A Web Services Composition Model for QoS Global Optimization

Minghui Wu1,2, Xianghui Xiong1,2, Jing Ying1,2, Canghong Jin2, and Chunyan Yu3
1 Department of Computer Science and Engineering, Zhejiang University City College, Hangzhou, P.R. China
2 College of Computer Science and Technology, Zhejiang University, Hangzhou, P.R. China
3 College of Mathematics and Computer Science, Fuzhou University, P.R. China
Email: {minghuiwu}@cs.zju.edu.cn

Abstract—QoS driven selection approaches for Web Services Composition are used to choose the best solution among candidate services which have the same functions. This paper focuses on the Web services composition problem, and introduces the multi-dimension QoS model. Based on the QoS model, the QoS-driven web service selection model was proposed and described in details. According to these models, service composition problem can be considered as single-objective multi-constraints optimization problem. A brief discussion of approaches to solve the optimization problem is given.

Index Terms—SOA; web services composition; QoS; global optimization;

I. INTRODUCTION

Web services distributed in various locations can be integrated into a composite service with more powerful function. Through services composition, resources could be reused and we could implement a complicated functionality rapidly at lower cost.

A composite service is assembled by several tasks to accomplish a mission. In internet there are maybe many available web services with various QoS (Quality of Service) providing the same functionality specific to a task. So a selection needs to be made. More about QoS, you can refer to [4]. During the composition, there are demands for QoS constraints to be met and QoS criterions to optimize. Therefore web service composition has to search for an optimal set of services to construct a composite service and result in a best QoS, under user’s QoS constraint and basic functionality claim. And how to construct the web service composition model is the main research subject of the paper.

The remainder of this paper is organized as follows: the section 2 provides some related work of web services composition, and the section 3 gives the web services composition model. A particular depiction of the QoS-driven web service selection mathematical model and algorithm has been given in section 4. Finally, the last section concludes this paper and prospects the future.

II. RELATED WORKS

Some approaches of web service selection are based on semantic web [2, 3], others are basis of QoS attribute computing [4-6]. The former has difficulties in global QoS evaluation. At present there are many approaches base on QoS attribute computing. Zeng [4] proposes two ways, local optimization and global planning by using integer programming. Local optimization will obtain optimal candidate services in one abstract service scope without considering constraints across multiple abstract services and contributions to optimize global QoS criterions. Consequently, the binding of abstract services is independent with each other. Unless QoS of service is changed or unavailable, the service binding do not need to redo. The time consumed will be stable if we keep identical scale of service composition. However, unlike global planning which can get better global QoS, local optimization could not follow global criterion. Nevertheless, global optimization also has its defect: the time cost will grow proportionally with the amount of execution routes for the same scale of services. Yu [6] defines the service composition problem as a Multi-dimensions Multi-choice 0-1 knapsack problem (MMKP) in combinatorial model and a Multi-constraint optimal path (MOP) problem in graph model, and solves it by using integer programming. Li [7] proposes a mapping framework to construct Service Overlay Network (SON), and translates web service composition problem to single constraint path selection problem. In the end, Diijistra shortest path algorithm could give an optimal selection.

We will construct the web service composition model and demonstrate its mathematical model as single-objective multi-constraints optimization problem.

III. WEB SERVICES COMPOSITION MODEL

A. Basic Definition

Definition 1: (Abstract Service). Abstract service has function descriptions without implementation and standard service interfaces across different service providers. An abstract service is corresponding to a workflow task.

Definition 2: (Service Instance). Service instances are concrete services published by service providers. They could give the function implementation specified by abstract services. And some service instances may have the same function, but different QoS.

Definition 3: (Candidate Relationship). While the function of several service instances $S_1, ..., S_n$ are consisted with the function description of abstract service $T$, we state that service instances $S_1, ..., S_n$ and abstract service $T$ have the candidate relationship. We also say
that $S_1,\ldots,S_n$ is the candidate services of $T$ which is labeled as $S_i \in T$ \quad (i = 1 \ldots n).

**Definition 4:** (Service Function Graph). Constructing abstract services as a workflow to fulfill user’s requirement in functionality will obtain a service function graph. In the service function graph, we have two additional special abstract services treated as Start label and End label, which have no functional meaning. An example is given as Fig.2.

