Improving Web Service Discovery using Community of Interest

Vandan Tewari, Nirmal Dagdee, Aruna Tiwari

Abstract— Web Services are the new breed of autonomous ubiquitous applications over Internet which are based on Service Oriented Architecture. As web services gain popularity and become more sophisticated and interdependent, the publication and discovery of web services become important area of research. This paper proposes a Client Centric Service Oriented Architecture (CCSOA) which emphasizes on Client and his preferences for web service discovery. The web service discovery can be improved using the information contained in a user's profile and his past usage history and taking advantage of the fact that people with similar profiles tend to prefer similar kind of services.
In this paper a proposal has been given for enabling Client Centric Discovery using service categorization, service rating, session tracking and profile management techniques. This will enable a user to search for an appropriate and effective web service. For selection of a service appropriate to a client his membership to community of interest (CoI) has been used. The communities of interest have been generated based on user profiles and past usage data to help a user discover services of his choice where a user may have a membership in various CoIs. An e-shopping domain has been used to demonstrate the implementation of proposed technique.

Index Terms— Client centric discovery, Community of Interest ,service categorization,SOA, User Centric, User Profiles, Web services,

1 INTRODUCTION

WEB Services[1] are the reusable software components over the Internet which provide the building blocks for developing the next generation applications using the Service Oriented Architecture SOA [2]. The basic Web Service Architecture is made of three entities: Service Provider, Service Requestor and Service Broker where the Service Provider advertises his service. One of such broker realization is UDDI [3] which is a specification for registry of information for Web Services. It is a platform independent XML based registry for the businesses spread across the world to list them on Internet. As the Web Service paradigm becomes popular, the issue pertaining to discovery of the most appropriate service becomes more significant. Service discovery is process of finding an appropriate service provider for the demanded or requested service. The service discovery procedure not only consists of sending and receiving simple information in a distributed environment but may have intricate technical issues to be considered. The focus of service discovery is also on where to find the appropriate services, and whether they are available and behave as advertised. To discover a service appropriate to a client, additional information can be generated about a client from his profile information and his past service usage data. The users having similar usage behavior and similarity in profile can benefit from experiences of their peers, and therefore communities can be generated based on user clusters for such recommendations.

2 A BRIEF SURVEY OF WEB SERVICE DISCOVERY

Earlier businesses on Internet essentially provided their services through a distributed paradigm where distributed object technologies such as DCOM and CORBA were used. These technologies presented some serious drawbacks of being dependent on the environment. To mitigate these drawbacks, the present SOA based Web Services were developed. The basic Web Service model was presented in year 2000 [4] where service could be published using WSDL[5] documents at UDDI registries, discovered from UDDI registries and then invoked by the service requestors.Today's scenario of Web Service discovery does not comprise of choosing a service with needed functionality but also appropriateness of the service for usage context. Since Web Service repositories can be tagged with a knowledge base, search or discovery now can be made intelligent. Web Service discovery mechanisms allow access to service registry – repositories that can warehouse information about business service and further details.The basic keyword based search techniques do not capture the underlying semantics of the web services and mainly focus on comparing keywords in user's query with syntactic description in WSDL. These techniques therefore may result in irrelevant search results. Further these techniques suffer from low recall and low precision [6]. So far many proposals have been given to improve keyword based discovery. In [7], the authors propose to use a custom build search engine Woogle to perform similarity search on finding similar web-service operations in as-

- *Vandan Tewari is currently pursuing Ph.D. program in Comuter engineering in RGPV University, M.P.,India, PH-0731-2541564 She is presently working as Asst. Prof. at Dept. of Comp.Tech.&Applications at Shri G.S.Inst. of Tech.&Sc.,Indore E-mail: vandantewari@gmail.com*
- *Dr. Nirmal Dagdee is working as Director and Professor in Dept. of Comp, Engg. at, S. D. Bansal College of Technology, Indore. He has completed his Ph.D. in Comp. Engg. from RGPV, Bhopal in year2003. His area of research includes Soft computing, Neural Network Learning Algorithms &, Web Services. E-mail: nirmal_dagdee@rediffmail.com*
- *Dr. Aruna Tiwari is working as Assistant professor at School of Computer Sc. & Engg. IIT Indore Indore. She has completed her Ph.D. in Comp. Engg. from RGPV, Bhopal in year 2009. E-Mail: atiwari@iiti.ac.in*

