International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013 184
ISSN 2229-5518

QoS-based Web Service Selection Using Filtering, Ranking and Selection Algorithm

Tajudeen Adeyemi Ajao, Safaai Deris, Isiaka Adekunle Obasa

Abstract— Appropriate web service selection procedure for determining optimal web service for a requester still remain an active area of research owing to the persisting upward trends in services with similar functionality. This paper proposes a QoS-based Filtering, Ranking and Selection Algorithm for the purpose of selecting the best web service for requester in line with his/her preferences. Experiments are conducted using real web services datasets and the outcome of the experiments confirms an improvement over existing methods.

Index Terms— Quality of Service, Service Filtering, Service Ranking, Web service, Web Service Selection

1 INTRODUCTION

The basic building block of Service Oriented Architecture (SOA) is the web service. In web service architecture, service provider presents web services that offers tasks or business procedures which are set up over the internet, for invocation by clients; a web service requester defines requirements for the purpose of finding web services of interest. Publishing, binding, and discovering web services are the three major tasks in web service architecture. The web service architecture in Figure 1 illustrates the service requester, providers, and discovery system with their interactions.

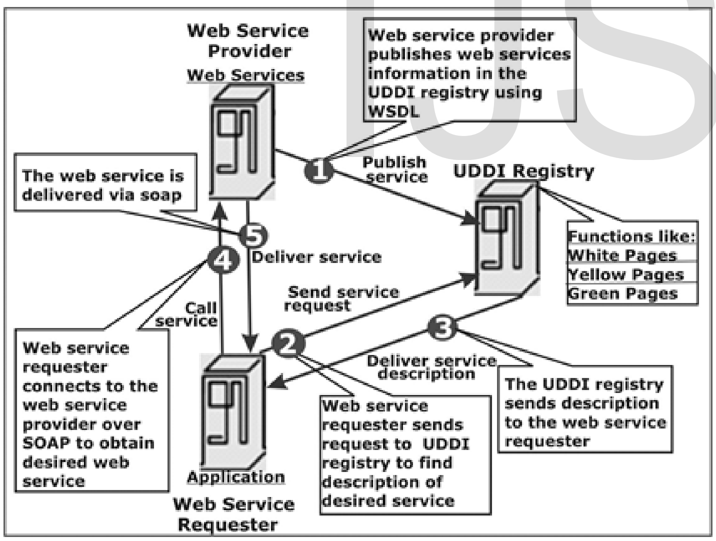

Figure 1: Web Services Architecture (Govatos, 2002)

As shown in figure 1, the service providers build web services that offer specified functions for users' which is made available on the internet for their consumption. The web service requester is any user of the web service who describes and sub mits requests for the purpose of finding a service. The web service registry is a centralized directory of services where service providers publish their service information. The specified information is kept in the registry and examined on submission of request by requester. Universal Description, Discovery and Integration (UDDI) is the registry standard for Web services.

1.1 Web Service

Varieties of definitions of web services are given by researchers and web service consortia. According to World Wide Web Consortium [1], "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards." Web Services are self-independent application that exhibits modular and distributed concepts [2]. Web services can be used by any application irrespective of platform in which it is developed. Web service description is provided in WSDL document, it can be accessed from internet using SOAP protocol. The primary aim of Web services is to demystify and normalize application interoperability within and across establishments, leading to growth in operational proficiencies and intimate partner relations [3]. In industry, many applications are built by calling different web services available on the internet which results in overwhelming acceptance of web services in recent years and the trend will continue for many years to come.

1.2 QoS Requirements for Web Services

According to W3C [4], Quality of Service (QoS) denotes the quality aspects of a web service such as performance, reliability, scalability, availability, etc. Constraints are defined on the QoS, and these constraints can be utilized to select an optimal service for a requester. For example, a requester can request for weather information service with availability of 96%. QoS plays vital role in all service oriented tasks, particularly in the discovery and selection of optimal services. In a situation

- *Tajudeen Adeyemi Ajao is currently pursuing masters degree program in computer science in Universiti Teknologi Malaysia, Malaysia, PH-601116297202. E-mail: taajao2@live.utm.my*
- *Safaai Deris is a professor of Computer Science in Universiti Teknologi Malaysia, Malaysia, PH-6075533033. E-mail: safaai@utm.my*
- *Isiaka Adekunle Obasa is currently pursuing PhD program in computer science in Universiti Teknologi Malaysia, Malaysia, PH-60146164635.E-mail: aiobasa2@live.utm.my*

where multiple services providing similar functionality can accomplish a user's functional requirement, QoS provides a reliable means of differentiating between the services, hence, QoS is an essential factor for choosing optimal service for requesters.

