

Semantic Driven Web Service Discovery (SDWSD) System

Chinmohan Nayak
Assistant Systems Engineer
Tata Consultancy Services
Bangalore-560066

Sambedita Nayak
IT Consultant
Fujitsu
Bangalore-560068

Sujit Ghosal
Senior Security Researcher
McAfee
Bangalore-560071

ABSTRACT

Service discovery is one of most challenging issues now-a-days because of the increasing number of services but a very few techniques are available for their efficient discovery. It is worthless if there are abundant services available but we cannot avail our desired services only because of the irrelevant discovery system. The semantic based service discovery is proposed to enhance the discovery of these elusive services. Web service standards, in their present format supports only keyword based search and many services which can fulfill the user's requirements are not retrieved. Basic requirement for efficient service discovery is to extract the contextual information provided in the service description. In such situation, the optimal solution is obtained by introducing semantics in the present Web Service Description Language (WSDL).

General Terms

Web Service Discovery, Sensor Network, Algorithms.

Keywords

Service Discovery, Sensor Network, WSDL, Web Services, SDWSD System.

1. INTRODUCTION

Web services are modular, self-describing, self-contained applications that are accessible over Internet. This is an emerging trend which has been identified as the technology for business process execution and application integration. There are increasing numbers of both publicly available Sensor Services (external) and Sensor services only exposed internally within an organization. It is becoming a kind of mainstream middleware technology of interoperation and integration between heterogeneous applications and resource sharing in Internet environment. Therefore it becomes a challenge for the external users or systems to discover and invoke the sensor derived data.

Various standards used by the Sensor web services like XML, SOAP, WSDL, UDDI helps software to discover and access the Sensor services available in the world wide web. But all this still need some kind of human intervention in selecting the most appropriate Sensor Web service. Currently, the industry standards for Web services are Web Services Description Language (WSDL) and Universal Description Discovery and Integration (UDDI) specifications. Web services are described using WSDL definitions and advertised in UDDI registries. The current discovery mechanism supported by UDDI is not powerful enough for automated discovery. The main inhibitor is the lack of semantics in the discovery process and the fact that UDDI does not use information in the service descriptions during discovery. This

makes UDDI less effective, even though it provides an interface for keyword and taxonomy based searching.

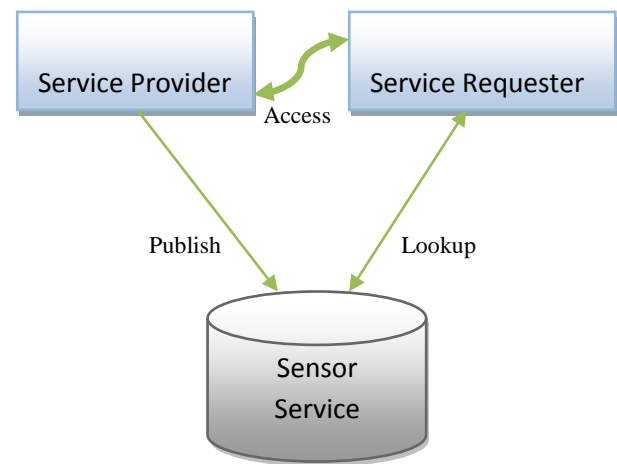


Fig 1: Overall Architecture Of Service Storage And Service Lookup.

The key to efficient discovery of Web services is having semantics in the description itself and then using semantic match making algorithms to find the required services.

In this paper, we develop a framework for semantically discovering Sensor Web services where we incorporate the semantics and integrate it with UDDI registries. Our aim is the discovery of Web services on a semantic comparison between a client query and available Sensor Web services. This architecture supports both service discovery and service publishing. The discovery contribution of this paper lies in three folds. First is the direct discovery by exact matching. If this step fails, the second is, the requested query for the service is matched with the semantic descriptions. Thirdly, we use a dictionary based approach to capture real world knowledge which is integrated in the algorithm based on the second fold.

2. LITERATURE SURVEY

The authors in [1] have proposed an algorithm for an efficient search but it is limited to only keywords and also they have not implemented the algorithm.

The authors in [2] proposed several important components on the sharing of sensor network data using the Web services and SWE standards. The Sensor Web proposed a registry for sensor network discovery and registration called Sensor Registry Service. The Sensor Registry Service is too abstract in the service oriented sensor web because too little attention

has been given to the detail functionality of the sensor registry service.

In [3] a mechanism to discover sensor web registry services based on functional requirements is proposed. However nonfunctional requirements of the services are not considered at all.

