ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

Two Stage Optimization Model to Semantic Service Discovery

Chellammal Surianarayanan,
Department of Computer Science,
Bharathidasan University Constituent Arts & Science College,
Srirangam Tk, Trichy-620009, TN, India

Abstract: **Discovering appropriate services quickly for dynamic service composition is a challenging issue. Clustering technique partitions the available services into clusters of similar services. During discovery of matched services for a query, semantic matching of service capabilities is performed only to a particular cluster which is most relevant to the query *and* other clusters are ignored as irrelevant. Thus clustering improves the performance of semantic discovery by eliminating irrelevancy. In one of our previous research work, two similarity models, one for computing similarity between services(called Output Similarity Model) while clustering them and the other(called Total Similarity Model) for finding matched services for a given query using clusters along with selection of similarity threshold and recommendation of complete linkage criterion for computing inter-cluster distance are proposed for service discovery using hierarchical agglomerative clustering. As an extension of our previous work, in this paper, an experimental evaluation has been performed to analyze the performance of OSM in regard to effective removal of irrelevancy and the strength of prioritizing parameters during discovery. Further, the clustering solutions obtained using Output Similarity Model are compared with those produced by standard methods such as syntactic similarity and Word Net similarity based methods. Though clustering improves the performance of discovery by eliminating irrelevant clusters, still is required to employ semantic matching to the services present in the relevant cluster. This involves invoking semantic reasoning during querying. To resolve this limitation, after clustering, an indexing technique is suggested to the resulting clustering solution. With this model, the invoking of semantic reasoning is completely eliminated.**

***Index Terms*—Agglomerative clustering, service clustering, similarity models, semantic service discovery, similarity threshold**

I. INTRODUCTION

Service Oriented Architecture (SOA) is an architectural style that promotes developing applications by reusing the existing interoperable software components (called *services*) having well defined interfaces, over network in a loose-coupled fashion. Web services, an open technology stack is predominantly used in implementing SOA due to its simplicity and use of existing transport protocol, Hyper Text Transfer Protocol (HTTP). Web services includes Web Service Description Language (WSDL) for describing interface of services, Simple Object Access Protocol (SOAP) for specifying the message format between service provider and consumer and Universal Description, Discovery and Integration(UDDI) as a means to publish and discover services. Though many services are available on the Web for use, atomic services may not be sufficient to implement complex business needs. Complex processes are implemented by combining atomic services from different domains via service composition.

Also, atomic services which could realize a given business process should be discovered by functional characteristics prior to composition itself so that composition can be accomplished successfully within the expected time as desired by clients. Service composition becomes complex due to the existence of several services. The needs of composition can be met by bringing automation into discovery and composition. Semantic service description languages such as [1-2] describe services with explicit semantics and make services as machine-process able entities.

Corresponding to semantic service description, various frameworks have been put forward for discovery using semantics. Typically a semantic discovery framework consists of two components, namely, a matcher and a semantic reasoner such as Pellet which infers semantic relations, viz., *exact, plugin, subsumes and fail*[3] between different concepts during matching. When a query is submitted, the matcher (a matching algorithm) matches each published service with the query and finds a list of matched

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

services with the help of semantic reasoner. Though semantic matching yields sufficient accuracy for business processes, the time involved in semantic reasoning is reasonably high of around 4-5 seconds even for a single service match of 10 concepts[4]. In general business transactions involve several services from different domains to be discovered and composed in a complex chain within relatively short intervals of time. Hence, semantic service discovery should be optimized. Different optimization techniques are discussed in [5].

Clustering is found to be an attractive method as it acts as a base for any other analysis. In a clustering based service discovery, prior to querying itself the available services are partitioned into different groups of similar services such as 'financial', 'weather, 'education, 'trading, etc. With services organized as clusters, when a query is submitted, the particular cluster which is most similar to the query alone will be chosen for semantic matching ignoring other clusters as irrelevant. For example, for the query '*Find temperature*', the cluster '*weather*' alone will be chosen for semantic matching.

