Approach and Impact of a Protocol for Selection of Service in Web Service Platform

Rajni Mohana
Dept. of Computer Science and Engineering
JUIT Waknaghat, Solan, H.P, India
rajni.mohana@juit.ac.in,

Deepak Dahiya
Dept. of Computer Science and Engineering
JUIT Waknaghat, Solan, H.P, India
deepak.dahiya@juit.ac.in

Abstract: Business agility is an important challenge while designing an enterprise application. Service oriented architecture is used to combine many outsourced web services to provide value added services to the users with agility. A service registry is maintained to keep track of various web service published by the service providers. The key challenge for the service requester is to pick the best web service among the various functionally equivalent web services in the service registry. This paper describes and analyses various service selection protocols given by the researchers. The approaches are classified into semantic and non-semantic approach. It also proposes a novel technique to look for the best web service based on QoS like reliability, throughput etc. The solution to the problem of selecting the best web service according to the requirements is designed as a fuzzy expert system. This rule based approach of Service registration and lookup is adaptive and responds dynamically to quality of service changes in the web services.

Categories and Subject Descriptors
D.3.3 [**Programming Languages**]: MATLAB

General Terms
Algorithms, Performance, Design, Reliability, Experimentation, Standardization.

Keyword
Web Services, SOA, Ontology, Fuzzy Clustering, PSO, Expert System.

1. INTRODUCTION
An enterprise application suffers from changing requirements. To provide the service to the consumer with agility the concept of *Service Oriented Architecture* (SOA) is used. The most common technology for implementing SOA-based systems is the SOAP-based *Web Services* [1]. Web services are the foundation of SOA to construct a variety of new strategic solutions. Web services appear as a black box behind the firewall and make the communication of data among organizations feasible, without intimate knowledge of each other's IT systems. It provides a standard means of interoperating between different software applications, running on a variety of platforms and frameworks. It facilitates corporate agility, to market for new products and services, reduce IT costs and improves operational efficiency [2].

The basic web service architecture is given in figure 1. For example, a movie booking web service composition be built by aggregating a movie booking web service, a car rental web service, an eating point web service, a payment web service, and an itinerary planning web service.

Participants in a Web services model are categorized into three types:

- Service Requester: Finding a web service of interest is an important task for a service requester.
- Service Provider: Publishing their web services in registries is the task of service provider.
- Service Registry: It maintains the details of all the web services which are published by the service requester. It also enables the service to lookup for the best service according to its requirements among large number of functionally-equivalent services.

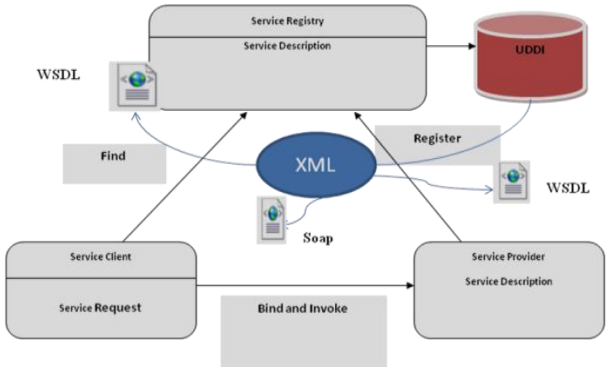

Figure 1: Web service Platform [3]

To provide interoperatability the basic web services platform consists of specifications (SOAP, WSDL and UDDI). In order to generate composite services, it is required to discover and select suitable web services for service requests. Some standards like Web Service Definition Language (WSDL) can be used to describe functional aspects of a Web service in form of a service description which is advertised in some Universal Description, Discovery and Integration (UDDI) registries [3].

The following organizations are working on standardizing web services specifications, guidelines, and tools to provide interoperatability are as follows:

- The World Wide Web Consortium (W3C)
- Organization for Advancement of Structured Information Standards (OASIS)
- Web Services Interoperability Organization (WS-I)
- Java Community Process (JCP)

There are many challenges in adoption of SOA like including adequate staff training, managing short term costs, maintaining the level of discipline required to ensure the services that are developed and reusable. Though having many challenges for its adoption it still is preferred by the developers as it provides benefits like:

- Reuse
- Efficiency
- Loose technology coupling
- Division of responsibility

The key challenge for the service requester is to pick the best web service among the various functionally equivalent web services in the service registry. We have studied and analyzed various service selection protocols given by the researchers. This paper describes the approaches, and classifies them into semantic and non-semantic approach. It also proposes a novel technique to look for the best web service based on

DOI: 10.1145/2088883.2088896 http://doi.acm.org/10.1145/2088883.2088896

QoS (Quality of Service) like reliability, throughput etc. The solution to the problem of selecting the best web service according to the requirements is designed as a fuzzy expert system. This rule based approach of Service registration and lookup is adaptive and responds dynamically to quality of service changes in the web services.