sumption that service operations are more important for using a service then the service metadata. They have defined the similarity in pairs of web service operations and then clustered these operations into semantically meaningful concepts so that a user can ask for a web service similar to given. Wu et al.[8] search for similar web services based on structure matching of data types in WSDL. However this matching will not suffice if two web service operations have multiple sub structures on data types. In another approach of web service discovery [9], agent based discovery was proposed. They used a group of agents for finding services related to them. In yet another proposal [10] ,the authors propose to cluster tags and WSDL to cluster similar services together so as to improve relevancy in search results. However since generation of tags depend on users who are using the services or services providers who are providing the services and hence depend on the information they supply for the services.

To improve keyword based search there are also proposals for using non functional parameters for and Web Service crawler engines[11][12] which have their own set of advantages and disadvantages. These techniques can actually overlook a user's context and semantics while discovering for services.

Ying Chen et al.[13] have proposed an extended UDDI architecture to enable service mining in UDDI where authors apply various data mining techniques to categorize the available services and rate them thereby allowing discovery of a service which is more useful to the requester. They have also proposed an architecture where service usage mining has been used to make dynamic Communities of Interest (COIs) and further allowing service access logs to be used for service rating thereby allowing selection of a service more appropriate to need of a user. They have stressed upon the fact that the users are in same community if they use similar kind of services which may not always suffice. Especially the new users can not benefit from such proposals. It would be fitting if users profile as well as usage data could be used for making communities. Further Balke et al. [14] have proposed a model for cooperative discovery for personalized Web Services while proposing an ontological algorithm, making use of user preferences. As a step towards improved selection, Mark Chang et al. [15] have proposed an improved user centric SOA for automated discovery of Web Services using concepts of COIs (Communities of Internet) for service matching where they propose to make communities of services offering similar type of functionality.

As semantics have started getting popularity for service description, service discovery can also be based upon semantics. In context with it, Pathak et al. [16] have proposed a framework for ontology based flexible discovery of semantic Web Services. Their approach relies on user supplied context specific mappings from user ontology to relevant domain ontologies to specify Web Services. In another proposal by Aabhas V. Paliwal et al. [17] the user request is enhanced semantically and an efficient matchmaking has been done using latent semantic indexing and service clustering.

3 The Proposed Approach

A framework for client oriented web service discovery has been proposed in this paper which our especially developed discovery cum publishing engine [18] which considers user profiles, and past service usage data to generate Community of Interest (CoI) for a group of users. It then uses user supported service rating to generate service ranking. To optimize and simplify the process of web service discovery, similar services have been clustered to form groups of services each represented by a set of keywords. We have used e-Shopping services to demonstrate the proposed design. The proposed design not tries to make the process of service discovery simpler for the client but also makes it efficient.

3.1 Client Centric Service Oriented Architecture

The proposed Client Centric Service Oriented Architecture (CCSOA) has a logical layer of discovery cum publishing engine that resides between the UBRs and client or providers. CCSOA is an extension of conventional SOA. The conventional SOA is governed by the services providers as service providers publish their services and service consumers must search from these available services. Based on SOA, CCSOA is built over conventional SOA to provide support for end users and to be able to provide services most appropriate to client's need. A comparision of conventional SOA and CCSOA has been given in Table 1.

