A Service-Merging-Enabled SOA Framework for Inter-Enterprise Collaboration*

Sen Zeng**
Department of Automation, Air Force Airborne Academy, Guilin 541003, China
Email: zengsen04@gmail.com

Mingguang Yang and Yimin Shen
Department of Computer Engineering, Chengdu Technological University, Chengdu 610071, China
Email: shen_yimin@sina.com

Yuanhua He
Department of Automation, Air Force Airborne Academy, Guilin 541003, China
Email: zs12181@163.com

Abstract—The functional constraints of service matching and performance requirements (i.e., quality of service, QoS) of service selection are separated distinctly under the traditional service-oriented architecture (SOA). Hence the service matching and service selection may be performed in turn and independently. And the complexities of the implementation of SOA are simplified also. In fact, the functional constraints and QoS are stretching and compromising frequently. And service providers with similar functions would come into being a merged-service with more powerful function and higher performance by collaboration with each other. This paper proposes an extended QoS (exQoS) system that contains some service functional constraints and traditional performance indicators. Then a service merging enabled SOA (meSOA) is established based on the exQoS, and the meSOA may be used to support service merging across enterprises. Furthermore, the problem of service matching and selection based on merged-service is formulated as a nonlinear planning (NLP) problem, and its corresponding solving algorithm is proposed also. At last, this NLP method and traditional method are compared with a numerical value experiment, which verified its correctness and the superiority of this method over the traditional one.

Index Terms—service merging, service-oriented architecture (SOA); inter-enterprise collaboration; extended-QoS

I. INTRODUCTION

Service-Oriented Architecture (SOA) unifies various resources as services, and a suit mechanics of service search, match, selection, binding, and running are proposed too[1][2]. With the emerging and rapid development of SOA and cloud computing, the flexibility and availability of forming new business solutions by service orchestration and choreograph with various business services and IT resources are enhanced evidently[3]. Especially with the economic globalization and the emergence of service-oriented manufacturing (SOM) paradigm, the merging of across enterprise manufacturing capability implemented by service merging usually provides more powerful service functions and higher level of QoS to the merged manufacture enterprise cluster [4][5][6]. For example, an enterprise cluster can produce larger quantity of products in a certain period to meet a customer’s requirement, but each of them can’t provide that quantity alone [7][8]. Correspondingly, an enterprise’s ability of across enterprise service merging is required urgently. That’s to say, if a service provided by an enterprise can not be merged with other enterprises’ services, a shared resource deployed on a cloud computing platform can’t be formed, thus restricting its being researched and selected.

The flow of service search, service matching, and service selection is described approximately as below. Function matching is carried out first to decide whether a service provider satisfies the service requester’s function requirement. Then an optimal one is selected from all the candidate service providers that satisfy the requester’s functional requirement by QoS, in order to optimize the overall performance of a business process or service composition that the service requester is orchestrated[9][10][11][12][13][14][15]. Therefore, the functional constraints and performance requirements play distinct roles in the flow of service search, matching, and selection.

The process of service matching is carried out under the rigid functional constraints of the service requester. That’s to say, if a service provider can’t reach the threshold of the functional constraint set up by the service requester, it will be washed out from the candidates.

The qualifiers will be selected further by performance indicators, i.e. QoS, in order to optimize the overall performance of a business process or service composition that the service requester is orchestrated[9][10][11][12][13][14][15]. Therefore, the functional constraints and performance requirements play distinct roles in the flow of service search, matching, and selection.

Manuscript received September 5, 2012; revised November x, 2012; accepted January x, 2013.