The remainder of this paper is organized as follows. Section 2 gives some background on knowledge service registry. Section 3 discusses various Service Selection Protocols. Section 4 presents the implementation of proposed approach for the service discovery and the dataset used. Section 5 describes the results and observations, Section 6 shows the conclusion, future scope is mentioned in section 7 and references are listed out in section 8.

2. SERVICE REGISTRY

Registries are expected to be the key technology for overcoming barriers in the adoption of global e-business. Its success is closely tied to an expected return from investing in technology for Internet e-commerce and the anticipated global electronic market. Many vendors (for example IBM, Oracle, Microsoft) [4] have started supporting and designing internet based registries. Various internet based registries [5] [6] are designed to enable businesses to discover other businesses, and the services they provide.

The web service platform provides a centralized discovery approach. Service registry in web service platform maintains the repository of the services which a service provider wants to publish as shown in figure 1. When a Service requester looks for a service he has to use a service discovery protocol, to choose the best web service among the large number of functionally equivalent web services published in the service registry.

3. SERVICE SELECTION PROTOCOLS

When a service request is issued, the requirement specifications are looked for in the UDDI as shown in figure 1 to find services that can provide expected functionality. Service discovery protocols minimizes the administrative overhead and increases usability by locating the best web service among the large number of functionally equivalent web services, published in UDDI. Research work on various centralized discovery approach as proposed by researchers [7-14] was analyzed during the literature survey.

The research work in the area of service selection protocols is directed in two different areas namely, semantic and non-semantic approaches both of which are focused around discovering the web service based on QoS in brief.

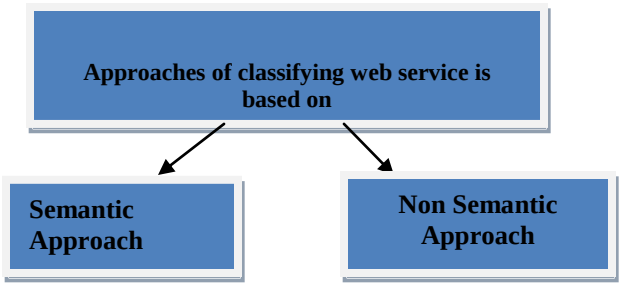

Figure 2: classification of approaches of service discovery

- Semantic approach: It performs web service searching based on the domain knowledge of the attributes. It requires prior systematic structuring of the contents such that requests are understood and responded based on their meaning.

- Non-semantic approach: It performs web service searching based on the syntax. It requires the data to be structured as per required format and the requests will be responded accordingly.

3.1 Semantic Approach

Segev, A. etal [7] analyzed different methods for automatically identifying possible service composition. They explored two sources for service analysis which are commonly used in service repositories i.e.

- WSDL description files
- Free textual descriptors,

They also investigated two methods for Web service classification for each type of descriptor:

- Term Frequency/ Inverse Document Frequency (TF/IDF)
- Context based analysis, and a baseline method.

Their work showed that the Web-based context extraction method by analyzing both the WSDL description and the textual description yields better results than the TF/IDF method and string matching.

In addition, the result proves the advantage of integrating the analysis of both the WSDL context descriptor and the service textual descriptor [7].

Kritikos k etal [8] have proposed QoS-based *WS Description and Discovery (WSDD)* to enable automatic discovery and composition of independently developed and deployed (Web) services. They have considered semantically rich QoS based in automatic discovery of Web Services (WSs) can be achieved by incorporating semantics into WS matchmaking and selection (i.e., discovery) process. Its main contribution is the analysis of the requirements for a semantically rich QoS-based *WSDM* and an accurate, effective QoS-based *WS Discovery (WSDI)* process. They also provided a road map to extend *WS * technologies* for realizing semantic, functional, and QoS-based WSDl.