1.3 Web Service Selection

Web service selection refers to the process by which a service implementation is chosen from numerous services discovered in response to requester's functional requirement. Service discovery is a prerequisite for service selection process; however, service selection is a core issue that must be addressed in order to retrieve appropriate service for a requester. Functional and Non-Functional properties especially QoS are the two main classes of requirements that are considered in selecting optimal service for a requester. Much work has been done in the domain of web service discovery, which mainly focusses on functional properties of web services. However, in view of large number of services with comparable functionalities, web service discovery alone is inadequate for selecting optimal service that would satisfy users' expectations, hence; efficient methodologies and procedures are required for appropriate web service selection, which is the main concern in the domain of service oriented computing.

1.4 Background of the Study

QoS related approaches for optimal selection of web services have been discussed in a number of recent literatures. The existing works examined various methods by which optimal web services can be identified from a set of candidates offering similar functionality using the QoS performance of the candidates and the preference of web service requesters. However, previous approaches failed in one way or the other in considering preferences of requesters which led to recommending the same service with highest score of QoS points to different requesters in spite of their diverse QoS inclinations. Some researchers arbitrarily assign unrealistic weights of zero to the weight of parameters not specified by the requester thereby putting the quality of output of such experiment into doubt. In addition, computation of normalized QoS by some of the existing approaches is debatable. Some expects users to specify constraints as well as assign weight to each constraint parameter which results in unnecessary burden on the requester and prone to conflicting representation.

An effective approach should recognize users' preferences and recommend appropriate service to each requester in line with his/her QoS inclination. This research work, attempts to address the above mentioned issues by using QoS-based service filtering, ranking and selection algorithm as a means of selecting best service for the requester with consideration of user's preferences and also deriving weights from user's specified constraints, thereby relieving users of the burden of providing weights for QoS parameters.

2 Literature Review

Service filtering is one of the methods used in managing numerous services discovered in response to requester's query. The aim of service filtering is to expose requesters to only those services that are relevant to their requests while blocking non-valuable services from requester's view. A number of approaches are proposed in the literature for web service selection using filtering techniques on QoS properties of the services. Some of the approaches in this area include reputation-aware/user ratings, Context-based and Knowledge-based. Reputation-aware selection of web service is a method of selecting a web service using reputation which emanates from the public's opinion about the quality of a web service (such as availability, response time, reliability etc). It is unbiased and denotes a combined evaluation of a collection of individuals. Reputation mechanism utilizes consumers' responses to winnow relevant services from those services that are redundant [5]. A number of trust and reputation techniques have been offered for web service selection. Most of these techniques depend on central QoS registry for the collection and storage of feedbacks from consumers. In these techniques, consumers send the data acquired from operating a web service (e.g. reliability, availability, response time values) to the central QoS registry in line with the QoS information and consumer's profile that shows the consumer's ratings for various QoS metrics (i.e. the importance attached to the QoS metrics by a consumer), the QoS registry computes an overall rating for each web service that agrees with the consumer's search request. The web service with the highest score is then recommended to the consumer.

Context-aware services are those services that have their description enriched with context information in line with the service execution environment [6]. Context-aware systems have been used in recommendation system and it aims at enhancing the quality of recommendations by taking into consideration available contextual information, such as location, time, mood, or presence of others. Context-based filtering approach explores users past consumption pattern and recommend based on such interest. The challenges of context based approach include temporal, mood, and social dimensions of context [7].

Knowledge-based filtering technique involves user's specification of his requirements. Recommendation of items is done in line with the user's specified constraints. The approach requires requester to submit requirements using predefined interfaces. This information is then compared with the descriptions of the available services to identify potential service candidates. The services suitable to requester are displayed in ranked order [8].

