A Quality-centric Scheme for Web Service Ranking using Fuzzified QoS Parameters

Mandar Shaha and Sowmya Kamath S

Department of Information Technology,
National Institute of Technology Karnataka, Surathkal
Mangalore - 575 025, Karnataka, INDIA
mandar.shaha90@gmail.com,sowmyakamath@nitk.ac.in

Abstract. Service composition, the process of combining already available basic services to provide a new, enhanced functionality, helps in serving diverse user requirements and promotes rapid application deployment. One of the premises for achieving service composition is, to consider the quality of service parameters like availability, response times etc, of the constituent services, so that effective ranking can be obtained. However, based on user need, multiple criteria may need to be considered during QoS based ranking, due to which it may be difficult to provide accurate and precise values with respect to a particular QoS parameter. In this paper, we address this problem by incorporating the theory of fuzzy logic, using fuzzy variables. We propose a new scheme that focuses on computing the combined values of various QoS parameters, for enhancing Web service recommendation. The proposed scheme has been applied to the real world datasets, with encouraging results.

Keywords: Web Service Ranking, Fuzzy Modeling, Quality of Service, Non-functional Requirements, Web Service Recommendation

1 Introduction

Owing to the wide-spread popularity of Web services as the technology of choice for Service oriented Architecture (SOA) based system design, there has been an exponential growth in the availability of varied services, many of which provided same or very similar functionalities. The trend is now towards novel application development based on the concept of composite Web services. Composite Web services are those specialized service workflows, where, basic services providing different but related functional services may be combined to implement a completely new, customized service. As there exist various limitations to the simple functionalities provided by a particular web service, it becomes a necessity to combine diverse web services, in view of attaining complex functionalities as required by envisioned business applications [1].

A significant requirement while developing a composite services is to effectively consider both functional and non-functional capabilities of their constituent services. The functional capabilities of a service is accessible through its service description in the form of WSDL (Web Service Description Language). It is a Web service standard for representing the inputs, outputs, message, operations, service endpoints etc, of a service. The non-functional capabilities of a service are most often expressed in the form of Quality of Service (QoS) parameters. QoS has been widely used as a standard way to model and evaluate the non-functional features of a Web service.

Typical QoS parameters considered for Web service based applications include reliability, response time, security, and invocation fee [2]. Hence, it is given that QoS factors play an essential role in various Web service management tasks such as - selecting a service that fulfill both functional and non-functional requirement specified by a user and ensuring a certain standard of user level performance [3]. QoS also serves as the primary criteria for differentiating between multiple Web services that offer similar functionality. As a result, many QoS-aware or QoS-based approaches have been proposed for problems like Web service discovery, selection and composition [4][5].

One of the crucial limitations in QoS based approaches, is that it is often counterproductive to merely provide the exact values for a considered QoS parameter. The most intuitive way in which users can indicate their quality preferences is by using relative performance metrics like 'good service', 'fast delivery' etc. In this paper, we aim to address this problem, by using the concepts of Fuzzy logic. We considered four QoS parameters - *response time* (total time taken to respond to a request for service), *latency* (time elapsed since user generated request till the desired output is displayed), *availability* (the total time a service is capable of being used during a given unit time interval) and *throughput* (number of requests handled by service per unit time). The proposed approach also uses a scheme for combining two different QoS parameters, viz. *response time* and *latency*, based on which Web services can be effectively ranked.

The remainder of the paper is organized as follows: Section 2 presents a discussion on relevant existing work in the area of QoS based service discovery and recommendation. Section 3 describes the proposed Fuzzy logic based Web service ranking approach using QoS parameters. Section 4 describes the observed results and the effect of the proposed composite QoS parameters on Web service recommendation, followed by conclusion and references.

2 Related Work

Web service discovery is an essential task in Web service application development and management. As the number of Web services grow exponentially, the emphasis is towards composing two or more different services to provide a new functionality, instead of creating another new service for providing new functionality. To combine services, a service provider essentially has to search in a wide range of service collection, and consider many available services providing the same functionality, and finally decide the ones most relevant for composition. For choosing candidate service for composition, a service provider can consider various QoS parameters, as required by the application under development.