This approach [8] comprises of

a. Description of the QoS aspect of WSs (i.e., QoS-based WS description)
b. Filtering of WS functional results based on user constraints on their QoS descriptions (i.e., QoS-based WS matchmaking) The algorithms were implemented using Mixed Integer Programming MIP, CP, and explanation-based CP
c. Sorting the results based on user-provided weights on QoS attributes/metrics (i.e., WS selection).

Pilioura etal[9] have proposed *PYRAMID-S* which uses a hybrid peer-to-peer topology to organize web service registries based on domains. It has a scalable framework for unified publication and discovery of semantically enhanced services over heterogeneous registries. In such a topology, each registry retains its autonomy, meaning that it can use the publication and discovery mechanisms as well as the ontology of its choice.

The features of *PYRAMID-S* are:

— unified Web service publication and discovery:
— over heterogeneous registries, thus alleviating the users from the burden of handling the diversion between different technologies;
— based on syntactic, semantic, and Quality of Service (QoS) information, improving in this way the precision and the recall;
— preservation of the autonomy of Web service registries by allowing the accommodation of different publication and discovery mechanisms; and
— Use of a scalable infrastructure which organizes registries based on domains [9].

Yin Baocai etal[10] proposed an approach to qualify a web service using its semantic description. They suggested a general ontology to describe the non-functional qualifies the web service using the semantic

DOI: 10.1145/2088883.2088896 http://doi.acm.org/10.1145/2088883.2088896

description of the web services using their non- functional specifications. This ontology can solve the interoperability of QoS description. The framework supports the automatic discovery of web services and it can improve the efficiency for users to find the best services. It uses *OWL-S* as the description language to represent QoS ontology.

3.2 Non-Semantic Approach

Zhang etal[11] has designed a broker-based architecture called *QBroker*, to provide end-to-end QoS management for distributed services. Functionalities of QBroker include service discovery, planning, selection, and adaptation. The efficiency of QBroker is dominated by the running time of the service selection algorithm. They also designed efficient algorithms for quality-driven Web service compositions. Their model defines multiple QoS criteria and takes global constraints into account. It ensures that the selected services always meet the QoS requirements. They have also proposed heuristic algorithms to find near-optimal solutions in polynomial time which is more suitable for making runtime decisions. They have mapped the service selection to a 0-1 multidimensional multi choice knapsack problem (MMKP) [11].

Vuong Xuan Tran etal [12] have suggested an approach where the web services are selected based on QoS. The web services are ranked based on their quality attributes, but the issue raised by the researchers was that it's difficult to provide a precise value to the quality attributes of the web service. Thus they suggested using fuzzy logic to support using imprecise QoS constraints. The benefit of this approach is that a user does not need to mention crisp values of QoS properties. Instead the user can use fuzzy linguistic concepts to express their expectation of service quality.

However a user has to define at most as many rules as there are degrees of acceptance that s/he wants to differentiate. When a number of QoS properties are involved, the number of rules can be large and it becomes a tedious task for the user. It does not consider that some QoS criteria can be defined by using only crisp form such as criteria having Boolean or string value type [12].

Maolin Tang etal[13] quoted service discovery as so-called optimal web service selection problem. This paper proposes a new hybrid genetic algorithm for the optimal web service between some web service implementations like dependency and conflict constraints. When an implementation is selected for one web service, a particular implementation for another web service must be selected. This is so called *dependency constraint*. Sometimes when an implementation for one web service is selected, a set of implementations for another web service must be excluded in the web service composition. This is so called *conflict constraint*. Thus, the optimal web service selection is a typical constrained combinatorial optimization problem from the computational point of view. The hybrid genetic algorithm has been implemented and evaluated. They compared various techniques used for service discovery like penalty-based genetic algorithm, the repairing-based genetic algorithm and the hybrid genetic algorithm. Its shown that hybrid genetic algorithm is better than above mentioned techniques. The hybrid genetic algorithm is more suitable for those web service problems with a large number of abstract web services and a large number of constraints[13].

Al-Masri, E[14] have used Artificial Neural Networks (ANN) to classify the best web service according to the user requirements. The ANN classifies the web service based on QoS. They have designed a Web Service Crawler Engine (WSCE) that provides an active monitoring tool that continuously collects the most recent and up-to-date QoS values. They have also provided a QWS data set generated using WSCE for the purpose of research. They obtained most Web services from public sources on the Web, including UDDI registries, search engines, and service portals. The dataset consists of 5,000 Web services, each with a set of nine QWS attributes that we measured using commercial benchmark tools [14].

