Journal of Computer Science 10 (5): 861-868, 2014
ISSN: 1549-3636
© 2014 Science Publications
doi:10.3844/jcssp.2014.861.868 Published Online 10 (5) 2014 (http://www.thescipub.com/jcs.toc)

QOS BASED SELECTION AND COMPOSITION OF WEB SERVICES-A FUZZY APPROACH

[1]N. Hema Priya, [1]A.M. Shobana Priya and [2]S. Chandramathi

[1]Department of Information Technology, PSG College of Technology, Coimbatore, India
[2]Department of Electrical Sciences, Hindustan College of Engineering and Technology, Coimbatore, India

Received 2013-10-22; Revised 2013-12-31; Accepted 2014-01-18

ABSTRACT

Web Services are emerging technologies which satisfy users' requests on the Web. Selecting appropriate services to suit the needs is a huge task. There are a huge number of service providers who offer same type of service, with differences in functional and non functional qualities. A few users may need multiple services at the same moment. So composition of services also becomes important. Moreover the consumers get confused in selecting apt services, which leads to the necessity of an automated service selection mechanism based on QoS metrics. It is tedious to find the service with the composite of QoS parameters along with the consumer vague perception. We take these constraints into account and implement a fuzzy rule based algorithm for selection and composition. The open source environment is preferred for a number of reasons. The simulation results show that this setup works well in Service oriented environments.

Keywords: Web Services, Selection and Composition, QoS, Fuzziness

1. INTRODUCTION

Web Services are the standard method for exchanging data and functionality over the Internet in a loosely coupled fashion. They form a predominant way of implementation in Service Oriented Architecture (SOA). SOA supports the changing nature of business requirements and is widely used in solving many real time problems. Web services are considered as self-contained, self-describing, modular applications that can be published, located and invoked across the Web.

Web Service architecture model consists of three major entities, Service Provider, Service requester and the Registry which is showed in **Fig. 1**. These are considered as the building block of Web services. The Service providers are many and the register their Service Level Agreements (SLAs) with the registry. The Service Requesters, based on their needs look up the registry and need to choose appropriate service(s). Then the requesters bind themselves with the providers and invoke their services. Some providers may need authentication based on the type of service.

Web services are implemented using the standards like Universal Description and Discovery Integration (UDDI), Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), Services are made available via the internet by a service provider and their description is published (using WSDL descriptor files with details stored in UDDI repositories); a service requester will query the UDDI or any middleware agent to find an appropriate service and then use SOAP to invoke that service.

2. PROBLEM DEFINITION

If the service requester is in need of a service, he has to surf the web (e.g., Google), a public or a private registry, or a portal and have to spend infinite time on that. Once he has found a list of services, he has to choose the best to suit his needs. So there is a necessity for an automated system, which stores the service level agreements of the service providers and incorporates algorithm to select user's request based service. The services differ by functional and nonfunctional aspects. Therefore QoS based selection becomes predominant. But it is tedious to composite the QoS for consumer vague perception with multi criteria QoS decision. So, all these constraints are considered in carrying out the work.

Corresponding Author: N. Hema Priya, Department of Information Technology, PSG College of Technology, Coimbatore, India

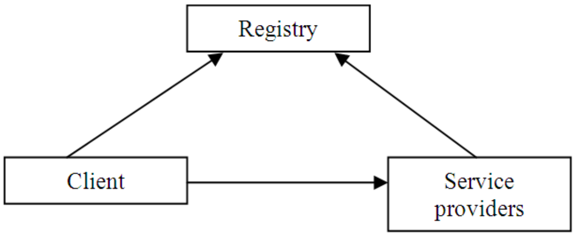

Fig. 1. Web service architecture

The rest of the paper is organized as it explores various related work done in this area, proposed architecture, fuzzy rule based algorithm, implementation details and the results.

