A Fuzzy Partial Ordering Approach for QoS-based Selection of Web Services

Dunlu Peng Qingkui Chen Huan Huo
School of Optical-Electrical and Computer Engineering,
University of Shanghai for Science and Technology, Shanghai, 200093, China
Email: dunlu.peng@gmail.com, chenqingk@tom.com, huoh@usst.edu.cn

***Abstract*—As the development of Service-Oriented Computing (SOC), more and more functional similar Web services are deployed over the Internet. Nowadays, Web service selection becomes a crucial issue for making SOC more applicable. Troublesome Web services will affect the reliability of the whole SOC application which invokes the service. Therefore, when choosing Web services, not only the functional attributes but also the Quality of Service (QoS) should also be considered. In this work, we develop a framework to collect the indices of service quality, including runtime and non-runtime indices. A fuzzy partial-ordering approach, which takes both quality indices and their uncertainties into consideration, is proposed to evaluate Web services. A series of partial-ordering models has been developed to rank Web services according to their qualities. Case study shows that the proposed approach is effective for selecting the service having the highest synthetic quality from the collections of functional similar services.**

***Index Terms*—Web service selection, QoS, fuzzy partial ordering, ranking approach**

I. INTRODUCTION

The objective of service selection is to choose the most relevant services that best meet consumers' requirements from the services collection. The paradigm of Service-Orient Computing (SOC) makes Web service enable businesses and organizations to collaborate in an unprecedented way. However, wide application of SOC brings service selection more and more challenging because of the rapid increasing number of functional similar Web services being made available on the Web.

In the Web service architecture currently in use, each incoming Web service was manually assigned to a pre-specified category by its provider at the registry center. If a consumer wants to find a service fulfilling some special requirements, he/she needs to browse the "right" categories. Unfortunately, for the lack of online evaluation approach, this category-based service discovery cannot guarantee the quality of selected services. This makes selecting and ranking Web services in terms of the Quality of Services (QoS) play an essential role in SOC architectures, especially when the semantic matchmaking process returns lots of services with similar functionalities.

In order to select the most superior Web services, there is a need to be able to distinguish the functional similar Web services using a set of well-defined QoS indices. Therefore, in this work, we define a series of indices for the measurement of QoS and catch the indices with an extension of tradition service architecture. The indices are classified into two classes: non-runtime indices, such as *price*, and runtime indices, such as *availability*, *mean response time*, *trustworthy* and *performance*. For the uncertainty of QoS, we present a fuzzy partial-ordering approach to evaluate services for ranking by taking overall consideration of QoS. Some service evaluation models are developed based on this approach.

The remainder of this paper is arranged as follows. Section II provides a short survey of related work. Section III presents the outline of the theory of fuzzy partial order in the context of Web services. Section IV describes the approach and framework exploited to obtain the quality indices. The case study of using our approach is given in Section V. Finally, the conclusion remarks are described in Section VI. For the sake of conciseness, we use "service(s)" to stand for "Web service(s)" in following parts.

II. RELATED WORK

Various types of matching can be exploited to judge how well a service satisfies a consumer's requirements. Signature matching is a way to matching two services only based on their function types without regarding their behavior [5]. Specification matching compares two services based on their behaviors description [6]. Syntactic matching uses syntax driven techniques to evaluate the similarities into data [4]. Semantic matching is to map the meaning of data [7]. These keyword-based approaches help the consumers to select services from different aspects. In [17], the authors present a probabilistic semantic approach for finding services. However, for the lack of the exact specifications, developing and testing methodologies, the above methods cannot assure they do not select the troublesome service which might affect the whole SOC application. Therefore, developing an effective QoS-based service selection technique is a crucial task for developing reliable SOC applications.

Currently, most quality-based service selection approaches reported in the literature addressed some dimensions which were advertised by the service providers, such as price, response time, availability, reliability and etc [16], and discussed how to promote the accuracy of these dimensions through optimizing service developing methods. A QoS standard and its editorial environment were proposed in [10]. Chen etc in [11] proposed a QoS normalizing approach and ranked functional similar services with the synthetical QoS. Finally, the service at the top rank was chosen for the

doi:10.4304/jsw.5.4.405-412

consumers. Based on [11], a QoS-based service selection has been studied for service composition in [14]. A Web services discovery model with QoS constraint has been developed in [15]. In that model, a set of QoS indices was defined, and service providers were allowed to advertise QoS information when they publish the services. Generally, the service with highest reputation was returned to users.

Obviously, for some of the dimensions, such as response time, it is undependable to merely rely on service providers' advertisement, because service providers are not in a "neutral" manner. This kind of dimensions has some uncertainty, that is, they might change with different invocation. Therefore, in a service selection model, it is profitable to take the indices of QoS and their uncertainty into consideration. However, in existing QoS-based service selection methods, few of them took both into account. Different from existing approaches, in our work, we employ a fuzzy partial-ordering approach to evaluate how a service satisfies the consumers' requirements according to both the values of the quality indices and their uncertainty, and rank the services with similar functions according to their partial-ordering relation.

III. Fuzzy Partial Ordering in Web Service Context

Evaluation generally means the behavior of specifying the objective measuring entity's attributes, and turning them into subjective. In other words, evaluation is the process of determining the value of entity by measuring its relevant attributes and the relationship among them [3].

A. Evaluation Models of Web Services

The evaluation patterns are different with different features and forms of evaluating entity. There are 2 typical evaluation models in service context.

Definition 3.1 A service value evaluation model is a triple (W, Q, F), where $W = \{w_1, w_2, ..., w_m\}$ is the set of services with similar functions, w_i is the i-th evaluating service; $Q = \{q_1, q_2, ..., q_n\}$ is the set of indices measuring the quality of services, q_j is the j-th quality index. $F = \{f_l : W \to V_l(l \le n)\}$ is the set of relations between services and the quality indices. Here, $f_l(w_i)$ is the measured value of service w_i on the quality index q_l, and V_l is the possible values for quality index q_l, called q_l's domain. If $f_l(l \le n)$ is in form of numerical value, the model is called a **service cardinal value evaluation model**; and if $f_l(l \le n)$ is in form of preferable position, the model is called a **service ordinal value evaluation model**.

