Optimizing Service Selection by User's QoS Expectation

Dongmei Liu, Zhiqing Shao, Caizhu Yu *

Abstract—***QoS* is of paramount importance for Web services to function. However, it is a challenging problem to construct services that meet user's *QoS* expectation. This paper proposes an optimization method for service selection in constructing applications through service composition. In order to make quality rating accurate, ingredients of service reputation and expectation similarity are included in quality evaluation. Optimization of Web services selection is performed at both basic service level and composite service level, which results in optimized service applications to satisfy user's *QoS* expectation. A case study demonstrates feasibility of the method.**

Keywords: QoS, service composition, expectation, reputation

1 Introduction

Service-oriented architecture (SOA) has been recognized as the next generation framework for building agile distributed applications over the Internet. Web services are considered as their components. Web services are self-contained, self-described and modularized. Selecting Web services with high quality while satisfying user's requirements is critical for SOA.

Since service selection is commonly driven by *QoS*, it is critical to evaluate QoS of service accurately and objectively. Zeng et al proposed two service selection methods driven by QoS [1]. Their methods collect quality ratings from the users of a service and then aggregate them using a simple arithmetic average to derive the quality of the service without considering the context where the ratings are derived [2, 3]. Moreover, they do not take into account the fact that some of the ratings may be irrelevant to a particular quality assessment request. Deora et al presented a range based similarity assessment of expectations [4, 5]. But the approach would be inaccurate because of single attribute matching and strict boundaries on expectation. To overcome the shortcomings, they then introduced a fuzzy based similarity assessment that allows calculating similarity on group of the "related" QoS attributes [6]. But it did not consider the weights of the attributes and the weights of the users.

We take user's expectation into account when assessing reputation and only the assessment with similar expectation has impact on the reputation rating for services. Following our previous work on *QoS* metrics [7, 8], we propose an optimized service selection method driven by user's *QoS* expectation for both elementary services and composite services in this paper. The optimal services are selected according to user's appetite and expectation.

The rest of this paper is organized as follows. Section 2 introduces reputation assessment method. Section 3 presents optimization technique for service selection. Section 4 presents a case study. Section 5 is the conclusion.

2 Reputation Assessment Method

2.1 Modeling Reputation Assessment

We introduce several definitions to model reputation assessment as follows.

Definition 1: Let U be a user and A_i be *QoS* attributes of service S. A rating triple on A_i by U is $< E(A_i), P(A_i), W(A_i) >$, where $E(A_i)$ represents the quality that U expects for A_i, $P(A_i)$ the actual quality of A_i perceived or experienced by U after using S, and $W(A_i)$ the weight that U assigns to A_i.

Definition 2: Service rate requester (SR) specifies quality exception $E_R(\varphi_i)$ on *QoS* attribute group φ_i. A similarity score ϕ_i can be obtained when assessing similarity between $E_R(\varphi_i)$ and $E_j(\varphi_i)$ which user j is expected on *QoS* attribute group φ_i. Only the *QoS* attribute group ratings with similarity score beyond some threshold are valid, where $E_R(\varphi_i)$=$E_R(\{\alpha_1, \alpha_2, \ldots, \alpha_n\})$, $\alpha \in [0, 1]$.

Definition 3: According to $E(A_i)$ and $P(A_i)$ in rating triple $< E(A_i), P(A_i), W(A_i) >$, we define $R(A_i)$ the quality rating that U gives to A_i as:

$$R(A_i) = \frac{P(A_i)}{E(A_i)} \quad (1)$$

*This work was partially supported by the NSF of China under grants No. 60773094 and 60473055, Shanghai Shuguang Program under grant No. 07SG32. The authors are with the Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China. The third author is also with Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai 201112, China. Email: dmliu@ecust.edu.cn

If quality rating $R(A_i) \geq 1$, it indicates that the user is satisfied with service quality. Otherwise he is not satisfied.

Definition 4: Let W_{ji}be weight that user j gives to the QoS attribute A_i, $R_j(A_i)$ be the rating that user j gives to the attribute A_i. The assessment on attribute group φ_i by user j is defined as:

$$R_j(\varphi_i) = \frac{\sum_{i=1}^{n} W_{ji} \times R_j(A_i)}{\sum_{i=1}^{n} W_{ji}} \tag{2}$$

Where n represents attributes number of QoS attribute group φ_i. Due to the difference among various users about reputation and trust, the importance of the assessments which are proposed by various users is different.