Some of the works done on web service selection with inclination on QoS-aware are given below:

Yu et al. [9] proposed two models for the QoS-based service composition problem: combinatorial and graph model. They used two heuristic algorithms based on linear programming for service procedures with a serial flow and a general flow structure to enable the selection of QoS-oriented services. The problem of the algorithms lies in scalability.

Xu et al. [10] combined reputation-enhanced service discovery algorithm, which utilizes consumer's feedbacks for QoS computation and a reputation management system used for building and maintaining service reputations for web ser-

ISSN 2229-5518

vice discovery and selection. Their algorithm discovers a set of services that match the consumer's requests. It also ranks the services using the QoS computed and reputation scores based on consumer's preferences in the service discovery request. However, the authenticity of the advertised QoS information presented may be doubtful.

Al-Masri & Mahmoud [11] proposed a solution by introducing the term -Web Service Relevancy Function (WsRF) which is used to measure the relevancy ranking of a specific Web service using QoS parameters and preference of requester in order to determine the best obtainable Web service during Web services discovery process based on a set of preferences specified by requester. The study though, recognizes user's preferences but, it places additional burden on user to specify weights for QoS parameters. Also the use of cost as one of the QoS parameters is subjected to argument as cost was excluded from QoS parameters specified by W3C.

D'Mello et al. [12] present a QoS broker based architecture for the purpose of selecting Web service dynamically. They compute QoS score using Quality Constraints Tree mechanism (QCT) for functionally similar Web Service. Min-Max normalization method was used for determining best Web Service for 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.

Zheng et al. [13] proposed a Web service recommender system (WSRec) which incorporates user-contribution machinery for Web service QoS information gathering with a hybrid collective filtering algorithm. They propose Web service QoS value forecast which can be used for Web service recommendation and selection. Their approach involves complex computations.

Malik & Bouguettaya [14] propose a reputation management framework for the establishment of trust between web services. Their framework made a provision for a cooperative model where sharing of experiences is established between web services regarding their service providers with their peers through response ratings. Various ratings are aggregated to derive service provider's reputation used in evaluating trust. A set of techniques was devised which aim at precisely accumulating the submitted ratings for reputation valuation.

Raj & Sasipraba [15] proposed web service selection model for selecting best web service based on QoS constraints. They stored the QoS attributes of web services in a database. The user specifies functional requirements, QoS values and threshold value for response time and throughput which are used for filtering the related service from the list. Min-Max method is used for normalizing QoS values. This approach places additional burden on the user for having to specify threshold aside from QoS constraints for his preferences. Also, the use of threshold for computation of normalized data for negatively inclined parameters e.g. response time is debatable.

Li et al. [16] present a selection approach tagged fast web service selection (FWSS) for service composition system. They used hierarchical fuzzy system and mixed integer programming to locate the most optimal service for the requester. This approach requires high computation which is a drawback.

Meng et al. [17] present requester's preferences obtained from past QoS values. They propose a QoS model in which users are allowed to specify their preferences while providing combination of multiple QoS properties to give an overall rating to a service. Then, the similarities between users are measured using the correlation between their rankings of services.

Maheswari & Karpagam [18] 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. The framework determines the Web Service Relevance Function (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

The proposed system is designed to carry out the process of selecting optimal service for a requester using service filtering, ranking and selection algorithm. The following four attributes –Response time, Reliability, Availability and Successability are used in this research work. Explanation of these attributes is given in table 1.

Table 1: Explanation on QoS Parameters used in Proposed Method

Sn	Parameter	Description
1	Response Time	Relates to performance factor of a web service which explains how fast a service request can be completed. It is measured in millisecond.
2	Reliability	It is the quality aspect of a Web service which represents the degree of being capable of maintaining the service and service quality. It is measured in percentage.
3	Availability	Relates to the absence of service downtime. It signifies the probability that the service is up and ready for immediate consumption. It is measured in percentage.
4	Successability	Represents the capability of the web service to serve the client's requests. It is measured in percentage.