Definition 3.2 A service relation evaluation model is a duality (W, R), where $W = \{w_1, w_2,..., w_m\}$ is the set of services to be evaluated, R is the pairwise relation between the members of W and can be expressed as a matrix as follows:

$$R = \begin{pmatrix} R(w_1,w_1) & R(w_1,w_2), & \ldots & R(w_1,w_m) \\ R(w_2,w_1) & R(w_2,w_2) & \ldots & R(w_2,w_m) \\ \ldots & \ldots & \ldots & \ldots \\ R(w_m,w_1) & R(w_m,w_2) & \ldots & R(w_m,w_m) \end{pmatrix} \quad (1)$$

where $R(w_i,w_j)$ represents some superior or inferior relation between service w_i and service w_j. Corresponding to the value evaluation model, if $R(w_i,w_j)$ is in form of preferable relation, the model is called an **ordinal service relation evaluation model**. Otherwise, if $R(w_i,w_j)$ is in form of numerical value, the model is called a **cardinal service relation valuation model**.

Notice that there is substantial difference between service value evaluation model and service relation evaluation mode. Service value evaluation model consists of the set of services under evaluation, the set of quality indices and the set of quality indices values in form of data table. Whereas, service relation evaluation model consists of the set of services being evaluated and bi-relation between services, which is in form of relation matrix. For service value evaluation model, information is integrated by means of building up service's quality indices values; for building up service relation evaluation model, information is integrated by means of building up services' bi-relation. Since multiple quality indices or services are involved, any service evaluation model is a kind of partial order. Information fusion is to turn partial order into total order, and get the superior and inferior order of the objects under evaluation.

The two kinds of service evaluation models can be mutually transformed. Let (W, Q, F) be a service value evaluation model, and

$$F = \{f_l : W \to V_l(l \le n)\} \quad (2a)$$

$$V = \{v = (v_1, v_2, ..., v_n) \mid v_l \to V_l(l \le n)\} \quad (2b)$$

then, $(V,\le)$ is partial order and denoted as

$$F(w_i) = (f_1(w_i), f_2(w_i), ..., f_m(w_i)) \quad (3a)$$

$$R(w_i,w_j) = \begin{cases} \mathbf{f}, & F(w_i) > F(w_j), \\ \mathbf{p}, & F(w_i) < F(w_j), \\ \approx, & F(w_i) = F(w_j), \\ ?, & others\ . \end{cases} \quad (3b)$$

Thus, the ordinal service relation evaluation model is generated.

Example 3.1 Let $W=\{w_1,w_2,w_3,w_4,w_5,w_6\}$ represent six services having similar functions. The ordinal service relation evaluation model is given as follows:

$$R(w_i,w_j) = \begin{pmatrix} \approx & \mathbf{f} & \mathbf{f} & \mathbf{f} & ? & \mathbf{p} \\ \mathbf{p} & \approx & \mathbf{p} & ? & ? & \mathbf{p} \\ \mathbf{p} & \mathbf{f} & \approx & \mathbf{f} & \mathbf{f} & \mathbf{p} \\ \mathbf{p} & \mathbf{f} & ? & \approx & ? & \mathbf{p} \\ ? & ? & \mathbf{p} & ? & \approx & \mathbf{p} \\ \mathbf{f} & \mathbf{f} & \mathbf{f} & \mathbf{f} & \mathbf{f} & \approx \end{pmatrix}$$

in above model, $R(w_i,w_j)$ is generated from the comparison between w_i and w_j and used to decide which is the superior one. For example, $R(w_3,w_2)$ = **f** represents w_3 is superior to w_2; $R(w_5, w_3)$ = **p** means that w_5 is inferior to w_3; $R(w_1,w_5)$ = ? indicates that w_1 and w_5 is incomparable and $R(w_i,w_i) = \approx (i \leq 6)$ shows that a service is an equivalent to itself.

Similarly, we can also present the cardinal service relation valuation model for above 6 services, i. e:

$$\begin{pmatrix} 0.50 & 1.00 & 1.00 & 1.00 & 0.85 & 0.00 \\ 0.00 & 0.50 & 0.00 & 0.85 & 0.85 & 0.00 \\ 0.00 & 1.00 & 0.50 & 1.00 & 1.00 & 0.00 \\ 0.00 & 1.00 & 0.85 & 0.50 & 0.85 & 0.00 \\ 0.85 & 0.85 & 0.00 & 0.85 & 0.50 & 0.00 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 0.50 \end{pmatrix}$$

in which $R(w_1,w_2)$=1.00 denotes w_1 is superior to w_2; $R(w_4,w_3)$=0.85 represents the degree of w_4 superior to w_3 is 0.85; $R(w_5,w_6)$=0.00 means w_5 is inferior to w_6. $R(w_i,w_i)=0.5(i \leq 6)$ shows that a service is equivalent to itself.

Service relation evaluation model can be transformed to service value evaluation model. As in service relation evaluation model, mutual relation is given between services, such relation should be quantified in information integration, and then get the complete compared relation. For a group of individual service evaluating relation given by multiple quality indices, after integration the model will be transformed to service value evaluation model.

B. Unifying Service Indices

In the service value evaluation model, the relations between evaluating services and service indices are different. Thus, before transforming, the type of service indices should be unified.

Let $f(w_i)$, $f(w_j)$, $f(w_k)$ be the values of quality indices for evaluating services w_i, w_j and w_k, respectively. For $w_i \geq w_k \geq w_j$, if

$$f(w_k) \geq f(w_i) \wedge f(w_j) \quad (4)$$

holds, f is called **an up-convex attribute function**; if

$$f(w_k) \leq f(w_i) \vee f(w_j) \quad (5)$$

holds, f is called **a down-convex attribute function**.

