Optimal Web Service Selection with Consideration for User's Preferences

Tajudeen Adeyemi Ajao[1], Safaai Deris[2]

**Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia
Skudai, 81310 Johor Bahru, Malaysia**

Abstract

The upward trends in web service providers, consumers as well as web services pose remarkable challenges in the area of web service description, discovery and selection. While remarkable work has been done in the area of publishing and discovery, selection still remains an area of challenge. Therefore, emphasis is being placed on how to find an optimal service that will satisfy requester's functional and non-functional requirements. Majority of the existing approaches either ignore the role of user's non-functional requirements or place unnecessary burden on the requester to provide weights for QoS parameters having specify QoS constraints while others assigned arbitrary value of zero to the weight(s) of parameter(s) not specify in the constraints by requesters. All these have the tendency of generating biased results. This research work proposes an enhanced method for selecting optimal service for requesters using Enhanced QoS-based Web Service Filtering Model. The approach of this work differs from the previous approaches in that user's preferences are taken into count, and the weights are derived from the constraints specified by the user which results to weights that are true representation of user's preferences while relieving user the burden of having to supply weights for QoS parameters. Experiments are conducted using Quality of Web Services datasets and the results confirm the model's ability for selecting best web service based on requester's preferences. The outcome of this research provides useful solution to service-oriented selection problems.

Keywords: *Quality of Service (QoS), Service Filtering, Service ranking, Web service, Web Service Selection.*

1. Introduction

Web services have made tremendous impact in the world of information technology and would make more in the years ahead as they combine the best aspects of component-based development and the web with the usage of internet protocols such as XML for coding and decoding data, and SOAP for the transportation which make them to be language and platform independent. 'The primary objective of Web services is to simplify and standardize application interoperability within and across companies, leading to increased operational efficiencies and tighter partner relationships' [1]. In industry, many applications are built by calling different web services available on the net. However, for web services to deliver the intended benefits, they must be published, discovered and bound. Remarkable work has been done in the area of publishing and discovery; requesters are now saddled with the problem of selecting optimal service from arrays of services discovered in response to their functional requirements. Hence, web service selection has become a crucial problem in service oriented computing [2].

Functional and Non-Functional properties especially Quality of Service (QoS) are the two main classes of requirements that are considered in selecting optimal service for a requester. Nevertheless, web service discovery mainly focusses on functional properties of web services. However, due to large number of services with similar functionalities, web service discovery alone is inadequate for efficient selection of optimal service that would satisfy users' expectations.

In addition, web service discovery should be complimented with effective web service selection approaches, which is a major issue in the field of service oriented computing. Many researchers [3], [4], [5], [6], [7], [8] propose that QoS is a key distinguishing factor between arrays of services offering similar functionalities. Therefore, effective selection approach must take into count users' preferences by considering non-functional properties of services.

1.1 Background of the Study

Web service selection is a process of choosing a service implementation from multiple services discovered by a discovery agent. The proliferation of web service providers, consumers as well as web services call for appropriate approaches to web service selection. The bottom line is that the web service requester should be able to control the discovery and selection process to locate services of interest. Although, service discovery is a pre-requisite requirement for service selection process, but, service selection is a major issue that must be addressed in order to retrieve optimal web service for a requester [9]. Since various users may have different inclinations on QoS therefore, it is imperative to represent QoS from the perspective of requesters' preferences. As an illustration, a requester may have preference for only response time owing to tight time constraint. Another requester may be mainly concerned with availability and successability, regardless of the response time and yet other service selection may be focused completely on reliability regardless of any other criteria. In the light of these, a single candidate cannot satisfy all the requesters because of different inclinations of the various requesters.

Majority of the existing approaches either ignore the role of user's non-functional requirements or place unnecessary burden on the requester to provide weights for QoS parameters while others assigned arbitrary value of zero to the weight of parameter(s) not specified by the requester in making his request. This has the tendency of generating biased result. As a solution, an enhanced QoS-based Filtering model for selecting best web service among discovered services with due consideration for users preferences is proposed.

2. LITERATURE REVIEW

A review of previous work undertaken by researchers in the area of web services selection is presented.