The four parameters are chosen because of their relative impact in assisting requesters to make reasonable selection decision as they relate to dependability and performance metrics [19] which are fundamental qualities of web services that are necessary for the fulfillment of web service requester's objectives. Dependability relates to building confidence about a web service. Reliability, availability and successability come under dependability metrics. The performance of a web service represents how fast a service request can be completed. Response time is categorized under performance. Reliability, availability and successability are considered crucial attributes in service selection; the response time also becomes very important since majority of consumers expects low response time in retrieving services of interest [20].

3.1 Service Filtering, Ranking and Selection

The service filtering, ranking and selection algorithm is concerned with filtering out redundant services, normalizing the QoS values of each parameter, computing overall QoS score for each service, arranging the relevant services in descending order of overall QoS scores and recommending the best service (service with the highest score) to the requester based on his preferences. Figure 2 depicts this algorithm.

```
/*input:
        a set of n candidates s(f) = (s1,s2,s3,…,sn) that each fulfills
        requester's functional requirements; //this is output from
        discovery agent
        a set of thresholds (default constraints) for desired services
        having 4 elements t = <t1, t2, t3, t4>; //where t1, t2, t3, t4 =
        response time, availability, successability and reliability
        values.
        a set of constraints for desired services having 4 elements
        c = <c1, c2, c3, c4>; //where c1, c2, c3, c4 = response time,
        reliability, availability and successability values*/
/*output:
        an optimal service sp ∈ sf that fulfills requester's function-
        al and nonfunctional requirements*/
// initialization:
        enter threshold data // (default values to be used if user did
//      not enter data for constraint parameter(s))
        store threshold data
// accept qos constraints
        user enters constraints requirements
//filtering: compare each candidate's qos value with user's
//constraints
        calculate total no of candidates (n)
        while i < = n do
            for j = 1 to 4
                if qij(si) < cj then filter out service si  // (filter out
//      the current candidate web service )
                endif
            end for
        end while
//ranking:
  compute normalized qos data for each filtered service
 compute weighted values for the constraints //based on re
//      quester's requirement
 compute product of weight and normalized QoS values for each
     service and get total scores for each service
 sort the services in descending order based on scores computed
 return the first service in the list
```

Figure 2: Filtering, Ranking and Selection Algorithm

3.2 Normalization of QoS values

In the normalization process, equation 1 is used for reliability availability and successability parameters that require maximization whereas equation 2 is used for response time that requires minimization [21].

$$qp = (q - qmin)/(qmax - qmin) \qquad \ldots\ldots(1)$$

$$qn = (qmax - q)/(qmax - qmin) \qquad \ldots\ldots(2)$$

qmax != qmin, 1 if qmax = qmin

qp, qn represent normalized value for positively and negatively inclined QoS parameter respectively, qmax and qmin represent the maximum and minimum QoS values for a set of QoS parameters and q is the QoS value of the parameter being considered.

The QoS values for the constraints are normalized using the following formula:

$$q = (qc - y)/(qmax - y) \qquad \ldots\ldots(3)$$

$$q' = (qmax - qc)/(qmax - y) \qquad \ldots\ldots(4)$$

qmax != y, 1 if qmax = y

where qc represents the QoS value for the parameter being considered, y is the default threshold value for the parameter being considered. q and q' depict normalized value for positively and negatively inclined parameter respectively.

3.3 Service Selection Process

When a service requester submits his query for a service of interest for example getting a map from global positioning system (gps), the Web service discovery agent returns those services that meet the requester's functional requirement. Before service selection is done, the consumer need to specify the constraints for the selection. As an example, the requester asks for a weather service satisfying these constraints: response time less than 350ms, availability not less than 85%, successability greater than 80%, reliability greater than 70%.

Based on Service Filtering, Ranking and Selection Algorithm, those services that fail to satisfy the requester's specified constraints are winnowed out. The QoS values of the filtered candidates are normalized using Min-Max method. The total QoS scores is computed and used for ranking the candidates. finally, the top n (n <=5) filtered candidates are arranged in the order of significance and presented to the requester with emphasis on the service with the highest score in the ranked list as the recommended optimal service.