*The work of the paper was granted by Guangxi Natural Science Foundation, China (No. 2010GXNSFB013057). **Corresponding author

© 2013 ACADEMY PUBLISHER
doi:10.4304/jsw.8.7.1586-1591
SOA for service-oriented enterprise suffers unexpected practical problems. First, the boundary between the functional rigid constraints and QoS is not so absolute. In fact, some seemingly rigid functional constraints are compromising. For example, the requirement of “delivery-cycle: two weeks” may be a rigid constraint for a service requester, but when the delivery-cycle postponed 1 day will make it obtain concession of price from the service providers which will make up the punishment of extended deliver-cycle for customers, the rigid constraint of delivery-cycle can be degraded to an ordinary performance indicator of QoS. The instance is frequent in service composition, i.e. almost all constraints of non-critic path can be transformed to negotiable performance indicators of QoS.

However, it is a pity that the existing architecture of SOA provides just composition methods of QoS (mainly for sequence, parallel, branch, and loop structures) for different functional services [10][11][15]. It can not support service-merging for services with the same functions in order to compose a merging-service that has not only higher level of quality of service but also more powerful business functions. This paper constructs a service merging-enabled SOA (meSOA) framework that makes service resources in SOA and cloud computing platform have higher level of service (LoS) in order to support across enterprise collaboration under SOA and cloud computing environment.

I. A SERVICE MERGING-ENABLED SOA FRAMEWORK

Suppose a service requester, its service performance is evaluated by n individual indicators of QoS, \( \{q_1, q_2, \ldots, q_n\} \). Among the m service providers arrival and response to the service requester in succession \( \{S_1, S_2, \ldots, S_m\} \), the value of corresponding QoS indicators of \( S_i \) is \( \{q_{i1}, q_{i2}, \ldots, q_{in}\} \).

For the selected service provider \( S_j \), the service requester requires it satisfying the constraints as below:

1. Its values of QoS indicators not less than pre-determined constant \( Q_i \):
   \[ q_{ij} \geq Q_i (i=1,2,\ldots,n) \]  

2. The service requester pursues the maximum of the linear composition of \( \{q_{ij}, q_{j2}, \ldots, q_{jn}\} \), i.e., the objective function is:
   \[ v = \max \left( \sum_{j=1}^{n} w_j q_{ij} \right)^m \]  

Where, \( \{w_i\}_{i=1,2,\ldots,n} \) are constants selected by the service requester with its own needs. Many references have discussed how to select the weighted constants such as Ref. [10][11][15][16]. Since the n indicators of QoS \( \{w_i\}_{i=1,2,\ldots,n} \) have different dimensions, normalization is not required.

The traditional methods of service matching and selection usually have two steps:

Step 1, the service providers of \( \{S_j\}_{j=1}^{m} \) that can’t satisfy equation (1) are filtrated;

Step 2, take traverse or other methods to pick out a service provider which will optimize equation (2) from the providers that satisfy equation (1).

In fact, with step 1, large numbers of service providers will be washed out thus constraint the available service providers significantly. However, on the premise of service having the capability of merging, this paper improves above traditional method in order to make the service requester have much better qualities of service. The essentials of this new method are:

1. When a service requester releases requests, specify whether the responding service providers should have the capability of merging;
2. Service providers will send respond to the service requester in succession, and they become the sources of service merging;
3. The server of service merging obtains the optimal scheme of service merging by the method of this paper, and notifies service requester of the QoS of the merged-services.
4. Service requester binds the sources of service merging through the server of service merging, and starts to consume them.

Note that since SOA has the feature of P2P, after the server of service merging have provided the merged-services to the service requester, it will not participate in the subsequent processes of service matching, selection, and binding. Consequently, if a service requester expects to know the details of sources of service merging (i.e., the service providers composing the merged-service) before its binding with merged-service, it may send the request to the server of service merging, and the information including the information of the service providers and the original QoS will be provided by the latter.

The service merging enabled SOA framework is shown in Fig.1.
II. EXTENDED QoS AND ITS NONLINEAR PLANNING PROBLEM

Although the practical QoS may be various in different application context, they are usually classified into five classes in literatures namely time, cost, reputation, success, and availability\(^\cite{10,11,15,16,17,18}\).