In one of our previous research works [6**],** a set of methodologies along with two similarity models namely, Output Similarity Model (OSM) and Total Similarity Model (TSM) have been proposed to enhance the discovery of semantic services using clustering. OSM computes similarity between two services based on their output parameters in terms of various levels Degree of Match (DoM) whereas TSM computes similarity between services using both the input and output parameters. The two models have different purposes. OSM is used during clustering and TSM is used during querying to find matched services from the relevant cluster of the query. In this paper, as extension to the above work, two experimental studies have been taken up. In one experiment, the two similarity models have been compared for their performance in regard to effective removal of irrelevancy and to show the strength of prioritizing the parameters of services for discovery. In another experiment, the clustering solutions produced by OSM have been compared with those of standard methods to study the appropriateness of clustering solutions produced by different methods. Though the previous work [6] enhances the performance of semantic discovery by employing semantic matching only to the cluster which is most similar to the query, it has an inherent limitation that it has to invoke semantic reasoning to the most similar cluster of the query during querying. To resolve this limitation an indexing based solution is also proposed in this work.

II. SIMILARITYMODELS

In this section, an overview of how similarity between two services is calculated using the similarity models proposed in [6] is presented

A. Output Similarity Model

In OSM, only outputs are considered for similarity computation. Two services are considered as similar if there is a high semantic relationship between their outputs. Consider two services, namely, s1 with 'm' number of output parameters and s2 with 'n' number of parameters. Let $\{op1i, 1 \le i \le m\}$and $\{op2\ j, 1 \le j \le n\}$ denote all output parameters of s1 and s2 respectively. Let $op1i$ denote ith output parameter of s1. Let $op2\ j$ denote jth output parameter of s2. Let *DoM* $(op1i, op2\ j)$ denote the Degree of Match between $op1i$ and $op2\ j$. For convenience, in this work, the standard levels of DoM as described below are used while finding similarity.

Exact: In this level, the type of $op1i$ is equivalent to that of $op2\ j$. The matching score of *exact* is 1.

Subsumes: In this level, the type of $op1i$ will be a subtype of $op2\ j$. The matching score of subsumes is 0.5

Plug-in: If the type of $op2\ j$ will be a subtype of $op1i$. The matching score assigned is 0.5

Fail: In this level, the type $op1i$ is different from that of $op2\ j$. The matching score of fail is 0.0.

Now the normalized similarity between between s1 and s2 is computed as described in[6]

B. Total Similarity Model

In Total Similarity Model, both input and output parameters of services are considered for similarity computation. For simplicity, here also, only conventional levels of DoM are taken into account. The computation of similarity is discussed in [6].

III. EXPERIMENTAL STUDY I

The aim of the first experiment is to study the performance of the two similarity models with respect to removal of irrelevancy. Towards this, an experimental setup as described in [6] is used. The similarity models are

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

implemented in Java. The OWLS-API which has in-built Pellet reasoner is used to find various DoMs among parameters.

A set of 100 services as given in Table I is built from the standard test collection, OWL-S Service Retrieval Test Collection version 3.0 is used to test the quality of the cluster partitions produced using OSM and TSM. The test data is constructed in such a way that it contains internal groups of similar services from different domains such as education, food, travel, and communication. It contains 5 singleton groups and 14 groups of services having similar outputs. Each group is given an ID.

Table I
Details of groups in test services

Group ID	Similar Outputs in the group	# of Services
Group-1	---	1
Group-2	---	1
Group-3	---	1
Group-4	price, taxed_price, recommended_price, recommended_price_in_dollar	19
Group-5	address, postal_address	4
Group-6	hotel	4
Group-7	film, comedy_film, low_comedy_film	11
Group-8	destination	8
Group-9	Food, preparedfood	7
Group-10	Icon, photograph	2
Group-11	researcher	2
Group-12	book	4
Group-13	author	8
Group-14	Funding, financing	7
Group-15	videomedia, vhs, dvd	6
Group-16	coffee, whisky, cola, drinks, irishcoffee, mixerycola	9
Group-17	lecturer_in_academia	4
Group-18	professor_in_academia	1

Average inter-cluster similarity, (*avg* □ *ircsim*) , average intra-cluster similarity (*avg* □ *iacsim*) and Silhouette Width (SW) of cluster partitions is used to evaluate the effectiveness of the models.

The similarity models and clustering procedures are employed to the test data. The test services are partitioned into 19 clusters by OSM and 13 clusters by TSM. The values of *avg* □ *ircsim* and *avg* □ *iacsim* for all partitions obtained using both the models are computed and presented in Fig. 1 and Fig. 2 respectively.

