Research Journal of Applied Sciences, Engineering and Technology 8(22): 2240-2247, 2014
DOI:10.19026/rjaset.8.1224
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 07, 2014 Accepted: September 20, 2014 Published: December 15, 2014

Research Article

Automatic Discovery of Relevant Web Services with Semantic Ranking

[1]N. Edison Rathinam, [2]Latha Parthiban and [3]G. Mariakalavathy
[1]Manonmaniam Sundaranar University, Tirunelveli,
[2]Department of Computer Science, Pondicherry University, Pondicherry,
[3]Department of Computer Science and Engineering, St. Joseph's College of Engineering, Chennai, India

Abstract: The main objective of this study is to discover and rank relevant web services from the given set of WSDL documents automatically for the specified user requirements through semantic search. Current web services are described using WSDL that only provides syntactic description of a web service. This syntactic description is related with the implementation details of a service and thus customized towards the requirements of the programmer. Semantic descriptions of the web services are necessary to automatically discover many relevant web services for a specific user request. In the proposed architecture, the semantic descriptions of a given set of WSDL documents are extracted using WordNet Ontology Framework and Suggested Upper Merged Ontology (SUMO) concepts and these semantic descriptions are stored in the database along with the WSDL documents. The given set of WSDL documents are semantically categorized in to six predefined categories based on the semantic descriptions. These semantically categorized WSDL documents are searched for discovering relevant web services for a refined service request. As these relevant web services are semantically ranked and displayed, the users are facilitated with the required information. The effectiveness of this proposed architecture is evaluated with set of WSDL documents that are collected from OWLS-TC test collection. The analysis of results shows that the proposed architecture provides better improvement on recall, precision and average query response time.

Keywords: Automatic service discovery, ranking of relevant services, semantic ranking, service categorization

INTRODUCTION

Web Services are the most promising implementation technology for Service Oriented Architecture that uses Internet as the communication medium and Internet-based standards such as Simple Object Access Protocol (SOAP) for transmitting data, Web Services Description Language (WSDL) for defining Web Services, Universal Description Discovery Integration (UDDI) for registering Web Services and Business Process Execution Language (BPEL) for orchestrating Web Services. A huge amount of web services with various functionalities in various areas are published that are used to create large scale distributed applications over the web. These web services are registered in UDDI under predefined categories such as business, educational, finance, scientific, etc. by the service providers. Sometimes, similar web services may be registered in different categories in UDDI registry by different service providers. The users have to manually search published services by category. The vast majority of available web services are described using WSDL with syntactic description that are related with the implementation details of a service and thus customized towards the requirements of the programmers those want to use the WSDL to compose services. The semantic descriptions that are related with conceptual aspects of a service for facilitating the end users were not explicitly specified. As a result many services that are relevant to a specific user service request with desired functionalities in different categories may not be discovered during service discovery. Therefore, automatic mechanisms such as service categorization and clustering are required based on the semantics of capabilities of web services rather than the classification provided by the service providers in the UDDI registry.

Reddy and Kamath (2012) discussed the probable discovery mechanisms of web services, the web service architecture and its related technologies. The semantic web concept and its characteristics and the realization of semantics to web services through ontology and ontological languages were also discussed. Fensel *et al.* (2007) focused mainly on Web Service Modeling Ontology (WSMO) which provides a comprehensive conceptual framework for the combination of Semantic Web technologies and Web services. The promising approach for automated service discovery is semantic web technology (Mcllraith *et al.*, 2001). Semantically tagged approaches such as WSDL-S (McIlraith and Martin, 2003) and OWL-S (Martin *et al.*, 2005) were used in most of the current semantic service discovery

Corresponding Author: N. Edison Rathinam, Manonmaniam Sundaranar University, Tirunelveli, India

process. The limitations on these approaches are that each new service should be semantically tagged and user does not know about all the semantics of the terms used in request. To address these problems, service categorization, selection of semantically relevant services with ranking are needed.