While analyzing non-semantic approaches it was observed that it involves soft computing techniques like *Fuzzy Logic, GA, ANN* to discover the best web service based on QoS. Their advantages and disadvantages are discussed in Table 1.

Soft computing technique used in non semantic approach	Advantages	Non - advantages
Neural networks	• High accuracy for float values	• Easy to over fit • Rules hard to extract and hard to understand
GA	• It updates the population and search for the optimum with random techniques.	• It is difficult to implement because of crossover and mutation. • System doesn't guarantee success
Fuzzy	• user does not need to specify concrete values of properties.	• a user has to define at most as many rules as there are degrees of acceptance that s/he wants to differentiate

Table 1: List of advantages and disadvantages of various soft computing techniques

Our research work proposes Particle Swarm Optimization (PSO), because of its advantages over other soft computing techniques.
The advantages of PSO are

- Easy to represent the interaction between attributes
- Consider several attributes once
- Balance between local exploitation and global employment
- PSO is easy to implement and there are few parameters to adjust.
- All the particles tend to converge to the best solution quickly even in the local version in most cases.

Subsequent research work [12] using fuzzy approach states that the problem while using only fuzzy is that a user has to define at most as many rules as there are degrees of acceptance that s/he wants to differentiate. When a number of QoS properties are involved, the number of rules can be large and it becomes a tedious task for the user. Additionally, for each QoS property, there are number of membership functions being modeled but they may not satisfy some users.

The significance of our approach is that it solves the above mentioned issues by generating rules automatically by using a dataset; it also

DOI: 10.1145/2088883.2088896 http://doi.acm.org/10.1145/2088883.2088896

reduces the number of rules by removing the rules having zero weight age according to the dataset.

4. PROPOSED PROTOCOL

Our work proposes a service selection protocol for the service registry based on non-semantic approach, which classifies the web services based on QoS. The QoS of the web service are dynamic in nature. The web service composition should be able to adapt dynamically to respond to such changes. The work involves designing a web service platform to behave as a fuzzy expert system which can formulate a relationship between QoS parameters and adaptation behavior of a service.

A fuzzy expert system [15] is application software that performs a task that would be performed by a human expert. It simply uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data [15]. A fuzzy expert system consists of four modules as shown in figure 3:

1. Fuzzification
2. Inference engine
3. Rule Base
4. Defuzzification

Rule base consists of rule used to classify the web service based on the QoS. However a user has to define at most as many rules as there are degrees of acceptance that s/he wants to differentiate. When a number of QoS properties are involved, the number of rules can be large and it becomes a tedious task for the user [12]. To solve the above mentioned problem we have used Fuzzy Clustering and PSO to generate the rule automatically through training data set.