Given a quality indices q_l, services are ranked from the superior to inferior order:

$$w_{i_1} \underline{\mathbf{f}} w_{i_2} \underline{\mathbf{f}} w_{i_m} \underline{\mathbf{f}} ... \underline{\mathbf{f}} w_{i_m} \quad (6)$$

The domain of valuating value of service quality index q_l, i.e, $f_l(w)(l$ £$n)$, is V_l = [0, 1].

Let service quality index q_l be a continuous up-convex function, the quality index can be classified as follows:

(1) For any $p < r < q$, if $f_l(w_{i_q}) \leq f_l(w_{i_r}) \leq f_l(w_{i_p})$ holds, q_l is called a benefit-type service quality index.

(2) If for any $p < r < q$, $f_l(w_{i_p}) \leq f_l(w_{i_r}) \leq f_l(w_{i_q})$ holds, q_l is called a service cost-type quality index.

(3) If $1 < p < q < m$, $f_l(w_{i_p}) = f_l(w_{i_q})$ and

$$f_l(w_{i_p}) \geq f_l(w_{i_u})(1 \leq u \leq p),$$
$$f_l(w_i) = f_l(w_{i_p})$$
$$f_l(w_{i_q}) \geq f_l(w_{i_v})(q \leq v \leq m)$$

holds, q_l is called a **ranged-type quality index.**.

Similarly, we can also define the aforementioned types of service quality indices for any down-convex continuous function. We declare that the benefit-type service quality indices of up-convex function are the cost-type service quality indices of down-convex function, and the cost-type service quality indices of up-convex function are the benefit-type service quality indices of down-convex function.

The three types of service quality indices for up-convex continuous function can be transformed to benefit-type service quality indices with following approach. For convenience, we assume V_l = [0, 1](l £m).

(1) No change is given to the benefit-type service quality indices, i.e,

$$f_l^{'}(w_i) = f_l(w_i) \quad (7)$$

(2) For any cost-type service quality index, we define

$$f_l^{'}(w_i) = 1 - f_l(w_i) \quad (8)$$

(3) For any ranged-type service quality index, let

$$f_l^{'} = \begin{cases} \dfrac{f_l(w_i)}{2}, & w_i \leq w_{i_p}, \\ \dfrac{1 - f_l(w_i) + f_l(x_{i_p})}{2}, & w_i > w_{i_p}. \end{cases} \quad (9)$$

With this approach, the domain of a service quality index is kept in [0,1] and a ranged-type service quality index is transformed to a benefit-type serviced quality index.

For any service quality index q_l, if $w_i \leq w_j$ and $f_l^{'}(w_i) \leq f_l^{'}(w_j)$ hold, transform all the service quality indices to benefit-type service quality indices. Therefore, the benefit-type service quality indices can be used to evaluation decision. The down-convex continuous function can be transformed in a similar way.

C. Evaluation Decision for Web Services

For a service value evaluation model (W, Q, F), if there are multiple service quality indices, the evaluating value of a service w_i is denoted as $F(w_i) = (f_1(w_i), f_2(w_i), \ldots, f_n(w_i))(i \leq m)$. Thus, each evaluating service is represented as an n-dimension vector, and there are m vectors under comparison. If $F(w_i) \leq F(w_j)$ stands for $w_i \underline{\mathbf{p}} w_j$, then $(W, \underline{\mathbf{p}})$ is a **quasi-order set.** For a quasi-order set $(W, \underline{\mathbf{p}})$, we employ $R = \{(w_i, w_j) \mid w_i \underline{\mathbf{p}} w_j\}$ to represent that w_i is inferior to w_j , and $R^{-1} = \{(w_i, w_j) \mid w_j \prec w_i\}$ to represent that w_i is superior to w_j . Thus, $R \cap R^{-1} = \{(w_i, w_j) \mid w_i \approx w_j\}$ represents the relation set in which w_i and w_j are equivalent. $\sim (R \cap R^{-1}) = \{(w_i, w_j) \mid (w_i, w_j) \approx ?\}$ represents the relation set that w_i and w_j are incomparable.

Definition 3.4 Let $(W, \prec)$ be a quasi-order set and the evaluation result $(W, \leq)$ be a set in total order. If for any $(w_i,w_j) \in R$, $w_i \leq w_j$ holds, and for any $(w_i,w_j) \in R^{-1}$, w_j£ w_i holds, then $(W, \leq)$ is a **reasonable evaluation**. For any w_i, $w_j \in W$, if $(w_i, w_j) \in R$ and $(w_i, w_j) \notin R \cap R^{-1}$, $w_i < w_j$ holds, and for any $(w_i, w_j) \in R^{-1}$ and $(w_i, w_j) \notin R \cap R^{-1}$, $w_i < w_j$ holds, then, $(W, \leq)$ is a **strict reasonable evaluation.**

Theorem 3.1 Assume $(W, \leq)$ to be a partial-order set, $(W, \leq)$ has following properties:

(1) $R \cap R^{-1}$ is an equivalence relation;

(2) $\sim(R \cup R^{-1})$ has symmetry and has no reflexivity and transitivity;

(3) R, R^{-1} has reflexivity and transitivity, but has no symmetry.

Proof: Theorem 3.1 can be proofed directly using above description.

Theorem 3.2 Let $(W, \prec)$ be a partial-order set, and there exists a total ordering set on W which makes total order set $(W, \leq)$ a strict reasonable evaluation.