**Definition 5:** (Service Selection Graph). Shown as Fig.3, all the service instances corresponding to abstract services in Service Function Graph of Fig.2 have been discovered, and a service function graph appears. In the process of service discovering, we will discard service instances that do not meet local constraints. Consequently, local constraints are already met in service selection graph, and we will not take account of local constraints in the following discussion.

**Definition 6:** (Execution Route). In the service function graph, execution route is a passageway between start node and end node, and only include one spur track for every condition structure. If there are $k$ execution routes with a probability $\rho_i (i = 1 \ldots k)$ in a service function graph, then $\sum \rho_i = 1$.

In Fig.2, there are two execution routes:

- $ER_1: (S_1, S_2, S_3, S_4, S_5)$, with probability $\rho_1$
- $ER_2: (S_1, S_2, S_3, S_4, S_5)$, with probability $\rho_2$ and $\rho_1 + \rho_2 = 1$.

**Definition 7:** (Execution Plan). For execution route $S_i$, we define $(T_1, \ldots, T_n)$ as a execution plan of execution route $(S_1, \ldots, S_n)$, if $T_i \in S_i$ \quad (i = 1 \ldots n), here $T_i$ represents abstract service.

According to the definition, an execution plan is an executable service chain which meets user’s requirement.

In Fig.3, execution plans of execution route $ER_i$ could be: $(S_1, S_{i1}, S_{i2}, S_{i3}, S_{i4}), \ldots, (S_{i2}, S_{i3}, S_{i4}, S_{i2})$.

**B. Web Service Composition Flow**

We abstract web service composition flow as three phases:

Firstly, compose abstract services as a service function graph to meet user’s function demand in abstract hierarchy.

Second, discover all the functionality demand in abstract hierarchy. Service instances which have candidate relationships with abstract services. And transform service function graph to service selection graph.

Last, select a service chain linking the start node and end node in service selection graph, and the chain could keep the QoS constraints.

**C. Multi-dimensions QoS Model**

QoS is a series of non-function attributes. In this paper we will just use three attributes: reputation, price and availability into account. Other attributes could be imported and will not impact our results. Due to article [4] has already given an explicit depiction of definition and calculation of QoS attribute, we will not give a duplicate description.

**C.1 QoS Attribute Standardization and Utility Function**

Different QoS values have different value ranges, it is unfair to calculate these values directly. So QoS attribute standardization is needed. And in order to evaluate quality of the overall QoS of service, utility function is imported.

In this paper, we states the QoS of the $j$th service instance $S_i$ included in abstract service $S_i$ as $\langle q_{i1}, \ldots, q_{in} \rangle$. QoS attributes can be separated into positive attributes and negative attributes. The positive attribute, as availability and reputation, means the higher value the better selection result. Negative attribute, as price and duration time, is contrast with positive one. So we standardize positive attribute with formula (1) and standardize negative attribute with formula (2).

$$q_{ik} = \frac{q_{ik} - \alpha_i}{\sigma_i} \quad 1 \leq k \leq n \quad \text{(1)}$$

$$q_{ik} = 1 - \frac{q_{ik} - \alpha_i}{\sigma_i} \quad 1 \leq k \leq n \quad \text{(2)}$$

Where $\alpha_i, \sigma_i$ are the average and standard deviation of the QoS values for all candidates of $k$th QoS attribute of service.

A service has multi-dimensions QoS value $\langle q_{1}, \ldots, q_{n} \rangle$, so we need to consider all the QoS in web service selection process. But multi-dimensions value is not easy for comparison. Thus we propose a utility function, mapping multi-dimensions value into a function value which is a scalar quantity, to give a comprehensive reference for our comparison. The definition of utility function is as follows:

$$UF(s) = \sum \omega_i q_i \quad \text{and} \quad \sum \omega_i = 1 \quad \text{(3)}$$

Where $\omega_i$ denotes the weight of the $i$th QoS attribute which reflects their importance. $q_i$ is the standard attribute value.
C.2 QoS of Execution Plan (Composite Service)

Execution plan $EPR(S_1, ..., S_n)$ is a powerful service composed by a series of atomic services. Like a normal service, execution plan has its own QoS. The QoS calculation formula of execution plan is displayed in Table 1.