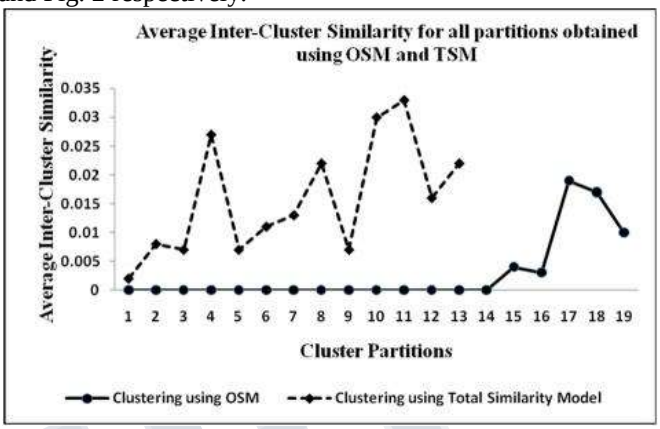

Fig. 1 (avg □ ircsim) for all partitions

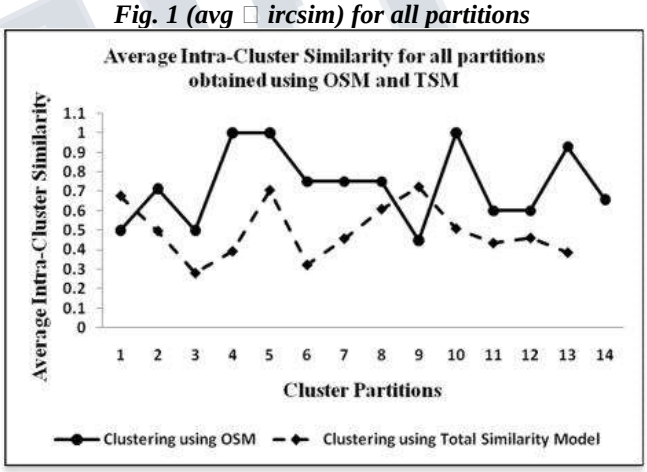

Fig. 2 avg □ iacsim for all partitions

From Fig. 1, it is found that the value of *avg* □ *ircsim* obtained using OSM is zero for 14 partitions and small (ranges from 0.01 to 0.017) for remaining 5 clusters. Whereas the value of *avg* *ircsim* of all partitions obtained using TSM have higher values ranging from 0.002 to 0.033. Out of 19 partitions produced by OSM, five are singleton clusters for which *avg* *iacsim* cannot be calculated. Hence, the value of *avg iacsim* is given for the remaining 14 clusters. From Fig. 2, it is seen that, the value of *avg* □ *iacsim* of all partitions obtained using OSM (except partition 9) is more than that of partitions obtained using TSM. From Fig. 1 and Fig. 2, it is clear that OSM yields partitions with low inter-cluster similarity (i.e. well separated clusters) and high intra-cluster similarity (i.e. highly cohesive within a cluster) when

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

compared with partitions obtained using TSM. In addition, the Silhouette Width of each partition produced by both the models is given in Table II. From Table II, it is found that out of 19 cluster partitions obtained using OSM, the Silhouette Width of 8 clusters is 1, another 7 clusters is greater than 0.75 and remaining 4 cluster partitions is greater than 0.5. Whereas the Silhouette Width of 7 partitions out of 13 cluster partitions obtained using TSM ranges between 0.3 to 0.5 and the Silhouette width of remaining 5 clusters ranges from 0.51 to 0.78.

Table II
Silhoutte width for all partitions

Clustering by OSM		Clustering by TSM	
Partition	**SW**	**Partition**	**SW**
Group-1	0.6473	Group-1	0.3764
Group-2	0.75	Group-2	0.4903
Group-3	0.75	Group-3	0.5268
Group-4	0.8125	Group-4	0.7201
Group-5	1	Group-5	0.5143
Group-6	0.8125	Group-6	0.6247
Group-7	1	Group-7	0.7419
Group-8	0.7857	Group-8	0.7719
Group-9	1	Group-9	0.3190
Group-10	0.7696	Group-10	0.3616
Group-11	0.7551	Group-11	0.4680
Group-12	0.6181	Group-12	0.4559
Group-13	0.5504	Group-13	0.4261
Group-14	0.5093	Group-14	0.4261
Group-15	1		
Group-16	1		
Group-17	1		
Group-18	1		
Group-19	1		

From the above discussion, it is understood that OSM produces partitions with low inter-cluster similarity, high intra-cluster similarity and high Silhouette Width. Based on the above study, we recommend OSM for precise clustering of services.