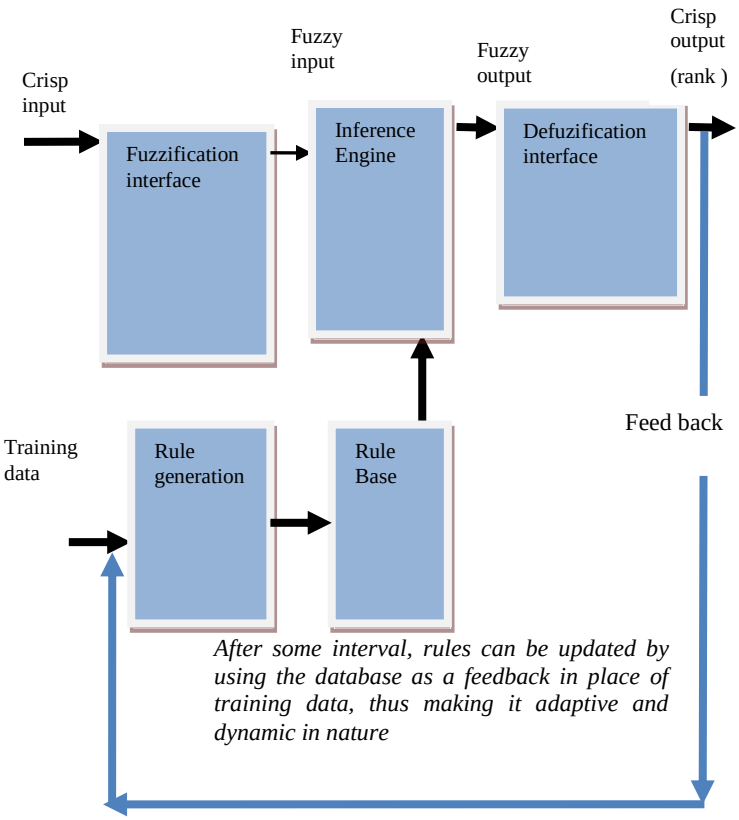

Figure 3: Architecture of service registry using proposed service selection protocol

Hence the system behaves as adaptive and dynamic in nature. The information flow for the service registry component which uses the proposed service selection protocol is shown in the figure 4.

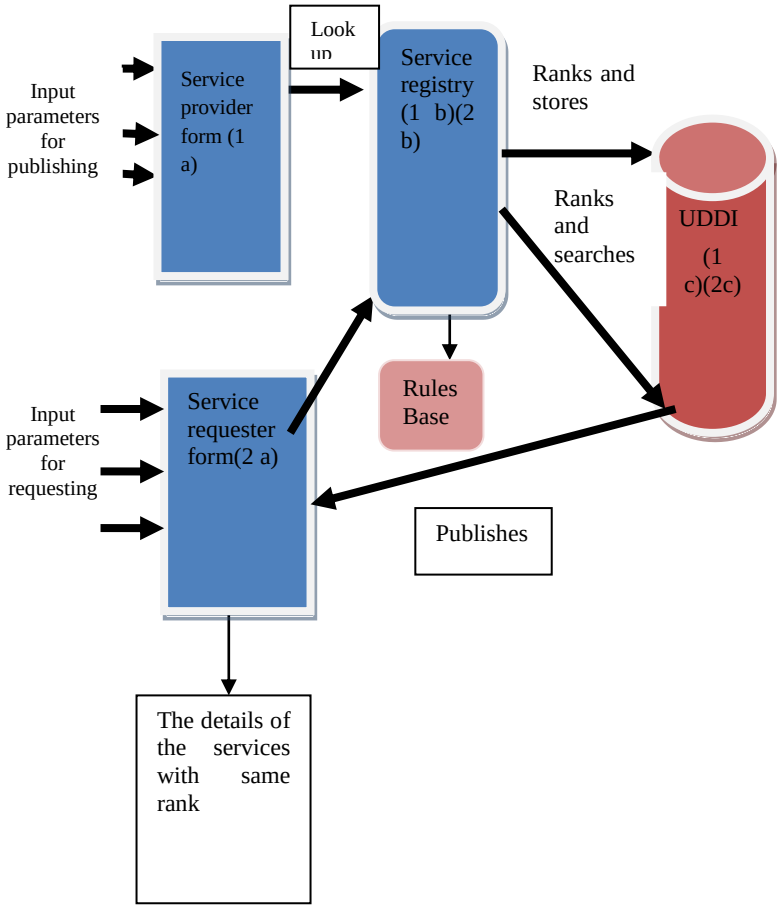

Figure 4: Information flow of the service registry component using proposed service selection protocol.

The data set we have used is QWS data set. This dataset was produced by testing each service over a ten-minute period for three consecutive days by the researchers [16].

We have considered a set of five quality attributes [17] as linguistic variables to simulate the proposed approach using Matlab fuzzy logic toolbox. This paper considers the dataset of *web service phone.*

I. Response Time (ms) ={high , average, low}

II. Throughput (hits/sec) ={ high , average, low}
III. Reliability (%) ={ high , average, low}
IV. Best Practices (%) ={ high , average, low}
V. Documentation ={ high , average, low}

One parameter is considered as output variable

I. Rank ={ Platinum (High quality) ,Gold ,Silver ,Bronze (Low quality)}

In our approach we have considered our system to be TSK model for generating fuzzy rules in the form of linguistic expressions. We have used a min function for the implication method and weighted average function for the defuzzification methods.

Accordingly the simulation of the system is done using *MATLAB fuzzy tool box* as depicted in figure 5, 6 & 7.

