Is Service Discovery necessary and sufficient – A Survey

K.V. Augustine, E.Rajkumar,

Department of Computer Science
Pondicherry University
Puducherry
augustine.k.v@gmail.com

Abstract— Service computing is a cross discipline technology emerged to fill the gap between the information technology and business process. A Service is an operation or set of operation provided by an entity to another entity though contracted interface. The success of its utility lies on its capability to abide to the requirment of its consumer. The serice said to be optimum if the preference of the puller is maximum or completely satisfied. In this paper we evaluate some of the apporaches that has emerged for and technically termed as service discovery. Here we also provide the challenges that has to be addressed when defining the discovery process and how far they has been satified. We aslo discuss some of the Quality of Service (QoS) factors that are to be considred.

Keywords – Service Computing, Service Discovery, QoS.

1 Introduction

Service oriented computing (SOC) has gained momentum with the paradigm shift from proprietary standard to global standard. The overview of the service computing is given in fig.1. The origin of SOC is from vast areas of computing such as component based, enterprises integration business modeling etc. But today it has gone far ahead of its origin into emergence of areas like Web2.0 applications, business process management, software as service, data as a service, and cloud computing. Still the need of technology to merge the business solutions and organization structure to find the organization cons and address them with a solution is enormous.

The research challenges that are faced today in the area are Dynamically (re-)configurable run-time architecture , Dynamic connectivity capabilities, Topic and content-based routing capabilities, End-to-end security solutions, Infrastructure support for application integration, Infrastructure support for data integration, Infrastructure support for process integration, Enhanced service discovery Moreover, quality properties need to be addressed. The competitive push of the providers and competence pull of consumers has made this environment an ever green research area.

In this papers we analyze the approaches in enhanced service discovery. As the number of services has increased the effort needed by the consumer to run through each service to find the service of his choice is taxing. To overcome this challenging an effective discovery mechanism is significant. The major challenge lies in digging out the optimal service precisely. The basic sources for discovery are repositories where the description and location of the service is depicted and service descriptions like WSDL where the interface, operations input and output parameters are provided.

The remainder of this paper is organized as follows : Section 2 Challenges in service discovery 3 Service discovery taxonomy 4 survey on discovery 5 Quality of Service 6Conclusion and future direction.

2 Challenges in Service Discovery

2.1 Preferred service tracing

Due to ubiquitous growth of services the major challenge lies in discovering the service that best suits to the user demand from among the available offer.

2.2 Uniform description

The specification of the provides and the goal demand of the consumer must have common standard so that the matching of the demand and offer can be made easier. The major challenges lie in developing a standard specification for both service and goal.

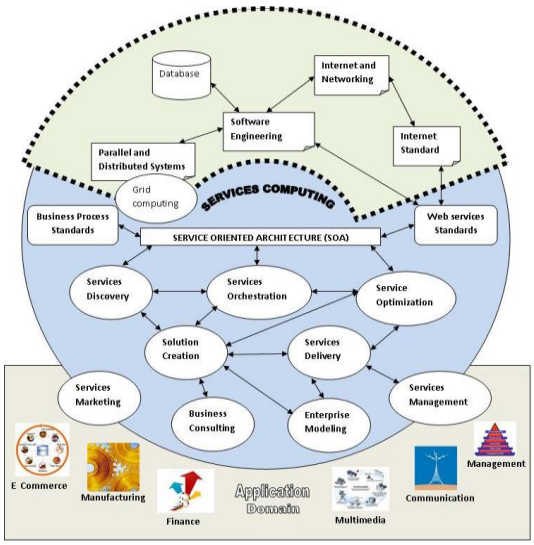

Fig1: Overview of Service Computing

2.3 Semantic annotation

The UDDI registry of a service support only keyword based search of the service. The underling semantic of the demand or the offer may improve the efficiency of the discovering mechanism. A better annotation techniques that provides the underlying semantics is a major challenge in discovery

2.4 Clustering approach

Clustering plays a vital role in reducing the search space for discovery. Most of the clustering approach are based on the functional aspects of service that is similarity measure of input, output, precondition and effects. A better clustering method that can incorporate both functional and non functional approach is a major challenge in service discovery.