[10] proposed a solution by introducing Web Service Relevancy Function (WsRF) used for measuring the relevancy ranking of a particular Web service based on QoS parameters and client preferences for the purpose of finding the best available Web service during Web services' discovery process based on a set of given client QoS preferences. The study though, recognizes user's preferences however, it places additional burden on user to specify weights for QoS parameters. [11], present a QoS broker based architecture for dynamic Web service selection. They used Quality Constrain Tree mechanism (QCT) to compute the QoS score for functionally similar Web Service using Min-Max normalization method and determine the best Web Service for the requester in response to his/her QoS requirements specification along with functional requirements. Also, their approach relies on user to supply weight for each QoS parameter which is a burden. [12]) proposed a Web service recommender system (WSRec) which integrates user-contribution mechanism for Web service QoS information collection with a hybrid collaborative filtering algorithm. They present Web service QoS value prediction which can be employed for Web service recommendation and selection. Their approach involves complex computation. [6] proposed a framework for QoS based Semantic Web Service Selection. The framework consists of four components -OWL-S converter, Semantic Repository, QoS Broker and Matchmaker. OWL-S converter. The framework determines the WsRF using the normalization process and then selects the relevant web services for requester. Their approach is dependent on perceived QoS by the users which in some cases may not be reliable.

3. METHODOLOGY

This work aims at selecting the best web service for re-quester. It proposes an enhanced QoS-based Filtering Model to pave way for the optimal selection of web service for requester based on his preferences.

3.1 Framework for Web Service Selection

The goal of this research is to select a service that maximizes the overall utility, while satisfying all the requester's specified constraints. The framework for web service selection is presented in Fig. 1.

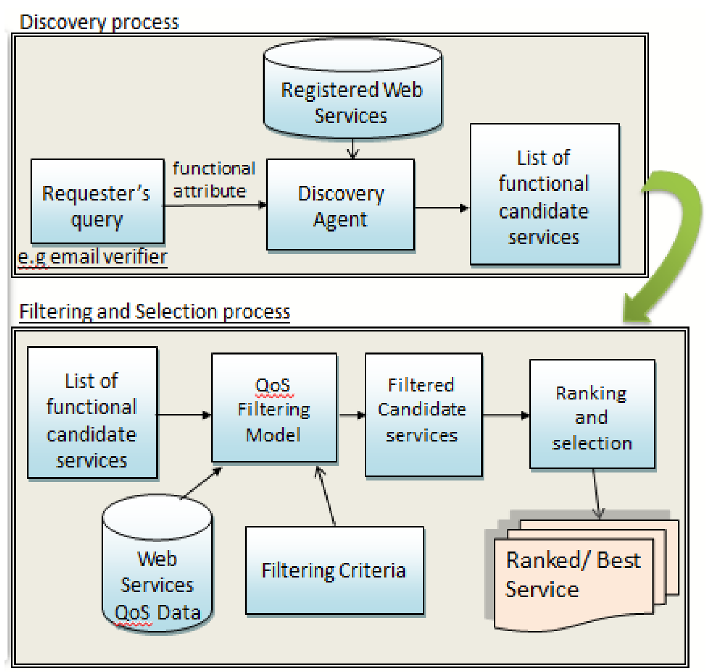

Figure 1: Enhanced QoS-based Web Service Filtering Model

In figure 1 above, the discovery process starts from the time a requester submits a query for a desired service with specific functional task such as "email verification". When the matchmaking process returns lots of services with comparable functionalities, then, the filtering model plays essential role of filtering out redundant services from relevant ones based on QoS constraints. As seen above, the

discovery agent generates a list of functional candidates in response to user's functional requirement.
Then, the list of functional candidate services, Filtering Criteria and the QoS data are accepted as input by the QoS Filtering Agent to generate list of Filtered candidates after removing those candidates that do not meet user's QoS constraints. The Filtered candidate services is accepted as input by the Ranking and Selection Agent computes the scores for each service and then ranks the services based on their scores to produce a final list of Ranked List/ Best Service.

3.2 QoS Tendency

QoS attribute can be positively or negatively inclined. The value of positive attribute needs to be maximized (e.g availability) since larger value is better, whereas the value of negative attribute requires minimization (e.g. response time) in this case the lesser value is preferred.
The table below shows the tendency of parameters being considered.