IV. EXPERIMENTAL STUDY II

The aim of second experiment is to analyze the appropriateness of clustering solution produced by OSM by comparing its clustering solution with that of standard methods. We have chosen two standard approaches namely syntactic similarity based clustering and Word Net similarity based clustering for comparison. As our clustering approach uses only output similarity for clustering, for comparing our approach with above two, we have tailored the above two approaches so that only output parameters are taken into account during similarity computation. In addition, as there is no logical semantic filters such as equivalent, subsumes or plug-in in both syntactic and WordNet based similarity computation, the similarity threshold is kept as (0 □ □ □ 1) . We use Resnik method [7] to compute similarity among output parameters using WordNet. The minimum value of Resnik similarity is 0. The similarity threshold is kept as (0 □ □ □ 1) . In our method, similarity threshold is keptas (0 □ □ □ 1) . This means any logical filter higher than *fail* will contribute to similarity. For simplicity, we denote the syntactic similarity based clustering as *SSM* and Wordnet similarity based clustering as *WSM*. The clustering partitions produced using SSM, WSM and OSM for the test data given in Table III, are compared against manually identified internal groupings of services in test data (please refer Table I). The manually identified grouping as per domain ontologism is taken as gold standard. The first three groups, Group-1, Group-2, Group-3 are services with no outputs. All the three approaches cluster them as singleton services as per gold standard.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

Table III
Clustering Solutions Produced By Different methods

Gold Standard	Clustering by SSM		Clustering by WSM		Clustering by OSM	
Group ID and number of services in each group(given in brackets)	Clusters produced for each group and number of services in each group	Outputs clustered together	Clusters produced for each group and number of services in each group	Outputs clustered together	Clusters produced for each group and number of services in each group	Output clustered together
Group-1 (1)	cluster-1 (1)	novel	cluster-1 (1)	novel	cluster-1 (1)	novel
Group-2 (1)	cluster-1 (1)	person	cluster-1 (1)	person	cluster-1 (1)	person
Group-3 (1)	cluster-1 (1)	link	cluster-1 (1)	link	cluster-1 (1)	link
Group-4 (19)	cluster-1 (19)	price	cluster-1 (19)	price	cluster-1 (19)	price
Group-5 (4)	cluster-1 (4)	address	cluster-1 (4)	address	cluster-1 (4)	address
Group-6 (4)	cluster-1 (4)	hotel	cluster-1 (4)	hotel	cluster-1 (4)	hotel
Group-7 (11)	cluster-1 (11)	film	cluster-1 (11)	film	cluster-1 (11)	film
Group-8 (8)	cluster-1 (8)	destination	cluster-1 (8)	destination	cluster-1 (8)	destination
Group-9 (7)	cluster-1 (7)	preparedfood	cluster-1 (7)	preparedfood	cluster-1 (7)	preparedfood
Group-10 (6)	cluster-1 (3)	videomedia	cluster-1 (3)	videomedia	cluster-1 (6)	videomedia, vhs, dvd
	cluster-2 (1)	dvd	cluster-2 (1)	dvd		
	cluster-3 (2)	vhs	cluster-3 (2)	vhs		
Group-11 (9)	cluster-1(5)	coffee	cluster-1(5)	coffee	cluster-1 (9)	coffee, cola, drinks
	cluster-2 (3)	cola	cluster-2 (3)	cola		
	cluster-3 (1)	drinks	cluster-3 (1)	drinks		
Group-12 (2)	cluster -1 (1)	icons	cluster -1 (1)	icons	cluster-1 (2)	icons, photograph
	cluster-2 (1)	photograph	cluster-2 (1)	photograph		
Group-13(4)	cluster -1 (4)	book	cluster -1 (12)	book author	cluster1(4)	book
Group-14 (8)	cluster -2(8)	author			cluster(8)	author
Group-15(7)	cluster -1 (4)	funding	cluster -1 (4)	funding	cluster(7)	funding, financing
	cluster -1 (3)	financing	cluster -1 (3)	financing		
Group-16 (4)	cluster -1 (6)	lecturer_in_academia	cluster -2 (8)	lecturer_in_academia + research	cluster(4)	lecturer_in_academia
Group-17 (1)					cluster(1)	professor_in_academia
Group-18(1)					cluster(1)	academic_support_staff