### Table 1.

<table>
<thead>
<tr>
<th>QoS attribute</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>$q_{\text{price}} = \sum_{i} q_{\text{price}}(i)$</td>
</tr>
<tr>
<td>Reputation</td>
<td>$q_{\text{reputation}} = \frac{1}{n} \sum_{i} q_{\text{reputation}}(i)$</td>
</tr>
<tr>
<td>Availability</td>
<td>$q_{\text{availability}} = \prod_{i} q_{\text{availability}}(i)$</td>
</tr>
<tr>
<td>Others</td>
<td>Formulas decided by attribute</td>
</tr>
</tbody>
</table>

C.3 QoS of Execution Route

In execution route $ER(S_1, ..., S_n)$, the candidates of abstract service $S_i$ are $S_{i_1}, ..., S_{i_z}$, and QoS of the $j$th service instance $S_j$ is $(q_{j_1}, ..., q_{j_z})$, so the QoS calculation formulas are as follows:

$$Q_{\text{price}} = \frac{\sum_{i=1}^{z} \sum_{j=1}^{z} q_{j_i} x_{ij}}{n}$$  \hspace{1cm} (4)

$$Q_{\text{reputation}} = \frac{\sum_{i=1}^{z} \sum_{j=1}^{z} q_{j_i} x_{ij}}{n}$$  \hspace{1cm} (5)

$$Q_{\text{availability}} = \prod_{i=1}^{z} q_{j_i} x_{ij}$$  \hspace{1cm} (6)

When the service $j$th is selected as an instance of the abstract service $i$th, we set $x_{ij} = 1$, otherwise we set $x_{ij} = 0$. The $i$ denotes the amount of candidate services of the abstract service $j$, Other QoS calculation formula of execution route could be imported by the same way.

IV. QoS-DRIVEN WEB SERVICE SELECTION MATHEMATICAL MODEL AND ALGORITHM

A. QoS-driven Web Service Selection mathematical Model

QoS-driven web service selection problem could be mapped into single constraint multi-objectives optimization problem in mathematics. We will first give web service selection mathematical model for single execution route, then web service selection mathematical model for multi execution routes will be displayed.

### A.1 QoS-driven Web Service Selection Mathematical Model for Single Execution Route

An execution route could make multi execution plans when abstract service is bound to different candidate services. So web service selection problem has to choose the optimal execution plan among all execution plans. In mathematics, it could map into a single constraint multi-objectives optimization problem.

First of all, topological sequence all the abstract services in an execution route, then each abstract service has its own QoS. We do the same things to all abstract services of each abstract service, and every candidate has an id $i$. We assume that abstract service $S_j$ ($1 \leq j \leq n$) in execution route $EPR(S_1, ..., S_n)$ has candidate services $(S_{i_1}, ..., S_{i_z})$, and $i$ is the amount of service instances included in abstract service $S_j$. Then, the corresponding single-objective multi-constraints optimization problem is:

1) **Objective function**

$$\text{MAX } f(ER) = UF(ER) = \sum_{i=1}^{z} \omega_i Q_i$$  \hspace{1cm} (7)

Where $Q_i$ is the $i$th QoS attribute value of execution route $ER$, derives from QoS calculation formula of execution route for standardization QoS. $\omega_i$ denotes the weight.

2) **Global QoS constraint**

Regarding that QoS values in global QoS constraints are actual data in application, so all the QoS values are not standardized in this part. Global QoS constraints can be divided into two groups.