Many researchers have worked upon linking statistical approaches for the web service recommendations using non-functional parameters. Dai et al [6] proposed to achieve automation in Web service recommendation to service providers to enhance the process of Web service discovery. Their focus was on extending a goal-oriented mechanism for Web service discovery, using rich functional and non-functional parameters, like the service's signature, domain ontology and preconditions etc. Mishra et al [7] devised a new system that considered the sequential information in Web navigational patterns, with the content information. Their system utilized a technique called the similarity upper approximation and then applied singular value decomposition (SVD) for providing service recommendations to users. In Benaboud et al's [8] work, the functionality match is carried out between services using OWL-S (Web Ontology Language for Services), then functionally similar services are ranked based on the QoS score.

Liu et al [9] proposed an algorithm called branch and bound for execution plan selection (BB4EPS), that combines services based on QoS attributes, where the availability and reliability are considered separately, but here, their combined effect is not evaluated. Lin et al [10] designed a heuristics based RQSS (Relaxable QoS-based Service Selection) algorithm which uses Multi-dimensional, multi-choice Knapsack problem, for composing services. Here, QoS attributes like Execution Time, Reliability, Availability, Reputation (based on user feedback) and Price were used for composing the services.

Gouscos et al [11] proposed a technique where QoS attributes were classified as static (price, promised response time and failure probability, which are stored in UDDI) and dynamic (actual response time, rate of failure, either stored in WSDL or provided by an information broker). But, they did not address the problem of dealing with out-of-date QoS information, once the data is stored in the UDDI. Huang et al [12] proposed a single QoS-based service discovery technique, where the service with best QoS attributes is selected and a service with best performance in an entire workflow is selected by a process called QoS-based optimization. Hwang and Chao [13] proposed several multi-agent based approaches with fuzzy group decision making methods and semantic Web technologies to assist both service providers and consumers in discovering appropriate services with respect to their expectations and preferences.

In contrast to the approaches discussed, we propose a fuzzy logic based approach that considers both the direct and combined effect of multiple QoS parameters on Web service ranking, during Web service discovery. In the proposed technique, a composite QoS parameter that includes both response time and latency for ranking Web services providing similar functionality.

3 Proposed Methodology

The overall methodology for ranking the web service is shown in Figure 1. The process involves applying a process of fuzzification to the crisp values of QoS parameters to get the fuzzy sets along with degree with which a crisp value is belonging to a particular fuzzy set. These are given as input to the next phase where inference rules are applied to generate the fuzzy sets for the proposed composite QoS parameter. Defuzzifying these resultant fuzzy sets will generate the crisp value of the proposed composite QoS parameter, using which the services are ranked.

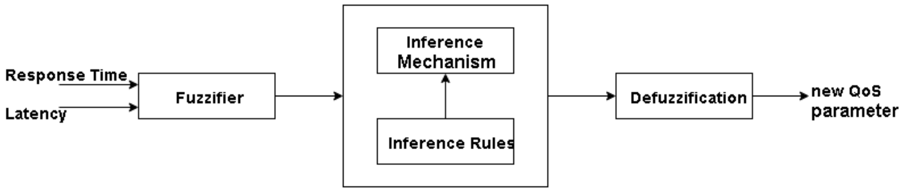

Fig. 1. Fuzzy Ranker layout

Fuzzy logic theory basically allows variables to take any real value between 0 and 1. Hence, it is a many-valued logic, in which the concept of partial truth or falsity is valid, unlike Boolean logic, where either complete truth or complete falsity are considered. The proposed fuzzy QoS ranker deals with both response time and latency based on a fuzzy rule based decision maker. This component aids in the computation of a new QoS value for all relevant Web services according to response time and latency. An overview of this process is shown in Figure 1. Each of the components are discussed in detail next.

3.1 QoS Fuzzifier

In this stage, the crisp values of QoS parameters considered - Response Time, Latency, Availability, Throughput etc., as taken as input, and then undergo a process of fuzzification. The result is in the form of QoS fuzzy sets and the degree to which the input crisp values belong to these fuzzy sets, which is determined by using membership functions.