3. RELATED WORK

Non-Functional Properties (NFP) play an important role in all service related tasks, especially in discovery, selection and substitution of services. It is simple to imagine a scenario in which multiple services which provide the same functionality can fulfill a user request. In this case the ability of the user to differentiate between the services depends upon their non-functional properties.

Masri and Mahmoud (2007) proposed to solve problems proposed by keyword based search techniques, Web Service Relevance Function (WSRF) for measuring the relevancy ranking of a Web Service is proposed. In order to provide quality driven ranking, the model uses a crawler engine. The model enables users to search and manage criteria based on their interest. Highest ranked service is considered the most relevant as per the user's interest. This reduces Cost of service selection.

An evolutionary based Bee Colony Optimization Metaheuristic (BCO) is projected in (Palanikkumar, 2012), for local optimization of QoS. The BCO is capable to solve deterministic combinatorial problems, as well as combinatorial problems characterized by uncertainty.

Shen and Su (2005) regarded a new behavior model for web services using automata and logic formalisms. Roughly, the model associates messages with activities and adopts the IOPR model in OWL-S to describe activities. A new query language is developed to express temporal and semantic properties on service behaviors.

Chengying (Mao, 2010) adopted a complexity measurement technique based on Petri Nets for web service composition. Based on business process representation, two metric sets are provided hrough analyzing the compositions' execution logics and dependency relations in workflow. The first one is count based metric set and the second one is execution path based metric set. Then an extension based on cognitive informatics is also discussed.

Data mining technique (WEKA) is used in (Susila and Vadivel, 2011) proposed such a scheme, which selects exact web service based on the information from WSDL files.

4. OPEN SOURCE ENVIRONMENT

Open source is a community with free and open source software. Open source is preferred for a number of reasons including interoperability, free and huge support. Web services can be built on the open source premise using web service components like SOAP, WSDL, UDDI and Eclipse IDE. The benefit of combining open source environment is that web service platform is widely available at minimum cost/free. Apache axis can be used for setting up the framework and creating web services or consuming real time web services. Use Apache Tomcat server for deploying the web service.

5. QOS CRITERIA

With the proliferation of web services as a business solution to enterprise application integration, the QoS for web services is becoming increasingly important to service providers. However, due to the dynamic and unpredictable nature of web services (Susila and Vadivel, 2011), it is not an easy task to provide the desired QoS for web service users.

Furthermore, different web service applications with different QoS requirements as in (Ran, 2003; Sora *et al.*, 2009), will compete for network and system resources such as bandwidth and processing time. Nevertheless, an enhanced QoS for a web service will bring competitive advantage for service provider. To provide users a better service, it is first necessary to identify the users' needs and then identify all the possible QoS metrics for web services.

5.1. Reliability

Web services should be provided with high reliability. Reliability here represents the ability of a web service to perform its required functions under stated conditions for a specified time interval.

5.2. Robustness

Web services should be provided with high robustness. Robustness here represents the degree to which a web service can function correctly even in the presence of invalid, incomplete or conflicting inputs.

5.3. Accuracy

Web services should be provided with high accuracy. Accuracy is defined as the error rate generated by the web service. The number of errors that the service generates over a time interval should be minimized.

5.4. Integrity

Integrity for web services should be provided so that a system or component can prevent unauthorized access to, or modification of, computer programs or data. There can be two types of integrity suggested in (Sumra and Arulazi, 1990) data integrity and transactional integrity. Data integrity defines whether the transferred data is modified in transit. Transactional integrity refers to a procedure or set of procedures, which is guaranteed to preserve database integrity in a transaction.

5.5. Accessibility

Web services should be with high accessibility. Accessibility represents, whether the web service is capable of serving the client's request. High accessibility can be achieved, e.g., by building highly scalable systems.

5.6. Availability

The web service should be ready (i.e., available) for immediate consumption. This availability is the probability that the system is up and related to reliability. Time-To-Repair (TTR) is associated with availability. TTR represents the time it takes to repair the web service. The service should be available immediately when it is invoked.