The groups, Group-4, Group-5, Group-6, Group-7, Group-8 and Group-9 are clustered by all the clustering approaches in the same manner as per gold standard. This is mainly due to all cluster groups contain keywords such as price, address, hotel and food in common. SSM and WSM fail to cluster Group-10 as a single cluster, as they are not able to find the semantic similarity among video media, dvd and vhs. Also SSM and WSM split Group-11 into three clusters as a single cluster as they are not able to find the semantic similarity among coffee, cola and drinks. But as OSM uses ontology based semantic similarity exactly identifies the similarity among video media, dvd and vhs as well as coffee, cola, and drinks

It produces only one cluster for Group-10 and Group-11 as per gold standard. For Group-12, both SSM and WSM are unable to discover the similarity among photograph and icon. They split the group into two clusters, whereas our approach could find the similarity among photograph and icon and produce only one cluster as in gold standard. Further for Group-15, both SSM and WSM are unable to find the similarity among financing and funding, cluster Group-15 as two clusters. But our method clusters them into one.

The groups, Group-13 and Group-14 have been combined as a single cluster by WSM. Because, 'book' and 'author' are found to be related in WSM. But SSM and OSM produced two clusters as expected. The groups, Group-16, Group-17, Group-18 and Group- 19 have been combined as a single cluster as WSM finds relatedness among 'lecturer', 'professor', 'academic_support_staff' and 'research'. SSM combines the groups Group-16, Group-17 and Group-18 as they have the common keyword, 'academic'. It produces one cluster for Group-16, Group-17 and Group-18 and another cluster for Group-19. But as OSM uses domain specific ontology based semantic similarity, as per ontology it does not find any similarity among the above output parameters and it produces 4 different clusters for Group-16, Group-17, Group-18 and Group-19 as expected in gold standard. From the above discussion, it is understood that, OSM produces 19 clusters as per gold standard. There are no spurious clusters or incorrect merging of groups in OSM. All the cluster partitions are found to be exactly matched with gold standard. SSM produces 23 clusters for the test data. Out of 23 clusters, only 12 clusters are produced as per gold standard and remaining 11 clusters are not correct. WSM produces 21 clusters, but only 9 clusters of them are as per gold standard. From this, it is seen that instead of using keyword based clustering or Word Net based clustering, we recommend OSM that uses domain ontology specific semantic similarity for clustering services. After clustering, to assist service discovery an indexing scheme is proposed as described in the subsequent section.

V. PROPOSED INDEXING SCHEME

In the proposed indexing method, two indices are created for each service cluster. They are output index and input index. The indexing scheme for a particular cluster, say C1 as in Fig. 3 is considered for discussion. The services of C1 have a set of output parameters, denoted by $\{op1i, 1 \square i \square m\}$ and a set of input parameters, denoted by $\{ip1\,j, 1 \square j \square n\}$ In output index of C1, the output parameters are used as keys of the index. Each key is linked to a list of three categories of services. The first category is a list of services that contain the key itself. The second category is a list of services that contain output parameters which are subtypes of the key. The third category is a list of services that contain output

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

parameters which are super-types of the key. In our indexing scheme, keys are used for retrieving services that contain *exact* output parameters, *subsumes* output parameters and *plug-in* output parameters with respect to a given key. Similarly, in the input index of C1, the input parameters are used as keys of the index. Each key is linked to a list of three categories of services, exact, subsumes and plug-in.