Membership functions (MFs) are the building blocks of fuzzy set theory, i.e., fuzziness in a fuzzy set is determined by its MF [14]. Accordingly, the shapes of MFs are important for a particular problem since they effect the fuzzy inference system. Different membership functions are available - Triangular Membership Function, Trapezoidal Membership Function, Gaussian Membership Function etc. In the proposed technique, the Trapezoidal Membership function was chosen, as it allows more number of services to belong to a particular class with higher values, when compared to triangular membership function. It also requires less computation when compared to Gaussian membership function.

Firstly, membership functions are defined for a range of values of a fuzzy set and provide a membership value (from 0 to 1) for each value in that range. This membership value indicates the degree to which the given crisp value belongs to a particular fuzzy set. Next, the range of values of QoS parameters are noted and this range is split into 5 equal parts, with each part corresponding to a Fuzzy

Set. A linguistic variable is assigned to each part, based on its lower limit and upper limit. For example, if the Response time range is from 1 to 100 seconds, it is split into 5 parts along with linguistic variables shown in Table 1.

Table 1. Fuzzy Sets for Response Time (Non-Overlapping and overlapping)

Fuzzy Sets	**Non-overlapping Ranges**	**Overlapping Ranges**
Very Low (VL)	1-20	1-26
Low (L)	20-40	14-46
Medium (M)	40-60	34-66
High (H)	60-80	54-86
Very High (VH)	80-100	74-100

Now, if the graph of membership degree versus Response Time is plotted with the ranges on the x-axis, the specific degree between 0 and 1 with which each Crisp value of Response time belongs to a particular fuzzy set (given by the y-axis value) can be found. The ranges shown in column 2 of Table 1 are non-overlapping and one of the disadvantage of such partitioning is that, it cannot provide proper mapping of degree values to boundary values. For example, if a Response Time value is 40, it belongs to fuzzy sets *"Low"* and *"Medium"* with a degree 0.0, which is incorrect. Hence, to provide proper justification to boundary values of these parts, it is better to define overlapping ranges, as shown in column 3 of Table 1.

The part range cannot should be in the range of 0 and 100. Trapezoidal Membership functions are redefined for these new part ranges and each function is represented in the form of 4 points, corresponding to the points of the trapezium. For a given value of Response Time, it is possible to get the membership degree of that value by plotting a vertical line from that value and finding the y-coordinate where this vertical line cuts one of the edges of a membership function. For values falling in overlapping ranges, this vertical line might produce two membership values corresponding to two different fuzzy sets it belongs to. In the same way, fuzzy sets corresponding to the QoS parameter *Latency* are also determined. These fuzzy sets for response time and latency are input to the next stage, *Inference Mechanism.*

3.2 Inference Mechanism

This stage takes as input the fuzzy sets of response time and latency and outputs fuzzy sets of new value of resultant fuzzy set for (RT, L) using Mamdani Style of Inference Mechanism [15]. To explain this inference mechanism, consider the below Inference Rules Matrix given in Tables 2 and . The fuzzy sets defined are as follows: VH (very high), H (high), M (medium), L (low) and VL (very low).

As can be seen from Table 2, if fuzzy set of response time is Very High and fuzzy set of latency is also Very High, then the associated rule indicates that the fuzzy set of new value of combined/composite QoS parameter (RT,L) is Very

Low. This is because, intuitively, it is desirable that a service should have low response time and low latency. But, this approach indicates that the new value of fuzzy set for resultant (RT,L) belongs to Very Low class.

Using this Inference table, each fuzzy set of latency is compared with each fuzzy set of response time to obtain a list of fuzzy sets of resultant (RT,L) value. The membership degree of each of the resultant fuzzy set is the minimum of the membership degree of two fuzzy sets being compared. This list of fuzzy sets and the corresponding membership degree is the input to next step, *Defuzzification*.