5.7. Interoperability

Web services should be interoperable between the different environments to implement services, so that developers using those services do not have to think about which programming language or operating system the services.

5.8. Security

Web services should be provided with the required security. With the increase in the use of web services which are delivered over the public Internet, there is a growing concern about security. The web service provider may apply different approaches and levels of providing security policy depending on the service requestor. Security for web services means providing authentication, authorization, confidentiality, data encryption, traceability, auditability and non-repudiation:

- Authentication: Users (or other services) who can access service and data should be authenticated
- Authorization: Users (or other services) should be authorized so that they only can access the protected services
- Confidentiality: Data should be treated properly so that only authorized users (or other services) can access or modify the data
- Accountability: The supplier can be hold accountable for their services
- Traceability and Auditability: It should be possible to trace the history of a service when a request was serviced
- Data encryption: Data should be encrypted for secured access
- Non-Repudiation: A user cannot deny requesting a service or data after the fact. The service provider needs to ensure these security requirements

6. FUZZINESS IN SELECTION

Each service has its own functional and non-functional properties. User has to select a service based on his/her requirements. For example, the user needs to select a service based on the cost factor and also with other preferences like response time of the service. Including soft computing techniques can yield better results (Tyagi and Sharma, 2012). So the composite of QoS properties is done with the help of fuzzy rule based service selection. **Figure 2** shows the architecture of the proposed system:

- End user gives vague request to the Fuzzy service discovery broker (middleware)
- Fuzzy service discoverer analyze the given request by passing it to Fuzzy engine
- Fuzzy engine calls the Fuzzy classifier and evaluate the QoS criteria's for the listed service
- Fuzzy engine then from the dependency matrix from the evaluated QoS criteria
- Next, Engine uses the inference rules from the repository and finds the weights for each service
- Dominance service is then selected from all the results of individual service score using Fuzzy ranking technique at Selection stage