Definition 5: Suppose T_j is weight of user j. The overall assessment on attribute group φ_i, $Q(\varphi_i)$ is defined by:

$$Q(\varphi_i) = \frac{\sum_{j=1}^{n} T_j \times R_j(\varphi_i)}{\sum_{j=1}^{n} T_j} \tag{3}$$

where n represents the number of users whose assessments are valid.

Definition 6: Reputation assessment on service S, $Q_{rep}(S)$is defined by

$$Q_{rep}(S) = \frac{\sum_{i=1}^{m} Q(\varphi_i)}{m} \tag{4}$$

where m represents number of attribute groups of the service S.

2.2 Similarity Computing

Fuzzy algebra allows representation of vague boundaries and QoS attribute concepts such as"fast", "slow", "high" and "low". Now we use membership function and similarity assessment based on fuzzy algebra to describe QoS attributes of service and model assessment on QoS of service. A QoS attribute group, for example, performance(pf) is represented by attributes response time (rt) and successful completion (sc). The membership function of response time (rt) and successful completion (sc) is shown in Figure 1 and Figure 2 respectively.

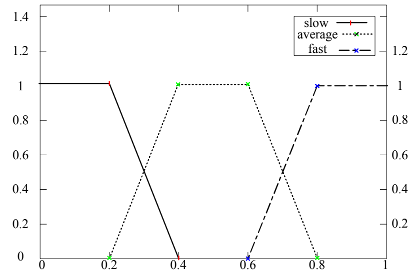

Figure 1: Membership function for rt

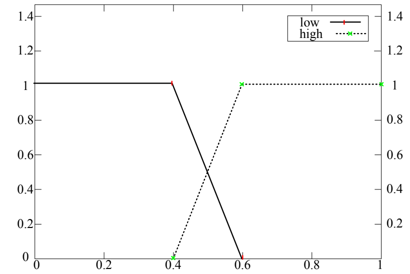

Figure 2: Membership function for sc

Each QoS attribute A_i can be described by m values. A QoS attribute such as *response time* may be represented using "*slow*", "*average*" and "*fast*" whereas another attribute such as successful completion may be represented as "*high*" and "*low*". The membership function value of each QoS attribute can vary and the number of presentations in membership function value depends on the granularity required. As shown in Figure 1 and Figure 2, the fuzzy set for each attribute can be described as follows:

$$A_{rt} = \{slow, avg, fast\} \tag{5}$$

$$A_{sc} = \{low, high\} \tag{6}$$

Using the above notation, we can formulate a fuzzy vector description for each QoS attribute group by

$$\varphi_i = \begin{pmatrix} \mu_{11}(A_1) & \mu_{11}(A_1) & \dots & \mu_{1m}(A_1) \\ \mu_{21}(A_2) & \mu_{22}(A_2) & \dots & \mu_{2m}(A_2) \\ & \dots & \dots & \\ \mu_{n1}(A_n) & \mu_{n2}(A_n) & \dots & \mu_{nm}(A_n) \end{pmatrix} \tag{7}$$

Using the theories of fuzzy mathematics, we can get the similarity score between two fuzzy sets as

$$\phi(E'_j(\varphi_i), E_R(\varphi_i)) = 1 - \frac{|\mu(E'_j(\varphi_i)) - \mu(E_R(\varphi - i))|}{max(E'_j(\varphi_i), E_R(\varphi_i))} \tag{8}$$

ISBN: 978-988-17012-2-0

Table 1: Similarity scores attributes for WS_A

User		Membership function for response time			Membership function for successful completion		Similarity score for Attributes Group
		Slow	Avg	Fast	Low	High	
U1	$E'(A)$	0.9			0.2		0.85
	μ	0	0	1.0	1.0	0	
U2	$E'(A)$	0.7			0.4		0.8
	μ	0	0.5	0.5	1.0	0	
U3	$E'(A)$	0.8			0.9		0.45
	μ	0	0	1.0	0	1.0	
U4	$E'(A)$	0.7			0.9		0.4
	μ	0	0.5	0.5	0	1.0	