Proof: Denote that

$$R(w_i, w_j)\begin{cases} 1 \quad (w_i, w_j) \in R^{-1}, (w_i, w_j) \notin R \cap R^{-1}, \\ d \quad d \in (0.5,1), (w_i, w_j) \in R \cup R^{-1} \ , \\ 0.5 \quad (w_i, w_j) \in R \cap R^{-1}, \\ 0 \quad (w_i, w_j) \in R, (w_i, w_j) \notin R \cap R^{-1}. \end{cases} \tag{10}$$

if $w_i > w_j$, then

1) when $w_i \succ w_j$, $w_i > w_j$ holds;

2) when $w_k ? w_j$, $w_i ? w_j$ or $w_i > w_j$ holds;

3) when $w_k \prec w_j$, $w_i < w_j$ or $w_i > w_j$ or $w_i ? w_j$ holds.

Therefore, for $R(w_i) = \sum_{j \neq i} R(w_i, w_j)$ when $w_i \succ w_k$, $R(w_i) > R(w_k)$ holds. If W is ordered by $R(w_i)$'s value, a strict reasonable evaluation will be obtained.

D. Fuzzy partial-ordering models for service evaluation decision

Although traditional evaluation relation model gives a partial-order bi-relation of services, it cannot catch the uncertainty of the service quality which leads to the model cannot catch the uncertainty of relation among services. Here, we apply a fuzzy partial ordering evaluation model which is an appropriate approach to capture the uncertainty of the relation among the evaluating services.

Model 3.1 Let (W, £) be a partial-order set, denote that

$$R(w_i, w_j) = \begin{cases} 1.0, & w_i > w_j, \\ 0.0, & w_i < w_j, \\ 0.5, & w_i \approx w_j, \\ 0.8, & w_i ? w_j. \end{cases} \tag{11}$$

Then, $R = R(w_i,w_j)$, where w_i, w_j Î Q, is a fuzzy strict partial-order relation.

Proof: We only need to prove that when $w_i \geq w_j$, $R(w_i, w_k) \geq R(w_j, w_k)$ holds. Three scenarios are needed to be considered: i) $w_i \geq w_j$; ii) $w_i \approx w_j$; iii) $w_j ? w_k$. When $w_j > w_k$, $w_i > w_k$ holds, for $w_i \geq w_j$; when w_j » w_k, w_i » w_k or $w_i > w_k$ holds; when $w_j ? w_k$, $w_i ? w_k$ or $w_i > w_k$ holds. In the above scenarios, $R(w_i, w_k)$, $R(w_j, w_k)$ holds. Similarly, we can prove when $w_i \geq w_j$, $R(w_k, w_j) \leq R(w_k, w_i)$ holds.

Model 3.2 Let (W, £) be a partial-order set, denote that

$$R(w_i, w_j) = \frac{|\{w_k \mid w_k \leq w_i\}|}{|\{w_k \mid w_k \leq w_i\}| + |\{w_k \mid w_k \leq w_j\}|} \tag{12}$$

where (w_i, w_j, $w_k \in W$) and $|\cdot|$ is the number of elements in the set, then, $R(w_i, w_j)$ is a fuzzy partial order relation.

Proof: It is obvious that $0 \leq R(w_i,w_j) \leq 1$ holds. Meanwhile, when w_i » w_j , $R(w_i, w_j) = 0.5$ holds; when $w_i \geq w_j$, then $R(w_i, w_j) \geq 0.5$ holds; when $w_i < w_j$, then $R(w_i, w_j) \leq 0.5$, $R(w_i, w_j) + R(w_j, w_i) = 1$ holds; when $w_i \geq w_j$, then $R(w_i, w_j) \geq R(w_j, w_i)$. Let us prove that when $w_i \geq w_j$, then $R(w_i, w_l) \geq R(w_j, w_l)$ holds as follows. Because

$$R(w_i, w_l) = \frac{|\{w_k \mid w_k \leq w_i\}|}{|\{w_k \mid w_k \leq w_i\}| + |\{w_k \mid w_k \leq w_l\}|} \tag{13a}$$

$$R(w_j, w_l) = \frac{|\{w_k \mid w_k \leq w_j\}|}{|\{w_k \mid w_k \leq w_j\}| + |\{w_k \mid w_k \leq w_l\}|} \tag{13b}$$

as we know when $w_i \geq w_j$, $|\{w_k|w_k \leq w_i\}| \geq |\{x_k|x_k \leq w_j\}|$ holds. According to the inequation $\frac{a}{a+c} \geq \frac{b}{b+c}$, $R(w_i,w_l) \geq R(w_j ,w_l)$ holds.

Model 3.3 Deem (W, Q, F) be an evaluation model, $F = \{f_l : W \to [0,1] (l \leq m)\}$,

$$R(w_i, w_j) = \frac{|\{q_l \mid f_l(w_i) \geq f_l(w_j)\}|}{n} \tag{14}$$

where n is the number of elements in Q (the set of indices of service quality), and q_l is a member of Q. Then, $R(w_i, w_j)$ is a fuzzy partial-set relation.

Proof: Obviously, $0 \leq R(w_i, w_j) \leq 1$ holds. We know that $F(w_i) = (f_1(w_i), f_2(w_i), ..., f_n(w_i))$ and denote $F(w_j) \leq F(w_i) \Leftrightarrow f_l(w_j) \leq f_l(w_i) (l \leq n)$. When $F(w_j) \leq F(w_i)$, $\{q_l|f_l(w_k) \leq f_l(w_j)\} \subseteq \{q_l|f_l(w_k) \leq f_l(w_i)\}$, then, $R(w_i, w_k)$ ³ $R(w_j , w_k)$

holds. Similarly, when $\{q_l|f_l(w_k) \geq f_l(w_j)\} \supseteq \{q_l|f_l(w_k) \geq f_l(w_i)\}$, then $R(w_k, w_i) \leq R(w_k, w_j)$ holds.