2.5 Mediator engine

The source of service description are heterogeneous in nature, a need for an mediator that can extract the description from the these heterogeneous sources and provide them for the discovery frame work is another challenging issues in service discovery.

2.6 Similar service discovery

Discovering set of services that are similar in nature to the service in hand is another challenging issue that has be addressed in recent year.

2.7 Storage model

Due to increase in number of services a service aggregator which acts a intermediate storage by clustering services based on functional and nonfunctional characteristics of service has became essential. A better storage model that can enable easy access and retrieval of service is a challenging issue.

2.8 Matching and Ranking

Matching is an important in aspect in service discovery. The challenge lies in providing a best matching method that matches the demand and the offer. Similar to matching ranking of service plays a vital is discovering the service that best suits the demand constrain. A ranking model based on quality is a challenging task in web service discovery.

3 Service discovery Taxonomy

In this section we discuss some the requirement and importance of service discovery together with some types and support for discovery. The overall taxonomy is depicted in fig 2.

3.1 Requirements of Service Discovery

We argue that the following requirements, over and above the generic requirements of services, are necessary to support service discovery in any context:

- Descriptions must be attached to different resources (services and workflows) published in different components (service registries, local file stores or databases)
- Publication of descriptions must be supported both for the author of the service and third parties
- Different classes of user will wish to examine different aspects of the available metadata, both from the service publisher
- There is a need for control over who make add and alter third party annotations
- We must support two types of discovery: the first using cross-domain knowledge; the second requiring access to common domain ontologies
- A single, unified interface for all these kinds of discovery should be made available to the user.

3.2 Importance of Service Discovery

To illustrate the importance of service discovery, the following impact shows the way for it. Alternatively one could assume to directly query the web service during the web service discovery process. However, this may lead to network and server overload and it makes a very strong assumption: in addition to data mediation, protocol and process mediation for the web service must be in place before the discovery process even starts. Without thinking that this is a realistic assumption, in consequence assumption is made that it is essential.

Taking the analogy with databases as illustration, web service discovery is about searching for databases that may contain instance data we are looking for, while service discovery is about finding the proper instance data by querying the discovered databases. The same analogy can be extended to consider services that imply some effects in the real world. Service discovery is about checking whether the ticket sellers offering such web services can really provide the concrete requested ticket. With this example, the importance of service discovery is known as clear crystal.

3.3 Support for retrieval of Service

3.3.1 Keyword-Based Retrieval:

Search based on keywords from the service request. This method is highly sensitive to the ‘zero or a million’ problem because keyword are a poor method to capture the semantics of a request. Keywords can be synonyms (i.e., syntactical different words can have the same meaning) or homonyms (i.e., equal words can have different meanings) leading to low precision and recall. Furthermore, the relationship between different keywords in a request cannot be handled.

3.3.2 Table-Based Retrieval

It consists of attribute value pairs that captures service properties (e.g., output = article name). Services and requests are both represented as tables with attribute-value pairs and then matched. Semantics are more precisely captured in this method than in keyword-based retrieval but still the problems with synonyms and homonyms exist.

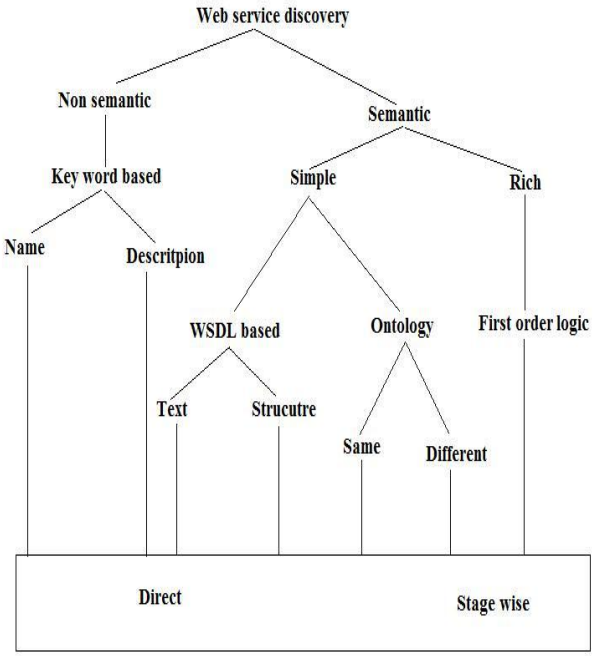

Fig 2: Service Discovery Taxonomy

3.3.3 Concept-Based Retrieval

Defines ontologies for classification of services thereby enabling retrieval on types rather than on keywords. This approach can give increased precision and recall difficult to develop and manage (i.e., difficult to define a consistent ontology of the world, how to combine ontologies with contradictory concepts).