Table 2: Similarity scores attributes for WS_B

User		Membership function for response time			Membership function for successful completion		Similarity score for Attributes Group
		Slow	Avg	Fast	Low	High	
U1	$E'(A)$	0.8			0.3		1.0
	μ	0	0	1.0	1.0	0	
U2	$E'(A)$	0.8			0.4		0.85
	μ	0	0.5	0.5	1.0	0	
U3	$E'(A)$	0.6			0.7		0.3
	μ	0	0	1.0	0	1.0	
U4	$E'(A)$	0.8			0.9		0.45
	μ	0	0.5	0.5	0	1.0	

2.3 Reputation Assessment Example

Suppose WS_A and WS_B are shopping stores on network. When customers go shopping, they first use searching engines of WS_A and WS_B to search the items that they want to buy. The respond time of item searching is denoted by rt while the probability of successful searching is denoted by sc.

A Service Rater (SR) assesses QoS of searching service of WS_A and WS_B, and focuses mainly on the attribute group that includes *response time* and *successful completion*. Initial expectation set by SR is E_R=$E_R(\{A_{rt}, A_{sc}\})$=$\{$0.8,0.3$\}$, and threshold $\geq$ 0.5. SR has found 4 users whose expectations are similar for WS_A and WS_B respectively. The similarity scores of attributes can derive according to formula (8), which are presented in Table 1 and Table 2 respectively.

We can infer from above tables that although all users have similar expectation on respond time, but only two users whose similarity scores for the attribute group are beyond 0.5. Therefore, only two user's assessments are satisfied to the request.

Moreover, we assign different weights to different attributes. The user's assessments on attributes group can be derived using formula (2), as shown in Table 3.

The overall quality assessments on attributes group can derived according to formula (3), are presented in Table 4.

Table 3: Attributes quality rating

Services provider		WS_A		WS_B	
User		U1	U2	U1	U2
response time (A_{rt})	$E(A_{rt})$	0.9	0.7	0.8	0.8
	$p(A_{rt})$	0.8	0.7	0.8	0.7
	$R(A_{rt})$	0.9	1	1	0.9
	$W(A_{rt})$	0.7	0.8	0.7	0.8
successful completion (A_{sc})	$E(A_{sc})$	0.2	0.4	0.3	0.4
	$p(A_{sc})$	0.2	0.3	0.2	0.3
	$R(A_{sc})$	1	0.8	0.9	0.9
	$W(A_{sc})$	0.3	0.2	0.3	0.2
Ratings for Attributes Group $R(\varphi_{pf})$		0.93	0.96	0.97	0.9

Table 4: Quality assessment rating

Services provider	User	Ratings for Attributes Group $R(\varphi_{pf})$	User's Weight W'_j	Aggregated Ratings $Q(\varphi_{pf})$
WS_A	U1	0.93	0.8	0.936
	U2	0.96	0.2	
WS_B	U1	0.97	0.7	0.949
	U2	0.9	0.3	

For simplicity, we assume the attributes group assessment on *respond time* and *successful completion* is the average value of all attributes group for the searching service. The above results are then treated as reputation assessment of the searching service.

In the example, we can see that only the user's assessments with similarity scores beyond the threshold are taken into account in assessing reputation of services. WS_B's reputation is higher than that of WS_A.

3 Optimizing Service Selection

3.1 Elementary Service Selection

QoS Attributes of Elementary Services We consider five generic quality criteria for services. For each criterion, we provide a definition and rules to compute its value for a given service.

(1) Execution price. Given a service s, the execution price $Q_{pr}(s)$is the fee that a service requester has to pay for invoking the service s.

(2) Response time. The response time $Q_n(s)$ measures the expected duration in seconds between a request and the corresponding response. The response time is computed by the expression $Q_{rt}(s)$=$T_{process}(s)$+$T_{trans}(s)$, meaning that the response time is the sum of the processing time $T_{process}(s)$ and the transmission time $T_{trans}(s)$. The transmission time is estimated from history of the service execution, the formula $T_{trans}(s)$=$\frac{\sum_{i=1}^{n} T_i(s)}{n}$ can used to compute it, where $T_i(s)$ is a past observation of

the transmission time, and n is the number of execution times observed in the past.

