2010 10th IEEE International Conference on Computer and Information Technology (CIT 2010)

QoS Assessment over Multiple Attributes

Hmood Al-Dossari, Jianhua Shao, Alun Preece
School of Computer Science and Informatics
Cardiff University
Cardiff CF24 3AA, UK
{H.Z.AL-Dossari,J.Shao,A.D.Preece}@cs.cardiff.ac.uk

Abstract

In an open service oriented computing environment, multiple providers may offer functionally identical services but with varying qualities. It is desirable therefore that we are able to assess the quality of a service (QoS), so that service consumers can be given additional guidance in selecting their preferred services. Various methods have been proposed to assess QoS using the data collected from monitoring tools, but they do not deal with multiple QoS attributes adequately. Typically these methods assume that the quality of a service may be assessed by first assessing the quality level delivered by each QoS attribute individually, and then aggregating them in some way to give an overall verdict for the service. In this paper, we show that this may lead to incorrect assessment, and suggest how existing methods may be improved to deal with multiple attributes more effectively.

1 Introduction

In an open service oriented computing environment, multiple providers may exist to offer functionally identical services but with varying qualities. To help service consumers select their preferred services, it is desirable that we are able to assess the quality of a service (QoS) accurately, so that consumers are given additional guidance.

In this paper, we consider the problem of QoS assessment, which may be broadly defined as attempting to predict, using the historical service provision data, the likely QoS level that a consumer may get from a service provider. Various methods have been proposed that use service performance data collected from monitoring tools to assess QoS [6, 9, 13]. Typically these methods assume that the quality of a service may be assessed by first assessing the quality level delivered by each QoS attribute individually, and then aggregating them in some way to give an overall verdict for the service. The following example illustrates this.

Suppose that we have a service S (e.g. a web hosting service) with two QoS attributes A_1 (e.g. access delay which is the round trip time between sending a request and receiving a response) and A_2 (e.g. failure rate which is the number of failures occurring per unit time). Suppose also that some historical data about the performance of S w.r.t. A_1 and A_2 have been collected as shown in Table 1. Each tuple in Table 1 represents a single service provision instance from S to a specific consumer, identified by SID, and $d(A_1)$ and $d(A_2)$ are the monitored quality levels delivered by A_1 and A_2, respectively, for that instance. For simplicity we assume that $d(A_1)$ and $d(A_2)$ are normalised into $[0, 1]$ with 0 representing the minimum level of quality and 1 the maximum.

Table 1: Monitored QoS Data for S

SID	$d(A_1)$	$d(A_2)$
si001	0.30	0.81
si002	0.27	0.72
si003	0.75	0.41
si004	0.37	0.80
si005	0.77	0.43
si006	0.32	0.78
si007	0.29	0.83
si008	0.71	0.38
si009	0.67	0.47
si010	0.25	0.87

To predict the likely QoS for S, we first calculate the quality level for A_1 and A_2 individually by averaging the observed data, i.e. $Avg(d(A_1)) = 0.47$ and $Avg(d(A_2)) = 0.65$, and then the two averages are aggregated to give $QoS(S) = 0.5 \times 0.47 + 0.5 \times 0.65 = 0.56$, assuming that the two attributes are equally important.

This type of calculation of QoS is meaningful if we assume that every level of quality is deliverable by a

978-0-7695-4108-2/10 $26.00 © 2010 IEEE
DOI 10.1109/CIT.2010.260

service. When this is not the case, then such methods can generate misleading verdicts. To see this, we re-arrange the order of tuples (service provision instances) in Table 1 to give us Table 2 below:

Table 2: Re-arranged QoS Data for S

SID	$d(A_1)$	$d(A_2)$
si001	0.30	0.81
si002	0.27	0.72
si004	0.37	0.80
si006	0.32	0.78
si007	0.29	0.83
si010	0.25	0.87
si003	0.75	0.41
si005	0.77	0.43
si008	0.71	0.38
si009	0.67	0.47

Now we can observe two groups of service provision instances in the table: one group with A_1 offered at around 0.3 and A_2 at around 0.8, and the other with A_1 delivered at around 0.7 and A_2 at 0.4. Let us assume that this grouping is not accidental, and S is in fact offered by the provider with these as the only two possible quality levels, perhaps as a result of some resource management requirement [9], then our calculation of $QoS(A_1) = 0.47$ and $QoS(A_2) = 0.65$ or $QoS(S) = 0.56$ is clearly misleading as it is not possible to get in practice.