3.3.4 Deductive Retrieval

In this approach, service semantics are expressed formally using logic. Retrieval then consists of deducing which service achieves the functionality described in the query. Theoretically, this method can achieve perfect recall and precision. The problems of this method are more practical. Formal modeling of service description and service request is very hard. Furthermore, the matching process in this method (proofing) can have high complexity and therefore operates slowly.

3.4 Types of Discovery

3.4.1 Goal Discovery

Users may describe their desires in a very individual and specific way that makes immediate mapping with service descriptions very complicated. Therefore, each service discovery attempt requires a process where user expectations are mapped on more generic goal descriptions. Notice that this can be hidden by the fact that a discovery engine allows the user only to select from predefined goals. However, then it is simply the user who has to provide this mapping i.e. who has to translate his specific requirements and expectations into more generic goal descriptions. This step can be called goal discovery,i.e., the user or the discovery engine has to find a goal that describes (with different levels of accuracy) his requirements and desires.

3.4.2 Service Discovery

Service discovery is based on the usage of web services for discovering actual services. Web service technology provides automated interfaces to the information provided by software artifacts that is needed to find, select and eventually buy a real-world service or simply find the piece of information somebody is looking for.

3.4.3 Web Service Discovery

Web service discovery is based on matching abstracted goal descriptions with semantic annotations of web services. This discovery process can only happen on an ontological level i.e., it can only rely on conceptual and reusable.

3.5 Service Discovery Approaches

Service Discovery is done based on the following ways:

- Service Discovery based on Keyword Matching
- Service Discovery based on Simple Semantic Description of Service
- Service Discovery based on Rich Semantic Description of Services

3.5.1 Service Discovery based on Keyword Matching

It is a basic ingredient in a complete framework for semantic web service discovery. By making a keyword base search the huge amount of service can be filtered or ranked rather quickly. In a typical keyword based scenario a keyword based engine is used to discover services. A query, which is basically a set of keyword, is provided as input to the query engine. The query engine matches the keywords from to query against the keyword used to describe the services. A query with the same meaning can be formulated by using a synonyms dictionary, like Word-Net. The semantic of the query remains the same but because of the different keyword used, synonyms of previous ones, more services that possible fulfill user request are found. Moreover, by using dictionaries like WordNet as well as natural language processing techniques increases of the semantic relevance of search result.

3.5.2 Service Discovery based on Simple Semantic Description of Service

It uses the controlled vocabularies with explicit, formal semantics. Ontologies, which offer a formal explicit specification of a shared conceptualization of some problem domain, are excellent and prominent conceptual means for this purpose. They provide an explicit and shared terminology, explicate interdependencies between single concepts and thus are well suited for the description of web services and requestor goals. Moreover, Ontologies can be formalized in logics, which enable the use of inference service for exploiting

knowledge about the problem domain during matchmaking and discovery**.**

3.5.3 Service Discovery based on Rich Semantic Description of Services

It does the reasoning over the first order formulae in set based modeling, which leads to the extension of set based modeling. This extension of the set based modeling approach for web service to rich semantic descriptions, which capture the actual relationship between inputs and outputs/effects of web service execution as well and gave the formalization in first-order languages. Instead of considering the state of the world, here consider the service as input, output, precondition, assumptions, Post conditions and effects of services.