In [4] a unique SOA approach is presented to design a sensor web registry that can be hosted on a special server called Sensor Name Server that cooperates and collaborates in searching a sensor network. However the author has given more emphasis on design of sensor web registry rather than sensor discovery process. This paper explains that the client is given a web based GUI interface to search a sensor network based on single or combination of parameters given in the registry. But it may not be an efficient approach for the casual users who have no idea about the above type of sensor search interface with parametric choice.

In [5] a sensor network registry is proposed and the query parameters for sensor network discovery are analyzed by 5W1H method. Here the authors have mentioned that the sensor network registry receives the discovery query using XML (XQuery). However XQuery and XPath are the advanced XML based technology which is very difficult for the novice requesters to understand.

In GEOSS [6], a community of researchers emphasized the need of sensor web registry with SWE compliance. It also discusses ad hoc network and moving (nomadic) sensor. But It does not provide any solution for design and discovery of sensor registry service.

In a similar effort, the authors in[7] proposed WOOGLE , a search engine which focus on retrieving WSDN operations. Woogle (which discontinued its service in 2006), collected services from accessible services registries and provided clients with capabilities to perform keyword-based search. However, the main underlying concept behind the method implemented in woogle was based on the assumption that web services belong to the same domain of interest and are equal in terms of their behavior in accomplishing the required functionality.

Other approaches focused on the semantic support for web services as presented in [8], the authors proposed a novel approach to integrate services considering only their availability, the functionalities they provide, and their non-functional QoS properties rather than considering the users direct request.

In[9] the authors proposed a solution for this problem and introduced the Web Service Relevancy Function (WSRF) that is used for measuring the relevancy ranking of a particular Web service based on QoS metrics and client preferences. However one of the challenges in this work is the clients ability to control the discovery process across accessible service registries for finding services of interest, yet semantic matching of services has not been considered.

3. PROPOSED WORK

We propose a genuine and effective system to overcome the limitations faced using traditional UDDI approach. We have also extended the scope of WSDL by adding the non-functional (QoS) parameters and the Semantic information along with the functionality offered by the WSDL. This system dynamically discovers and selects the Web services using a series of proposed algorithms.

The framework gives more importance to the design of the core part i.e., Semantic Driven Web Service Discovery (SDWSD) system which carries out all the operations, beginning from the registration of a service by the service provider to the results obtained by the service requestor while he fires his queries to the system. The proposed architecture of SDWSD model is shown in Fig. 2.

3. 1 Framework of the Proposed Model

The Proposed framework of SDWSD model can be divided into three major parts mentioned below.

3.1.1 GUI Layer

GUI Layer is the interaction layer between the end users and the SDWSD system. In this model we have considered the Service Providers as well as the Service Consumers to be the end users. The Service Providers are provided with a GUI interface to register their web services into the system's database whereas the Service Consumers requests for those services and finally uses those registered web services.

3.1.2 SDWSD System

SDWSD System is the heart of the proposed model, without which the system is no more than a paralyzed entity. This controls the registration as well as the dynamic discovery and selection process. This consists of four main important parts explained below

QoS Validation And Certification- QoS Validation And Certification as the name suggests, it validates the QoS information of the web services and issues a certificate. A copy of the QoS certificate is stored in the QoS database and the other one is sent to the Web service provider.

Service Selector- Service Selector plays a very important role in web service discovery. It receives the request from the Service Consumer and then it applies appropriate algorithms to retrieve all the relevant web services.

Service Rank- Service Rank is used to calculate the rank of those services retrieved by the Service Selector. The ranking formula is given in Proposed Algorithm.

Rank Information- Rank information is a temporary storage location where all the relevant services found out by the Service Selector is stored along with their calculated ranks which will be emptied as soon as a single transaction gets over.

3.1.3 Data Layer

Data layers stores all the information related to services along with some other information for the discovery of best possible web services from the results obtained in the first phase. It consists of four repositories explained below.

Service Database- Service Database includes attributes such as service name, service address and information related to the web services.

Semantic Database- Semantic Database stores the semantic description of the web services so as to make the discovery of a service easier by semantic based searching.