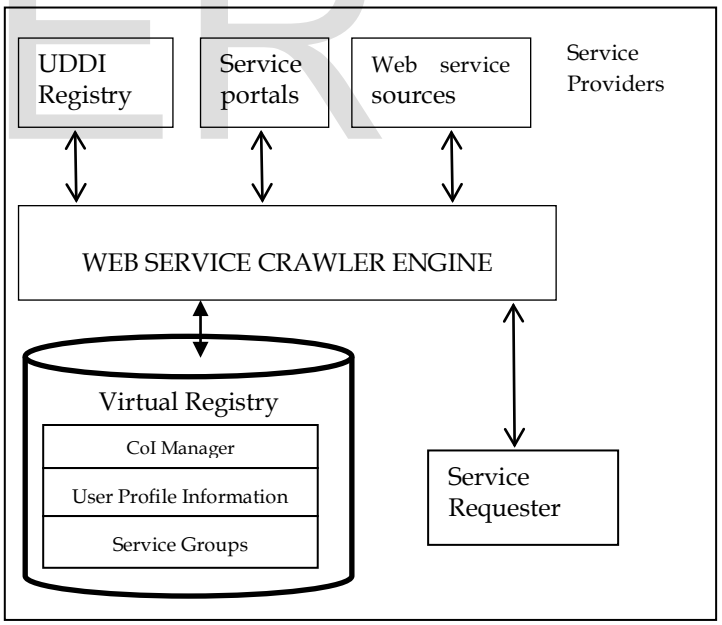


Fig.1: High Level Architecture of Client Centric SOA

TABLE 1. Comparision of SOA and Client Centric SOA

	SOA	Client Centric SOA
Service related-Publishing Information	Service descriptions including methods, input output operations in WSDL	In addition to specifications of SOA, Users profile information along with his interests and service tags supplied by the provider.
Where to publish	UDDI data structures	Modified UDDI data structures as well as CoI Manager keeps the information about profiles and service manager keeps information about categorization.
Service Matching	Matching is done on WSDL Contents	Matching is done using extended index of keywords
Service Discovery	Using Inquiry API on Known Parameters	Based on knowledge of user profile and service usage data
Recommendation	None	Recommendations based on CoI information as well as services categories.
Service Ranking	None	Services are rated by the users and offered services are ranked accordingly

3.2 Publishing Web Services in Proposed Framework

To make a web service available to a user, it necessarily needs to be published in a UDDI Registry. The proposed design of engine eases the task of publishing a web service. Each service provider has to under go a registration process which requires simple information to be furnished by the service provider. During publishing a publisher is asked to supply keywords as a description to the service he is providing in addition to the required information. This helps the search engine later on in discovering appropriate services using a keyword based search.

3.3 Discovering Web Services in Proposed Framework

In current web service architecture, finding a web service of interest means scanning through service registries. As web services proliferate, size and magnitude of UDDI Business Registry (UBRs) are likely to increase. With multiple services having similar functionality, it is difficult for a user to select appropriate services. It may even happen that none of the discovered services may actually fulfill his needs.Therefore a framework has been designed to enable a client oriented web service search which uses a specially designed web crawler engine [18] that facilitates a better search/discovery based on user's interest and other parameters .For demonstrating the proposed design an e-shopping domain has been chosen. Service rating module, user profile management and service registration module have been used for necessary functions as proposed and described later.

Consider a scenario of a user visiting an e-shopping site where purchasing each product is a web service. Since different products are being provided by different service providers, these web services are published at different UDDI Registries. For purchasing a product, a user has to register first and the information provided by him will be further used for discovering the services according to his interest. For generating communities of interest ,the attributes have been taken from the user profiles such as age, salary, interest, ethnicity, location etc. Further data mining techniques have been used to make search more appropriate e.g.the search results for a product will be presented in a ranked list as per ratings provided by other members of the community to which this user belongs. A product which has been purchased the most and has been rated best by the members of his community will appear on top of the result. This helps a user select a better service.In addition since similar services are clustered together, the number of services to be searched fairly reduces.

3.4 Application of Dynamic User Profiles for Generation of CoIs

User profiling is the process of gathering information specific to each visitor, either explicitly or implicitly. A user profile includes demographic information about the user, her interests and even her behavior when searching for a web service. In order to construct an individual user's profile, information may be collected explicitly, through direct user intervention, or implicitly, through agents that monitor user activity. Profiles that can be modified or augmented are considered dynamic, in contrast to static profiles that maintain the same information over time. This information is exploited in order to customize the web service search results to a user according to his needs. User profile allows discovering service according to user's interest and criteria. It not only provides user's brief information but also helps to generate results which are more appropriate to the user. As an example, if a user wants to search for a restaurant booking service, with the help user profile we can match the user's city with provider's city so that search result can contain same city products on the top of the search results. Similar kind of rules can be applied to user's income and his age. Usages of data mining techniques have been used to make clusters of users on basis of their profile information and past usage history. The schema design for this user profile table is as follows.