4 Survey on Discovery

4.1 Centralised and Decentralised registries

Discovery of Web services is of an immense interest and is a fundamental area of research in ubiquitous computing. Many researchers have focused on discovering Web services through a centralized UDDI registry [5-7]. Although centralized registries can provide effective methods for the discovery of Webservices, they suffer from problems associated with having centralized systems such as a single point of failure, and bottlenecks. In addition, other issues relating to the scalability of data replication, providing notifications to all subscribers when performing any system upgrades, and handling versioning of services from the same provider have driven researchers to find other alternatives

Other approaches focused on having multiple public/private registries grouped into registry federations [8,9] such asMETEOR-S for enhancing the discovery process. METEOR-S [9] provides a discovery mechanism for publishing Web services over federated registry sources but, similar to the centralized registry environment, it does not provide any means for advanced search techniques which are essential for locating appropriate business applications. In addition, having a federated registry environment can potentially provide inconsistent policies to be employed which will significantly have an impact on the practicability of conducting inquiries across the federated environment and can at the same time significantly affect the productiveness of discovering Web services in a real-time manner across multiple registries.

4.2 Non Semantic approach

The growing number of web services available within an organization and on the Web raises is a challenging search problem: locating desired web services. In fact, to address this problem, several simple search engines have implementes [1, 2, 3, 4]. These engines provide only simple keyword search on web service descriptions. Keyword based search techniques do not consider the semantic description of services. Thus, they suffer from poor precision and recall.

Text-based method is the most straightforward way to conduct Web service discovery. The most widely used text-based is the keyword matching built in the UDDI public registry. The UDDI API allows developers to specify keywords of particular interests and it then returns a list of Web services whose service description contains those keywords..Beyond the literal keyword matching, research in XML schema matching[11] has applied various string comparison algorithms (e.g. *prefix*, *suffix*, *edit distance*) to match those interchangeable keywords but with slightly different spellings. Although keyword matching methods (i.e. broad, phrase, exact, and negative) may partially support the discovery of Webservices, they do not provide clients with efficient ways for articulating proper service queries (i.e. consider input/output values of service operations).

4.3 Semantic based discovery

A comprehensive description of the SWS is given in [12]. The fundamental idea underlying current SWS community is that in order to achieve machine-to-machine integration, a markup language (e.g. annotation) must be descriptive enough that a computer can automatically determine its meaning. Following this principle, many semantic annotation markup languages for Web services have come into existence and use such as OWL-S [13], (formerly known as DAML-S [14]), and WSDL-S [15] that have gained great momentum in recent years. The main goal of both OWL-S and WSDL-S is to establish a framework within which service descriptions are made and shared.

4.3.1 WSDL-based and Ontology-based.

The rationale is that WSDL is the defacto standard for representing a Web service's functional capability and technical specifications "on the wire" [16]. It is then natural to discern service discovery methods that centre upon WSDL with those do not. It should be noted that these two categories are not absolutely orthogonal with each other. For example, in the ontology-based method WSDL-S [17], annotation have been made to reference to a domain ontology through the standard WSDL extension mechanism. Hence, we define that WSDL based refers to those methods that take regular WSDL files 'as-is' without further augmenting. Ontology based methods[18], on the other hand, aim to provide a 'semantically enriched' version of WSDL files in order to automate complicated tasks such as service composition.

4.3.2 WSDL based approaches

In this section, we succinctly survey WSDL-based approaches In [19] VSM is used to build a Web service search engine. [20] has attempted to leverage LSA, a variant of VSM, to facilitate web services discovery. However, both [19] and [20] rely on existing UDDI public registries. In [22], a WSDL file is treated as a structural tree that can be compared based on the structures of the operations' input/output messages, which in turn, is based on the comparison of the data types delivered contained in these messages. Likewise, the interface similarity defined in [23] is computed by identifying the pair-wise correspondence of their operations that maximizes the sum

total of the matching scores of the individual pairs. The author in [21] calculated the similarity of complex WSDL concepts given similarity scores for their sub-elements. Using the maximum-weighted bipartite matching [24] algorithm from the graph theory, the author defined a number of coefficiencies to determine the ultimate structural similarity score between two parts in a matching pair. Most of these WSDL structural matching methods are inspired from the signature matching [25], a software component retrieval method from software engineering research. Although a standard WSDL does not provide semantic information, identifiers of messages and operations do contain information that can potentially be used to infer the semantics. when comparing two operations in [22], WordNet is used for deriving the synonyms for the semantic similarity calculation. The lexical similarity defined in [23] and [26] is also based on the concept distance computed from the WordNet sense hierarchy. Interestingly, research in [21] indicates that using WordNet may bring many false correlations due to its excessive generality.