This proposed approach is more generic in handling the set of web services that are basically described using WSDL documents. First, all the web services are semantically classified into various categories such as finance, weather, scientific, business, education and entertainment using WordNet (Miller, 1995) and SUMO Ontology (Niles and Pease, 2003). This service categorization is performed offline on a regular basis and is independent of the service request. Secondly, semantically matched category is selected for the service request. At last, relevant web services from the selected category is ranked and displayed to the users based on the service request. Using this proposed approach, the user will be able to get all the relevant web services even if the user is unaware of all the relevant terms included in the web services. The main advantage of this approach is that the user intervention is not required during service discovery process. They just want to specify the request and will be facilitated with the required relevant web services.

LITARTURE REVIEW

As web services and service providers proliferate, there will be a large number of candidate services that are spread over the distributed environment and likely competing, services for fulfilling a desired task. Hence, effective service discovery mechanisms are required for identifying and retrieving the most relevant web services. Normally the requirements of the users can be collected and the required services can be searched and composed for access by the end user. To overcome the limitations of keyword-based search, several semantic-based approach using ontologies for enhancing the service descriptions were proposed. The services were described using WSDL-S, OWL-S and WSMO and the match making were addressed as a logic inference task (Paolucci *et al*., 2002). The similarity between requested and offered inputs and outputs were assessed by comparing classes in associated domain ontology (Cardoso, 2006; Skoutas *et al*., 2007). Bellur and Kulkarni (2007) proposed match making algorithm based on matching bipartite graphs to match the requested and offered parameters. Zhang *et al*. (2009) proposed the method of service discovery based on bipartite matching of semantic message similarity for remote medical systems. Ontologies, user profiles, query expansion and relaxation along the given service ontology and the personal preferences of the users were suggested by Balke and Wagner (2003). The hybrid match maker OWLS-MX (Klusch *et al*., 2006) utilized both logic based reasoning and IR techniques for semantic Web services in OWL-S. Another matchmaker WSMO-MX proposed by Klusch and Kaufer (2009) performs hybrid semantic service matching based on both logic programming in F-Logic and syntactic similarity measurement. The Content-based matching for web service discovery and ranking prototype called BASIL (Caverlee *et al*., 2004) was presented that supports the personalized views of data-intensive web services through source biased focus. Specifically, it probes the candidate service and measures the relevance based on the actual data returned, rather than on the metadata in the service description.

The discovery framework incorporating semantic description by extending current UDDI architecture was proposed by Yu *et al*. (2004). This architecture supports publishing and discovering of services but the users are expected to give their preferences. The ranked relevant web services for the service request were not displayed to the users. Bernstein and Klein (2004) designed a language called PQL (Process Query Language) for retrieving process models. One key issue involved is that the service provider might model a service in a way that is semantically equivalent to, but not a syntactic match with, a given PQL query. Another key issue concerns rapid service modeling that is the service providers routinely created process models for many services, so that the models were translated often automatically into service descriptions suitable for PQL retrieval. Creating PQL queries, as with many query languages, requires some technical expertise. Ragone *et al*. (2007) proposed a tractable greedy concept covering algorithm to perform an automatic semantic web services composition and jUDDI+, a framework implementing their approach in OWL-S. They extended Concept Covering definition and re-defined it in terms of Concept Abduction. The semantic specification of the service was based not only on the Inputs Outputs Preconditions Effects model but also endowed with a semantic description of the provided functionalities. Tsetsos *et al*. (2006) have proposed generalized evaluation schemes, based on soft computing techniques like Fuzzy Sets for evaluating semantic web service match making system.

