The Method for Service Recommendation in Ubiquitous Computing

Yun-Young Hwang¹, Jungsun Yoon, Kyu-Chul Lee², *

¹ Korea Institute of Science and Technology Information, Korea
{yyhwang, jsyoon}@kisti.re.kr
² Chungnam National University, Korea
kclee@cnu.ac.kr

Abstract. The arrival of ubiquitous computing environment, while providing services in many devices and users has been a lot of movement. In this environment, users want „seamless” service. A lot of mobility in ubiquitous computing environment to provide „seamless” service to users, many studies was used to substitute the service. An existing service substitution does not consider the characteristic of ubiquitous computing environment. Furthermore, it does not support ordering substitution list for fit of the user requirements. This paper proposed the approach for the operation recommendation considering the ubiquitous computing environment. This approach is based a Semantic Web technique. In addition, we investigated ubiquitous in the environment quality of services (QoS) that it has been used to research. We identified QoS of the existing ubiquitous computing environment that affects the user based on the substitution of alternative operations execution time, execution cost, availability and location. In addition, we defined new QoS that is suitable in substitution and based on QoS list identified.

Keywords: service substitution, service recommendation, ubiquitous computing, semantic service

1 Introduction

Recent Advances in computing and network technology, various services have increased exponentially. As a result, users can have a wide choice of services, and they choose the better quality of them among services, which have the same functionality. The arrival of ubiquitous computing environment, while the services are provided, devices, including these services, and users can have a number of moving. Even under such environment, users want to provide ‘seamless’ service. To address this requirement, many researches used the method of service substitution. Substitution means replacing a component with another component, as long as the replacing component produces the same output and satisfies the same requirements as the replaced component [1]. In ubiquitous computing, we considered the following points. The fist, the user shall not be limited by the communication protocol. Each of the ubiquitous devices is conformed to different communication protocols. The
second is frequently movement of devices and user. This paper proposed the operation recommendation approach providing „seamless” service to user. The following scenario will help the understanding our approach.

Scenario: The music streaming service of smartphone should be always used with sound output device such as speaker. If the user goes to company, he can continue to provide service with sound output device of the company. This device is located near the user, and it can communicate with his smartphone.

In order to realize this scenario, a list of alternative services is composited services, which have same functionality and are satisfied the user requirements. In this paper we propose an enhanced service substitution method of service brokering approach (universal service broker: US-Broker) in [2]. The approach addresses interoperability and composition between heterogeneous ubiquitous services. The basic concept of US-Broker composition is to derive a mapping between the target operation that should be substituted and substitute operation that offers similar functionality through a different interface. In addition, [2] represents the operation grouping information grouped by operation functionality. If target operation and another operation are belongs to same operation group, the operation is substitute operation. It is very simple, but it does not guarantee substitute service is executable because it does not consider I/O message structure and I/O parameter type. Furthermore, it does not take account of replacement costs or operation position. To solve this problem, we suggest the method for service recommendation of semantic based service substitution in ubiquitous computing. Our approach is to enable the mapping between input of the one operation and output of the other operation. In addition, we can search the substitute operation based on matching between I/O structure messages. Our approach can provide a list of ranking services considering replacement cost, degree of satisfaction of user requirements, availability, execution time of operation, and execution cost, and location of user and operation. To make ranking service list, we defined the weights to user requirement. The rest of the paper is structured as follows. Next section introduces related work. Third section discusses our approach for substitution of stateful services. Finally, we present the conclusions of this study and further research related to this study.