DOI: 10.1145/2088883.2088896 http://doi.acm.org/10.1145/2088883.2088896

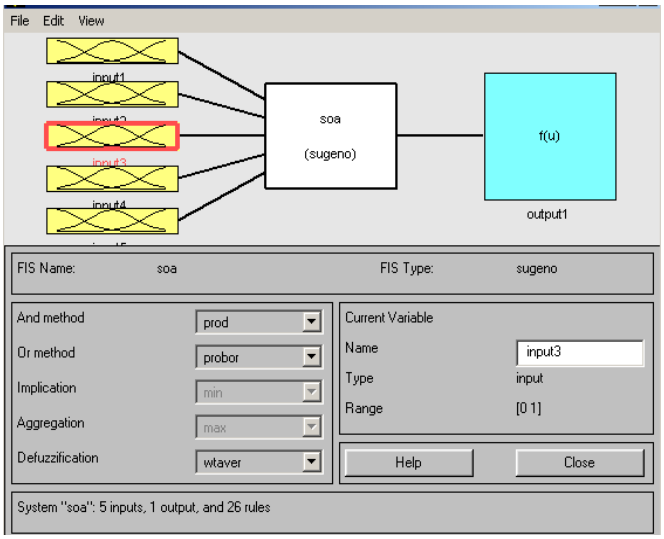

Figure 5: TSk model of the problem domain using FIS in Matlab fuzzy toolbox

The linguistic values of the variables considered are high, average and low.
The membership functions of linguistic value high are taken as Z, average is Triangular and that of low is S shaped.

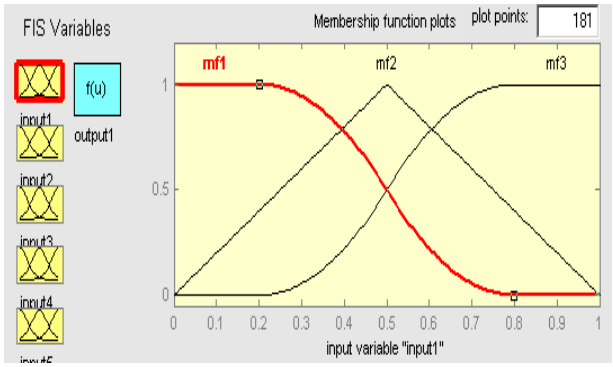

Figure 6: membership functions of the input variables using FIS in Matlab fuzzy toolbox

The membership function for output variable is a constant.

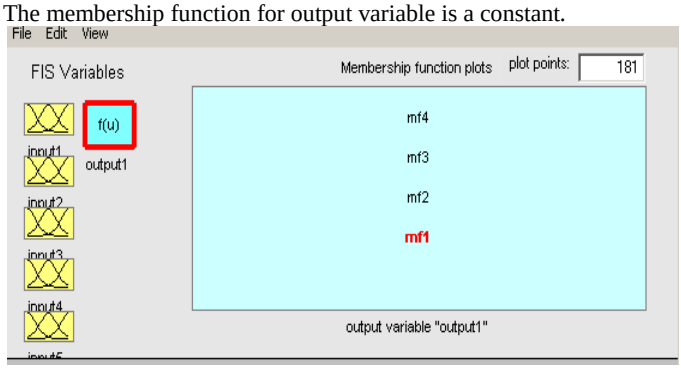

Figure 7: membership function of the Output variables using FIS in Matlab fuzzy toolbox

5. RESULTS AND OBSERVATIONS

A data set which was generated on using the demo [13] and searching for a *web service phone* [16].

The training dataset is shown in table 2.

Name	Response Time (ms)	Through put (hits/sec)	Reliability (%)	Best Practices (%)	Docume ntation (%)	Rank
DOTS GeoPhone	126.2	12.3	78.7	80	86	4
Phone	150.45	7.4	82.1	82	37	3
DOTS Phone Append	118.5	0.7	70.2	80	90	3
Phone Verify	131	1.6	65.9	72	41	3
Phone Notify	437.62	1	68.4	69	93	2
Phone Service	133	1.4	64.7	82	10	1
Phonebook	464	3.1	43.2	80	2	1

Table 2 : Data set on searching the web service phone

The rules are generated using the above given training dataset and the algorithm mentioned by the authors in [17] using Matlab 7.0 are shown in table 3.