Table 2. Inference Rules for Response Time and Latency

Latency \ Response Time	**VH**	**H**	**M**	**L**	**VL**
VH	VL	VL	VL	L	M
H	VL	L	L	M	M
M	L	L	L	M	H
L	M	M	H	H	VH
VL	M	H	H	VH	VH

Table 3. Inference Rules for Availability and Throughput

Availability \ Throughput	**VH**	**H**	**M**	**L**	**VL**
VH	VH	VH	H	H	M
H	VH	H	H	M	M
M	H	M	L	L	L
L	M	M	L	L	VL
VL	M	L	VL	VL	VL

3.3 Defuzzification

Defuzzification is the last step in Fuzzy Inference System. This step takes as input the fuzzy sets of resultant (R,TL) QoS parameters and generates new value in the crisp format. Several different defuzzification methods [16] like Maximum Membership principle, Center of Gravity technique, Weighted Average method etc, are available. The Center of Gravity (CoG) Defuzzification method is the most accurate of all and hence was chosen for this problem.

The CoG method is also known as Centroid method or Center of Area (CoA) Defuzzification. In this method, an aggregate area is determined corresponding to the resultant fuzzy sets and their membership degrees determined in the previous step. A vertical line dividing this aggregate area into two equal masses is determined and the x-coordinate of this vertical line gives the new value of

QoS values in crisp format, which is the required new value. Mathematically, the Centroid value is obtained by the summation formula as shown in Equation 1.

$$COG = \frac{\sum_a^b \mu_A(x)xdx}{\sum_a^b \mu_A(x)dx} \tag{1}$$

Figure 2 shows degree of membership with which the resultant variable belongs to each class, and can be used to illustrate how equation 1 maps fuzzy sets to crisp values. Here, resultant variable belongs to three classes with a degree of membership 0.1, 0.2 and 0.5 respectively. Now, to defuzzify the value of a resultant variable applying COG formula to the graph in figure 3, we get -

$$COG = \frac{(0+10+20)*0.1+(30+40+50+60)*0.2+(70+80+90+100)*0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} \tag{2}$$

which gives the values as 67.4. In this way, the individual fuzzified QoS parameters for Response time and Latency, can be mapped to the proposed composite QoS parameter (RT,L), again represented in fuzzy sets. Finally, these fuzzy sets are defuzzified into crisp composite (RT,L) QoS values, which is used for ranking Web service composition candidates as per user's requirements.

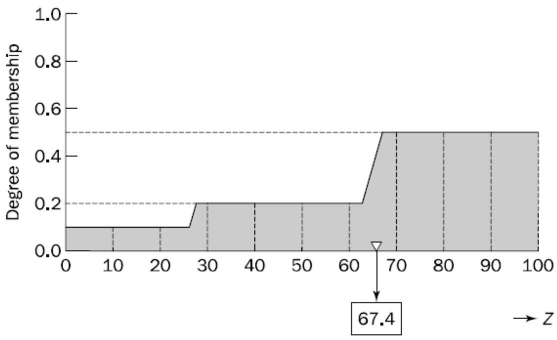

Fig. 2. Defuzzifying the resultant fuzzy sets

4 Experimental Results

To evaluate the proposed approach in ranking Web service during service discovery, a QWS[1] dataset [17], in which we have considered response time, latency, availability and throughput QoS parameters for 1064 real world Web services. To illustrate the effectiveness of the proposed approach, we consider a set of 8 different web services with the QoS parameter (Response Time, Latency, Availability and Throughput) as shown in Table 4. The range of Response Time is noted and split into 5 sub-ranges of equal width, with each sub-range representing a

[1] Available at `http://www.uoguelph.ca/~qmahmoud/qws/`

fuzzy set. The fuzzy sets (VL, L, M, H and VH) for response time, formed using trapezoidal membership function are as shown in Figure 4. Similarly, fuzzy sets are identified for the other QoS parameters considered - Latency, Availability and Throughput also.