Table 3.1 QoS Tendency

S/N	*Attribute*	*Tendency*
1	Response Time	Negative
2	Reliability	Positive
3	Availability	Positive
4	Successability	Positive

3.3 QoS Data Normalization

According to [13], there are two purposes of normalization: one is to allow for a uniform measurement of service qualities independent of units; the other is to put all values (qi) in a range from 0 to 1.

Since QoS attributes vary both in magnitude and direction, attributes values must be transformed to a standard range and the same sense of direction in order to have a fair evaluation.

One of the most widely used approaches for this transformation is Max-Min method as depicted below:

$$qp = \frac{q - qmin}{qmax - qmin} \quad (1)$$

$$qn = \frac{qmax - q}{qmax - qmin} \quad (2)$$

qmax != qmin, 1 if qmax = qmin
qp, qn represent normalized value for positively and negatively inclined QoS parameter respectively, qmax and qmin are the maximum and minimum QoS values for a set of QoS parameters.

3.4 Computation of weights

To compute weight for each constraint, the normalized value of the constraint parameter is divided by 4 since we have 4 constraint parameters.
Those constraints which user does not enter values for will be assigned default values say di/4 where di represents the corresponding default value in the threshold table. This becomes necessary because of the implication that these parameters may have on the overall output. e.g. recommending a service with negligible percentage of availability is undesirable (Majority of the previous approaches did not take this into consideration).

3.5 Computation of Total score for each service (SCi)

Total score for each service is computed using equation (3).

$$SCi = \sum_1^4 (wi*qi) \quad (3)$$

Where SC_i is the score for service i, wi is the weight of constraint i and qi is the corresponding QoS value for service i.

4. Experiment and Result

The model is implemented as a console-based application, using C# language, in Microsoft Visual Studio 2010 environment. The experiments were carried out on HP Pavilion dm4 Notebook PC with the following configuration: Intel (R) Core (TM) i5 CPU 2.53 GHz, 4 GB RAM, and installed Microsoft Windows 7 (64-bit) operating system. QWS dataset provided by [10] which contains details of QoS parameters for over 2,000 web services available on the Web was used in carrying out the experiment.
Requester enters a functional requirement: *email verifier*
The system then displays a list of available services in response to the query.

Table 4.1 List of Functional candidate services (i.e. services that meet user's functional requirement)

WS	*Resp*	*Availab.*	*Success.*	*Reliab.*
S1	249.50	91	97	60
S2	136.00	83	84	83
S3	152.00	96	99	60
S4	539.00	95	98	60
S5	575.50	86	95	67
S6	126.00	99	100	73
S7	95.38	62	62	73

The user enters the following constraints

Table 4.2 User's QoS constraints

Resp	*Availab.*	*Success.*	*Reliab.*
250.00	90	93	60

The constraints are used to filter redundant candidates and the list of relevant candidates is returned.

Table 4.3 List of Filtered candidates (i.e services that met all the constraints)

WS	*Resp*	*Availab.*	*Success.*	*Reliab.*
S1	249.50	91	97	60
S3	152.00	96	99	60
S6	126.00	99	100	73
Const	*250.00*	*90*	*93*	*60*
Max	*249.50*	*99*	*100*	*73*
Min	*126.00*	*91*	*97*	*60*

The QoS values for the Filtered candidates are normalized using equation 2 for Response time and equation 1 for others.

Table 4.4 List of Filtered Candidates with Normalized QoS

WS	*Resp*	*Availab.*	*Success.*	*Reliab.*
S1	0.00	0.00	0.00	0.00
S3	0.79	0.63	0.67	0.00
S6	1.00	1.00	1.00	1.00
Const	*0.00*	*0.84*	*0.88*	*0.44*

The weight is computed by dividing each normalized constraint value by 4.

Table 4.5 Filtered Candidates with constraints weight

WS	*Resp*	*Availab.*	*Success.*	*Reliab.*
S1	0.00	0.00	0.00	0.00
S3	0.79	0.63	0.67	0.00
S6	1.00	1.00	1.00	1.00
weight	*0.00*	*0.21*	*0.22*	*0.11*

Total score of each service is computed by aggregating the product of weight and the normalized QoS using equation 3.