Model 3.4 Let $(W,\leq\cdot)$ be a partial-order set, $[w_i]^< = \{w_j| w_i\leq w_j\}$ represents the class that is superior to w_i, then the superior relation

$$R^<(w_i, w_j) = \frac{|\sim [w_i]^< \cup [w_j]^<|}{|W|} \quad (15)$$

is a fuzzy partial-order relation on $(W, \leq)$. Similarly, $[w_i]^> = \{w_j|w_i\leq w_j\}$ stands for the class inferior to w_i and it is transferred into the superior relation as

$$R^>(w_i, w_j) = \frac{|\sim [w_i]^> \cup [w_j]^>|}{|W|} \quad (16)$$

Proof: If $w_i\leq w_j$, then $[w_i]^< \subseteq [w_j]^<$, thus, $\sim [w_i]^< \cup [w_j]^< = W$, therefore, $R^<(w_i, w_j) = 1$. Here, we get $\sim [w_i]^< \cup [w_k]^< \supseteq \sim [w_j]^< \cup [w_k]^<$, therefore, $R^<(w_i,w_k)$, $R(w_j, w_k)$. Similarly, when $w_i\geq w_j \geq w_k$, $R^<(w_k,w_j) \geq R^<(w_k, w_i)$ holds. Therefore, model 3.4 (see equations 15 and 16) exist.

Model 3.5 For any continuous information system (***W***, ***Q***, ***F***),

$$\begin{aligned} R_l(w_i, w_j) &= 1 \wedge (1 - f_l(w_i) + f_l(w_j)), \\ R(w_i, w_j) &= \min_{l\leq n} R_l(w_i, w_j)(l \leq n) \\ R(w_i, w_j) &= \frac{1}{n}\sum_{l=1}^{n} R_l(w_i, w_j)(l \leq n) \end{aligned} \quad (17)$$

is a fuzzy partial-order relation on $(W, \leq\cdot)$.

Proof: For R_l is a fuzzy partial-order relation on W and therefore R is a fuzzy partial-order relation on W.

Model 3.6 For any continuous information system (*W*, *Q*, *F*), formula

$$R(w_i, w_j) = \frac{\sum_{l=1}^{n}(f_l(w_i) \wedge f_l(w_j))}{\sum_{l=1}^{n} f_l(w_j)} \quad (18)$$

is a fuzzy partial-order relation.

Proof: Firstly, $0 \leq R(w_i, w_j) \leq 1$ holds, when $w_i\geq w_j$, $f_l(w_i) \geq f_l(w_k)$, $f_l(w_i) \wedge f_l(w_j) = f_l(w_j)(l\leq n)$, then $R(w_i, w_j) = 1$. Here, $f_l(w_i)\wedge f_l(w_k) \geq f_l(w_j) \wedge f_l(w_k)(l\leq m)$, thus, $R(w_i,w_k) \geq R(w_j, w_k)$. Similarly, when $w_i \geq w_j\geq w_k$, then $R(w_k, w_j) \geq R(w_k, w_i)$ which means model 3.6 is a fuzzy partial-order relation.

E. From partial order to total order

Let (*W*, *Q*, *F*) be a continuous information system, where $F=\{f_l:W\rightarrow V_l(l\leq n)\}$, $V_l=[0,1]$ and $F(w_i)=(f_1(w_i), f_2(w_i),\ldots, f_n(w_i))$ $(w_i\in W)$. We introduce the following denotation:

$$w_i \geq w_j \Leftrightarrow F(w_i) \geq F(w_j) \geq f_l(w_i) \geq f_l(w_j) \ \ (l\leq n),$$
$$w_i \approx w_j \Leftrightarrow F(w_i)= F(w_j) \Leftrightarrow f_l(w_i)= f_l(w_j) \ \ (l\leq n)$$

then, $(W, \leq)$ is a partial-order set.

For any continuous information system, after we use Models 3.1~3.6 to establish a fuzzy partial-order relation, we can get the total order in a service context *W* using following formula:

$$R(w_i) = \sum_{j=1}^{n} R(w_i, w_j) \quad (19)$$

IV. Service Quality Indices and Their Collection

QoS is a combination of several qualities or properties of a service [1]. In recent years, the number of functional similar services being made available over the Internet is increasing rapidly. A set of well defined indices of QoS is needed to be able to distinguish the services. In this section, we firstly define the QoS quality indices used in our model, and then present a feasible approach to obtain them.

A. Service Quality Indices

A set of non-functional attributes can be employed to measure the quality of a service. Here, these attributes are called as service quality indices. We consider five quality indices which can be measured objectively for services: (1)*Price*, (2)*Availability*, (3)*Mean Response Time*, (4)*Trustworthy* and (5) *Performance*. For the sake of illustration, the number of QoS indices discussed in this paper is limited. However, our model is extensible. New indices can be added without fundamentally altering the underlying computation mechanism as shown in Section III.

Price: The *price* of a service is the money that a service consumer has to pay for requesting the service, such as checking a credit, getting a commodity and etc. Service providers either directly advertise the price of their services, or provide ways for potential consumers to query it. It may be charged per the number of service requests, or could be a flat rate charged for a period of time. Let w_i be one service, then $q_{i,1}$ is the price for requesting w_i.

Availability: The *availability* of a service is the probability that the service will be available at some period of time [2]. It is measured with the times of successfully invoking the service to the total times of invoking the service. The availability of a given service may very depending on a particular application; therefore, we use the mean value rather than an exact value of availability to stand for the QoS. We denote the availability of service w_i as $q_{i,2}$ in following sections.

Mean Response Time: The *response time* is defined with the time interval from a request arriving at the service to the instant the corresponding reply beginning to appear at the consumers' terminal. It is determined by two factors: the quality of network transmission, and the processing capacity of the service. The former depends largely on the network traffic, which makes the response time varies widely for different service request. The later may be a constant for a given service. Therefore, we use the mean response time standing for the response time for a service. Let w_i be one service, then $q_{i,3}$ is the mean response time for requesting w_i during the testing time.