Skoutas *et al*. (2010) addressed ranking and clustering of web service search results and proposed methods based on the notion of dominance, which apply multiple matching criteria without aggregating the match scores of individual service parameters. Paliwal *et al*. (2012) proposed an ontology guided categorization of web services for semantic-based service categorization. An ontology linking and LSI was also employed for extending the indexing procedure from syntactical information to a semantic level for semantic-based service selection. Ma *et al*. (2008) utilized Probabilistic Latent Semantic Analysis

(PLSA), a machine learning method, to capture the semantics hidden behind the words in a query and the descriptions in the services, so that service matching can be carried out at the concept level. The semantic web service selection mechanism which distinguishes semantically similar web services based on the Quality of Service (QoS) and Business Offerings (BO) was proposed by D'Mello and Ananthanarayana (2009). Chen *et al*. (2013) proposed a novel clustering facilitated web services discovery model (CFWSFinder), which introduces current machine learning technologies into the services representing, services clustering and services matching processes. A semantic web services framework (Samper *et al*., 2008) have been designed and implemented to support mobile web services. This framework in combination with the use of the matchmaking algorithm was used to find services based on their similarities and this algorithm uses ontologies and account concepts to find the web services.

These works focused on matching pairs of parameters from the requested and offered services. Based on the diversity of these approaches, it is evident that there are many matching criterion that constitutes the semantic web service discovery problem. Therefore, the approach proposed in this study provides an efficient architecture that semantically categorizes the web services through service database creation and ranks the services that are matched with the service request based on keywords, semantic and service parameters. In addition, the proposed approach does not require prior knowledge of the preferences of the users.

SEMANTIC CATEGORIZATION OF WSDL DOCUMENTS

Normally, the capabilities of all the web services are described using abstract interface called WSDL including the name of the service, name of the methods, input-output parameters and other descriptions. As the semantic description of the capabilities of the web services were not defined using WSDL, the users are overwhelmed by the large number of irrelevant web services for their request. Therefore, semantic categorization of WSDL documents is necessary. This semantic categorization is performed offline on a given set of WSDL documents and is independent of the service request. As a result of this semantic categorization, each WSDL document is assigned to any one of the six predefined categories. The predefined categories such as finance, weather, scientific, business, education and entertainment are selected as sample categories in this proposed approach.

In the proposed approach, the following steps are used to categorize the web services as shown in Fig. 1:

- Parse WSDL for extracting service name, methods name and input-output parameters.
- Define a database using extracted terms as shown below in Table 1.
- Infer the related concepts for each service and methods available in a web service using WordNet Ontology and Suggested Upper Merged Ontology (SUMO) concepts and enter them into the database.
- Calculate similarity score by comparing related concepts with predefined six categories.
- Assign a web service to the category which has highest similarity score.

The details about these steps are explained in the following sections.

Parsing WDSL and web service database creation: The given set of WSDL documents are parsed to extract

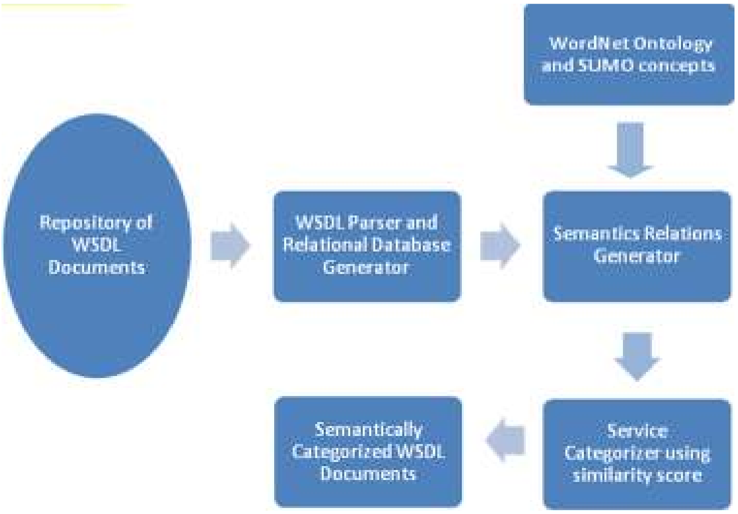
WordNet Ontology and SUMO concepts
Repository of WSDL Documents
WSDL Parser and Relational Database Generator
Semantics Relations Generator
Semantically Categorized WSDL Documents
Service Categorizer using similarity score