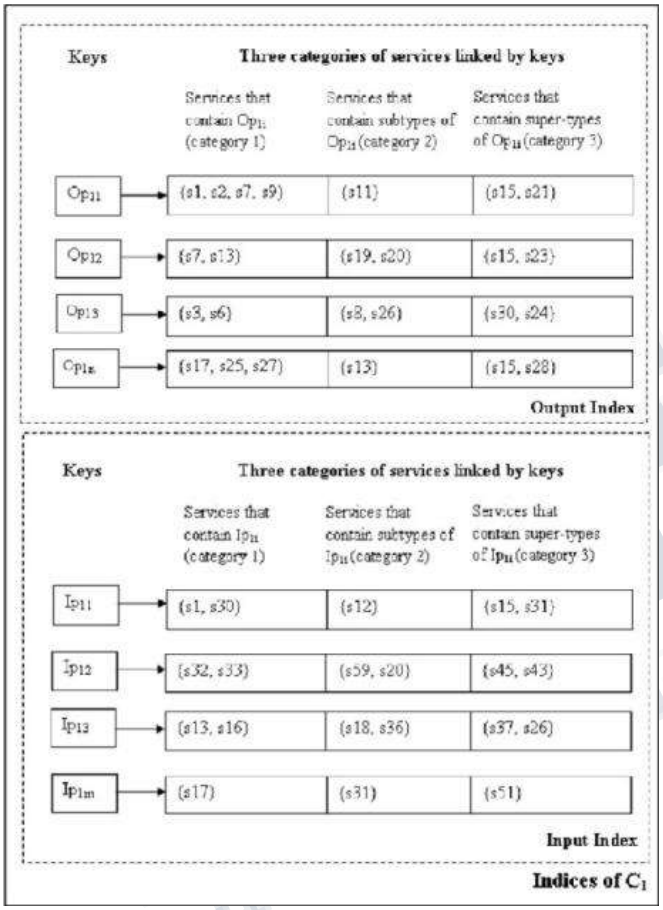

Fig. 3 indexing scheme

When a query is submitted, following steps are performed to retrieve the matched services from the indices.

Step 1: Find the cluster which is most similar to the query as relevant cluster of the query

Step 2: Fetch exact, subsumes and plug-in matches for each output from the output index of the relevant cluster. Combine the corresponding categories of services obtained for all outputs. This results in three sets of services, denoted by, *OE* ,*OS* and *OP* where *OE* contains matched services with exact DoM, *OS* contains matched services with subsumes DoM and *OP* contains matched services with plug-in DoM.

Step 3: Fetch exact, subsumes and plug-in matches for each input from the input index of the relevant cluster. Combine the corresponding categories of services obtained for all inputs. These results in three sets of services denoted as *IE* , *IS* and *IP* where *IE* contains matched services with exact DoM, *IS* contains matched services with subsumes DoM and *IP* contains matched services with plug-in DoM.

Step 4: Matched services of the query are returned as follows. Equivalent matches for the query, (*ME*) are given as

ME □ *OE* □*IE*

Subsumes matches for the query, (*MS*) are given as

Ms □ (*OS* □ *IS*) □ (*OE* □ *IS*) □ (*OS* □ *IE*) .

Plug-in matches for the query, (*MP*) are given as

( ) ( ) ( ) ( )

( ) ( )

P P P S P E P P E

P S P P

M O I O I O I O I

O I O I

□ □ □ □ □ □ □ □

□ □ □ □

VI. CONCLUSION

In this paper, an extension to one of our previous research work[6] has been taken up. Two experiments have been conducted to analyze the performance of Output Similarity Model in regard to effective elimination of irrelevancy and appropriateness of clustering for service discovery. In Experiment I, the OSM is found to be better than TSM in eliminating irrelevancy and hence OSM is suggested for clustering. In Experiment II, the clustering solution produced by OSM is compared with standard approaches namely syntactic approach and Word Net based approaches. OSM is found to produce more appropriate clusters. Further, an indexing scheme is proposed to eliminate the invoking of semantic reasoning during query. This speeds up the performance of semantic discovery.

REFERENCES

[1] Ruiqiang Guo, Jiajin Le, XiaoLing Xia, "Capability matching of Web services based on OWL-S", Proceedings of 16th International Workshop on Database and Expert Systems Applications", 22-26 August 2005, pp. 653-657.

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

[3] Massimo Paolucci, Takahiro Kawamura, Terry R Payne and Katia Sycara, "Semantic Matching of Web Service Capabilities", International Semantic Web Conference, Springer Verlag, LNCS, Vol. 2342, 2002, pp. 333-347

[4] Richi Nayak, Bryan Lee, "Web service discovery with additional semantics and clustering", WI '07 Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2007, pp.555-558.

[5] Sonia Ben Mokhtar, Anupam Kaul Nikolaos Georgantas and Valerie Issarny, "Towards Efficient Matching of Semantic Web Service Capabilities," *Proc. of International Workshop on Web Services Modeling and Testing* (WS-MaTe 2006), pp.137-152, 2006

[6] Chellammal Surianarayanan, Gopinath Ganapathy, "An Approach to Computation of Similarity, Inter-Cluster Distance and Selection of Threshold for Service Discovery using Clusters", IEEE Transactions on Services Computing, no. 1, pp. 1, PrePrints, doi:10.1109/TSC.2015.2399301

[7] Philip Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy,", *Proc. of the 14th International Joint Conference on Artificial Intelligence* (IJCAI), vol. 1, pp. 448-453, 1995.