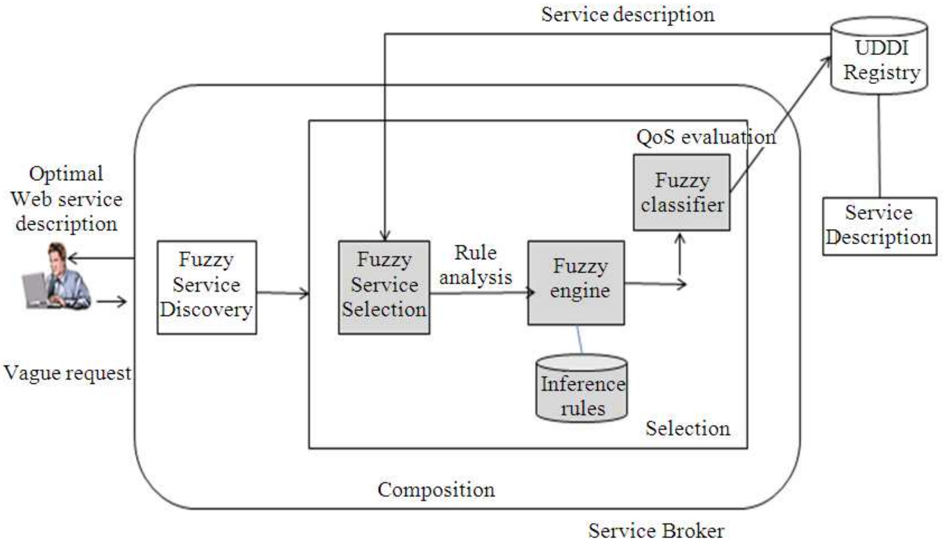

Fig. 2. Architecture of the optimal web service selection

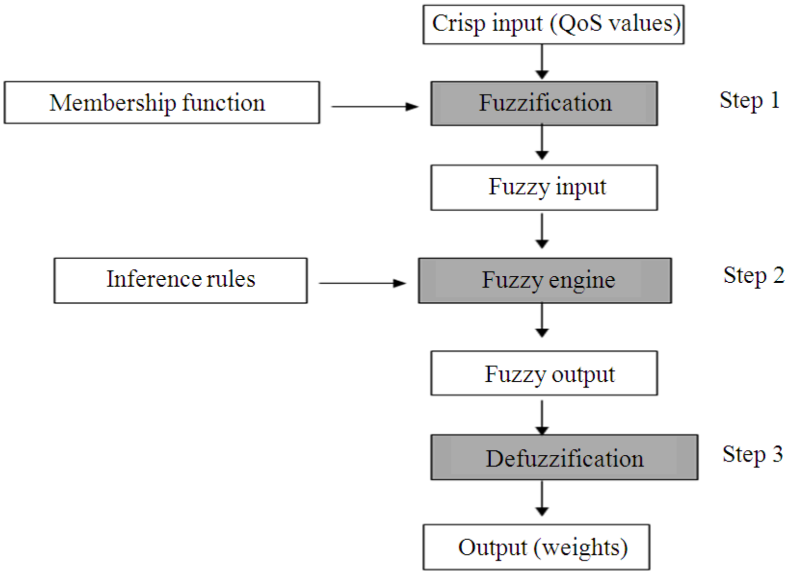

Fig. 3. Fuzzy system flow diagram

6.1. Fuzzification

The crisp input which is QoS values is fuzzified using membership function and passes to fuzzy engine where it uses the inference rules to obtain weights and then it is defuzzified to crisp output. The total flow is described in **Fig. 3**. The first step is to take the crisp inputs and determine the degree (between 0 to 1) to which these inputs belong to each of the appropriate fuzzy sets using membership functions. The center of the Fuzzy modeling technique is the idea of a linguistic variable which is found using membership function. At its root, a linguistic variable is the name of a Fuzzy set. The main membership functions used are triangular and trapezoidal membership function.

6.1.1. Traingular Membership Function

It takes three parameters {a, b, c}. The parameters {a, b, c} (with a<b<c) determine the x coordinates of

the three corners of the Triangular Membership Function as represented in Equation (1):

$$\text{Traingle}(x;a,b,c)=\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) \qquad (1)$$

6.1.2. Trapezoidal Membership Function

It takes four parameters {a,b,c,d}. The parameters {a,b,c,d} (with a<b <= c<d) determine the x coordinates of the four corners of the underlying trapezoidal Membership Function as represented in Equation (2):

$$\text{Traingle}(x;a,b,c,d)=\max\left(\min\left(\frac{x-a}{b-a},1,\frac{d-x}{d-c}\right),0\right) \qquad (2)$$

6.2. Fuzzy Engine

The result of membership function is sent to fuzzy engine which has two steps. First is the creation of dependency matrix. Then based on the inference rules in the repository find the exact optimal service using the dominance degree.

6.2.1. Creation of Dependency Matrix

The obtained membership values are listed in the matrix from with the available list of services at one end and the list of QoS parameters considered at the other end:

$$\begin{array}{cccccc} & q_1 & q_2 & .. & .. & q_n \\ S_1 & a_{11} & a_{12} & .. & .. & a_{1n} \\ S_2 & a_{21} & a_{22} & .. & .. & a_{2n} \\ . & .. & .. & .. & .. & .. \\ . & .. & .. & .. & .. & .. \\ S_n & a_{ml} & a_{m2} & .. & .. & a_{mn} \end{array}$$

where, s_1 to s_m are the services available. q_1 to q_n are the QoS attributes and a_{ij} is the Fuzzy linguistic variable