(3) Reputation. The reputation $Q_{rep}(s)$ of a service is a measurement of its trustworthiness. It mainly depends on the end user's experiences on service. We can derive more accurate and meaningful reputation assessment on Web services based on the definitions from 3 to 6 which present the reputation rating method driven by user's expectation.

(4) Successful completion. The successful completion $Q_{sc}(s,k)$ of a service s is the probability that a request is correctly responded, the value of the successful completion come from data of past invocations using the expression $Q_{sc}(s,k)$=$N_c(s)/k$, where $N_c(s)$ is the number of times that the service s has been successfully completed within the maximum expected time frame, and k is the total number of invocations.

(5) Availability. The availability $Q_{av}(s,\theta)$ of a service s is the probability that the service is accessible. The value of the availability of a service s is computed using the following expression $Q_{av}(s,\theta)$=$T_a(s)/\theta$, where $T_a(s)$ is the total amount of time in which service s is available during the last time period θ. The value of θ may vary depending on a particular application.

Given the above considerations, the quality vector of a service s is defined as

Q(s)=$(Q_{pr}(s), Q_{rt}(s), Q_{rep}(s), Q_{sc}(s,k), Q_{av}(s,\theta))$

Aggregated *QoS* Rating for Elementary Services By applying Multiple Criteria Decision Making (*MCDM*) theory and SAW (simple Additive Weighting) technique, we merge multiple dimensions of *QoS* into a whole that can calculate entire value for *QoS* of Web service [9, 10].

Given a task t_k in a composite service, there is a set of candidate Web services S_k=$\{s_{k1}, s_{k2}, \ldots, s_{kn}\}$ that can be used to execute this task. By merging the quality vectors of all these candidate Web services, a matrix Q=$Q_{i,j}(1 \le i \le n, 1 \le j \le 5)$is built, where n is the number of candidate Web services and j is the number of quality dimensions, in which each row Q_i corresponds to a candidate Web service $s_{k,i}$ while each column corresponds to a quality dimension. There are two phases in calculating the entire *QoS* value of an elementary Web service.

(1)Scaling Phase

Some criteria could be negative, i.e., the higher the value, the lower the quality. This includes criteria such as execution time and execution price. Other criteria are positive, i.e., the higher the value, the higher the quality. For negative criteria, values are scaled according to (9), and for positive criteria, values are scaled according to (10).

$$V_{ij} = \begin{cases} \frac{Q_j^{max} - Q_{i,j}}{Q_j^{max} - Q_j^{min}}: & if\ Q_j^{max} - Q_j^{min} \neq 0 \\ 1: & if\ Q_j^{max} - Q_j^{min} = 0 \end{cases} \tag{9}$$

$$V_{ij} = \begin{cases} \frac{Q_{i,j} - Q_j^{min}}{Q_j^{max} - Q_j^{min}}: & if\ Q_j^{max} - Q_j^{min} \neq 0 \\ 1: & if\ Q_j^{max} - Q_j^{min} = 0 \end{cases} \tag{10}$$

In the above formulae, Q_j^{max} is the maximal value of a quality criterion in matrix Q, i.e., Q_j^{max}=$Max\{Q_{i,j} : 1 \le i \le n\}$. While Q_j^{min} is the minimal value of a quality criteria in matrix Q, i.e., Q_j^{min}=$Min\{Q_{i,j} : 1 \le i \le n\}$. By applying formulae (9) and (10) on Q, we obtain matrix V=$(V_{i,j}; 1 \le i \le n, 1 \le j \le 5)$in which each row V_i corresponds to a candidate Web service $s_{k,i}$ while each column corresponds to a quality dimension.

After scaling phase the value of each element in V is in [0, 1].

(2) Weighting Phase

After scaling phase, the end users can express their preferences regarding to *QoS* by providing different weights for different quality dimensions. The following formula is used to compute the overall quality score for each candidate Web service:

$$Score(s_i) = \sum_{j=1}^{5} (V_{i,j} * W_j) \tag{11}$$

where $W_j \in [0,1]$ and $\sum_{j=1}^{5}$=1. W_j represents the weight of criterion j.

Finally, the system will choose the Web service with maximal score.