Profile {Uid, Age, Occupation, Salary, Location, Interest, usage}

For capturing this information the GUIs have been generated and the final data set contained preprocessed and normalizeed data. The preprocessing had been done so that the clus-

tering algorithm can be applied. For example in our case we represented salary which is a continuous attribute by an ordinal categorical attribute using rule based classification on suitable split points and then converted it to binary values. Similarly the interest attribute was predefined to be a domain of six values such as {cars, shoes, holidays, sports, electronics, books}. It was coded as a binary vector of length six as per choices of users. The 'usage' contains past service usages of the user for the considered period of three days. E.g. If a user has used a service as {0,0,1,0,0,1} with weights {0,0,2,0,0,3}, it represents that this user has used a holiday service twice while he has used a book buying service thrice. This information can actually tell us about the user's behavior according to which a usage profile can be generated. However if user is new the same algorithm can work on the supplied data at the time of registration. To find the similarity between profiles a K-Means clustering algorithm [19] was used with a value of K=4 i.e. initially four cluster centroids were initialized. Since past service usages of a client keep on changing, the membership of a user can change in CoI and hence the profiles that are being considered are dynamic.

4 Proposed Architecture of the System

The framework for client oriented web service discovery has been designed using various modules. There are two major components of the designed system: an offline component which is a batch processing application and runs offline. It generates the dynamic profiles by adding the last service usages to static profile information. Second is an online component which allows the search engine to access these dynamic profiles for producing personalized selection of services for a user. The description for the same has been given below.

Log parsing: this offline component takes a log file as its input and parses it to extract information including user_id, requesttype, service_name, service_key etc. If a user wants the personalized selection, his user_id should be present in the log to collect his activities.

Web Service Retrieval and categorization: After a service is identified by its service_key and name, the metadata of WSDL and keywords from service_description are used to categorize services in clusters using supervised clustering. We have categorized services into six categories namely buying services related to different domains. Note that this construction of category cluster is done only once. Each category cluster is represented by a vector of about fifteen to twenty words presently. For improving this categorization , a semi supervised approach may be used which can be done by using clustering on available service documents in order to create desired clusters of services and once such clusters are identified using a lexical database such as Wordnet[20] to define a category of services. We have used a vector space model to represent services where each service is represented by a term vector of form $K=(k_1,k_2,\ldots k_n)$.We have then used a common approach of finding document similarity i.e. tf-idf[21]. We have modified the technique to suit our cause by representing the term vector as the service category vector.Each term k_i has a weight associated with as w_i, where w_i represents the frequency of occurrence of keywords in the given six categories. Each category is represented by synonyms in this case. In cosine normalization each term's weight is calculated as

$$w_i = tf_i.idf_i / (\sum (tf_i.idf_i)^2)^{1/2}$$

Where tf_i represents the term frequency of k_i and idf_irepresents the inverse document frequency.The inverse document frequency represents the inverse of number of services in which the given keyword category is present in training set. Now the similarity between two services can be found by calculating cosine distance between service descriptions with given weight vectors. The cosine distance between two vectors s_1 and s_2 is given as

$$cosine(s_1,s_2) = s_1.s_2 / |s_1||s_2|$$

Generation of Communities of Interest: In a community of interest, each client is represented as a vector of service categories and its associated weights along with the data supplied by the client at registration. Members of a group represent a user community which share similar kind of profile as well as service usage. These communities are stored in the virtual registry and are used while offering services to a user. A user can belong to more than one community and initially belong to all communities to whom the profile matches.

4.1 Publish Module

This module simplifies the tedious process of publishing by providing a simple interface for the process of publishing .It is responsible for publishing web service on a group of cooperating UDDI nodes in the framework. For publishing web services, publisher profile and his last publication history is used to make publishing process fast and efficient. Publisher routine publications are used to speed up publishing process. During publication the service provider is also asked to supply tags regarding the description of the services that he has published to help categorize the services in virtual registry. This tag information is stored in UDDI Registry by extending the data structures.

4.2 Search Module

The search module helps discovering appropriate and desired web service using the CoI information. This module communicates and interacts with other modules for clustering all related web services in each category so that most highly ranked service is on top most position on the selected offeredservices.