So in more complicated service provision scenarios such as the one outlined above where some grouping of quality levels exist across multiple attributes, calculating QoS data for each attribute individually first can result in incorrect assessment. Current QoS assessment methods do not deal with such scenarios adequately. In this paper, we address this issue. We show that a simple solution to this problem exists, but it is unlikely to perform well in practice due to the way QoS data is typically collected from monitoring tools, and we discuss how it may be improved.

The remainder of the paper is organised as follows. We first discuss the problem of QoS assessment in general in Section 2. In Section 3 we explain the shortcoming of existing approaches in dealing with multiple attributes. A simple extension to existing methods is given, and we explain why such a simple solution is unlikely to do well in practice and discuss how it may be improved. We report some experimental results in Section 4 to demonstrate the issues we discuss in this paper. Finally in Section 5 we conclude the paper.

2 QoS Assessment Process

To discuss how effective QoS assessment methods may be developed, it is useful to understand what is involved in the process of QoS assessment in general first. As outlined in Figure 1, we consider a QoS assessment process involving four fundamental tasks: data collection, data selection, data aggregation and service ranking.

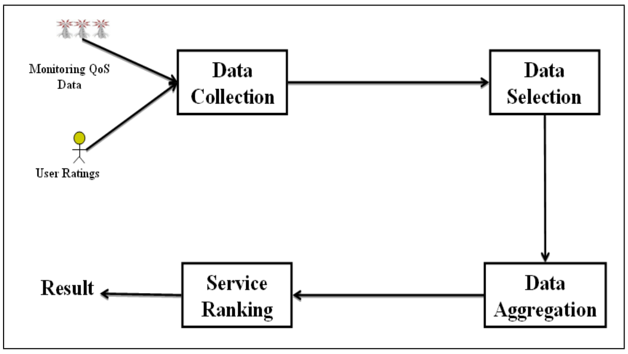


Figure 1: QoS Assessment Process

Data Collection

This is about what and how data relevant to the quality of a service should be obtained. Broadly, there are two types of QoS data, user ratings and monitored QoS data, that may be used for assessment. User ratings are collected from service consumers. They reflect users' view of service quality and thus are subjective in nature [2, 5, 11]. Monitored QoS data, on the other hand, is collected using automated tools and hence is more objective [6, 9, 13]. Existing approaches also differ in terms of what they assume about the collected data. For example, some assume that the collected data is trusted [2, 9], some assume that QoS attributes are independent of each other and QoS data for multiple attributes may be collected synchronously w.r.t. time [6, 13], and some assume that additional contextual information, such as user expectations, is also collected [2, 8, 9]. These assumptions determine, to a large extent, the power and applicability of a specific assessment method.

In our work, we use monitored QoS data as we are interested in automated processes. We do not assume that QoS data for multiple attributes is synchronously collected as it is unrealistic and unlikely that this will be the case in practice [10]. We do not assume that QoS data in different attributes is independent of each other either, because it is quite possible that some quality patterns or groupings will exist, as our example in Introduction has shown.

Data Selection

Not all collected QoS data may be relevant to a particular assessment request, for example, when quality of a service is offered at different levels. Thus we must determine which data should be selected for use in assessment. Previous studies have proposed different mechanisms. A simple one is to consider all data as relevant [6, 13]. More advanced techniques use various heuristics. For example, Deora et al. [2] introduced expectation-based selection, where consumers are asked to state expected QoS levels as part of their assessment request, and only the data that has similar expectation to those stated in the request will be selected and used. This approach was also adopted in [8, 9]. For rating based assessment, mechanisms for detecting and removing unfair ratings have been considered [11] and collaborative filtering, a more semantic heuristic, has also been used to determine the relevance of collected data to assessment requests [12]. In this paper, we concentrate on expectation-based data selection.

Data Aggregation

Selected QoS data must then be aggregated to give a QoS level that the service provider is likely to deliver. Various methods may be used, depending on the type of QoS data involved. For example, beta probability density functions may be used to aggregate binary data representing satisfied/unsatisfied service provisions [5, 11], simple or weighted averages can be used to summarise numerical QoS data into a single verdict [2, 6, 12], and forgetting or damping factors can be employed to help discount past performances [5]. For multiple attributes, however, existing works largely assume that the data in each attribute may be aggregated individually, which can lead to incorrect assessment [8]. In this paper we discuss how this may be improved.

