Lynn Andrea Stein, "OWL web ontology language"- *http:\\www.w3.org/TR/owl-ref/* , 2004.

[18] L Matthias Kiusch, Patrick Kapahnke, Benedikt Fries, Mahboob Alam Khalid and Martin Vasileski, "OWLS-TC - OWL-S Service Retrieval Test Collection Version 4.0",- *http://projects.semwebcentral.org/projects/owls-tc/*, 2010.

[19] D.K.Lin, "WordNet: An electronic lexical database", *Computational Linguistics*, Vol. 25, No. 2, 1999, pp.292-296.

[20] Hai Dong and Farookh Khadeer Hussain " Focused Crawling for Automatic Service Discovery, Annotation and Classification in Industrial Electronics", *IEEE Transcations on Industrial electronics*, Vol. 58, No. 6, 2011, pp.2106-2116.