- Single selection constraint

In the web service selection process, there is only one candidate service of each abstract service can be selected into composition flow. So single selection constraint can formalize as below:

$$\forall i \in n \hspace{0.5cm} \sum_{j=1}^{z} x_{ij} = 1$$  \hspace{1cm} (8)

Where $n$ denotes the number of abstract services in execution route. When $j$th candidate service in abstract service $i$ is selected, $x_{ij} = 1$; Otherwise, $x_{ij} = 0$, $i$ is the amount of service instances of abstract service $i$. Other QoS value constraint

QoS value constraints are constraints proposed by users, such as service price don’t go beyond 100S, service availability don’t be under 99.9%. If there are $h$ QoS value constraints $C^{i}$ $1 \leq k \leq h$, and $k$ denotes the $k$th QoS attribute, then

$$\forall k, 1 \leq k \leq h \hspace{0.5cm} Q_i \leq C^i$$  \hspace{1cm} (9)

Where $Q_i$ is the QoS value with attribute $k$ of execution route $i$.

### A.2 QoS-driven Web Service Selection Mathematical Model for Service Function Graph

There are may be multiple execution routes in a service function graph. Web service selection of service function graph with multiple execution routes is more close to the application in reality. The problem can map into single-objective multi-constraints optimization problem in mathematics.
A.2.2 Web Service Selection Mathematical Model of Service Function Graph with Multiple Execution Routes

Abstract web service selection problem of multiple execution routes into single-objective multiple-constraints optimization problem.

The execution routes of service function graph are $ER_1, ER_2, \ldots, ER_s$, and execution probability of $ER_i$ ($1 \leq i \leq s$) is $\xi_i$, where $\sum_{i=1}^{s} \xi_i = 1$.

1) Objective function

$$\max \ f(SFG) = \sum_{i=1}^{s} \xi_i f(ER_i) \quad (10)$$

Where $f(ER)$ is defined by formula (7).

2) Global QoS constraints

- Selection constraints
  It is the same as formula (8).

- QoS value constraints
  In order to meet the QoS value constraints strictly, we enforce every execution route to satisfy the constraint requirements. If there are $h$ QoS value constraints $C_1, C_2, \ldots, C_h$, and $k$ denotes the $k$th QoS attribute, then

$$\forall i, k \ 1 \leq i \leq s \ 1 \leq k \leq h; \ Q_i^k \leq C^k \quad (11)$$

Where $Q_i^k$ is the $k$th QoS attribute value of the execution route $i$.

B. QoS-driven Web Service Selection Algorithm

Integer programming and evolutionary algorithm, such as genetic algorithm, are two methods adopted broadly to solve optimization problem of web service selection. The strongpoint of QoS attribute computing based on integer programming lies on the maturation of theory and plenty of approaches. But it claims that the QoS constraints and QoS criterions should be linear while there are many non-linear QoS attributes just like availability. Genetic Algorithm also be applied to web service selection [5,8,9]. In contrast to Integer Programming, it has no requirement on whether the QoS constraints and objective function are linear or not, which extends the field of application rooting from Genetic Algorithm’s dependence to problem areas. On the other hand, Genetic Algorithm has better strongpoint of QoS attribute computing based on integer programming and evolutionary Computation, such as genetic algorithm, are two methods adopted broadly to solve optimization problem of web service selection. The strongpoint of QoS attribute computing based on integer programming lies on the maturation of theory and plenty of approaches. But it claims that the QoS constraints and QoS criterions should be linear while there are many non-linear QoS attributes just like availability. Genetic Algorithm also be applied to web service selection [5,8,9]. In contrast to Integer Programming, it has no requirement on whether the QoS constraints and objective function are linear or not, which extends the field of application rooting from Genetic Algorithm’s dependence to problem areas. On the other hand, Genetic Algorithm has better time consumed than Integer Programming as the increasing of composition scale [8].

V. CONCLUSION AND FUTURE WORK

Our work at present mainly focuses on constructing a platform for SOA application development: SMICE (Semantic Model-driven Integrated Construction Environment). Research of service composition model is an important part in this project. At the early stage we had developed a QoS driven service composition ontology framework [10], which provides a basis for web service composition and selection. Our work in this paper mainly lays on giving a QoS-driven web service composition model in detail and map it into single-objective multi-constraints problem. In future how to combine service semantics and QoS attribute to integrate more efficiently is an important goal of our work.

ACKNOWLEDGMENT

This work was supported in part by the National High-Tech Research and Development Plan Foundation (863), China (Grant No.2007AA01Z187) and NSFC (Grant No. 60805042)

REFERENCES