It is obvious that the indicators of quantity and quality are far more important than the five classes of QoS enumerated above. The reason why they are not listed above is just about that these indicators are functional indicators with the feature of rigidity. The meSOA framework shown in Fig.1 makes it possible for the first time that these rigid indicators may be merged into SOA.

In fact, with the across enterprise business collaboration represented by service merging, the performance indicators including quantity and quality will be enhanced again possibly. For example,

- In a period of time, two manufacturing enterprises can produce 1000 parts and 1200 parts individually. Obviously, by the merging of their capability of manufacturing, it is possible for them to finish a task of manufacturing 1800 parts together, though each one of them can not do it by itself.

- The ratio of one time correction commonly is 99%, i.e., the ratio of error is 1%, in the service domain of data input in business process outsourcing (BPO) industry. With the addition of stage of QA, the ratio may be improved up to 99.9%. But it is possible that an enterprise still can not reach the requirement of correct ratio for some critic services. However, with the across enterprise business collaboration by merging more enterprises’ capabilities into the stage of QA, they will increase the correct ratio up to 99.99% for a key customer.

The QoS system comprising quantity, quality, and other rigid performance indicators is denoted as extended QoS (exQoS) in this paper.

Let \( x_j \) denote the quantity or workload assigned to the \( j \)th service provider, and \( \Psi \) the merging operator of the \( l \)th indicator of exQoS when these service providers are comprising a virtual enterprise. Thus, the performance indicators of exQoS of the virtual enterprise merged with these service providers are:

\[
q^l = \Psi(q_j, j=1,2,...,m)
\]

Where \( \Psi \) may be various linear and (or) nonlinear function format.

Combined with equation (2), a service selection problem of meSOA can be formulated as a non-linear planning (NLP) problem as below:

\[
\min \ -\sum_{i=1}^{n} w_i \Psi_i \left( \left[ x_j \right]_{j=1}^m \right)
\]

\[
\begin{align*}
\Psi_i \left( \left[ x_j \right]_{j=1}^m \right) &= Q_i \\
\Psi_i \left( \left[ x_j \right]_{j=1}^m \right) &\geq Q_i \\
x_j &\geq 0
\end{align*}
\]

Where, constraint equation (3) insures to complete the whole tasks, constraint equation (4) insures the satisfaction of the requirements of the other six performance indicators of exQoS, constraint equation (5) indicates that the quantity of task of each service provider is no less than zero.

It is obvious that the possible solution of the above NLP problem depends on the representation format of \( \Psi \). Referring to the classification of operators of service composition in\(^\cite{10}\), the mathematical format of service-merging-operators for service requester to estimate its value of performance objective is recommended in Table 1.

<table>
<thead>
<tr>
<th>Extended QoS indicators</th>
<th>Merging operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity (q(_1))</td>
<td>( \Psi_1 = \sum_{j=1}^{m} x_j )</td>
</tr>
<tr>
<td>Quality (q(_2))</td>
<td>( \Psi_2 = \sum_{j=1}^{m} \frac{q_j x_j}{Q_i} )</td>
</tr>
<tr>
<td>Cost (q(_3))</td>
<td>( \Psi_3 = \sum_{j=1}^{m} \frac{q_j x_j}{Q_i} )</td>
</tr>
<tr>
<td>Time (q(_4))</td>
<td>( \Psi_4 = \max \left( \frac{q_j x_j}{Q_i} \right)_{j=1}^{m} )</td>
</tr>
<tr>
<td>Reputation(q(_5))</td>
<td>( \Psi_5 = \sum_{j=1}^{m} \frac{q_j x_j}{Q_i} )</td>
</tr>
<tr>
<td>Success (q(_6))</td>
<td>( \Psi_6 = \sum_{j=1}^{m} \frac{q_j x_j}{Q_i} )</td>
</tr>
<tr>
<td>Availability(q(_7))</td>
<td>( \Psi_7 = \sum_{j=1}^{m} \frac{q_j x_j}{Q_i} )</td>
</tr>
</tbody>
</table>

Among them, for the indicators of cost and time, the lower values mean better performance, so that the operators of \( q_1, q_2 \) always comprise minus.