1.	if Response Time is low and throughput is low and reliability is average and best practices is low and documentation if low then output is bronze 0.121455
2.	if Response Time is low and throughput is low and reliability is average and best practices is low and documentation if average then output is silver 0.467807
3.	if Response Time is low and throughput is low and reliability is average and best practices is average and documentation if low then output is silver 0.034978
4.	if Response Time is low and throughput is low and reliability is average and best practices is average and documentation if average then output is silver 0.134724
5.	if Response Time is low and throughput is low and reliability is average and best practices is average and documentation if high then output is platinum 0.000811
6.	if Response Time is low and throughput is low and reliability is average and best practices is high and documentation if high then output is gold 0.439231
7.	.....
8.	.....
24.	if Response Time is high and throughput is low and reliability is high and best practices is low and documentation if high then output is silver 0.426456
25	if Response Time is high and throughput is average and reliability is low and best practices is average and documentation if low then output is platinum 0.000285
26.	if Response Time is high and throughput is average and reliability is low and best practices is high and documentation if low then output is platinum 0.15462

Table 3: Rule generated from the data set and there corresponding weight

Considering the available data set, the number of rules reduces from 243 (product of linguistic values of input and output variables) to 26,

DOI: 10.1145/2088883.2088896 http://doi.acm.org/10.1145/2088883.2088896

and are automatically generated. The lesser the number of rule, faster will be the processing of ranking. The proposed service selection protocol can be used in the service registry to publish and search the web services. The architecture is adaptive in nature as any change in QoS of a web service will change the rank of the web service

The rule view and the surface view are shown in figure 8 and 9.

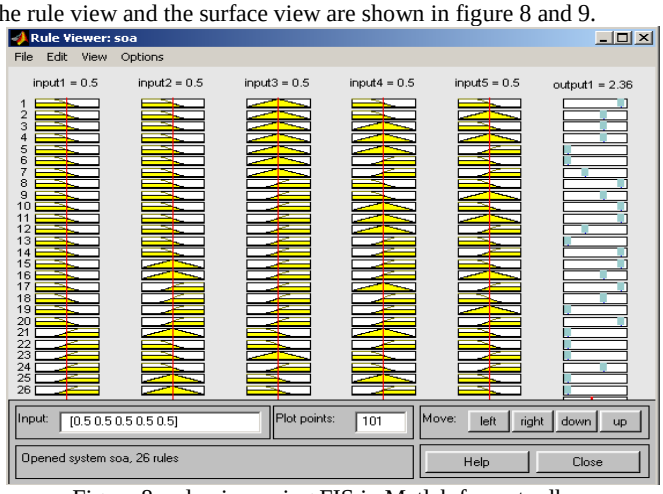

Figure 8: rule view using FIS in Matlab fuzzy toolbox

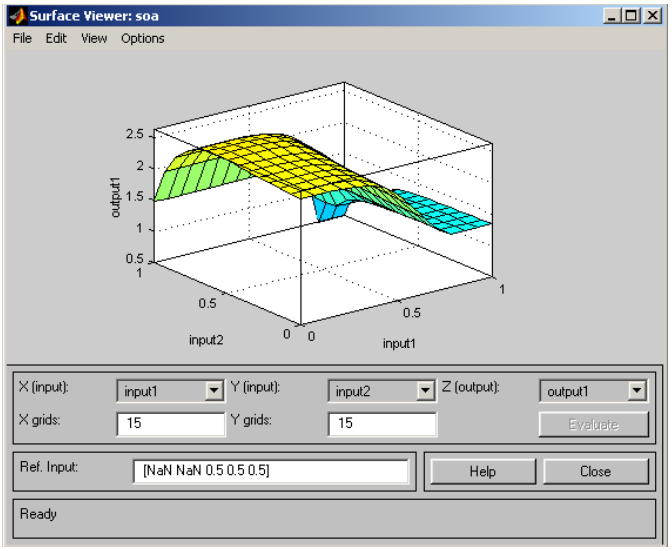

Figure 9: surface view using FIS in Matlab fuzzy toolbox

6. CONCLUSION

The proposed optimized service registry will enable one to develop a better B2B or a B2C e-commerce application with agility. The service requester can compare among the list of web service and choose the appropriate service provider based on its requirements. The problem domain is designed as a fuzzy expert system and this technique is a rule based approach which responds dynamically to quality of service changes in the web services. The rules used in the fuzzy expert system are generated automatically by using the dataset. It also reduces the number of rules by removing the rules having zero weight age and doesn't have any impact on the system. If there are changes in the quality attributes of the web service the rules are changed dynamically, considering the output as a feedback**.** Hence the system is adaptive and dynamic in nature.