Trustworthy: The *trustworthy* of a service can be measured with its reputation which mainly lies on the consumers' experiences of requesting the service. The opinion varies different among consumers on the same service. The value of the trustworthy is computed with the average ranking given to the service by consumers. For a service w_i, we use $q_{i,4}$ to represent its trustworthy.

Performance: Performance is the measure of the speed to complete a service request. It is measured by two metrics: *latency* and *throughput*. *Latency* is the delay between the arrival and completion of a service request and *throughput* is the number of requests completed over a period of time. We use $q_{i,5}$ and $q_{i,6}$ to represent the latency and throughout of service w_i respectively.

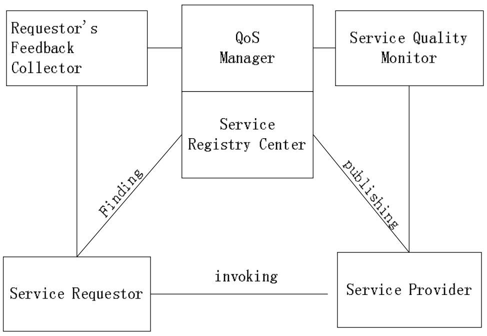

Figure 1. The extended Web service architecture

B. Collecting Service Quality Information

Many approaches have been proposed to collect service quality information [14], [9], [12], [13]. In our work, we classify service quality indices into two categories, non-runtime quality indices and runtime quality indices. The non-runtime quality indices, which keep constantly during the service being invoked, can be provided by service providers as they publish the service at the register center. On the contrary, the runtime quality indices, which vary with the circumstance in which the service is invoked, are measured at the runtime. Obviously, in the set of our service quality indices, *price* is a non-runtime quality index, and *availability*, *mean response time*, *trustworthy* and *performance* are the non-runtime quality indices of a service.

With the extension of the traditional service architecture, we make a mechanism for service providers advertise the non-runtime quality indices and collect the runtime quality indices from users' feedback. The framework shown in Figure 1 depicts our idea about the collection of service quality indices. From the figure, we can see besides the three roles: service providers, service requesters and service registry central, there are two additional ones, service quality monitor and requestor's feedback collector. There former three roles have the same names as in the traditional service architecture. However, in our framework, their responsibilities are different from in the traditional service architecture, that is, service providers are needed to advertise the values of non-runtime quality indices when they publish services, service requestors are required to give feedback on the runtime quality indices after they invoke the service, and the service registry center has to manage the quality indices besides the services themselves.

To make service providers publish the non-runtime quality indices conveniently, we embed a fragment tagged with "<non-runtimeQoS>…</non-runtimeQoS>" into the standard WSDL document of a service. In the fragment, each tag describes a non-runtime quality index for the service. For example, in Figure 2, the element tagged with "price" is to describe the value of price for service "TelephoneCall" at "http://example.com/telephonecall". With this extended WSDL, the service providers advertise their service non-runtime quality indices at the same time as they publish their service to the registry center (or UDDI center). At the registry center, the QoS fragment is extracted from the service description document and handed to the QoS manager to storage and manage it.

```
… …
<service name="TelephoneCall">
    <documentation>telephone call service</documentation>
    <port name="TelephoneCallPort" binding="tns:TelephoneCallBinding">
        <soap:addresslocation="http://example.com/telephonecall"/>
    </port>
    <non-runtimeQoS>
        <price metric="minute ">0.25$ </price>
    <non-runtimeQoS>
…
</service>
… …
```

Figure 2. An extended WSDL document for registering non-runtime service quality indices

Collecting the values of runtime indices is the responsibility of service quality monitor and requestors' feedback collector. Service quality monitor is assigned to monitor the runtime indices that can be observed at the service provider side when the service is invoking, such as *mean response time*, *performance* and *availability*. Requestor's feedback collector is to obtain the values of runtime indices whose evaluation needs service consumer's participation, such as *trustworthy* of a service.

V. Case Study

In this section, we examine the evaluation models with a set of QoS indices values which we obtained by running the prototype system of our proposed Web service architecture proposed in Section IV-B. The prototype system was written in Java and performed on 32 PCs and each of which has an Intel Core 2 Duo 1.8 GHz processor (2GB RAM) running Windows XP professional operating System. One of these PCs acted as the registry center which runs s requestor's feedback collector, service quality manager and QoS manager, and Six of them play the role of service providers and others act as the service requestors.

A. Experimental Data and its Normalization

To investigate the effectiveness of our proposed approach, we apply it to choose the most superior service from the set of six services with the same functions. The services are denoted as w_1, w_2, w_3, w_4, w_5 and w_6. Their raw values of quality indices are shown in Table I, where q_1, q_2, q_3, q_4, q_5 and q_6 represents the indices we discussed in Section IV-A respectively, that is, *price* (q_1), *availability* (q_2), *mean response time* (q_3), *trustworthy* (q_4), *latency* (q_5) and *throughput* (q_6).

Before applying our evaluation model, we normalize the quality indices whose raw values are not in the range [0,1] using following formula:

$$q'_{i,j} = \frac{q_{i,j} - \min(q_j)}{\max(q_i) - \min(q_j)} \quad (20)$$

for any benefit-type quality index, and

$$q'_{i,j} = 1 - \frac{q_{i,j} - \min(q_j)}{\max(q_i) - \min(q_j)} \quad (21)$$

for any cost-type quality index. The definitions of quality indices types are given in Section III. Using Equation 20, 21 and the index type transformation formulas (see Equation 7,8, 9), the quality indices are transferred into benefit-type quality indices and bounded in [0,1]. The normalized data are shown in Table II.