6.2.2. Fuzzy Mapping Rules

The second step is to take the fuzzified inputs and apply them to the antecedents of the fuzzy rules. So for selecting a service based on the user preferences and criteria's, we use Fuzzy rule based selection of service which is done using Fuzzy IF THEN rules as discussed in (Mao, 2010; Atanassov, 2012). Fuzzy rules allows selecting a service that matches the user preferences more. The input of such Fuzzy systems can be 'Fuzzy', like temperature having values as (very cold, cold, moderate, hot, very hot) and 'crisp values/exact' like temperate value is 30 degree Celsius. The return value for the input is a service that highly matches the users' preferences. If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. To evaluate the disjunction of the rule antecedents, we use the OR fuzzy operation as in Equation (3):

$$\mu_{A\cap B}(x)=\text{Min}\left[\mu_A(x),\mu_B(x)\right] \qquad (3)$$

In order to evaluate conjunction of the rule antecedents, we apply AND operation as in Equation (4):

$$\mu_{A\cap B}(x)=\text{Min}\left[\mu_A(x),\mu_B(x)\right] \qquad (4)$$

For the QoS parameter (like availability and rating) the inference rules generated are:

IF availability IS poor AND rating IS low THEN weight IS low
IF availability IS poor AND rating IS medium THEN weight IS low
IF availability IS poor AND rating IS high THEN weight IS low
IF availability IS medium AND rating IS low THEN weight IS low
IF availability IS medium AND rating IS medium THEN weight IS medium
IF availability IS medium AND rating IS high THEN weight IS medium
IF availability IS good AND rating IS low THEN weight IS medium
IF availability IS good AND rating IS medium THEN weight IS high
IF availability IS good AND rating IS high THEN weight IS high

The reason for using the Fuzzy rule based service selection is it uses natural knowledge uniform representation since it uses IF THEN structure and it is easy to understand, read, add and modify the services. Moreover it separates knowledge from processing so that time complexity is reduced.

Fuzzy rule-based systems usually execute faster than conventional rule-based systems and require fewer rules. With the ability to explain their reasoning, they provide an ideal way of addressing these difficult problems. So, to solve real time problems like Web service selection based on QoS composition, this fuzzy rule-based systems show faster response.

6.3. Defuzzification

The last step in the fuzzy inference process is defuzzification. Fuzziness helps us to evaluate the rules, but the center of gravity defuzzification technique is used and it is expressed as:

$$x = \frac{\sum_{i=1}^{n} m^{l} w^{i}}{\sum_{i=1}^{n} m^{i}}$$

Where:

x = The defuzzified output
m^{i} = The membership of the output of each rule
w_{i} = The weight associated with each rule.

This method is used since it is faster, easier and gives fairly accurate result.

6.4. Algorithm for QoS aware Web Service Selection using Fuzzy Logic

Input: list of web services with QoS attributes.
Output: weights for each Web Service.

1. Extract the QoS parameters for each Web Service.
2. Compute Fuzzy values for QoS parameter.
3. Construct the Fuzzy judgement matrix A = $(a_{ij})_{m*n,}$ Where m = list of QoS attributes, n = Web services and a_{ij} = Fuzzy values
4. Find the Fuzzy values using triangular or trapezoidal membership method.
5. Calculate the QoS Weights using the inference rules in Fuzzy engine by taking Fuzzy judgement matrix as input
6. Defuzzify the values for each Web service using center of Gravity method
7. The Result with Larger value indicates the high rank web service

The above said technique is followed in implementing the case of selection of hotel services, which comprises of composite services like room booking, food and cab facilities.

7. IMPLEMENTATION DETAILS

The selection and composition of atomic candidate services is modeled based on the given below architecture in **Fig. 4**.