4 Experiment and Results

Experiments are conducted on the proposed QoS-based Filtering Ranking and Selection Algorithm (QFRSA) and the output compared with the output from Web Service Selection and Ranking Model (WSSRM) proposed by Raj and Sasipraba (2010) which also used max-min method of normalization but computation for minimizing negatively inclined parameters is done using equation 5.

$$qn = (qthreshold - q)/(qthreshold - qmin) \qquad \ldots..(5)$$

Three Requesters (A, B and C) were considered with each of them having the same functional requirement but different

QoS constraints. The functional requirement for each of the users is email verifier.
The list of services returned by discovery agent satisfying the functional requirement (email verifier) for all the three requesters is presented in Table 2 below.

Table 2: List of functional candidates for requesters A, B & C with their QoS values

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

The QoS constraints presented by each of the requesters are shown in Table 3.

Table 3: QoS constraints for Requesters A, B and C

Requester	Respon.	Availab.	Success.	Reliab.
A	600	60	60	66
B	550	80	80	60
C	500	80	65	0

The requesters specify their QoS parameters for response time, availability, successability and reliability in that order and each of the users' QoS constraints are utilized in turn on the two models. Requester C is indifferent regarding the reliability parameter; in this case, the default value of 50 is used in carrying out the test. The subsititution of 0 for the default value is necessary in assisting the requester to select a suitable service owing tio the fact that, selecting a service with negligible reliability is unrealistic. The simulations are presented in scenarios 1 – 3.

4.1 Test Scenario 1: Requester A

Based on constraints (600,60,60,66) given by requester A, the ranked filtered candidates in response to his query with the overall score for each service is shown in Table 4 and 5 for QFRSA and WSSRM respectively.

Table 4: Ranked weighted Sum of Normalized Filtered Candidates for Requester **A**, returned by QFRSA

Service	Resp.	Avail.	Succ.	Reliab.	Scores
S1	0.0732	0.0284	0.0290	0.12000	**0.25056**
S7	0.0749	0.0500	0.0500	0.04500	0.21990
S3	0.0800	0.0000	0.0000	0.04500	0.12500
S4	0.0000	0.0324	0.0434	0.00000	0.07585

Table 5: Ranked weighted Sum of Normalized Filtered Candidates for Requester A, returned by WSSRM

Service	Respon.	Availab.	Success.	Reliab.	Scores
S1	0.07356	0.02838	0.02895	0.1200	**0.25089**
S7	0.07515	0.05000	0.05000	0.0450	0.22015
S3	0.08000	0.00000	0.00000	0.0450	0.12500
S4	0.00388	0.03243	0.04342	0.0000	0.07974

S1 having the highest score (0.25056 in QFRSA and 0.25089 in WSSRM) in the lists is recommended as the best service for requester A by both methods.

4.2 Test Scenario 2: Requester B

For requester B with constraints (550,80,80,60), the ranked filtered candidate services with the scores in response to his query are shown in Table 6 and Table 7 for QFRSA and WSSRM respectively.

Table 6: Ranked weighted Sum of Normalized Filtered Candidates for Requester **B** returned by QFRSA

Service	Response	Availab.	Successa.	Reliabi.	Scores
S7	0.07000	0.15000	0.15000	0.04522	**0.41522**
S2	0.06559	0.12188	0.14063	0.00000	0.32809
S6	0.04907	0.07500	0.12188	0.00000	0.24594
S5	0.00000	0.11250	0.13125	0.00000	0.24375
S1	0.06831	0.00000	0.00000	0.08000	0.14831

Table 7: Ranked weighted Sum of Normalized Filtered Candidates for Requester **B** returned by WSSRM

Service	Respons.	Availab.	Success.	Reliab.	Scores
S7	0.07000	0.15000	0.15000	0.04522	**0.41522**
S2	0.06513	0.12188	0.14063	0.00000	0.32763
S6	0.04689	0.07500	0.12188	0.00000	0.24376
S1	0.06813	0.00000	0.00000	0.08000	0.14813
S5	Service winnowed for not meeting the threshold of 500ms for response time				

S7 having the highest score (0.41522) in the list is recommended as the best service for requester B from both QFRSA and WSSRM.