Fig. 1: Categorization of WSDL documents

Table 1: Sample service table

S.no.	Service name	Method names	Input parameters	Output parameters	Relevant concepts	Category
1	Weather_ Summary	Get summary, update weather, weather notification	Zipcode	Temperature, wind, pressure	Temperature, cold, hot, wind, weather, influence, breathe, musical instrument, pressure, force, imperative, insistence, distress	Weather
2	Weather	Weather service	Zipcode, date, time	Temperature, humidity, pressure, rain fall, precipitation	Temperature, cold, hot, wetness, moisture, sogginess, wateriness, pressure, force, imperative, insistence, distress, rain fall, precipitation, downfall, city, date, time	Weather
3	CityInfo	Info service	City	Population, temperature, wind	Population, people, group, collection, temperature, hot, cold, wind, weather, influence, breathe, musical instrument,	Weather

the name of the service, name of the methods and input-out parameters. The Java code segment for parsing a single WSDL document is shown as follows:

```
WSDLFactory factory = WSDLFactory.newInstance();
WSDLReader reader = factory.newWSDLReader();
reader.setFeature("javax.wsdl.verbose", true);
reader.setFeature("javax.wsdl.importDocuments",true);
Definition def = reader.readWSDL(null,
"C:\\Documents and Settings\\Administrator\\My
Documents\\ wsdlfiles\\StockQuote.wsdl");
Map ss = def.getAllServices();
Service service = def.getService(new QName(tns1,
ser1)) ;
Port port = service.getPort(pt);
List <String> operations = new ArrayList
<String>(portType.getOperations());
System.out.println("no.of operations =
"+operations.size());
Operation operation = (Operation)opIterator.next();
System.out.println("\noperation name = "+ operation.
getName();
System.out.println("\noperation input = "+ operation.
getInput());
System.out.println("\noperation output = "+ operation.
getOutput());
```

The extracted terms from each WSDL document are stored into the database as shown in Table 1 which shows the sample data after processing three WSDL documents.

Categorization of web services: The related concepts of the name of the service, methods name and input-output parameters are added into the database. In addition, the category in which the WSDL document belongs are added into the database as explained in the following algorithm Add Related Concept Category. WordNet noun database with SUMO mappings (Niles and Pease, 2003) are used in this algorithm to enhance the semantic relationships between the service name and predefined categories.

Algorithm: Add Related Concept Category
Aim: To add related concepts and category for services into the database

```
Input: Web service database which includes service table (S)
Output: Web service database with related concepts and category
begin
    for each service $s_i \in S$ do
    begin
        extract the set of WordNet elements for service name, input and output parameters
    for each WordNet element ($e_i$) in the above set
        begin
          find SUMO concept $c_i \in C$ for $e_i$
         If found then
          begin
            update service table by adding $c_i$ as related concept for service $s_i$
            if Relevance ($R_i$) exists between $c_i$ and service name, input and output parameters
              calculate Similarity Score $SS[i,j]$ between $e_i$ and each predefined categories ($g_j$)
        end
    end
 for each predefined category $g_j$
        calculate score $S[j] = \sum SS[i,j]$
      find the category $j$ which has maximum score
    assign the category $j$ to the service $s_i$ in the table $S$
  end
end.
```

The first step in this algorithm involves adding relevant concepts for each service by extracting WordNet elements and maps them into SUMO concepts represented by set C. The relevance R is calculated to specify the relevance between any two concepts ti and tj. The relevance predicate R(ti,tj) evaluates true if the concepts ti and tj are linked to a common concept in the upper ontology. If there is such relevance between mapped SUMO concept and service name, the Similarity Score (SS) is computed for each WordNet element with six predefined categories. Probability-based Measures of Semantic Relatedness (P-MSR) is called Normalized Similarity Score (NSS) which is used to measure the similarity score between the two words. NSS is derived from Normalized Google Distance (NGD) as below:

$$NSS_{i,j} = 1\text{-}NGD_{i,j} \tag{1}$$

where, *NGD* is derived by (Cilibrasi and Vitanyi, 2007) as follows:

$$NGD_{i,j} = \frac{max\,\{log\,f(i), log\,f(j)\}\,log\,f(i,j)}{log\,M\,min\,\{log\,f(i), log\,f(j)\}} \tag{2}$$

where *M* is the total number of web pages searched by Google; *f*(*i*) and *f*(*j*) are the number of hits for each search term *i* and *j* respectively; and *f*(*i*, *j*) is the number of hits for both *i* and *j*.