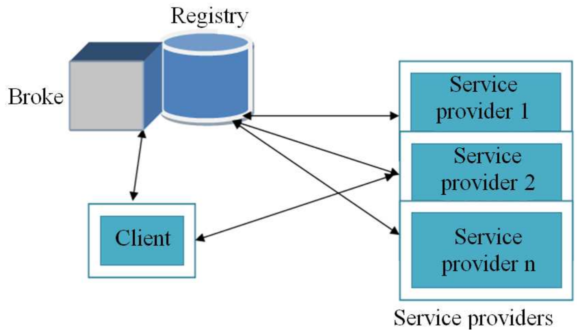

Fig. 4. Web service selection setup

The architecture constitutes client, who requests for service available in the registry, pool of service providers in the registry and a middleware. The middleware constitutes a broker and a registry. The broker employs a fuzzy rule based service selection and composition algorithm. This algorithm takes QoS parameter input from the details stored in the registry and finds the weight for each service by composing the QoS parameters. Finally the ranked list of service providers is sent to the client.

The setup is purely built on Open Source tools, which provide good interoperability at nil cost. The tools used are Java, Apache Tomcat and Apache Axis 2, on Eclipse IDE. Ten similar Service providers with differences in their QoS are considered as shown in **Table 1**. QoS parameters taken into account are Availability and Reliability, based on customer's feedback.

In the **Table 1**, a few listed services are referred from xmethods.net and others are simulated. All the ten candidate services are allowed to undergo the above said processes of QoS aware web service selection using Fuzzy logic and finally ranked with the obtained defuzzified value.

8. RESULTS

A set of 10 functionally similar candidate services are chosen and it is allowed to run in the proposed setup. Also the QoS values are made to change dynamically after every 10 minutes. It was found that the proposed algorithm has eliminated fuzziness and 80% accuracy is achieved in selecting the best available service. The top ranked service is provided to the end user, based on the dynamic changes in QoS values. Also the fuzzy rule based system provides selection in 40 s when executed in a LAN, which is a valuable measure in computing the response time.

Table 1. Service Providers and their details

S. No	Services	Services Offered	Availability	Reliability (Rating)
1	Indian Holiday Service	Service based on dates (Cab, Food, Stay) (Indian service)	High	High
2	Bharath Hotel Services	Service based on dates (Cab, Food, Stay) (Indian service)	High	High High
3	A-Z Holiday Service	Service based on dates (Food, Stay) (Indian service)	Poor	High
4	Holiday Service	Service based on dates (Food, Stay) (Country wise)	Poor	Medium
5	US Holiday Service	Service based on dates (Cab, Food, Stay) (US Service)	Poor	Low
6	Australian Holiday Package	Services offered based on dates Cab, food, Stay)- (Australian service)	High	Medium
7	GBSCT Holiday Dates	Services offered based on dates (Food, Stay)-Scotland service	High	Low
8	GBEAW Holiday Dates	Services offered based on dates (Cab, Food, Stay)-Scotland service	High	Low
9	GBEAW Holiday Service	Services offered based on specific holidays (Cab, Stay)-Great Britain Service	High	Medium
10	Swiss Holidays	Services offered based on specific holidays (Cab, Food, Stay)-Switzerland service	Poor	Medium