4.3 Test Scenario 3: Requester C

For requester C with constraints (500,80,65,50), (a default value of 50 is used to substitute 0 since requester C is indifferent to reliability parameter) the ranked filtered candidate services in response to his query are shown in Tables 8 and 9 for QFRSA and WSSRM.

Table 8: Ranked weighted Sum of Normalized Filtered Candidates for Requester **C** returned by QFRSA

Service	Respons.	Availab.	Success.	Reliab.	Scores
S7	0.25000	0.1500	0.0800	0.0000	**0.48000**
S2	0.19737	0.1219	0.0750	0.0000	0.39424
S1	0.22976	0.0000	0.0000	0.0000	0.22976
S6	0.00000	0.0750	0.0650	0.0000	0.14000

Table 9: Ranked weighted Sum of Normalized Filtered Candidates for Requester **C** returned by WSSRM

Service	Response	Availab.	Success.	Reliab.	Scores
S7	0.25000	0.15000	0.08000	0.00000	**0.48000**
S2	0.23262	0.12188	0.07500	0.00000	0.42950
S6	0.16745	0.07500	0.06500	0.00000	0.30745
S1	0.24332	0.00000	0.00000	0.00000	0.24332

4.4 Discussion

In the following section, summary of the results generated from the models are presented for better evaluation. The web service selection and ranking model (WSSRM) proposed by Raj and Sasipraba (2010) is used as baseline. The results of ranking by the proposed method (QFRSA) and the WSSRM proposed by Raj & Sasipraba are presented in table 10.

Table 10: Comparison of outputs from existing and proposed models

Req.	WSSRM (Existing)	QFRSA (Proposed)	Remarks
A	**S1**>S7>S3>S4	**S1**>S7>S3>S4	Same ranking from WSSRM and QFRSA
B	**S7**>S2>S6>S1	**S7**>S2>S6>S5 >S1	S5 excluded by WSSRM
C	**S7**>S2>S6>S1	**S7**>S2>S1>S6	In QFRSA, **S1**>S6, in WSSRM **S6**>S1

WSSRM and QFRSA generate similar ranking list of services for requester A and the scores for the services are almost at par for all services except service S4 whose score is slightly higher in WSSRM than that of QFRSA. However, for requester B, there is an exclusion of S5 in the rank list from WSSRM which was ranked higher than S1 in the list from QFRSA. A close examination of QoS values of all parameters for services S1 and S5 using Euclidean distance shows that S5 has closer similarity to the optimal service returned by both models as shown in table 11.

Table 11: Normalized QoS values for S1, S5, S7 (optimal Qos value) and Euclidean distance value (Case of Requester B)

Service	Respo.	Availa.	Success.	Reliab.	Euc.
S1	0.975787	0	0	1	1.291
S5	0	0.75	0.875	0	**0.528**
S7 (Optimal service)	1	1	1	0.56522	

From table 11, euclidean distances of S1 and S5 from the optimal service returned for requester B are 1.291 and 0.528 respectively. This result shows that service **S5** is preferred to service S1 which agrees with the output from QFRSA.

Also for requester C, in the list generated by QFRSA, service S1 has higher ranking position than S6 whereas they appeared in opposite order in the list generated from WSSRM. A further scrutiny of QoS values of the parameters for services S1 and S6 using Euclidean distance indicates that S6 has closer similarity to the optimal service returned by the two models as depicted in table 12.

Table 12: Normalized QoS values for S1, S6, S7 (optimal Qos value) and Euclidean distance value (Case of Requester C)

Service	Respo.	Availa.	Success.	Reliab.	Euc
S1	0.91903	0	0	1	**0.481**
S6	0	0.5	0.81250	0	0.539
S7 (Optimal service)	1	1	1	0.56522	

Euclidean distance of S1 and S6 from the optimal service returned for requester C gives 0.481 and 0.56522 respectively. This confirms that service S6 is preferred to service S1 and this agrees with the output from QFRSA. The cases illustrated above clearly indicate that the QFRSA approach performs better than the WSSRM proposed by Raj and Sasipraba.

5 Conclusion

In this paper QoS-based Filtering and Ranking Algorithm is proposed for selecting best web service for requesters. The procedure allows users to specify their QoS constraints which are used to filter off redundant services and compute total QoS utility score for each filtered candidate. It filters off redundant services, ranks relevant web services and assists users in selecting the best web service in response to their specified preferences.