3.2 Service Selection by Global Planning

QoS Attributes of Composite Services For each task t_j in a composite service, there is a set of candidate services that execute task t_j. Assigning a candidate service to each task t_j in a composite service leads to a possible execution plan. In the global planning approach, all possible plans are generated and the one which maximizes the user's preferences while satisfying the imposed constraints is then selected.

The quality criteria, defined in terms of elementary Web services, are also applicable to composite cases. The following *QoS* descriptions provide aggregation functions for the computation of the *QoS* of a composite service when executed using plan p=$\{< t_1, S_1 >, < t_1, S_1 >, \ldots, < t_n, S_n >\}$.

(1) Execution price. The execution price $Q_{pr}(p)$ of an execution plan p is the sum of execution prices of the services involved, which is computed by $Q_{pr}(p)=\sum_{i=1}^{n} Q_{pr}(s_i)$.

(2) Response time. The response time $Q_{rt}(p)$ is computed by using the Critical Path Algorithm (*CPA*). The response time is computed by the expression $Q_n(p) = CPA(p, Q_n)$.

(3) Reputation. The reputation $Q_{rep}(p)$ is the average of the reputations of the services that participate in p, i.e. $Q_{rep}(p)=\frac{1}{n}\sum_{i=1}^{n} Q_{rep}(s_i)$.

(4) Successful completion. The successful completion $Q_{sc}(p)$ is the product of the factors $Q_{sc}(s_i)^{z_i}$, where z_i is equal to 1 if service s_i is a critical service in the execution plan p, or 0 otherwise. The formula used to compute successful completion is $Q_{sc}(p)=\prod_{i=1}^{n}(Q_{sc}(s_i)^{z_i})$.

(5) Availability. The availability $Q_{av}(p)$ is given by the product of the factors $Q_{av}(s_i)^{Z_i}$, where $Q_{av}(s_i)$ is the availability of service s_i and z_i indicates whether the service is a critical service or not. The formula used to compute availability is $Q_{av}(p)=\prod_{i=1}^{n}(Q_{av}(s_i)^{z_i})$.

Given the above definitions, the quality vector of a composite service's execution plan is defined as

Q(p)=$(Q_{pr}(p), Q_{rt}(p), Q_{rep}(p), Q_{sc}(p), Q_{av}(p))$

Aggregated *QoS* Rating for Composite Service The selection of an execution plan also relies on the application of *MCDM* technique. A quality matrix $QP=(QP_{i,j}; 1 \leq i \leq n, 1 \leq j \leq 5)$ is built the same way as the local selection approach. In this matrix, a row corresponds to the quality vector of a possible execution plan p. The SAW technique is used to select an optimal execution plan. The two phases of applying *SAW* are:

(1) Scaling Phase

We first scale the values of each quality criterion. In order to compute the maximum execution price of all the execution plans, we select the most expensive Web service for each task and sum up all these execution prices. We select the service with the shortest response time for each task and use *CPA* to compute the minimum response time of all the execution plans. After the scaling phase, we obtain another matrix $VP=(VP_{i,j}; 1 \leq i \leq n, 1 \leq j \leq 5)$

(2) Weighting Phase

The following formula is used to compute the overall quality score for each execution plan p_i:

$$Score(p_i) = \sum_{j=1}^{5} VP_{i,j}(W_j) \tag{12}$$

where $W_j \in [0, 1]$ and $\sum_{j=1}^{5}=1$, W_j represents the weight of criterion j.

End users can give their preferences on *QoS* to select a desired execution plan by adjusting the value of W_j. The execution plan with maximal *Score*(p_i) will be chosen.

Table 5: Quality of candidate services

Task	t_1			t_2			
CS	RT (ms)	Price	Repu.	CS	RT (ms)	Price	Repu.
s_{11}	125	27.5	0.94	s_{21}	710	7.4	0.89
s_{12}	120	27.7	0.73	s_{22}	810	9.3	0.83
s_{13}	600	8.5	0.85	s_{23}	188	29	0.93
s_{14}	240	16	0.9	s_{24}	591	8.7	0.75
s_{15}	380	12	0.78	s_{25}	390	8.3	0.92
s_{16}	308	15	0.87	s_{26}	560	8.1	0.84
s_{17}	830	8.2	0.95	s_{27}	780	7.8	0.9
s_{18}	540	9	0.82	s_{28}	794	6.7	0.86
s_{19}	430	9.1	0.89	s_{29}	770	8.9	0.79