Service Ranking

If the goal of QoS assessment is to help a consumer to choose a preferred one among those functionally identical but quality-wise varying services, then it is essential that we are able to rank a set of services somehow at the end of assessment. One obvious approach is to attempt to deliver a single numeric verdict for each service under assessment, and then rank the services under assessment based on their numerical order [6, 13]. Unfortunately, it is not always desirable or possible to derive a single verdict, for example, when the quality of each attribute must be considered and compared separately. In such cases, more sophisticated solutions based on multiple criteria decision making principles [3] must be considered. The issue of how services may be ranked is however beyond the scope of this paper.

3 Assessment of Multiple Attributes

In this section, we discuss the limitations of current approaches when used to assess QoS involving multiple attributes, and we suggest how such limitations may be overcome. We will first define what we mean by quality, and then explain the relevant issues through two representative methods.

3.1 A QoS Model

A number of definitions of quality are possible and in our study we adopt a conformance view of quality [2]. Let $S(A_1, A_2, \ldots, A_m)$ be a service where each $A_i, 1 \leq i \leq m$, is a QoS attribute. Suppose that the service provider is required to deliver S to a consumer with $\{e(A_1) = \alpha_1, e(A_2) = \alpha_2, \ldots, e(A_m) = \alpha_m\}$, where $e(A_i) = \alpha_i \in [0, 1]$ represents the quality expected by the consumer, possibly as part of a service level agreement. Now we suppose that during the service delivery we monitored the following $\{d(A_1) = \beta_1, d(A_2) = \beta_2, \ldots, d(A_m) = \beta_m\}$, where each $d(A_i) = \beta_i$ is the actual quality level delivered to the consumer. We define quality for a single attribute to be the difference between the expected and delivered values:

$$QoS(A_i) = |d(A_i) - e(A_i)|$$

where both $d(A_i)$ and $e(A_i)$ are normalised values in $[0, 1]$[1]. The smaller the difference, the higher the quality.

Now we assume that a set of past service provision performance data has been collected, and each instance in the dataset is recorded as $\langle s_k, e(A_i), d(A_i)\rangle$ in a QoS database, where s_k is a service instance identifier, $e(A_i)$ the expected quality of A_i and $d(A_i)$ the delivered. The problem of QoS assessment can then be described as follows: given the content of a QoS database and assuming that s_k is functionally adequate, determine how likely s_k will meet or conform to a consumer quality requirement γ using the observed past performances.

[1] We note that in practice, questions of how QoS data may be monitored and normalised may not be straightforward to answer [10]. In this study, however, we do not consider such issues, but simply assume that the data has already been monitored and normalised.

3.2 Averaging All

To calculate QoS for S, the simplest method is to average all the delivered quality values observed for each attribute of S first, and then average the aggregated values across attributes [6, 13]. This method was already introduced in Section 1, but is described here again for completion and ease of references. That is, we calculate QoS as follows:

$$QoS(S) = \begin{cases} \sum_{j=1}^{m}(w_j \times \sum_{i=1}^{n} \frac{\beta_{ij}}{n}) & \text{if } n > 0 \\ default & n = 0 \end{cases}$$

where m is the number of attributes, n the number observed instances of S in the database, β_{ij} the observed delivered quality value for A_j in the i-th instance, and w_j a weight indicating the significant of each attribute in aggregation. The closer the calculated $QoS(S)$ is to the γ requested by a consumer, the higher the quality of S is considered to be for that consumer when no previous provisioning instance has been observed, a default value will be returned. For example, if we apply this method to Table 1 given in Section 1 and assuming that A_1 and A_2 are equally important, we have $QoS(S) = 0.56$. If this is close to what a consumer expects (e.g. $\gamma = 0.5$), then this service is considered to offer good quality.

This method works fine if we assume that any level of quality may be offered by each of the attributes. When this is not the case, for example, when A_1 is actually offered at two distinct levels at around 0.3 and 0.7 , then $QoS(A_1) = 0.47$ is unrealistic to get and the overall prediction of $QoS(S) = 0.56$ is unlikely to materialise in practice[2].

3.3 Using Expectations

In [9] a multiple quality-space mapping (MQSM) approach is proposed to provide a more accurate assessment when a service provides multiple levels of quality. This approach attempts to classify service provision instances into clusters before aggregating delivered values, and it works as follows. First, service instances whose delivered values satisfy $|\beta_{ij} - \gamma_j| \leq \delta$ are selected from the database, where γ_j is the consumer's expectation on A_j and δ denotes a bound intended to capture similar values. Then for the selected instances we find minimum and maximum expectation values, α_{min} and α_{max}, and we visit the database again to retrieve a new set of instances whose expectation values satisfy $\alpha_{min} \leq \alpha_{ij} \leq \alpha_{max}$, where α_{ij} denotes the expectation value for A_j in the i-th instance. Finally, we average the corresponding β_{ij} values in this set of instances to obtain a prediction for A_i.