4.4 Clustering in service discovery

More recently, clustering approaches are used for discovering Web services [27, 28, and 29]. Dong [29] puts forward a clustering approach to search Web services where the search consisted of two main stages. A service user first types keywords into a service search engine, looking for the corresponding services. Then, based on the initial Web services returned, the approach extracts semantic concepts from the natural language descriptions provided in the Web services. In particular, with the help of the co-occurrence of the terms appearing in the inputs and outputs, in the names of the operations and in the descriptions of Web services, the similarity search approach employs the agglomerative clustering algorithm for clustering these terms to the meaningful concepts. Through combination of the original keywords and the concepts extracted from the descriptions in the services, the similarity of two Web services can be compared at the concept level so that the proposed approach improves the precision and recall.

Arbramowicz [28] proposes an architecture for Web services filtering and clustering. The service filtering is based on the profiles representing users and application information, which are further described through Web Ontology Language for Services (OWL-S). In order to improve the effectiveness of the filtering process, a clustering analysis is applied to the filtering process by comparing services with related the clusters. The objectives of the proposed matchmaking process are to save execution time, and to improve the refinement of the stored data. Another similar approach [27] concentrates on Web service discovery with OWL-S and clustering technology, which consists of three main steps. The OWL-S is first combined with WSDL to represent service semantics before a clustering algorithm is used to group the collections of heterogeneous services together. Finally, a user query is matched against the clusters, in order to return the suitable services.

Another approach [30] focuses on service discovery based on a directory where Web services are clustered into the predefined hierarchical business categories. In this situation, the performance of reasonable service discovery relies on both service providers and service requesters having prior knowledge on the service organization schemes.

The approach of CPLSA [31] has some similarities to the older approaches [27, 28, and 29] in that keywords are used to first retrieve Web services, and extract semantic concepts from the natural language descriptions in the Web services. This work differs from other works in several ways. Firstly, we eliminate irrelevant service via exploiting a clustering algorithm to diminish the size of services returned; this approach shows some potential applications like over mobile uses. Secondly, based on the characteristics of Web services with a very limited amount of information, we regard the extraction of semantic concepts from service description as a problem of dealing with missing data. Therefore, this work utilizes Probabilistic Latent Semantic Analysis (PLSA) a machine learning method, to capture the semantic concept hidden behind the words in a query and the advertisements in services.

Another recent approach [32] is discovering homogeneous service communities through web service clustering. It gathers the features for a WSDL file is not as simple as collecting description documents when assuming no central UDDI registries. Another closely related area is the conventional document or web page clustering. They both involve the discovery of naturally-occurring groups of related documents. Web service files do not usually contain sufficiently large number of words for use as index terms or features. Moreover, the small numbers of words present in the web service files are erratic and unreliable. Hence, conventional, detailed linguistic analysis, and statistical techniques using local corpora cannot be applied directly for web service files clustering. The use of link analysis between WSDL files to discover related web services has also been studied. In our experiments, we employed Google API's search options for discovering web page referral or citation. However, it is discovered that most of the WSDL files do not have related web pages that provide hyperlinks to them. For the few that have hyperlinks referring to them, such WSDL files are typically educational examples for teaching how to program in a service-oriented paradigm.

In short, the individual existing techniques borrowed from related research areas such as information retrieval are inadequate for the purpose of discovering functionally-related web service clusters. While there is a small number of existing approaches dedicated to the discovery of web services as mentioned above, most of them remain hypothetical in nature, and have yet to be implemented and tested with real-world datasets.

4.5 Similar service discovery

In a approach [32] in discovering similar service operation where for a given service the service similar to services are discovered from the repositories by matching the input, output and operation of the service The approach proposed schema matching techniques for the input, output datatypes schemas.

In another approach [33] Xion Dong et.al proposed a search engine woogle which searches for a similar service operation given an operation as input. In their approach the matching problem is consider similar to the text document matching problem, database schema matching problem and software component matching problem in their approach they match input, output as concepts by using A-prior algorithm for association rules and agglomeration algorithm for clustering similar concepts. In [34] a recent approach the similar approach as [33] was used with improvement in the clustering techniques using domain taxonomy. They also proposed a "service pool" to store similar service which may be single service or composite service using Graph based techniques.