2 Related Work

We represent two types of related work. One is about service substitution method the other is related of quality of services. Among the main challenges of [3], is the issue of service substitution for the application execution in such heterogeneous environments. In this article, we define a generic service model and describe the equivalence relations between services considering the functionalities they propose and their non-functional QoS properties. In order to determine the degree of similarity between two services, they compared between the concepts, operations, and interfaces. The similarity has four types: exact, plug-in, subsume, and fail. They measure the degree of similarity, after then discover the alternative service. However, this approach supports only the list of services, which have same I/O message structure.
and I/O parameter type. [4] presents an approach whose objective is to support web services substitution. To perform Web services substitution with less impact on the ongoing, and sometimes critical, business processes, the approach proposes deploying communities of Web services. [4] uses the term “mapping rule” to express a single correspondence between the I/O parameters of operations to be adapted. However, if target service is not predefined into the mapping rule, it cannot support service substitution. It cannot consider the heterogeneous of I/O types, either. [5] defined the QoS such as capacity, availability, and security. The capacity includes throughput, responsiveness, information capacity, scalability, variability, and consistency. Service availability considers availability, reliability, maintainability, resilience, variability, and constancy. Security has confidentiality, integrity, availability, authenticity, authority, and nonrepudiation. However, they assume that all levels of quality factors are given and did not propose a specific guideline to determine levels. [6] defined the QoS for selection services are participated in the service composition. It considers execution price, execution duration, reputation, successful execution rate and availability. [7] enhanced ISO/IEC9126 for ubiquitous computing. ISO/IEC9126 is standard about quality of services. [7] redefined ISO/IEC9126 in the light of the mobility. However, ISO/IEC9126 is mostly suitable for stand-alone applications, and has not been massively tested against web-based applications and ubiquitous computing applications. QoS-based operation composition has been the subject of various research efforts. Usually, the consumer’s quality requirements are represented with quality properties. Then, mathematical or rule-based approaches are employed to find the best combination of operations for given quality constraints. [6] proposed a set of guidelines for service composition using user utility functions for QoS. However, they did not present a mechanism for resolving conflcits between quality factors. These approaches give only limited consideration to qualities such as response time and availability. In addition, they evaluate quality by an objective function to be minimized (e.g., response time or cost) or maximized (e.g., availability) with constraints to be satisfied. To resolve conflcits between quality factors, we propose a quality-driven operation composition methodology for ubiquitous applications.

3 Input/Out Mapping Approach

Ubiquitous environment forms from the convergence of a variety of services and devices, and therefore, various quality factors are involved to provide transparent and seamless services. The substituted operation can communicate with the front and the rear operation. To facilitate this, we propose I/O mapping using SAWSDL (Semantic Annotation for WSDL). Figure 1 shows the example of heterogeneous I/O message structure between operation D and operation A, and heterogeneous I/O parameter type between operation D and operation B. In order to operation D instead of operation C, we need I/O mapping between heterogeneous I/O messages. We use SAWSDL component for I/O mapping. We define the ontology (operation group ontology), which describes operation functionality. Each I/O parameter refers ontology. The output message of operation A has same functionality as input message of operation B.
Figure 2 shows how to resolve the heterogeneity between I/O messages. The process to find substitutable operations consists of four steps. The first step is making first candidate operations list. WSUN extracts services from OperationGroup of target operation should be substituted, and makes candidate substitutable operation list. Next, WSUN checks whether between the model reference of candidate operation and existing operation, which will connect to substituted operation.

Figure 3. Type matching process
According the location of services will connect to substituted service; candidate service satisfies equations, and then it does lifting and lowering process between different parameter types. Figure 3 shows the parameter type matching process. The sendResource2Onto.xslt can be used as a schema for mapping from SendResourceToPrinter.wsdl to concepts in Ontology. The type of output parameter in WSDL document is postscript, and the normal type of output parameter defined PCL in Ontology. We support sendResource2Onto.xslt to mapping between different parameter types. In addition, we address the problem about heterogeneous structures between parameters. This process is displayed in Figure 4. Assuming SendResourceToPrinter.wsdl in Figure 4 must connect to ObjectPush.wsdl, ObjectPush.wsdl required the Count input is integer type and mandatory parameter. In this case, we support sendResource2Onto.xslt to allocate default value into Count input parameter automatically.

4 Quality Criteria for Elementary Services

In ubiquitous environment, we have to guarantee „seamless“ service to the user. To address this problem, we define the Quality-of-Services (QoS) as a set of perceivable characteristics expressed in user-friendly language with quantifiable parameters that may be subjective or objective. The characteristics of quality and their parameters are based on the user or client requirements. In addition, we consider the where the service is. The user decides whether the location is important or not. Our system makes the candidate service list according the user decision. We consider four generic quality criteria for elementary services:

- execution time: the execution time of a operation is defined as the time spent by the system execution that operation, including the time spent execution
run-time and the time spent lifting and lowering schema mapping. The execution time is computed using the expression

\[ q_{\text{time}}(op) = T_{\text{process}}(op) + T_{\text{mapping}}(\text{onto}, op), \]  

\[ T_{\text{mapping}}(\text{onto}, op) = T_{\text{lifting}}(\text{onto}, o) + T_{\text{lowering}}(o, i), \]  

meaning that the execution time is the sum of the processing time \( T_{\text{process}}(op) \) and the schema mapping time \( T_{\text{mapping}}(\text{onto}, op) \). The schema mapping time is estimated based on executions of lifting and lowering schema mapping, i.e.,

\[ T_{\text{lifting}}(\text{onto}, o) \] is lifting schema mapping time between operation group ontology \( \text{onto} \) and operation output parameter \( o \), and \( T_{\text{lowering}}(o, i) \) is lowering schema mapping time between operation \( \text{onto} \) and operation input parameter \( i \).