To illustrate how this method works, we expand Table 1 with expectation values and new data is shown in Table 3. Suppose that a consumer has the following expectation: $\gamma_1 = 0.8$ and $\gamma_2 = 0.9$ and we are asked to assess how likely S will meet this expectation. Assuming $\delta = 0.1$, MQSM first selects instances from Table 3 for A_1 based on $|d(A_1) - 0.8| \leq 0.1$ which gives us $\{si003, si005, si008\}$. From this set, we have $\alpha_{min} = 0.65$ and $\alpha_{max} = 0.85$. We then retrieve a new set of instances based on $0.65 \leq e(A_i) \leq 0.85$ which gives us $\{si003, si005, si008, si009\}$. Finally, we aggregate the delivered values in this set to find $QoS(A_1) = 0.73$. Quality for A_2 is similarly assessed and we have $QoS(A_2) = 0.81$. Assuming the two attributes are equally important, the overall prediction for S is computed to be $QoS(S) = 0.77$.

Table 3: Historical QoS Data with Expectations

SID	$\langle e(A_1), d(A_1) \rangle$	$\langle e(A_2), d(A_2) \rangle$
si001	$\langle 0.27, 0.30 \rangle$	$\langle 0.92, 0.81 \rangle$
si002	$\langle 0.31, 0.27 \rangle$	$\langle 0.85, 0.72 \rangle$
si003	$\langle 0.80, 0.75 \rangle$	$\langle 0.35, 0.41 \rangle$
si004	$\langle 0.40, 0.37 \rangle$	$\langle 0.85, 0.80 \rangle$
si005	$\langle 0.85, 0.77 \rangle$	$\langle 0.40, 0.43 \rangle$
si006	$\langle 0.36, 0.32 \rangle$	$\langle 0.75, 0.78 \rangle$
si007	$\langle 0.25, 0.29 \rangle$	$\langle 0.80, 0.83 \rangle$
si008	$\langle 0.65, 0.71 \rangle$	$\langle 0.21, 0.38 \rangle$
si009	$\langle 0.77, 0.67 \rangle$	$\langle 0.28, 0.47 \rangle$
si010	$\langle 0.19, 0.25 \rangle$	$\langle 0.87, 0.87 \rangle$

If we consider the accuracy of the assessment of a single attribute, MQSM performs better than the Averaging-All approach as both $QoS(A_1) = 0.73$ and $QoS(A_2) = 0.81$ are clearly meaningful. However, if we assume that quality is offered with some "packaging" across multiple attributes, e.g. one package offers around 0.3 for A_1 and around 0.8 for A_2, and another offers around 0.7 for A_1 and around 0.4 for A_2, then the predicted combination $QoS(A_1) = 0.73$ and $QoS(A_2) = 0.81$ is clearly unattainable and $QoS(S) = 0.77$ is misleading.

[2] We note that using average as a prediction is rather incomplete, as it is unlikely that a future provision of the service will have a quality at the indicated average level exactly. For a more complete prediction, confidence interval surrounding the predicted (average) value can be calculated.

3.4 Dealing with Multiple Attributes

To deal with multiple attributes correctly, we can apply a simple "adjustment" to the approaches discussed above. Instead of selecting and aggregating $d(A_1)$ and $d(A_2)$ separately first in Table 3 and then combining $QoS(A_1)$ and $QoS(A_2)$ into a single verdict, we can select individual instances based on both γ_1 and γ_2 first, and then aggregate the qualified $d(A_1)$ and $d(A_2)$ into a single prediction for S. That is, we perform

$$QoS(S) = \begin{cases} \sum_{i=1}^{k}(\sum_{j=1}^{m} w_j \times \beta'_{ij})/k & \text{if } k > 0 \\ default & k = 0 \end{cases}$$

where k is the number of instances that satisfy $|\beta'_{i1} - \gamma_1| \leq \delta, |\beta'_{i2} - \gamma_2| \leq \delta, \ldots, |\beta'_{im} - \gamma_m| \leq \delta$ simultaneously. Applying this to our running example, it is easy to verify that no instances satisfy $|d(A_1) - 0.8| \leq 0.1$ and $|d(A_2) - 0.9| \leq 0.1$ simultaneously, hence a default result will be reported. This verdict is clearly more accurate.