4.3 Rating Module

This module undertakes the responsibility of collecting rating of services and then ranking them on basis of client ratings which is then used by search module for service selection from CoI information. The service user is asked to rate a service between 1-5 using the GUI, where 1 represents the worst case while 5 represents the best case. These ratings from the members of each CoI are collected and rank aggregation is done using average ranking model. When a new service is

published and has never been used, it is aged and given a default service rating of 2.0. Later if this service is being used at least ten times and cannot improve its rating ,it is penalized by 0.5 each time. E.g. If a service was new and was given initial rating of say 2.0 and it was used six times and given ratings as (2,3,4,3,2,4). The average rank in this case would be 3.0. Since the service has improved its rating, its current rating would be 3.0. However if its rating on six usages was say (2,2,1,3,1,3). The average rank becomes 2.0.Since the service was not able to improve its rating, it will be penalized by 0.5 and the current rating would become 1.5.

4.4 Updating Module

The updating module is responsible for service review management and transaction history management .Updating module is also responsible for client profile updating by initializing offline process of log collection and creating service categories.This in term is also responsible for shopping cart and session management to collect the service usage data.

4.5 CoI Generator

The CoI Generator contains two modules: **A CoI Manager** module uses data mining techniques on client profile and captured information about client to find out his community of interest. Configured CoI is used to select the web services appropriate to a client. Further it also determines the membership of each user for different CoIs. The change in membership is done using past service usages by a given service user over a period of three days interval. **A Profile Manager** module manages user profiles who can be either service requestors or service providers. This is further used by CoI generator and search module.

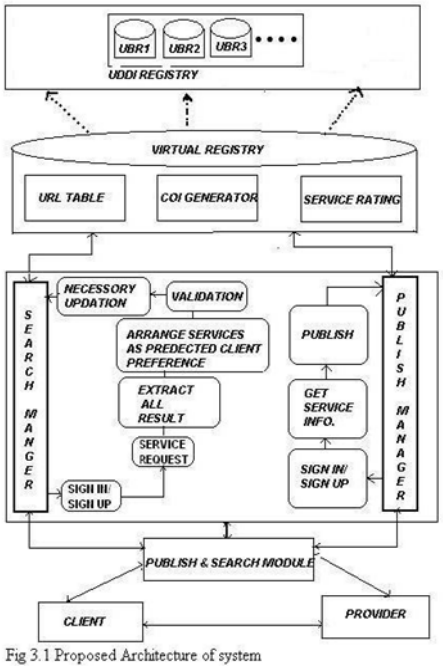


Fig 3.1 Proposed Architecture of system

Fig 2. Proposed Architecture of the System

4 Algorithms Used for Demonstrating the Proposed Work

Here we are presenting the basic algorithms used in publication and discovery of services.

4.1 Algorithm for Publishing Web Service

Step 1: Start
Step2: Sign in/sign up publisher.
Step 3: Ask for routine publishing.
Step 4: If yes
Ask required information, Ask the Publisher for tagging the services ;Publish in selected UDDI Nodes ; Keep the Category information about services in Virtual registry;.
Else
Select UDDI Registry and feed necessary information publish.
Step 5: Stop.

4.2 Algorithm for Searching Web Service

Step 1: Start
Step2: Sign in/sign up client.
Step 3: Accept Query and parse it to generate keywords
Step 4: Select the category of required service and fetch results from Virtual Registry
Step 5: Select the results as per user's CoI Membership.
Step 6: Rank the search results as per ratings given by other members of chosen CoI.
Step 7: Present user the ranked results for selection of service
Step 8: If user has used a service, ask for the feedback.
Step 9: Perform necessary updations like Rating management, Shopping cart etc.
Step 10: Send the service usage data to the COI Generator.
Step 11: End