The next step is to find the sum of Similarity Score for each category. The category which has the highest Similarity Score is the category of the web service and that is entered into the database.

PROPOSED ARCHITECTURE FOR AUTOMATIC DISCOVERY OF RELEVANT WEB SERVICES (ADRWS)

The proposed architecture for Automatic Discovery of Relevant Web Services (ADRWS) shown in Fig. 2 facilitates the user with the required relevant web services that are ranked using semantic relationships between service request and matched web services.

The user request is parsed and refined to get the required information such as input and output parameters using Service Request Refinement (SRR) module. This SRR module preprocesses the service request and filters the required terms that are used to find the required services. The preprocessing includes punctuation removal, stop words removal and stemming. For example, after preprocessing, the set of terms that are extracted from the Service Request (SR): “find the temperature, rain-fall and pressure of the city Chennai” are {temperature, rain-fall, pressure, city}. Next, the SRR module finds the related terms for this set of terms using WordNet ontology.

The related terms of the service request and set of semantically categorized WSDL documents are given to Semantic Service Match Maker (SSM) module. The SSM finds the relevant category for the request from the available six categories such as finance, weather, scientific, business, education and entertainment, in which the set of WSDL documents are semantically categorized. Next, it computes similarity score between the related concepts of web services in the selected category and the related terms of service request. The resultant web services are ranked based on this similarity score and the services with the highest similarity score are displayed to the users.

Proposed Semantic Service Match Maker (SSM) module: The proposed match maker finds the matched services for the user request using match making procedure called Semantic Service Match Maker (SSM). The set of relevant terms for each predefined category are stored in the database using WordNet ontology. For e.g., the relevant set for weather category is given as follows:

Weather = {temperature, rain-fall, pressure, humidity, precipitation, wind, UV-index, snow fall, city, district, zipcode, date and time} (3)

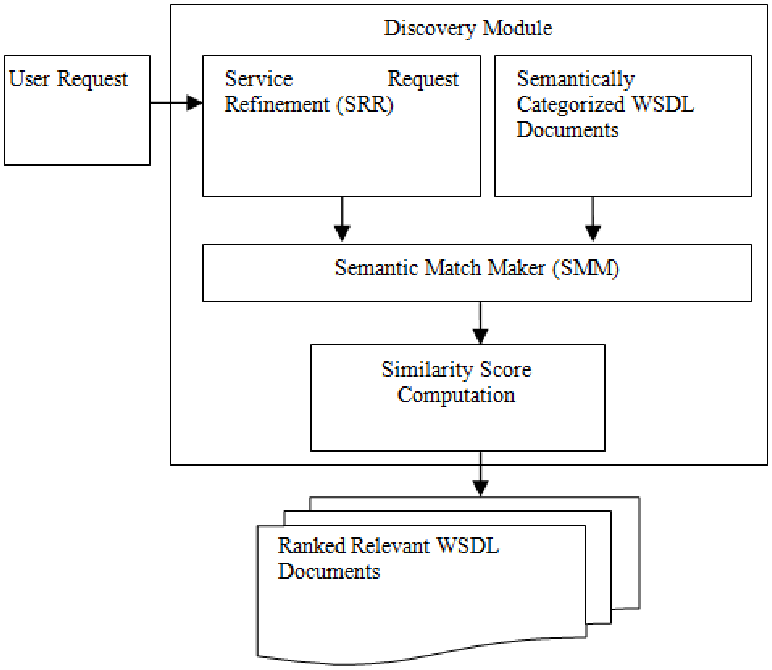
Discovery Module
User Request
Service Request Refinement (SRR)
Semantically Categorized WSDL Documents
Semantic Match Maker (SMM)
Similarity Score Computation
Ranked Relevant WSDL Documents

Fig. 2: Proposed architecture for automatic service discovery of relevant web services (ADRWS)