For the given crisp values of Response Time and Latency, fuzzy sets are identified such that every crisp value may belong to at most two fuzzy sets. These fuzzy sets along with the degree by which the value belongs to a particular fuzzy set are given as inputs to the inference mechanism. Based on the inference rules, the fuzzy set of the output is determined. Now, using CoG defuzzification technique, a crisp value is obtained for this output fuzzy set, with the help of membership values of input variables. As per the inference rules, the web service with the highest defuzzified value is the best.

Table 4. QoS Metrics for various available Web Services

Services	**Response Time(ms)**	**Latency (ms)**	**Availability (%)**	**Throughput (Req/s)**
WS1	302.75	187.75	89	7.1
WS2	482	1	85	16
WS3	126.17	22.77	98	12
WS4	107.87	58.33	87	1.9
WS5	107.57	18.21	80	1.7
WS6	255	40.8	98	1.3
WS7	136.71	11.57	76	2.8
WS8	102.61	41.66	91	15.3

Table 5. Fuzzy set ranges for QoS Parameters

Fuzzy Set	**Response Time (ms)**	**Latency (ms)**	**Availability (%)**	**Throughput (%)**
VL	102 - 200.8	1 - 49.62	0 - 26	1 - 4.9
L	155.2 - 276.8	27.18 - 87.02	14 - 46	3.1 - 7.9
M	231.2 - 352.8	64.58 - 124.42	34 - 66	6.1 - 10.9
H	307.2 - 428.8	101.98 - 161.82	54 - 86	9.1 - 13.9
VH	383.2 - 482	139.38 - 188	74 - 100	12.1 - 16

To determine the best service, the Response Time and Latency values of each Web service are combined to its (RT,L) value and its Availability and Throughput values are combined to get its (A,TP) value. Table 5 depicts the resultant fuzzified values considering (RT,L) and (A,TP). For web service WS1, Response Time is 302.75 which is High and Latency is 187.75 which is Low. Based on Inference Rules, if RT is high and Latency is low, the output (RT,L) should be of medium fuzzy set (M). Hence, the corresponding defuzzified crisp (RT,L) value is 273.41 as shown in Table 6. For web service WS8, Availability is 91% which is Very High and Throughput is 15.3 which is also Very High.

Based on Inference Rules, if A is Very high and TP is Very High, the output (A,TP) should be of Very High fuzzy set. The corresponding crisp (RT,L) value is 110.33 (as can be seen from Table 6). Hence, higher the fuzzified value, higher the ranking of a service. Based on the results of the service ranking that can be achieved from the fuzzified values tabulated in Table 6, it can be seen that the proposed fuzzy logic based approach works well.

Table 6. QoS Metrics for various available Web Services

Web Service	(RT,L) Fuzzified values	(A,TP) Fuzzified Values
WS1	273.41	95.17
WS2	386.65	107.30
WS3	606.13	110.33
WS4	608.15	88.31
WS5	613.13	80.97
WS6	537.89	95.65
WS7	603.12	80.97
WS8	613.13	110.33