5 Result and Test Cases

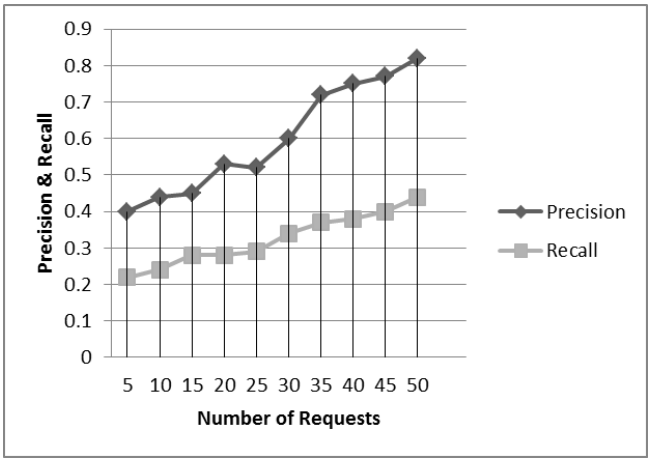
To illustrate the usefulness of proposed approach, an illustrative case study was taken on synthetic data. We have deployed the proposed system for an e-Shopping domain. We collected the data by asking a random group of five people to register and (some of them were aware of web services some were not) to classify the search on relevancy for any search results.The people were asked to give a yes or no for the fetched results. Each user was asked to invoke ten to fifteen searches each. In total 162 web services were deployed on 10 UBRs.When using simple search on keywords versus the proposed system, the results are presented as per relevancy table.E.g when a user searched for buying cars, he was presented by 18 results out of which he picked out only 11 to be relevant in simple search. However when he searched it using the proposed approach, he was presented with 12 results, all of which were relevant according to him. The recommendations were measured using precision and recall ,where precision-

measures the fraction of relevant items retrieved among those retrieved while recall represents the fraction of relevant and retrieved items in retrieved. We are presenting the average case scenarios for all the searches in Figure 2 which represents the precision and recall calculated for first 50 requests submitted to the system by five users. In Table 2 we presen comparision of simple search and our system using the relevancy of fetched results in average case.

TABLE 2: Results for Relevancy of Fetched Web Services

Category of Service	Simple search		Searching with Proposed approach in Client Centric SOA	
	No. of services in result	Relevant services fetched	No. of services in result	Relevant services fetched
Buying cars	18	11	12	12
Buying shoes	17	12	13	11
Buying holiday	12	8	9	8
Buy sports goods	15	10	11	10
Buying electronics	16	10	12	10
Buying books	18	11	12	12

Fig 2: Performance of Proposed system for first 50 requests



6 Conclusion and Future Work

On the advent of service-oriented computing, exploring appropriate business applications published as web services in UBR is mainly done by scanning through UBR. As the number of web services increases, the success of business will depend on publishing a service, service discovery and performance time when searching multiple UBRs. In this proposal using concept of **Client Centric Service Oriented Architecture, E-Shopping** domain has been presented for the purpose of effective and appropriate discovery of web services with in acceptable performance time. The presented searching scheme incorporates intelligence mechanism which hire client Centric approaches like profile management and generation of Community of Interest(CoI) to narrow down the search space of UBR hence reduces the search response time. As resultant, system saves cost and time and try to present the services according to user's interest. Our system has an edge over the other existing discovery mechanisms in number of ways: first of all it has employed dynamic user profile mininig technique which suggests appropriate service to client and assures that search result will feasible to end user. Further a large variety of discovered services can be offered to a requestor since it uses an enhanced web service crawler engine[18] which dynamically finds services available on Internet therby enhancing the quality of services. In future this work may be extended for Business to business interactions also where client can be a user or a business.

References

[1] Web Services, www.w3schools.com/webservices/ws_intro.asp

[2] Service Oriented Architecture http://java.sun.com/developer/Books/j2ee/jwsa/JWSA_CH02.pdf www.service-architecture.com/web-services/articles/

[3] Universal Description Discovery and Integration (UDDI) Version 3.0.2 UDDI SpecTechnical Committee Draft, Dated 20041019, Available at:http://www.uddi.org/pubs/uddi_v3.htm, Accessed on 18.04.2013.

[4] "Web Services Architecture", David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Iona Michael Champion, Chris Ferris and David Orchard ,W3C Working group, http://www.w3.org/TR/WS_arch

[5] Web Services Description Language (WSDL) version 2.0 Part 1 Core Language, W3C working draft 26 March 2000 www.w3.org/TR/2004/WD-wsdl20-20040326/wsdl20.pdf (IEEE Transactions)

[6] E. Stroulia, Y. Wang, "Structural and Semantic Matching for Assessing Web- Service similarity", IJCIS 5:14, pp. 1-30 (2004).