The SSM procedure selects relevant category from predefined six categories and then selects relevant services. This service selection is executed online and in real time on a per request basis. The relevant terms for the extracted terms from the service request are found using WordNet. The relevant set for the Service Request (SR) “find temperature, rain fall and pressure of the city Chennai” which includes five terms such as temperature, rain-fall, pressure, city and Chennai are:

SR = {(temperature, cold, hot), (rain fall, precipitation, downfall), (pressure, force, imperative, insistence, distress), (city, municipality, administrative district), {Chennai, Madras} (4)

This SR includes five relevant term set for the five terms that are extracted from the service request. The relevant set (3) of weather category is compared with relevant set (4) of SR. Out of five related term set in SR, one from each of four related term set are matched with the terms in the weather set (3). Therefore the SR is matched 80% with weather category. In the same way, SR is compared with other five categories such as finance, scientific, business, education and entertainment. The category which has the highest degree of match is selected as most relevant category.

Finally, the services in the most relevant category are selected and each service in that category is compared with the service request. This comparison is between the related concepts of the service from the database and related terms of the service request. The similarity score is computed for each service in the category based on this comparison. This similarity score is used to semantically rank the services. The services are displayed to the users in the order of highest rank.

For example, the weather category includes two services such S1 and S2. S1: web service which returns weather information such as temperature, humidity, pressure, rain fall, precipitation for the given zip code, date and time. S2: web service which returns population, temperature, wind for the given city. The service S1 and SR is matched with four parameters, but the service S2 is matched with two parameters. Therefore the service S1 is ranked higher than service S2. These services are displayed to the users with the highest rank first.

EXPERIMENTAL RESULTS

The OWLS-TC 4.0 test collection is used for evaluating the relevant services discovery performance of this proposed approach. The OWLS-TC4.0 test collection is a public test collection which consists of 1083 semantic web services described with OWL-S 1.1 and 1076 WSDL files. The majority of these services were retrieved from public IBM UDDI registries and semi-automatically transformed from WSDL to OWL-S. This test collection can be accessed via the http://projects.semwebcentral.org/projects/owls-tc/.

The experiments are conducted on this test set of WSDL files for evaluating the service discovery performance of this proposed approach. The micro-averaging the individual precision-recall curves (Van Rijsbergen, 1979) is adopted for this evaluation. Let Q be the set of Service Requests (SR) in Owls-TC:

$$Q = \{SR_1, SR_2, \ldots SR_N\} \quad (5)$$

R_i be the relevant WSDL document of the service request $SR_i \in Q$, A be the collection of all the relevant WSDL documents of all requests in Q:

$$A = \{R_1, R_2, \ldots R_n\} \quad (6)$$

For each service request SR_i, relevant documents retrieved (recall) $B_{\lambda SR}$ is measured for λ = 20 steps up to its maximum recall value. Similarly, related precision B_λ of retrieved documents at each step λ is measured. The micro-averaging of recall and precision at step λ over all requests is defined as:

$$Rc_\lambda = \sum_{SRi \in Q} \frac{|A \cap B\lambda SR|}{|A|} \quad (7)$$

$$Pc_\lambda = \sum_{SRi \in Q} \frac{|A \cap B\lambda SR|}{|B\lambda|} \quad (8)$$

The micro-averaged R-P curves and average response time of ADRWS together with match makers for OWL-S services and without using semantic discovery are displayed in Fig. 3 and 4, respectively.