4 Experiments and Analysis

4.1 Case Study

Suppose composite service contains 2 tasks, and each task has 9 candidate services (*CS*). Service's reputation assessment is calculated by the method proposed in Section 2. Suppose quality dimensions that user care about are response time (*RT*), price and reputation. The response time, price and reputation of candidate services for each task are listed in Table 5. Firstly, according to the local optimization method described in Section 3.1, the quality matrixes for the two tasks can then be built as Q_1 and Q_2. Scaling the quality matrixes based on formulae (9) to (10), we can derive quality matrixes as V_1 and V_2.

$$Q_1 = \begin{bmatrix} Q_{rt} & Q_{pr} & Q_{rep} \\ 125 & 27.5 & 0.94 \\ 120 & 27.7 & 0.73 \\ 600 & 8.5 & 0.85 \\ 240 & 16 & 0.9 \\ 380 & 12 & 0.78 \\ 308 & 15 & 0.87 \\ 830 & 8.2 & 0.95 \\ 540 & 9 & 0.82 \\ 430 & 9.1 & 0.89 \end{bmatrix} \quad Q_2 = \begin{bmatrix} Q_{rt} & Q_{pr} & Q_{rep} \\ 710 & 7.4 & 0.89 \\ 810 & 9.3 & 0.83 \\ 188 & 29 & 0.93 \\ 591 & 8.7 & 0.75 \\ 390 & 8.3 & 0.92 \\ 560 & 8.1 & 0.84 \\ 780 & 7.8 & 0.9 \\ 794 & 6.7 & 0.86 \\ 770 & 8.9 & 0.79 \end{bmatrix}$$

According to preference of end users, we assign different weights to three quality dimensions as W=(0.5, 0.3, 0.2). Finally, the overall *QoS* scores are derived to be (0.688, 0.5, 0.564, 0.749, 0.604, 0.693, 0.5, 0.575, 0.711) for t_1

ISBN: 978-988-17012-2-0

$$V_1 = \begin{bmatrix} Q_{rt} & Q_{pr} & Q_{rep} \\ 0.99 & 0.01 & 0.95 \\ 1 & 0 & 0 \\ 0.32 & 0.98 & 0.55 \\ 0.83 & 0.6 & 0.77 \\ 0.63 & 0.81 & 0.23 \\ 0.74 & 0.65 & 0.64 \\ 0 & 1 & 1 \\ 0.41 & 0.96 & 0.41 \\ 0.56 & 0.95 & 0.73 \end{bmatrix} \qquad V_2 = \begin{bmatrix} Q_{rt} & Q_{pr} & Q_{rep} \\ 0.16 & 0.97 & 0.78 \\ 0 & 0.88 & 0.44 \\ 1 & 0 & 1 \\ 0.35 & 0.91 & 0 \\ 0.68 & 0.93 & 0.94 \\ 0.4 & 0.94 & 0.5 \\ 0.05 & 0.95 & 0.83 \\ 0.03 & 1 & 0.61 \\ 0.06 & 0.9 & 0.22 \end{bmatrix}$$

and (0.527, 0.352, 0.7, 0.448, 0.807, 0.582, 0.476, 0.437, 0.344) for t_2 by using formula (11).

According to the global optimization method described in Section 3.2, we can derive service selection result. For simplicity we suppose the two tasks execute in sequence and only 16 possibilities of overall 81 are presented in the quality matrix QP. After scaling the quality matrix QP, we can derive another quality matrix VP as