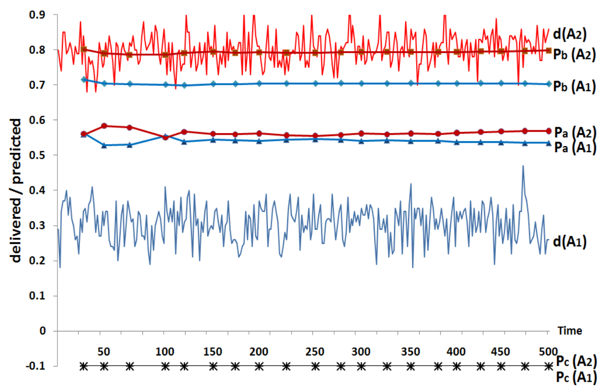
The problem with this simple adjustment is that it implicitly assumes that QoS data for the multiple attributes involved are synchronously collected, as we require $|\beta'_{i1} - \gamma_1| \leq \delta, |\beta'_{i2} - \gamma_2| \leq \delta, \ldots, |\beta'_{im} - \gamma_m| \leq \delta$ to be simultaneously satisfied. This is a rather restrictive assumption which is unlikely to be held in practice, because multiple QoS attributes are more likely to be monitored independently at different time points and at different rates [10]. If we allow QoS data to be collected asynchronously, then we can anticipate that the number of instances in the database that satisfy our required condition will be significantly reduced, particularly when we have a large number of attributes. This in turn can seriously reduce the confidence of assessment [7].

4 Experiments

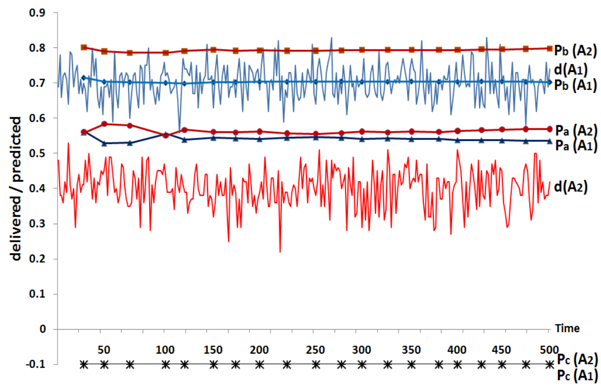
We have carried out some simulation to examine the performance of the methods discussed in Section 3 for assessing QoS over multiple attributes. Our test data was generated to simulate a single service S consisting of two attributes, A_1 and A_2, and offering two quality packages, `P1` with $d(A_1) \in [0.2, 0.4]$ and $d(A_2) \in [0.7, 0.9]$, and `P2` with $d(A_1) \in [0.6, 0.8]$ and $d(A_2) \in [0.3, 0.5]$. `P1` is offered to consumers with expectation $e(A_1) \leq 0.5$ and $e(A_2) \geq 0.6$, and `P2` for consumers with $e(A_1) > 0.5$ and $e(A_2) < 0.6$.

First, we considered prediction accuracy of the three methods (Averaging-All, MQSM, and our Simple Extension) we discussed in the previous section for assessing QoS over multiple attributes. We randomly generated 500 tuples in the form of $\langle e(A_1), d(A_1), e(A_2), d(A_2)\rangle$ in our experiments, each representing a past service instance and satisfying the conditions given above, and the values were normally distributed within each quality package. We then assumed that S was to be assessed for a specific consumer request: $e(A_1) = 0.8$ and $e(A_2) = 0.9$, i.e. how likely S will meet this requirement.

To observe the effect of different data sizes on assessment, we repeated our quality calculation for every addition of 25 service instances using the three methods. The result is shown in Figure 2, where $d(A_1)$ ($d(A_2)$) marks the actual quality delivered by S for A_1 (A_2), $p_a(A_1)$ ($p_a(A_2)$) marks the quality predicted by Averaging-All, $p_b(A_1)$ ($p_b(A_2)$) marks the quality predicted by MQSM, and $p_c(A_1)$ ($p_c(A_2)$) marks the quality predicted by our Simple Extension, respectively. As can be seen, the Averaging-All approach predicted qualities for A_1 and A_2 which are far from their real behaviours, i.e. the predicted values ($p_a(A_1)$ and $p_a(A_2)$) do not overlap with the delivered values ($d(A_1)$ and $d(A_2)$) at all in both packages. This error



(a) Prediction vs Quality Package `P1`



(b) Prediction vs Quality Package `P2`

Figure 2: Assessment for $e(A_1) = 0.8$ and $e(A_2) = 0.9$

was caused by the fact that there existed two levels of QoS for each attribute in our scenario and each level was only available to consumer requests in certain ranges, but the method wrongly aggregated them together in assessment.