The following conclusions are arrived based on these experimental results:

- As a set of relevant services is subjected to continuous change due to increase in number of services published, the precision is more important to the user than recall for discovering semantic web services. Since the semantic categorization and match maker module of this proposed approach uses all possibilities of relevant terms of the service request and terms used in the WSDL documents, the recall and precision is better compared to semantic discovery in OWL-S services as shown in the Fig. 3.
- In terms of average query response time, the proposed approach ADRWS shows better result than OWL-S matchmakers as shown in the Fig. 4. This is due to the additional computational efforts required by OWL-S matchmakers to determine the subsumption relationships based on the imported large ontologies the OWL-S services refer to.
- As the predefined categories specified by the service providers are considered for the services

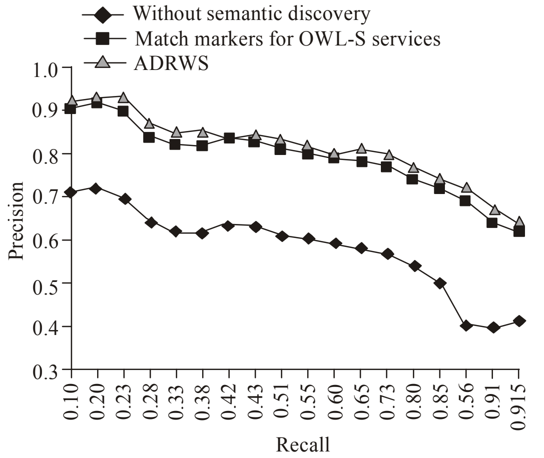

Fig. 3: Recall-precision performance of ADRWS with match makers for OWL-S and without using semantic discovery

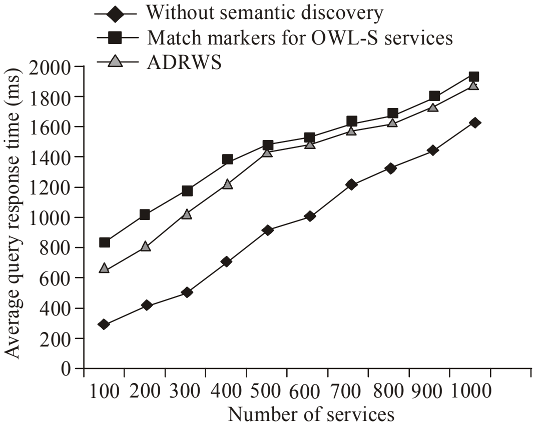

Fig. 4: Average response time of ADRWS with match makers for OWL-S and without using semantic discovery

discovery without any semantic categorization and semantic matching between service request and published services, the average query response time of discovery without semantic matching is better.

CONCLUSION

The relevant web services are discovered and ranked for the user request automatically from the given set of WSDL documents using this proposed architecture. This facilitates the user with required information for their selection. Specifically, this approach addresses two major aspects of semantic-based service discovery such as semantic categorization of given set of WSDL documents and semantic match making between service request and WSDL documents from selected category. The effectiveness of service categorization and semantic matching process is measured using R-P curves and average query response time. The experimental results show that recall, precision and average query response time are quite better than other match makers.

REFERENCES

Balke, W.T. and M. Wagner, 2003. Cooperative discovery for user-centered web service provisioning. Proceeding of the IEEE International Conference on Web Services (ICWS, 2003), pp: 191-197.

Bellur, U. and R. Kulkarni, 2007. Improved matchmaking algorithm for semantic web services based on bipartite graph matching. Proceedings of the IEEE International Conference on Web Services (ICWS, 2007), pp: 86-93.

Bernstein, A. and M. Klein, 2004. Toward high-precision service retrieval. IEEE Internet Comput., 8(1): 30-36.

Cardoso, J., 2006. Discovering semantic web services with and without a common ontology commitment. Proceedings of IEEE Services Computing Workshops (SCW), pp: 183-190.

Caverlee, J., L. Liu and D. Rocco, 2004. Discovering and ranking web services with BASIL: A personalized approach with biased focus. Proceeding of the 2nd International Conference on Service Oriented Computing (ICSOC, 2004), pp: 153-162.

Chen, L., G. Yang and W. Zhu, Y. Zhang and Z. Yang, 2013. Clustering facilitated web services discovery model based on supervised term weighting and adaptive metric learning. Int. J. Web Eng. Technol., 8(1): 58-80.

Cilibrasi, R. and P.M.B. Vitanyi, 2007. The Google similarity distance. IEEE T. Knowl. Data En., 19(3): 370-383.