TABLE I.
SERVICES AND THE RAW VALUES OF THEIR QUALITY INDICES

services	q_1	q_2	q_3	q_4	q_5	q_6
w_1	220.00	0.80	120	0.91	80	800
w_2	187.00	0.50	150	0.95	85	752
w_3	120.00	0.71	135	0.83	90	900
w_4	230.00	0.65	101	0.76	60	600
w_5	135.00	0.31	90	0.70	52	502
w_6	210.00	0.98	70	0.98	70	700

TABLE II.
THE NORMALIZED VALUES OF SERVICE QUALITY INDICES

services	q_1	q_2	q_3	q_4	q_5	q_6
w_1	0.09	0.80	0.38	0.91	0.26	0.75
w_2	0.39	0.50	0..00	0.95	0.13	0.63
w_3	1.00	0.71	0.19	0.83	0.00	1.00
w_4	0.00	0.65	0.61	0.76	0.79	0.25
w_5	0.86	0.31	0.75	0.70	1.00	0.00
w_6	0.18	0.98	1.00	0.98	0.53	0.50

B. Ordering Services Using Partial-Ordering Models

Before choosing the services, we need to know which is the most superior one, that is, the services need to be ordered according to their quality. We use the models presented in Section III-D and Equation 19 (see Section III-E) to order the service.

Method 1: We get the following fuzzy partial-order relation using Model 3.1 (see formula 11):

$$\begin{pmatrix} 0.5 & 1.0 & 1.0 & 1.0 & 0.8 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.8 & 0.8 & 0.0 \\ 0.0 & 1.0 & 0.5 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 0.8 & 0.5 & 0.8 & 0.0 \\ 0.8 & 0.8 & 0.0 & 0.8 & 0.5 & 0.0 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 0.5 \end{pmatrix}$$

With formula 15, we get

$R(w_1) = 4.30$, $R(w_2) = 2.10$, $R(w_3) = 3.50$,
$R(w_4) = 3.10$, $R(w_5) = 2.90$, $R(w_6) = 5.50$.

Then, the total order is: $w_6 > w_1 > w_3 > w_4 > w_5 > w_2$.

Method 2: Using Model 3.2 (see formula 12) and formula 15, we get

$R(w_1) = 3.14$, $R(w_2) = 2.71$, $R(w_3) = 3.14$,
$R(w_4) = 2.58$, $R(w_5) = 2.83$, $R(w_6) = 3.60$.

Thus, the order of the services is $w_6 > w_1 \sim w_3 > w_5 > w_2 > w_4$.

Method 3: Using Model 3.3 (see formula 14), we get

$R(w_1) = 3.67$, $R(w_2) = 3.00$, $R(w_3) = 3.67$,
$R(w_4) = 3.00$, $R(w_5) = 3.17$, $R(w_6) = 4.50$.

Therefore, we order the services as follows $w_6 > w_1 \sim w_3 > w_5 > w_2 \sim w_4$.

Method 4: Using Model 3.4 (see formula 15 and formula 16), we get

$R^<(w_1) = 5.16$, $R^<(w_2) = 3.83$, $R^<(w_3) = 5.50$,
$R^<(w_4) = 4.67$, $R^<(w_5) = 4.83$, $R^<(w_6) = 6.00$,
$R^>(w_1) = 0.84$, $R^>(w_2) = 0.73$, $R^>(w_3) = 0.84$,
$R^>(w_4) = 0.67$, $R^>(w_5) =0.81$, $R^>(w_6) =2.50$.

Thus, the descending order of the services is $w_6 > w_3 > w_1 > w_5 > w_4 > w_2$ using $R^<(w_i)$, and $w_6 > w_1 \sim w_3 > w_5 > w_2 > w_4$ using $R^>(w_i)$.

Method 5: With Model 3.5 (see formula 17) and formula 16), we get

$R(w_1) = 4.71$, $R(w_2) = 4.32$, $R(w_3) = 4.80$,
$R(w_4) = 4.35$, $R(w_5) = 4.70$, $R(w_6) = 5.5$.

The services are ordered as follows: $w_6 > w_3 > w_1 > w_5 > w_4 > w_2$.

Method 6: With Model 3.6 (see formula 18), we get

$R(w_1) = 3.50$, $R(w_2) = 2.89$, $R(w_3) = 3.71$,
$R(w_4) = 2.97$, $R(w_5) = 3.47$ and $R(w_6) = 4.23$.

Therefore, we totally order the services as follows: $w_6 > w_3 > w_1 > w_5 > w_4 > w_2$.

According to the partial orders, we can decide the service ranked at the first is the most superior service and its synthetical quality is the highest. In our example, w_6 is the most satisfying service for the consumers.

In our example case, the indices are normalized to some value in [0,1], in some cases they might be given in form of an interval. We use following approach to process this case.

Let (W, Q, F) be an interval model, we denote

$$f_l(w_i) = [a_l(w_i), b_l(w_j)]$$

$$R_l(w_i,w_j) = \begin{cases} \frac{1}{2}\left(\frac{(a_l(w_i)-a_l(w_j))+(b_l(w_i)-b_l(w_j))}{|a_l(w_i)-a_l(w_j)|+|b_l(w_i)-b_l(w_j)|}+1\right) & f_l(w_i) \le f_l(w_j) \\ 1, & f_l(w_i) = f_l(w_j) \end{cases} \quad (22)$$

then the relations computed with following formula 23 and 24 are fuzzy partial order relation on fuzzy set $(W,\le)$.

$$R(w_i, w_j) = \min_{l \le n} R_l(w_i, w_j) \quad (23)$$

$$R(w_i, w_j) = \frac{1}{n}\sum_{l=1}^{n} R_l(w_i, w_j) \quad (24)$$

Similar to information integration in continuous systems, we can get information integration methods for interval type of fuzzy partial order relation.

VI. CONCLUSION

Large-scale SOC application becomes one of the key computing paradigms that enable the implementation of Internet-based integration of e-Science and e-Business at the global level. Service selection is an important issue for realizing SOC applications with high performance. Choosing services with high quality is the guarantee of developing efficient SOC applications.