[7] Dong, X.,Halevy, A.,Madhavan,J.,Nemes,E. and Zhang, J., "Similarity search for web services", International Conference on Very Large Databases. pp. 372-383, 2004. (Conference proceedings)

[8] Wu, Jian; Zhaohui Wu, "Similarity-based Web service matchmaking," IEEE International ConferenceonServices Computing, 2005, pp.287-294vol.1. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1531265&isnumber=32669.(Conference proceedings)

[9] Paul Palathingal and Sandeep Chandra, "Agent Approach for Service Discovery and Utilization", Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4 - Volume 4 , Page 40111.3 ,IEEE Computer Society Washington, DC, USA. (Conference proceedings)

[10] Liang Chen, Liukai Hu, ZibinZheng, and Jian Wu, "Utilizing Tags for clusteringWeb services", In Proceedings of. 9th International Conference on Service Oriented Computing, Paphos, Cyprus,Dec. 5-8, 2011.(Conference Proceedings)

[11] Al-Masri, E.; Mahmoud, Qusay H., "Toward Quality-Driven Web Service Discovery," IT Professional,vol.10,no.3,pp.24-28,May-June2008 doi:10.1109/MITP.2008.59
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4525538&isnumber=4525527. (URL for Conference Proceedings)

[12] E. Al-Masri, and Q. H. Mahmoud, "WSCE: A crawlerengine for large-scale discovery of web services," ICWS pp.1104-1111, 2007.(Conference Proceedings)

[13] Ying Chen, Brad Cohen. Booz Allen Hamilton "Data Mining and Service Rating in Service-Oriented Architectures to Improve Information Sharing", In Proceedings of Aerospace Conference,March,2005 ieeexplore.ieee.org/iel5/10432/33126/01559624.pdf?arnumber=1559624.(URL for Conference Proceedings)

[14] W.-T. Balke, M. Wagner. "Towards Personalized Selection of Web Services". In Proceedings of the IEEE12th Int. World Wide Web Conference (WWW 2003), Budapest, Hungary, 2003.http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.5811&rep=rep1&type=pdf (URL for Conference Proceedings)

[15] Chang, M.; He, J.; Tsai, W.-T.;Bingnan Xiao; Yinong Chen, "UCSOA: User-Centric Service-Oriented Architecture," IEEE International Conference one-Business Engineering, 2006. ICEBE '06,pp.248-255. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4031658&isnumber=4031614(URL for Conference Proceedings)

[16] Jyotishman Pathak, Neeraj Koul,Doina Caragea ,Vasant G Honavar, "A Framework for Semantic Web Services Discovery",, In Proceedings of ACM International Workshop on Web Information and Data ManagementWIDM'05,November 5, 2005, Bremen, Germany.(Conference Proceedings)

[17] A.V. Paliwal, B. Shafiq,J. Vaidya, HuiXiong,N. Adam, "Semantics-Based Automated Service Discovery", IEEE Transactions on Services Computing,pp. 260-275,April 2012.(IEEE Transactions)

[18] Tewari, V.; Dagdee, N.; Singh, I.; Garg, N.; Soni, P., "An Improved Discovery Engine for Efficient and Intelligent Discovery of Web Service with Publication Facility," Services - II, 2009. World Conference on SERVICES-2 '09., vol., no., pp.63,70, 21-25 Sept. 2009 doi:10.1109/SERVICES-2.2009.21
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5305984&isnumber=5305957 (URL for Conference Proceedings)

[19] Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y. (2002). "An efficient k-means clustering algorithm: Analysis and implementation". *IEEE Trans. Pattern Analysis and Machine Intelligence* **24**: 881–892. doi:10.1109/TPAMI.2002.1017616. Retrieved 2013-04-24.

[20] George A. Miller (1995). WordNet: A Lexical Database for English,Communications of the ACM Vol. 38, No. 11: 39-41. http://wordnet.princeton.edu (ACM Communications)

[21] Salton G, Buckley C (1988). "Term-weighting approaches in automatic text retrieval". *Information Processing and Management* **24** (5): 513–523. doi:10.1016/0306-4573(88)90021-0 (Journal Proceedings)