The system was simulated using a fuzzy inference engine and it was seen that reducing the number of rules doesn't affect the system, because of the various advantages of PSO it's a better technique than other soft computing techniques.

7. FUTURE SCOPE

Our future work will encompass development of the service registry using proposed service selection protocol. We will be using Netbeans IDE, XML, J2EE framework and comparing it with other approaches in terms of time duration to search the right web service.

Our future work will also involve enhancing our approach in terms of semantic approach. Later analyzing and comparing our approach with respect to the various semantic approaches.

8. REFERENCES

[1] Gruschka, N.; Jensen, M.; Lo Iacono, L.; Luttenberger, N.;" *Server-Side Streaming Processing of WS-Security"* , IEEE Transactions on Services ComputingIssue:99 On page(s): 1 - 14Date of Publication: 06 January 2011Digital Object Identifier: 10.1109/TSC.2010.61

[2] Understanding SOA with Web Services By Newcomer published by Pearson Education India

[3] Qi Yu, Xumin Liu, Athman Bouguettaya and Brahim Medjahed "Deploying and managing Webservices: issues, solutions, and directions" The VLDB Journal — The International Journal on Very Large Data Bases Volume 17 Issue 3, May 2008

[4]http://www.webservicessummit.com/News/UDDI2006.htm last accessed on June 2011

[5] http://webservices.seekda.com last accessed on June 2011

[6] http://uddi.xml.org last accessed on June 2011

[7] Segev, A.; Toch, E. *Context-Based Matching and Ranking of Web Services for Composition* IEEE Transactions on Services Computing, 2009 vol 2 Issue: 3 July-Sept. 2009 pp: 210 – 222 doi: 10.1109/TSC.2009.14

[8] Kritikos, K.; Plexousakis, D *Requirements for QoS-based Web Service Description and Discovery* IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 2, NO. 4, pp 321-336 OCTOBER-DECEMBER 2009

[9] Pilioura, T. and Tsalgatidou, A. 2009. Unified publication and discovery of semantic Web services. ACM Trans. Web. 3, 3, Article 11 (June 2009), 44 pages. DOI = 10.1145/1541822.1541826

[10] Yin Baocai; Yang Huirong; Fu Pengbin; Gu Liheng; Liu Mingli; *A framework and QoS based web services discovery* In proc. of the IEEE International Conference Software Engineering and Service Sciences (ICSESS), 2010. Beijing, pp 755 – 758 16-18 July 2010

[11] Yu, T., Zhang, Y., and Lin, K.-J. 2007. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Trans. Web 1, 1, Article 6 (May 2007), 26 pages. DOI = 10.1145/1232722.1232728

[12] Vuong Xuan Tran; Tsuji, H.*QoS based Ranking for Web Services: Fuzzy Approaches* In proc. of the 4th International Conference on Next Generation Web Services Practices, Seoul, pp 77 – 82, 20-22 Oct. 2008

[13] Maolin Tang , Lifeng Ai *; A Hybrid Genetic Algorithm for the Optimal Constrained Web Service Selection Problem in Web Service Composition* In proc. Of IEEE Congress on the Evolutionary Computation (CEC), 2010 pp 1 – 8 18-23 July 2010

[14] Al-Masri, E.; Mahmoud, Q.H.; *Discovering the Best Web Service: A Neural Network-based Solution* In proc. of the IEEE International Conference on Systems, Man and Cybernetics, 2009. San Antonio, TX, USA, pp 4250 – 4255 11-14 Oct. 2009

[15] Fuzzy Logic: Intelligence, Control, And Information by George J. Klir, Bo Yuan published by Pearson Education India

[16] http://www.uoguelph.ca/~qmahmoud/qws/index.html as on Jan 2011

[17] Rajni Mohana and Deepak Dahiya. "Optimized Service Discovery using QoS based Ranking: A Fuzzy Clustering and Particle Swarm Optimization Approach," Paper published in the IEEE Proceedings of the 35th IEEE Computer Software and Applications Conference, Munich, Germany (IEEE COMPSAC 2011), pp. 452 – 457.