5 Quality of Service attributes

Based on the behavior the quality of a service can be classified as given in [10]

Computational behavior: These QoS includes Execution Attributes (Latency, Accuracy, Throughput, Reliability, Extendibility, Capacity, and Exception Handing), Security (such as Encryption, Authentication, and Authorization), Secrecy, Availability etc.

Business behavior: These QoS mainly refers to Execution Cost, Reliability of the provided service, Punishment on condition that SLA could not be sufficed with.

Metadata restriction: These QoS includes Constraints that have to be followed regarding UDDI /WSDL /SOAP/WSLA parameters such as location, specific companies , schema .

5.1 Execution attributes

1. *Response Time*: time elapsed from the submission of a request to the time the response is received.

2. *Accessibility*: represents the degree that a service is able to serve a request.

3. *Compliance*: represents the extent to which a WSDL document follows WSDL specification

4. *Successability*: represents the number of request messages that have been responded.

5. *Availability*: represents the percentage of time that a service is operating.

5.2 Security

It is related to the ability of a given Web service to provide suitable security mechanisms by considering the following three parameters.

1. *Encryption*: the ability of a Web service to support the encryption of messages.

2. *Authentication*: the capacity of a Web service to offer suitable mechanisms dealing with the identification of the invoking party and allow operation invocation.

3. *Access control*: whether the Web service provides access control facilities to restrict the invocation of operation and the access to information to authorized parties.

5.3 Business attributes

Like QoS properties, they are relevant for differentiating Web services having the same functional characteristics

1. *Cost:* represents money that a consumer of a Web service must pay in order to use the Web service.

2. *Reputation*: measures the reputation of Web services based on user feedback

3. *Organization arrangement*: includes preferences and history (ongoing partnerships)

4. *Payment method*: represents the payment methods accepted by a Web service, i.e. transfer bank, Visa card etc.

5. *Monitoring*: required for a number of purposes, including performance tuning, status checking, debugging and troubleshooting.

6 Conclusion

Service Computing has gained momentum during the years and its proliferation has bridged the gap between the business and IT industries to make them work close to each other. The advent made has forced the challenges faced as given in section 1 to be addressed with more significance and enhanced its research scope. Due to the exponential growth of service and the relation between demand and supply made the aspects like service discovery more prominent. The work that has been undertaken in that has directions reveals the necessity of it. So necessity part has become vital. In sufficiency point of view does the promising approaches that are discussed in section 4 satisfies. The two measure that are commonly used to measure the efficiency of the approaches are precision and recall. Keyword based approaches is not unto the expectation more prominence is on semantic based approaches. The efficiency of the semantic based approaches has significantly increased as for the precision and recall are concerned, still the lack in optimality. More work has to done with the advent in technology to make the discovery process more and more optimal. Our survey above is to motivate the researcher to work on this area has this has become a promising area of research.

References

[1]. Binding point. http://www.bindingpoint.com/.

[2]. Grand central. http://www.grandcentral.com/directory/.

[3]. Salcentral. http://www.salcentral.com/.

[4]. Web service list. http://www.webservicelist.com/.

[5]. U. Thaden, W. Siberski, and W. Nejdl, "A semantic web Based Peer-to-Peer Service Registry Network," TechnicalReport, Learning Lab Lower Saxony, 2003.

[6]. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, "Semantic matching of web services capabilities," ISWC, pp. 333-347, 2002.

[7]. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara "Importing the semantic web in UDDI," InternationalWorkshop on Web Services, E-Business, and the Semantic Web, pp. 225-236, 2002.

[8]. K. Sivashanmugam, K. Verma, and A. Sheth, "Discovery of web services in a federated registry environment," ICWS, pp. 270-278, 2004.

[9]. C. Zhou, L. Chia, B. Silverajan, and B. Lee, "UX- an architecture providing QoS-aware and federated support for UDDI," ICWS, pp. 171-176, 2003.

[10]. Quality Model for Web Services v2.0 Committee Draft, September 2005

[11]. H. H. Do and E. Rahm, "COMA - A system forflexible combination of schema matching approaches,"presented at 28th VLDB Conference, Hong Kong, China,2002.