D'Mello, D.A. and V.S. Ananthanarayana, 2009. Semantic web service selection based on service provider's business offerings. Int. J. Simul. Syst. Sci. Technol., 10(2): 25-37.

Fensel, D., H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman and J. Domingue, 2007. Enabling Semantic Web Services. Springer, ISBN: 978-3-540-34520-6.

Klusch, M., B. Fries and K. Sycara, 2006. Automated semantic web service discovery with OWLS-MX. Proceeding of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '06), pp: 915-922.

Klusch, M. and F. Kaufer, 2009. WSMO-MX: A hybrid semantic web service matchmaker. Int. J. Web Intell. Agent Syst., 7(1): 23-42.

Ma, J., Y. Zhang and J. He, 2008. Efficiently finding web services using a clustering semantic approach. Proceeding of the International Workshop Context Enabled Source and Service Selection, Integration and Adaptation (CSSSIA), pp: 1-8.

Martin, D., M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGunneess, B. Barsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan and K. Sycara, 2005. Bringing semantics to web services: The OWL-S approach. In: J. Cardoso and A. Sheth (Eds.), Semantic Web Services and Web Process Composition (SWSWPC). LNCS 338, Springer Verlag, Berlin, Heidelberg, pp: 26-42.

Mcllraith, S., T. Son and H. Zeng, 2001. Semantic web services. IEEE Intell. Syst., 16(2): 46-53.

McIlraith, S. and D. Martin, 2003. Bringing semantics to web services. IEEE Intell. Syst., 18(1): 90-93.

Miller, G.A., 1995. WordNet: A lexical database for English. Commun. ACM, 38(11): 39-41.

Niles, I. and A. Pease, 2003. Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology. Proceeding of the IEEE International Conferences on Information and Knowledge Engineering (IKE '03), DOI: 10.1.1.10.4486.

Paliwal, A.V., B. Shafiq, J. Vaidya and N. Adsam, 2012. Semantics-based automated service discovery. IEEE T. Serv. Comput., 5(2): 260-275.

Paolucci, M., T. Kawamura, T.R. Payne and K.P. Sycara, 2002. Semantic matching of web services capabilities. Proceeding of the 1st International Semantic Web Conference on the Semantic Web (ISWC'02), pp: 333-347.

Ragone, A., T.D. Noia, E.D. Sciascio, F.M. Donini, S. Colucci and F. Colasuonno, 2007. Fully automated web services discovery and composition through concept covering and concept abduction. Int. J. Web Serv. Res., 4(3): 1-31.

Reddy, A.A. and S. Kamath, 2012. Research on potential semantic web service discovery mechanisms. Proceeding of the International Conference on Recent Trends in Computer Science and Engineering (ICRTCSE' 2012), pp: 1-6.

Samper, J.J., F.J. Adell, L. van den Berg and J.J. Martinez, 2008. Improving semantic web service discovery. J. Networks, 3(1): 35-42.

Skoutas, D., A. Simitsis and T. Sellis, 2007. A ranking mechanism for semanticWeb service discovery. Proceeding of IEEE Congress on Services, pp: 41-48.

Skoutas, D., D. Sacharidis, A. Simitsis and T. Sellis, 2010. Ranking and clustering web services using multi-criteria dominance relationships. IEEE T. Serv. Comput., 3(3): 163-177.

Tsetsos, V., C. Anagnostopoulos and S. Hadjiefthymiades, 2006. On the evaluation of semantic web service matchmaking systems. Proceeding of the European Conference on Web Services (ECOWS '06), pp: 255-264.

Van Rijsbergen, C., 1979. Information Retrieval. Butterworth, London.

Yu, S., J. Liu and J. Le, 2004. Semantic based web services discovery. In: Chi, C.H. and K.Y. Lam (Eds.), AWCC 2004. LNCS 3309, Springer-Verlag, Berlin, Heidelberg, pp: 388-393.

Zhang, Y., B.Y. Liu and H. Wang, 2009. A method of web service discovery based on semantic message bipartite matching for remote medical system. J. Theor. Appl. Electron. Commer. Res., 4(2): 79-87.