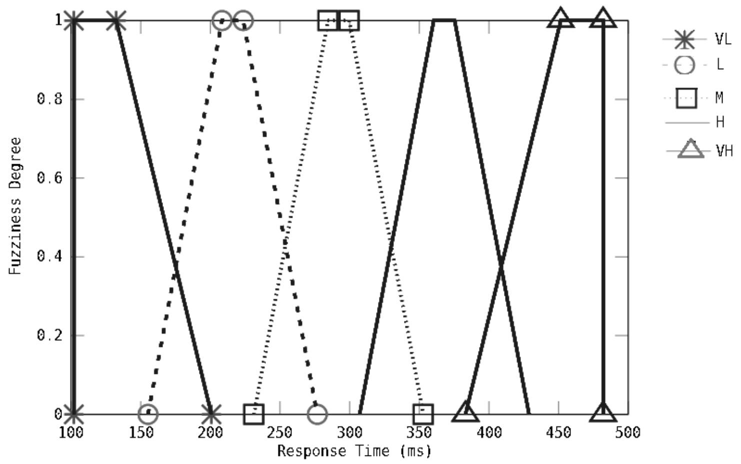

Fig. 3. Fuzzy set ranges for Response Time

5 Conclusion and Future Work

In this paper, we proposed a fuzzy logic based technique for Web service ranking based on composite QoS parameters, as per user requirement. The aim is to solve the problem of ranking the web services based on different QoS parameters. Normally, when the user is given the choice of selecting web services with same functionality and different QoS parameters like Response time and Latency, its difficult to choose the best one. In order to obtain an efficient ranking system, a rule based fuzzy decision approach was presented that deals with both response time and latency. This approach effectively combines both the QoS parameters and computes a new composite QoS parameter, using which functionally similar services can be ranked. As part of future work, we intend to achieve new composite QoS parameters, which can help an application designer in deciding the best service among a pool of functionally similar Web services with significantly less time and effort.

References

1. Gabrel, V., M. Manouvrier, and C. Murat. "Web services composition: Complexity and models." Discrete Applied Mathematics (2014).
2. Barbon, Fabio, et al. "Run-time monitoring of instances and classes of web service compositions." Web Services, International Conference on. IEEE, 2006.
3. Rajeswari, M., et al. "Appraisal and analysis on various web service composition approaches based on QoS factors." Journal of King Saud University-Computer and Information Sciences 26.1 (2014): 143-152.
4. Yan, Hai, Wang Zhijian, and Lu Guiming. "A novel semantic Web service composition algorithm based on QoS ontology." Computer and Communication Technologies in Agriculture Engineering (CCTAE), IEEE International Conference On, 2010.
5. Alexandre Sawczuk, Hui Ma, and Mengjie Zhang. "A graph-based particle swarm optimisation approach to qos-aware web service composition and selection." Evolutionary Computation (CEC), 2014 IEEE Congress on. IEEE, 2014.
6. Dai, Bixiang, and Xinke Li. "An enhanced goal-based semantic web service discovery." Knowledge Acquisition and Modeling, 2009. KAM'09. Second International Symposium on. Vol. 1. IEEE, 2009.
7. Mishra, Rajhans, Pradeep Kumar, and Bharat Bhasker. "A web recommendation system considering sequential information." Decision Support Systems 75 (2015): 1-10.
8. Rohallah Benaboud, Ramdane Maamri, Zaidi Sahnoun. Towards scalability of reputation and qos based web services discovery using agents and ontologies. In Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services 2011;ACM;262269.
9. Min Liu, Mingrui Wang, Weiming Shen, Nan Luo, Junwei Yan. A quality of service (QoS)-aware execution plan selection approach for a service composition process. Future Generation Computer Systems 2012;28(7);1080-1089.
10. Chia-Feng Lin, Ruey-Kai Sheu, Yue-Shan Chang, Shyan-Ming Yuan. A relaxable service selection algorithm for QoS-based web service composition. Information and Software Technology 2011;53(12);1370-1381.
11. Gouscos, Dimitris, Manolis Kalikakis, and Panagiotis Georgiadis. "An approach to modeling Web service QoS and provision price." Web Information Systems Engineering Workshops, 2003. Proceedings. Fourth International Conference on. IEEE, 2003.
12. Huang, Angus FM, Ci-Wei Lan, and Stephen JH Yang. "An optimal QoS-based Web service selection scheme." Information Sciences 179.19 (2009): 3309-3322.
13. Chen, S., Hwang, C., Hwang, F. Fuzzy Multiple Attribute Decision Making: Methods and Applications, Springer- Verlag, Berlin, New York, 292-323, (1992)

14. Zadeh, Lofti A. "Fuzzy logic." Computer 4 (1988): 83-93.
15. Mamdani, Ebrahim H., and Sedrak Assilian. "An experiment in linguistic synthesis with a fuzzy logic controller." International journal of man-machine studies 7.1 (1975): 1-13.
16. Negnevitsky, Michael. Artificial intelligence: a guide to intelligent systems. Pearson Education, 2005.
17. Al-Masri, E., and Mahmoud, Q. H., "QoS-based Discovery and Ranking of Web Services", IEEE 16th International Conference on Computer Communications and Networks (ICCCN), 2007, pp. 529-534.