Synonym Database- Synonym Database stores all possible

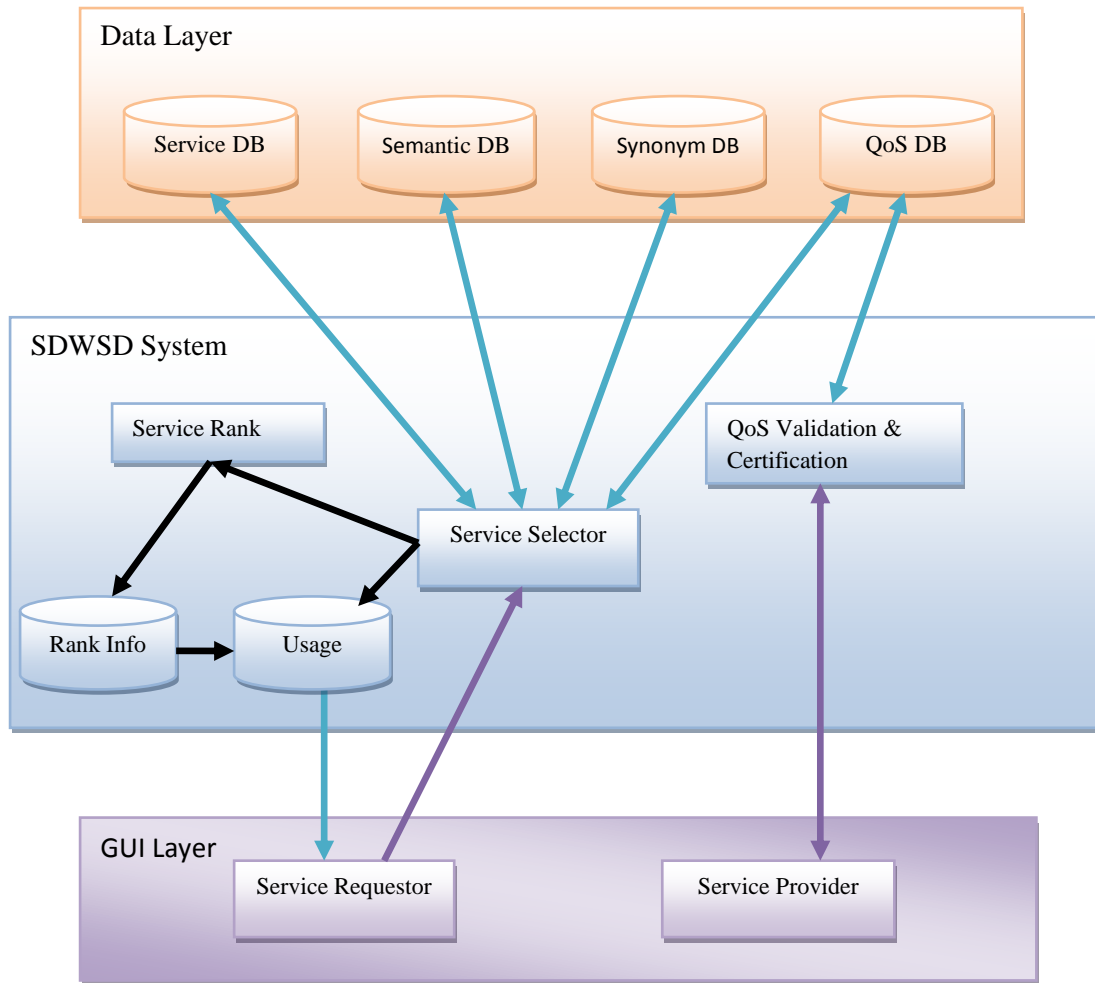


Fig 2: The Proposed architecture of SDWSD System.

words along with their synonyms that is used in our proposed algorithm.

Quality Of Service (QoS) Database- QoS Database stores all of the non-functional parameters of a web service registered.

3.2 Arrow Perception Of SDWSD's Proposed Architecture

There are three kinds of arrows found in the architecture (Fig. 3). The first one is the violet arrow which emerges from the GUI layer signifies the flow of interaction starts from the GUI Layer (by the end users) and the ending point signifies the end of interaction. On the other hand the ocean blue arrow emerges from the middleware i.e., from the SDWSD System. But the arrow represented in black signifies the flow within a layer. Bidirectional arrow represents the flow of data in both directions.

3.3 Proposed Algorithm

Here we have proposed two algorithms

- Keyword Search Algorithm
- Semantic Search Algorithm

Keyword Search Algorithm is a very short algorithm which performs direct check operation to find a Web Service. This

algorithm is implemented only to save the execution time and enhance the performance of SDWSD System. However, the second algorithm i.e., the Semantic Search Algorithm can be applied if the former algorithm fails to search Web Services or if the Service Consumer is not satisfied with the retrieved Web Service result. The Algorithms are given below in a Step-wise manner.

3.3.1 Keyword Search Algorithm

1. Request raised by the Service Consumer
2. System receives the Natural Language Query
3. Matches the query with the Service Name
4. If match found, retrieve all the details of the service and display it to the consumer. Else, Auto invokes the Semantic Search Algorithm.