$$QP = \begin{bmatrix} Q_{rt} & Q_{pr} & Q_{rep} \\ 835 & 34.9 & 0.92 \\ 935 & 36.8 & 0.89 \\ 313 & 56.5 & 0.94 \\ 716 & 36.2 & 0.85 \\ 830 & 35.1 & 0.81 \\ 930 & 37 & 0.78 \\ 308 & 56.7 & 0.83 \\ 711 & 36.4 & 0.74 \\ 1310 & 15.9 & 0.87 \\ 1410 & 17.8 & 0.84 \\ 788 & 37.5 & 0.89 \\ 1191 & 17.2 & 0.8 \\ 950 & 23.4 & 0.9 \\ 1050 & 25.3 & 0.87 \\ 428 & 45 & 0.92 \\ 831 & 24.7 & 0.83 \end{bmatrix} \qquad VP = \begin{bmatrix} Q_{rt} & Q_{pr} & Q_{rep} \\ 0.52 & 0.53 & 0.9 \\ 0.43 & 0.49 & 0.75 \\ 0.99 & 0.005 & 1 \\ 0.63 & 0.5 & 0.55 \\ 0.53 & 0.53 & 0.35 \\ 0.44 & 0.48 & 0.2 \\ 1 & 0 & 0.45 \\ 0.63 & 0.5 & 0 \\ 0.09 & 1 & 0.65 \\ 0 & 0.95 & 0.5 \\ 0.56 & 0.47 & 0.75 \\ 0.2 & 0.97 & 0.3 \\ 0.42 & 0.82 & 0.8 \\ 0.33 & 0.77 & 0.65 \\ 0.89 & 0.29 & 0.9 \\ 0.53 & 0.78 & 0.45 \end{bmatrix}$$

We assign different weights to the three quality dimensions again. The overall QoS scores for each plan are derived to be (0.599,0.512, 0.697, 0.575, 0.494, 0.404, 0.59, 0.465, 0.385, 0.571, 0.451,0.616, 0.526, 0.712, 0.589) by using formula (11). Finally, the plan with highest QoS score will be selected.

4.2 Result Analysis

We consider the context alongside the expectation of users who used the services. Users are assigned different weights according to user's degree and reputation, so users' influence on overall assessment of QoS is different. Based on it, we can derive a more accurate and meaningful measure for reputation of service. Therefore, overall QoS assessment methods for both elementary service and composite service are derived.

5 Conclusion

Service orientation computing is currently a hot research area in Internet application. The ability to measure quality of service objectively and accurately is critical for SOC. When rating reputation of service, we consider the context alongside the expectation of services' user. Based on similarity score between the expectation of user who used service and the expectation set by SR, only the ratings with similar expectation have influence on the aggregate rating for reputation of service. Therefore, we can derive a more objective and accurate QoS measurement of Web services. Moreover, an optimized service selection method driven by user's QoS expectation is proposed. It selects optimal service for Web service composition which satisfies user's appetite and expectation.

References

[1] Zeng, L. Z., Benatallah, B., et al, "QoS-aware middleware for Web services composition, solutions, and directions," *IEEE Transaction on Software Engineering*, V30, N5, pp. 311-327, 2004

[2] Schillo, M., Funk P., Rovatsos, M., "Using trust for detecting deceitful agents in artificial societies," *Applied Artificial Intelligence, Special Issue on Trust, Deception and Fraud in Agent Societies*, pp. 825-848, 2000

[3] Yu B., Singh, M., "An evidential model of distributed reputation management," *Proceedings of First International Conference on Autonomous Agents and Multi Agent System*, pp. 294-301, 2002

[4] Deora, V., Shao, J., Gray,W. A., et al., "A quality of service management framework based on user expectations," *Proceedings of the First International Conference on Service Oriented Computing*, pp. 104-114, 2003

[5] Roozbeh, F., Uwe, G., Mona, V., "Expectation-based quality of service assessment," *International Journal on Digital Libraries*, V30, N5, pp. 311-327, 2006

[6] Deora, V., Shao, J., Gray,W. A., et al., "Modelling quality of service in service oriented computing," *Proceedings of the Second IEEE International Symposium on Service-Oriented System Engineering*, 2006

[7] Liu, D. M., Shao, Z. Q., "An extended QoS model based on fuzzy set in service-oriented computing," *Computer Sciences*, 2008

[8] Liu, D. M., Shao, Z. Q., "A Optimized service selection driven by user's expectation on Qo," *Journal of East China University of Science and Technology (Natural Science Edition)*, 2008

[9] Chen, T., *Decision Making and Analysis*, Science Press, 1987.

[10] Hwang, C. L., Yoon, K., "Multiple attribute decision making and applications," *Lecture Notes in Economics and Mathematical Systems*, 1981

ISBN: 978-988-17012-2-0