The MQSM approach worked better and correctly predicted the quality for A_2 ($p_b(A_2)$) in `P1` and for A_1 ($p_b(A_1)$) in `P2`. However, it failed to predict the quality for the other attribute correctly in the respective package. This is because MQSM assessed two attributes individually, and then mistakenly paired $p_b(A_2)$ in `P1` with $p_b(A_1)$ in `P2` suggesting that the requested $e(A_1) = 0.8$ and $e(A_2) = 0.9$ could be served by S at around $p_b(A_1) = 0.7$ and $p_b(A_2) = 0.8$, despite the fact that this *combination* would not be offered by S. Our Simple Extension, on the other hand, correctly identified the fact that the required quality level is unlikely to be met by S, and correctly returned a default value of -0.1 in all assessment, suggesting that no verdict could be reached.

5 Conclusions

In this paper, we analysed the problems associated with QoS assessment over multiple attributes. We explained the shortcomings of existing QoS assessment approaches and have shown that a simple extension can be made, but it works well only when data is assumed to be synchronously collected.

There are a number of possible directions for future research to address this issue. One possible approach is to relax the definition of synchronisation. That is, instead of using the most strict form of synchronisation where one seeks exact matching of time stamps, we may allow monitored QoS data to be synchronised within a “time window”. This method is commonly used in data stream applications [1]. Another approach is to treat asynchronous values as missing values and we attempt to predict such missing values as commonly exercised in data mining [4]. We plan to investigate these methods in our future work.

Acknowledgments

The first author is sponsored by King Saud University, Kingdom of Saudi Arabia.

References

[1] B. Babcock, S. Babu, M. Datar, R. Motwani and J. Widom. Models and Issues in Data Stream Systems. In *21st ACM Symposium on Principles of Database Systems*, pp 1-16, 2002.

[2] V. Deora, J. Shao, W. A. Gray and N. J. Fiddian. A Quality of Service Management Framework Based on User Expectations. In *1st International Conference on Service Oriented Computing*, pp 104-114, 2003.

[3] J. Figueira, S. Greco and M. Ehrgott. Multiple Criteria Decision Analysis: State of the Art Surveys. Springer Verlag, 2005.

[4] J. W. Grzymala-Busse and M. Hu. A Comparison of Several Approaches to Missing Attribute Values in Data Mining. In *Rough Sets and Current Trends in Computing*, pp 378-385, 2001.

[5] A. Jøsang and R. Ismail. The Beta Reputation System. In *15th Bled eCommerce Conference*, pp 324-337, 2002.

[6] Y. Liu, A. H. H. Ng and L. Zeng. QoS Computation and Policing in Dynamic Web Service Selection. In *13th International World Wide Web Conference*, pp 66-73, 2004.

[7] J.S. Milton and J. C. Arnold. Introduction to Probability and Statistics: Principles and Applications for Engineering and Computing Sciences. McGraw-Hill Higher Education, 2002.

[8] W. Sherchan, S. Krishnaswamy and S. W. Loke. Relevant Past Performance for Selecting Web Services. In *5th International Conference on Quality Software*, pp 439-445, 2005.

[9] G. Shercliff, J. Shao, W. A. Gray and N. J. Fiddian. A Multiple Quality-Space Mapping Approach to QoS Assessment. In *6th International Conference on Computer and Information Technology*, pp 26-31, 2006.

[10] P. Stockreisser, J. Shao, W. A. Gray and N. J. Fiddian. Supporting QoS Monitoring in Virtual Organisations. In *4th International Conference on Service Oriented Computing*, pp 447-452, 2006.

[11] J. Zhang and R. Cohen. A Personalized Approach to Address Unfair Ratings in Multiagent Reputation Systems. In *5th Workshop on Trust in Agent Societies* , pp 89-98, 2006.

[12] Z. Zheng, H. Ma, M. R. Lyu and I. King. WS-Rec: A Collaborative Filtering Based Web Service Recommender System. In *7th International Conference on Web Services*, pp 437-444, 2009.

[13] C. Zhou, L. T. Chia, B. Silverajan and B. S. Lee. UX- An Architecture Providing QoS-Aware and Federated Support for UDDI. In *1st International Conference on Web Services*, pp 171-176, 2003.