It can be verified, with the definition of service-merging-operators in Table 1, the NLP problem above is a convex planning problem and its available solution domain is convex. And its local optimal value is the
The method of this paper has well available calculation. Many common solution algorithms can be used to solve the NLP problem [19].

Generally speaking, the NLP problem can be run off-line, i.e., by waiting for a span of time or a certain numbers of service providers coming and responding to the service requester, users construct the above NLP problem and solve it to obtain the optimal scheme of service merging solutions at one time.

Under the context of this customer’s demand, \( m=1,000 \) service providers are constructed randomly. Values of their performance indicators are all subject to normal distribution with parameters of mean and variance as Table 2.

### IV. CASE STUDY

Suppose a customer demands \( Q_1=1,000 \) parts in - \( Q_2=25 \) days, the cost no more than - \( Q_3=6000 \) dollars, and the relative quality up to \( Q_4=0.99 \). For the supplier, its reputation must better than \( Q_5=0.9 \), the success and availability must larger than \( Q_6=0.95 \), \( Q_7=0.99 \) respectively. The customer wants the weights of quality, cost, time, reputation, success, and availability as \( w_1=0.05 \), \( w_2=0.0002 \), \( w_3=1 \), \( w_4=0.99 \), \( w_5=1 \), \( w_7=1 \).

Under the context of this customer’s demand, \( m=1,000 \) service providers are constructed randomly. Values of their performance indicators are all subject to normal distribution with parameters of mean and variance as Table 2.

<table>
<thead>
<tr>
<th>Parameters of Performance Indicators of Service Providers Constructed Randomly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality ( q_1 )</td>
</tr>
<tr>
<td>Quality ( q_2 )</td>
</tr>
<tr>
<td>Cost ( q_3 )</td>
</tr>
<tr>
<td>Time ( q_4 )</td>
</tr>
<tr>
<td>Reputation ( q_5 )</td>
</tr>
<tr>
<td>Success ( q_6 )</td>
</tr>
<tr>
<td>Availability ( q_7 )</td>
</tr>
</tbody>
</table>

The results of the above numerical experimentations repeated 300 times are shown in Table 4.

### V. EXPERIMENTATION REPEATED 300 TIMES

<table>
<thead>
<tr>
<th>Items</th>
<th>Method of this paper</th>
<th>Improve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of average available schemes of service composition</td>
<td>8.0266666667</td>
<td>6,357 times</td>
</tr>
<tr>
<td>Average value of overall performance objective</td>
<td>2.305285518</td>
<td>23.71%</td>
</tr>
</tbody>
</table>

In fact, probability distributions of the two methods are shown as Fig.2.
and expand the candidate service providers for service requester of service matching and service selection across enterprise significantly. And the performance indicators including quantity and quality can be enhanced again possibly.

With a nonlinear planning (NLP) problem, this paper has formulated the merging-service based service matching and selection problem. By comparing with traditional method we verified the correctness of exQoS and meSOA and the superiority of our method.

Future work is mainly to improve the practicality of the exQoS, meSOA and NLP proposed in this paper. First is the problems of selection and rationality of performance indicators of rigid functional constraints and traditional QoS included in exQoS. The real world problem is much more complex than the numerical experimentations demonstrated in this paper. It is therefore difficult to select appropriate performance indicators and quantify their values. Then is the implementation of meSOA, for example, how to describe the feature of service merging for service requesters and service providers, how to construct server of service merging, and so on.

V. REFERENCES


*Sen Zeng* received the Ph.D. degree in control theory and engineering from Tsinghua University, Beijing, China in 2009. His research interests include enterprise architecture, business process modeling methods and optimization analysis, complex system modeling and simulation, performance analysis and optimization, service-oriented computing, Web services, etc.