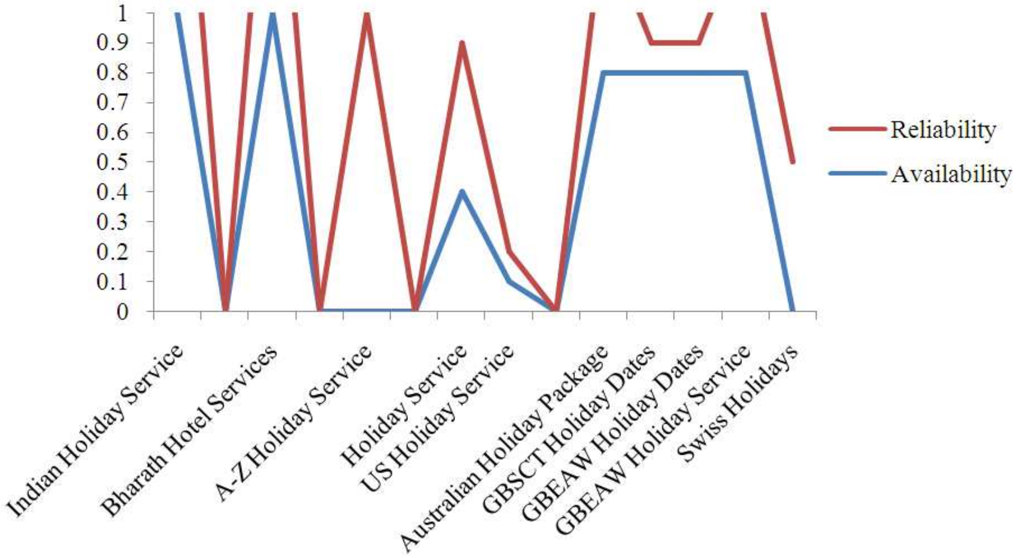

Fig. 5. Ranking of Serices

From the graph **Fig. 5** we can visualize the ranking of top five services, out of ten candidate services. The service with highest reliability and availability will be selected and recommended for the customers. At one moment, Indian Holiday Services and Bharat Services are ranked top, so it is given to the users.

9. CONCLUSION

We have formulated a fuzzy logic rule based dynamic web service selection with respect to the users' preference using open source software. The need for having an automated web service selection at the broker, based on the users' query is discussed. A detailed study about various QoS criteria while selecting the web service is described. We have proposed a fuzzy rule based algorithm, which composes QoS criteria, for automated selection of web service at the broker, is done at a low cost by using open source environment. At last, an implementation procedure and the results using rule based web service selection and composition for selection is given. Reliable and faster service selection is achieved by using this setup. Also this project aims at promoting open source technology.

10. FUTURE WORK

We have concentrated on two QoS criteria, availability and reliability for web service selection. As future work, we concentrate on selecting service by taking more QoS attributes into account. Also service composition can be taken into account along with QoS composition.

11. REFERENCES

Atanassov, K.T., 2012. Intuitionistic fuzzy sets. Fuzzy Sets Syst., 20: 87-96. DOI: 10.1016/S0165-0114(86)80034-3

Mao, C., 2010. Control flow complexity metrics for petri net-based web service composition. J. Software, 5: 1292-1299. DOI: 10.4304/jsw.5.11.1292-1299

Masri, E. and Q.H. Mahmoud, 2007. QoS-based discovery and ranking of web services. Proceeding of the 6th International Conference on Computer Communications and Networks, Aug, 13-16, IEEE Xplore Press, Honolulu, HI., pp: 529-534. DOI: 10.1109/ICCCN.2007.4317873

Palanikkumar, D., 2012. An algorithmic evaluation of optimal service selection using BCO. Eur. J. Scientific Res., 68: 4-4.

Ran, S., 2008. A model for web services discovery with QoS. ACM SIGecom Exchanges, 4: 1-10. DOI: 10.1145/844357.844360

Shen, Z. and J. Su, 2005. Web service discovery based on behavior signatures. Proceedings of the IEEE International Conference on Services Computing, Jul. 11-15, IEEE Xplore Press, pp: 279-286. DOI: 10.1109/SCC.2005.107

Sora, I., D. Todinca and C. Avram, 2009. Translating user preferences into fuzzy rules for the automatic selection of services. Proceeding of the 5th International Symposium on Applied Computational Intelligence and Informatics, May 28-29, IEEE Xplore Press, Timisoara, pp: 497-502. DOI: 10.1109/SACI.2009.5136299

Sumra, R. and D. Arulazi, 1990. Quality of service for web services-demystification, limitations and best practices. Proceedings of the IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, New York.

Susila, S. and S. Vadivel, 2011. Agent based discovery of web service to enhance the quality of web service selection. Int. J. Comp. Sci. Network Security.

Tyagi, K. and A. Sharma, 2012. Reliability of component based systems-a critical survey. WSEAS Trans.