Based on this research work, it can be concluded that, the proposed QFRSA provides solution to dynamic web service selection at run time. The output from the proposed QFRSA demonstrates that the probability of selecting a service that best meet user's requirements is improved if the user specifies QoS constraints, and that user's specification of his/her preferences is a key towards selecting optimal service for his/her request. Also, QFRSA produced better quality result in comparison to the existing methods. The future scope of this work is to include additional QoS parameters.

REFERENCES

[1] W3C Working Group *Web Services Architecture. Available at* http://www.w3c.org/TR/ws-arch/ *[Accessed on 05/11/2012].*

[2] IBM WSAT, *Web Service Architecture. Available at:* http://www.ibm.com/developerworks/webservices/library/w-ovr/ *[Accessed on 23/10/2012].*

[3] 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 on 21/12/2012].*

[4] W3C Working Group, *QoS for Web Services:Requirements and Possible Approaches. [Available at* http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/ *[Accessed on 15/11/2012]*

[5] Wang, Y. and J. Vassileva. *A review on trust and reputation for web service selection*. in *Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference*. 2007. IEEE.

[6] Kirsch-Pinheiro, M., Y. Vanrompay, and Y. Berbers. *Context-aware service selection using graph matching*. in *2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC'08), ECOWS. CEUR Workshop proceedings*. 2008.

[7] Said, A., S. Berkovsky, and E.W. De Luca. *Putting things in context: Challenge on context-aware movie recommendation*. in *Proceedings of the Workshop on Context-Aware Movie Recommendation*. 2010. ACM.

[8] Godse, M., R. Sonar, and A. Jadhav, *A hybrid approach for knowledge-based product recommendation*, in *Information Systems, Technology and Management*. 2009, Springer. p. 268-279.

[9] Yu, T., Y. Zhang, and K. Lin, *Efficient algorithms for Web services selection with end-to-end QoS constraints.* ACM Transactions on the Web (TWEB), 2007. **1**(1): p. 6.

[10] Xu, Z., P. Martin, W. Powley, and F. Zulkernine. *Reputation-enhanced qos-based web services discovery*. in *Web Services, 2007. ICWS 2007. IEEE International Conference*. 2007. IEEE.

[11] 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.

[12] D'Mello, D.A., V.S. Ananthanarayana, and T. Santhi. *A qos broker based architecture for dynamic web service selection*. in *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference*. 2008. IEEE.

[13] Zheng, Z., H. Ma, M.R. Lyu, and I. King. *Wsrec: A collaborative filtering based web service recommender system*. in *Web Services, 2009. ICWS 2009. IEEE International Conference*. 2009. IEEE.

[14] Malik, Z. and A. Bouguettaya, *Rateweb: Reputation assessment for trust establishment among web services.* The VLDB Journal, 2009. **18**(4): p. 885-911.

[15] Raj, R.J. and T. Sasipraba. *Web service selection based on QoS Constraints*. in *Trendz in Information Sciences & Computing (TISC)*. 2010. IEEE.

[16] Li, J., S. Wang, Q. Sun, and F. Yang, *A Fast Web Service Selection Approach.* Computing and Intelligent Systems, 2011: p. 502-507.

[17] 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.* 2011. IEEE.

[18] Maheswari, S. and G.R. Karpagam, *QoS Based Efficient Web Service Selection*. European Journal of Scientific Research, 2011. **66**(3): p. 428-440.

[19] Marzolla, M. and R. Mirandola, *Qos analysis for web service applications: a survey of performance-oriented approaches from an architectural viewpoint*, 2010, Technical Report UBLCS-2010-05, University of Bologna.

[20] Sathya, M., M. Swarnamugi, P. Dhavachelvan, and G. Sureshkumar, *Evaluation of qos based web-service selection techniques for service composition.* International Journal of Software Engineering, 2010. **1**(5): p. 73-90.

[21] Hu, J., C. Guo, H. Wang, and P. Zou. *Quality driven web services selection.* in *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference.* 2005. IEEE.