3.3.2 Semantic Search Algorithm

1. Take user's input.
2. Faster slicing of query by whitespaces

3. Extract only the singular and plural nouns after the slicing and store them in an array (all other words should be excluded).
4. Frame new arrays which will store the possible synonyms of the nouns (extracted in the previous step) in the same order as that of the nouns.
5. Match every words present in the arrays with the semantic description of the services in a word by word fashion.
6. Calculate the Service Rank.

Rank= Number of words matched/ Total number of words

NOTE:- Maximum value of Service Rank is 1.

7. Store the retrieved services with their ranks calculated as the result in Usage (Fig. 2).
8. In the Usage we can find the repetition of the same services with different Ranks. Extract the distinct services obtaining top ranks.
9. Modify the Usage DB by arranging the services according to their Service Rank in descending order.
10. Display the results to the Service Requestor.

3.4 QoS Parameters Specification

An efficient Sensor Web Registry should contain the following functional and non-functional parameters for executing the sensor web service discovery query efficiently.

The Functional Parameters are namely Sensor Service Name, Sensor Service Address, and Sensor Service Description. The Non-functional Parameters are QoS Data like Response Time (RT), Throughput (TP), Availability (AV), and Cost of Service(C).

Table 1. QoS Parameters Specification

QoS Parameters	Type	Unit
Response Time	Integer	Milleseconds
Throughput	Float	Req/min
Availability	Integer	Degree
Accessibility	Integer	Degree
Interoperability Analysis	Integer	Degree
Cost	Float	USD \$

4. EXPERIMENTAL EVALUATION

The Natural Language Interface system for discovering Sensor Web Service is programmed using PHP technology, a distributed, loosely-coupled, Platform-independent system, which can run on multiple operating systems, such as Linux, Windows, or Solaris. The Sensor Service storage system is designed using MySQL database package.

Let us consider a query “**New Features of ipad**” fired by a Service Consumer. This particular query can have three different combinations of queries (say). Combinations Preparing Process (using Synonyms):-

Query-1:- New Features of ipad

Query 2:- Latest Features of ipad

Query-3:- Latest Specification of ipad

Let us consider three services are present in the Service Database. The nouns extracted from their Semantic Descriptions of the services are as follows (say):-

Service-1:- latest, specification, ipad

Service-2:- new, ipad

Service-3:- ipad

While the proposed algorithm is executed for the Query 1, we will get service 1 & 2 & 3 because the individual nouns present in Query 1 will match the nouns in the Semantic Description of the services. For Service 1, only one noun will match i.e. “ipad”, two nouns will match for Service 2 and one noun for Service 3.

Table 2: Service Ranks for the Experiment

Query No.	Service Name	Service Rank
Query 1	Service 1	1/3=0.33
Query 2	Service 2	2/3=0.66
Query 3	Service 3	1/3=0.33
Query 4	Service 1	2/3=0.66
Query 5	Service 2	1/3=0.33
Query 6	Service 3	1/3=0.33
Query 7	Service 1	3/3=1
Query 8	Service 2	1/3=0.33
Query 9	Service 3	1/3=0.33

After the Match Process is over, we will be eligible to calculate Service Ranks of the results.

As per the above explanation, when Query 1 will be fired, only one word out of three will match the Semantic

Description of Service 1 i.e. “ipad”. So the Service Rank of the result will be

$$\text{Service Rank} = 1/3 = 0.33$$

Similarly all the possible results is given in the Table 2. From Table 2, we must extract the maximum value of each service and arrange the services in the descending order of their Service Ranks, displayed in Table 3. For Service 1 the maximum value is 1 and for services 2 and 3 the maximum values are 0.66 and 0.33 respectively.

Here two exceptional cases may arise. These cases with their solutions are given below in Table 4:-

Case I:- If the Service Rank of two or more services is equal then calculate the QoS Parameters Average (excluding Service Cost) by the given formula and store them according to the decreasing order of QoS Parameter Average.

Table 3: Services arranged in the descending order of their Service Rank

Sl.	Service Name	Service Rank	RT	TP	AV	AC	IA	Cost
1	Service1	1	560	1.2	70	91	99	3.8
2	Service2	0.66	900	2.7	88	89	70	9.2
3	Service3	0.33	980	9.9	90	81	90	7

Case II:- If the QoS Parameter Average is still found to be equal then arrange them in an ascending order of their Service Cost because consumers prefers the service with low cost among the services which have almost same features.