Table 4.6 Total Scores for each QoS attribute

WS	*wi x Resp*	*wi x Avail*	*wi x Succ*	*wi x Reli*	*Total Σ(wi qi)*
S6	0.00	0.21	0.22	0.11	0.54
S3	0.00	0.13	0.15	0.00	0.28
S1	0.00	0.00	0.00	0..00	0.00

Finally, the services are ranked based on the scores of each service.

Table 4.7 Ranked/Best Service (RESULT)

WS	*Total*	
S6	0.54	**Best Service**
S3	0.28	
S1	0.00	

S6 is then recommended to the requester based on his preferences.

5. Conclusion and Future Work

Most of the literature reviewed, agree that as web services are increasing in number and popularity, the number of web services performing the same functionality is also increasing in large number. Therefore, emphasis is being placed on how to find the optimal service that will satisfy requester's functional and non-functional (QoS) requirements. As a solution, an enhanced QoS-based Filtering Model is proposed for the purpose of effective optimal service selection. All web services returned by discovery agent are evaluated in terms of their inclination to the user's constraints values of each quality attribute, and those services that are redundant in terms of meeting user's requirements are winnowed from those that are relevant. The QoS of the filtered candidates are normalized and overall scores for each service is computed using the normalized QoS of the filtered services and requester's specified constraints. The model then performs ranking and selection of optimal services. Experiments are conducted using real datasets, and the result confirms the model's ability for selecting best web service for the requesters with consideration for their preferences. The future work should focus on integrating effective management of QoS with the model.

References

1. Govatos, G. *UDDI is Yellow Pages of Web services. Network World Web Applications Newsletter. Available at* http://www.networkworld.com/news/tech/2002/0527tech.html *[Accessed 21/12/2012]*. 2002.
2. Meng, L., J. Li, and H. Sun. *WSRank: A Collaborative Ranking Approach for Web Service Selection*. in *Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on*. 2011. IEEE.
3. Papazoglou, M.P., et al., *Service-oriented computing: State of the art and research challenges.* Computer, 2007. **40**(11): p. 38-45.
4. Toma, I., et al., *A multi-criteria service ranking approach based on non-functional properties rules evaluation.* Service-Oriented Computing–ICSOC 2007, 2007: p. 435-441.
5. Li, J., et al., *A secure collaboration service for dynamic virtual organizations.* Information Sciences, 2010. **180**(17): p. 3086-3107.
6. Maheswari, S. and G. Karpagam, *QoS Based Efficient Web Service Selection.* European Journal of Scientific Research, 2011. **66**(3): p. 428-440.
7. Priya, N.H. and S. Chandramathi, *Optimal Selection and Composition of Web Services-A Survey.* International Journal of Computer Applications, 2012. **49**(2): p. 1-5.
8. Jamous, M.M. and S.B. Deris, *Web Services Non-Functional Classification to Enhance Discovery Speed.* International Journal of Computer Science Issues, 2011. **8**(4).
9. Sathya, M., et al., *Evaluation of qos based web-service selection techniques for service composition.* International Journal of Software Engineering, 2010. **1**(5): p. 73-90.
10. Al-Masri, E. and Q.H. Mahmoud. *Discovering the best web service*. in *Proceedings of the 16th international conference on World Wide Web*. 2007. ACM.
11. D'Mello, D.A., V. Ananthanarayana, and T. Santhi. *A qos broker based architecture for dynamic web service selection*. in *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on*. 2008. IEEE.
12. Zheng, Z., et al. *Wsrec: A collaborative filtering based web service recommender system*. in *Web Services, 2009. ICWS 2009. IEEE International Conference on*. 2009. IEEE.
13. Hu, J., et al. *Quality driven web services selection*. in *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on*. 2005. IEEE.

Tajudeen Adeyemi Ajao has a Bachelor degree in Computer Science/Economics from Obafemi Awolowo University, Ile-Ife in 1988 and MBA from Ahmadu Bello University, Zaria, Nigeria in 1999 and currently undergoing MSc degree in Computer Science at Universiti Teknologi, Malaysia. Research interests include Web services, Cloud Computing and Software Engineering and Database Management.

Safaai Deris is a full professor of Computer Science at Universiti Teknologi Malaysia. His research interests include but not limited to Software Engineering, Service-Oriented Architecture, Bioinformatics, Artificial Intelligence and Intelligent Systems, Planning and Scheduling, Computational Biology and System Biology.