• execution price: Given an operation \( op \) of service, the execution price \( q_{\text{pr}}(op) \) is the fee that an operation requester has to pay for invoking the operation \( F_{\text{invoke}}(op) \). In addition, it includes the fee to use software \( F_{\text{invoke}}(\text{soft}) \) which supports schema mapping.

\[ q_{\text{pr}}(op) = F_{\text{invoke}}(op) + F_{\text{invoke}}(\text{soft}) \]

• successful execution rate: The successful execution rate \( q_{\text{rat}}(op) \) of an operation \( op \) is that how many the operation is completed without failed or error occurred. The value of the success rate is computed from data of past invocations using the expression

\[ q_{\text{rat}}(op) = \frac{N_{\text{completed}}(op)}{T}, \]

where \( N_{\text{completed}}(op) \) is the number of times that the operation \( op \) has been successfully completed, and \( T \) is the total number of invocations.

• use frequency: Given an operation \( op \) of service, the use frequency \( q_{\text{fr}}(op) \) is that how many the operation is invoked \( N_{\text{selection}}(op) \) by the operation requester. The expression is

\[ q_{\text{fr}}(op) = \sum_{1}^{n} N_{\text{onto}}(\text{onto}) / N_{\text{selection}}(op, \text{onto}), \]

where \( \sum_{1}^{n} N_{\text{onto}}(\text{onto}) \) is the total number of invoked operation in same operation group ontology \( \text{onto} \).

Our system makes the recommend operations list used by the weight of user requirement. The users are able to decide that how much each qos criteria are important. Our system supports ranked operation list that is the result to calculate with weighted qos criteria. The ranked operation list is made by the expression

\[ Q = (q_{\text{time}}(op) \times W_{\text{time}}(op)) + (q_{\text{pr}}(op) \times W_{\text{pr}}(op)) + (q_{\text{rat}}(op) \times W_{\text{rat}}(op)) + (q_{\text{fr}}(op) \times W_{\text{fr}}(op)), \]

where \( W_{\text{time}}(op) \) is the weight of execution time that is the value entered by user. For example, there are two candidate operations (Table 1). The candidate operation B is free, many users used this operation. The candidate operation C is faster than candidate operation B, but user has to pay using this operation. The user, who needs seamless services now, prefers a good software performance.

<table>
<thead>
<tr>
<th>Table 1. Example of candidate operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>QoS criteria</td>
</tr>
<tr>
<td>----------------------------------------</td>
</tr>
<tr>
<td>Execution time</td>
</tr>
<tr>
<td>Execution price</td>
</tr>
<tr>
<td>Successful execution rate</td>
</tr>
<tr>
<td>Use frequency</td>
</tr>
</tbody>
</table>
Table 2 shows the calculated result according to expression (11). Although many users had used the operation B, operation C is fit to current user, who wants software which supports good performance.

<table>
<thead>
<tr>
<th>Table 2 The result of Q</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Candidate operation B</strong></td>
</tr>
<tr>
<td>( Q_{\text{time}}(\text{op}) \times W_{\text{time}}(\text{op}) )</td>
</tr>
<tr>
<td>( Q_{\text{opt}}(\text{op}) \times W_{\text{opt}}(\text{op}) )</td>
</tr>
<tr>
<td>( Q_{\text{ref}}(\text{op}) \times W_{\text{ref}}(\text{op}) )</td>
</tr>
<tr>
<td>( Q_{\text{cost}}(\text{op}) \times W_{\text{cost}}(\text{op}) )</td>
</tr>
<tr>
<td>Q</td>
</tr>
</tbody>
</table>

5 Conclusion

When ubiquitous environment occur the mobility of heterogeneous services, some services have to be replaced to another service. Therefore, the user spends a lot of time to find the appropriate service. To provide 'seamless' service to users, many studies were used to substitute the service. However they do not consider the characteristic of ubiquitous computing environment. Furthermore, they do not support ordering substitution list for fit of the user's requirements. This paper proposed the approach for the operation recommendation considering the ubiquitous computing environment. This approach is based a Semantic Web technique. In addition, we investigated ubiquitous in the environment QoS(quality of services) that it has been used to research. We identified QoS of the existing ubiquitous computing environment that affects the user based on the substitution of alternative operations execution time, execution cost, availability and location. In addition, we defined new QoS that is suitable in substitution and based on QoS list identified.

References