[12]. Processes and Applications," in Semantic Web and Beyond:Computing for Human Experience, R. Jain and A. Sheth, Eds.: Springer, 2006.

[13]. D. Martin, "OWL-S: Semantic Markup for Web Services," in Releases of DAML-S / OWL-S, 2004.

[14]. M. Paolucci, K. Sycara, T. Nishimura, and N.Srinivasan, "Using DAML-S for P2P Discovery," Proceedings of First International Conference on WebServices, ICWS, 2003.

[15]. IBM and UGA, "Web Service Semantics," IBM and UGA 2005.

[16]. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language.(WSDL) 1.1," 2001.

[17]. " Web Service Semantics, "http://www.w3.org/ Submission /WSDL-S/.

[18]. J. Cardoso and A. Sheth, "Semantic Web Services,Processes and Applications," in Semantic Web and Beyond:Computing for Human Experience, R. Jain and A. Sheth,Eds.: Springer, 2006.

[19]. C. Platzer and S. Dustdar, "A vector space search engine for Web services," presented at Third IEEE European Conference on Web Services, Sweden, 2005.

[20]. N. Kokash, W.-J. v. d. Heuvel, and D. A.,Vincenzo, "Leveraging Web Services Discovery with Customizable Hybrid Matching," Technical Report,University of Trento, vol. DIT-06-042, 2006.

[21]. N. Kokash, "A Comparison of Web Service Interface Similarity Measures," University of Trento 2006.

[22]. E. Stroulia and Y. Wang, "Structural and Semantic Matching for Accessing Web Service Similarity," International Journal of Cooperation Information Systems, vol. 14, pp. 407 - 437, 2005.

[23]. J. Wu and Z. Wu, "Similarity-based Web Service Matchmaking," presented at IEEE International Conference on Service Computing, 2005.

[24]. L. Lovasz and M. D. Plummer, Matching Theory.North-Holland: Elsevier Science Publisher, 1986.

[25]. Zaremski and J. Wing, "Signature Matching of Software Components," ACM Transactions on SoftwareEngineering and Methodology, pp. 333-369, 1997.

[26]. Z. Zhuang, P. Mitra, and A. Jaiswal, "Corpus-based web services matchmaking," presented at Workshop on Exploring Planning and Scheduling for Web services, Grid, and Autonomic Computing, 2005.

[27]. Richi Nayak Bryan Lee, "Web Service Discovery with additional Semantics and Clustering" 2007 IEEE/WIC/ACM International Conference on Web Intelligence, 2007

[28]. W. Abramowicz, K. Haniewicz, M. Kaczmarek and D. Zyskowski "Architecture for Web services filtering and clustering. In Internet and Web Applications and Services", (ICIW '07), 2007.

[29]. X. Dong, A. Halevy, J. Madhavan, .E. Nemes and J. Zhang. Similarity Search for Web services. In Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

[30]. Constantinescu, W. Binder and B. Faltings. Flexible and efficient matchmaking and ranking in service directories. In Proceedings of the IEEE International Conference on Web Services (ICWS'05), 2005

[31]. Jiangang Ma, Yanchun Zhang, Jing He "Efficiently Finding Web Services Using a Clustering Semantic Approach" CSSSIA 2008, April 22, ACM ISBN 978-1-60558-107-1/08/04, 2008

[32]. Wei Liu and Wilson Wong "Discovering Homogenous Service Communities through Web Service Clustering" SOCASE 2008, LNCS 5006, pp. 69–82, 2008.Springer-Verlag Berlin Heidelberg 2008

[33]. "Discovering homogenous web service community in the user centric web environment", IEEE transition of service computing VOL.2 No@ April June 2009

[34]. " Similarity search fo web service" Porceeding of the 3oth VLDB conference, Totronto Canda 2004

Author Biographies

Augustine.K.V received his Masters degree in Mathematics in the year 1996 Later on in 2005 completed his Masters in Computer Application. He is doing his Maters of Technology in Computer Science and Engineering. He has also completed Post Graduate diploma in computer science, Statistical and Research Methodology. Currently he is working for the Government of Puducherrry as statistician His research interest are computational statistics, Service Computing, data mining and Discrete Mathematics.