Table 4: Example of an exceptional case

Sl.	Service Name	Service Rank	RT	TP	AV	AC	IA	Cost
1	Ipad Store	1	560	1.2	70	91	65	3.8
2	Apple Ipad	0.66	900	2.7	88	89	70	7.2
3	Mobile Corner	0.66	900	2.7	70	88	89	6

In Table 4, as the Service Rank of Apple Ipad is same as Mobile Corner, we have to calculate their averages of QoS Parameters. After calculation, QoS Parameter’s average of Apple Ipad is again same as that of the average of Mobile Corner. Now we should consider the Cost factor of Mobile Corner and Apple Ipad. As Mobile Corner is having all the features similar to that of Apple Ipad with less Cost, it is the desired service after Ipad Store. Also all the consumers will prefer Mobile Corner rather than Apple Ipad because its cheaper. So it should be displayed second and Apple Ipad should be displayed third. After rearranging the final result is shown in Table 5.

Table 5: Final arrangement of the exceptional case

Sl.	Service Name	Service Rank	RT	TP	AV	AC	IA	Cost
1	Ipad Store	1	560	1.2	70	91	65	3.8
2	Mobile Corner	0.66	900	2.7	70	88	89	6
3	Apple Ipad	0.66	900	2.7	88	89	70	7.2

5. CONCLUSION

This paper presents a semantically enhanced service repository for service oriented application system development. The two parts of the repository are service registry and semantic service repository. Service registry provides the basic service management function and act as the navigation point of the existing services. The service model used in this registry is thoroughly analyzed. The semantic service repository encapsulate the service with semantic information. Two algorithms facilitating service discovery are introduced in this paper. Our approach extends semantic-based discovery capability to mainly used enterprise application components such as Web services, EJBs, and POJOs. This approach can greatly improve the user’s awareness of various types of available services and therefore enhance the reusability and integration of those components. This paper demonstrates the algorithms very eloquently with all possible exceptional cases. Calculations and tabulations of the cases are also done for a clear transmission of idea.

This model can be extended in areas of application security and advance web technology concepts.

6. REFERENCES

- [1] Rajender Nath, Harish Kumar(2009), “Building Software Reuse Library with Efficient Keyword based Search Mechanism”– International Journal of Computing Science and Communication Technologies, VOL. 2, NO. 1, July 2009. (ISSN 0974-3375).
- [2] Chu X., Buyya R., (2007), "Service Oriented Sensor Web, in Sensor Networks and Configuration: Fundamentals, Standards, Platforms and Applications", N. P. Mahalik, (ed). Springer- Verlag, 2007, pp. 51-74, 978-3-40-37364-3
- [3] Manoranjan Parhi, B.M. Acharya, B. Puthal “An Effective Mechanism to Discover Sensor Web Registry Services for Wireless Sensor Network under x-SOA Approach”, Proceedings of 2nd IEEE International Conference on Trendz in Information Science & Computing[TISC-2010], Chennai, India, December 17th -19th, 2010.
- [4] Pandey, K.K. Patel, S.V. (2009) “A Design of Sensor Web Registry for Wireless Sensor Networks with SOA Approach”, Proceedings of the 1st IEEE international conference on Computational Intelligence, Communication Systems and Networks(CICSYN2009), Indore, India, July 23-25, pp.247-252
- [5] Jeongkyu Park; Jiung Han; Kibong Kang; Keung Hae Lee.,(2007), "The Registry for Sensor Network Discovery", Proceedings of the 12th IEEE International

Conference on Engineering Complex Computer Systems (ICECCS2007), Auckland, New Zealand, July 11-14, pp.129-137.

- [6] Ingo Simonis " Meraka Institute, CSIR, South Africa , Johannes Echterho! " IfGI, University of Muenster, Germany June 2008 , GEOSS Sensor Web Workshop Report.
- [7] X. Dong, A. Halevy, J.Madhavan, E. Nemes, and J. Zhang, "similarity search for web services," in proceedings of the Thirtieth international conference on

Very large data bases,ser. VLDB '04 VLDB Endowment,2004,pp. 372-383.

- [8] N. Ibrahim*, F. Le Mouël** and S. Frénot, "Mysim: a spontaneous service integration middleware for pervasive environments," in proceedings of the 2009 international conference on pervasive services,ser. ICPS '09,New York,NY,USA: ACM,2009,pp. 1-10.
- [9] E. Al-Masri and Q.H. Mahmoud, "Discovering the best Web service," in proceedings of the 16th international conference on World Wide Web,ser. WWW '07. New York,NY, USA : ACM, 2007, pp. 1257-1258