In this paper, we divide the quality indices of services into two classes: run-time quality indices and non-runtime quality indices, and propose a framework to collect the quality indices through extension the traditional service architecture. In view of the uncertainty of QoS, a set of models based on fuzzy partial ordering is presented for service selection.

Compared with the existing QoS service selection methods, our method has better overall consideration on the uncertainty of service quality and the relations between services and their qualities. The example case study shows that the proposed approach is effective to select the most superior service from a functional similar service collection. In our future work, we try to develop a service selection and ranking approach in which we will take both functional and non-functional attributes of Web services into consideration.

ACKNOWLEDGEMENT

This work was partially supported by the National Hi-tech R&D Program of China under Grant No. 2006AA01Z103, the Innovation Program of Shanghai Municipal Education Commission under Grant No. 08YZ98 and the Nominated Excellent Youth-Teacher Program of Shanghai Higher Education under Grant No. SLG08012.

REFERENCES

[1] D. A. Menasc, "QoS Issues in Web Services", *IEEE Internet Computing*, 6(6), pp. 72-75, November/December 2002.

[2] A. Mani and A. Nagarajan, "Understanding quality of service for Web services", *http://www.106.ibm.com/ developerworks/library/ws-quality.html*, January 2002.

[3] Z. Wenxiu and Q. Guofang. *Uncertain Decision Making Based on Rough Sets* (*in Chinese*). Beijing: Tsinghua Publishing House, 2005.

[4] D. Xin, H. Alon, M. Jayant, N. Ema and Z. Jun, "Similarity Search for Web service". *In the Proceedings of the 30th International Conference on Very Large Database (VLDB2004)*, Toronto, Canada. Morgan Kaufmann, August 31 - September 3, 2004.

[5] Z. Hai Z and L. Jun, "Flexible retrieval of Web service". *The Journal of Systems and Software,* vol. 70, pp 106-117, 2004.

[6] W. Yiqiao and E. Stroulia, "Flexible Interface Matching for Web-Service Discovery". *In the Proceedings of the Fourth International Conference on Web Information Systems Engineering(WISE 2003)*, Rome, Italy. IEEE Computer Society, December 10-12, 2003,

[7] The OWL Services Coalition. "OWL-S specification version 1.0", *http://www.daml.org/services/*, November 2003.

[8] L. Yutu L, H.H.N Anne and Z. Liangzhao, "QoS Computation and Policing in Dynamic Web Service Selection". *In Proceedings of the 13th International World Wide Web Conference (WWW2004)*, New York, USA. ACM Press, May 17-20, 2004.

[9] Z. Liangzhao, B. Boualem, D. Marlon, K. Jayant and Z.S. Quan, "Quality Driven Web Services Composition". *In Proceedings of the 12th international conference on World Wide Web (WWW2003)*, Budapest, Hungary. ACM Press, May 2003.

[10] M. Tian, A. Gramm, H. Ritter and J. Schiller, "Efficient selection and monitoring of QoS-aware Web services with the WSQoS framework". *In the Proceedings of the IEEE International Conference on Web Intelligence* (*WI 2004*). Beijing, China. IEEE Computer Society, September 20-24, 2004.

[11] C. Hongan, Y. Tao, L. Kwei-Jay, "QCWS: An implementation of QoS-capable multimedia Web services". *In the Proceedings of the 5th International Symposium on Multimedia Software Engineering*(*MMSE'2003*). Taichung, Taiwan, China. IEEE Computer Society, December 10-12, 2003

[12] R. Shuping, A Model for Web Sevices Discovery with QoS. *ACM SIGecom Exchanges*, 4(1):1-10, 2003.

[13] S. Amit, C. Jorge, M. John and K. Krys, "Qos for service-oriented middleware". *In Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics* (*SCI2002*), Orlando, Florida, USA. July 14 - 18, 2002.

[14] L. Yutu, H.H.N. Anne and Z. Liangzhao, "QoS computation and policing in dynamic Web service selection". *In Proceedings of the 13th International Conference on World Wide Web* (*WWW 2004*). New York, USA. ACM Press, May 17-20, 2004.

[15] Y. Wensheng and C. Meilin, A model for Web service discovery with QoS constraints". *Chinese Journal of Computers* (*in Chinese with English abstract*). 28(4), pp 589- 594, 2005.

[16] D.P. Flavio, P. Matteo, C. Marco and M. Andrea, "A Meta-model for Non-functional Property Descriptions of Web Services". *in Proceedings of 2008 IEEE International Conference on Web Services* (*ICWS 2008*). Beijing, China. IEEE Computer Society, September 23-26, 2008.

[17] M. Jiangang, C. Jinli and Z. Yanchun, "A probabilistic semantic approach for discovering web services. *In Proceedings of the 16th International Conference on World Wide Web* (*WWW 2007*). Banff, Alberta, Canada. ACM Press, May 8-12, 2007.

Dunlu Peng is an associate professor of University of Shanghai for Science and Technology, Shanghai, China. He received his Ph.D degree from Fudan University, Shanghai, China in June 2006.He served as a PC member of SCC, EDOC, APSCC, WHICEB, CLOUD, etc. His research interests include Web applications, service-oriented computing, XML data management and Web mining.

Qingkui Chen is a professor and Ph.D supervisor of University of Shanghai for Science and Technology (USST), Shanghai, China. He is a senior member of China Computer Federation (CCF) and Professional Committee of Computer Support Cooperative Work of Shanghai Computer Federation in China. He also served as a program committee member of IFIP, NPC and the Technical Program Committee Co-Chairs of ICISE. His research interests include network computing, parallel computing, parallel database and computer network.

Huan Huo is a lecture of University of Shanghai for Science and Technology, Shanghai, China. She received her Ph.D degree from Northeastern University, Shenyang, China in September 2007. Her research interests include XML data management and stream data processing.