

## The Service Recommendation Method based on Semantic Technology in Ubiquitous Computing

Yun-Young Hwang<sup>1</sup>, Jungsun Yoon<sup>1</sup> and Kyu-Chul Lee<sup>2,\*</sup>

<sup>1</sup>Korea Institute of Science and Technology Information, Korea

<sup>2</sup>Chungnam National University, Korea

<sup>1</sup>{yyhwang, jsyoon}@kisti.re.kr, <sup>2</sup>klee@cnu.ac.kr

### Abstract

*In ubiquitous computing environments, users usually compose several services to achieve their goals. The service composition method of this paper is based on WSUN(Web Services on Universal Networks). WSUN approach supports dynamic service composition which takes account of the dynamic factors of ubiquitous environments. In service composition, an operation of a service should be substituted by another operation when the operation is not available. The substitutable operation should support same functionality of unavailable operation. Moreover, it should be able to communicate with pre operation and post operation. The service composition mechanism of WSUN considers the functional similarity between unavailable operations and substitutable operation, but it does not take account of I/O message structures and parameter types of operations. In order to enhance WSUN service substitution method, we use SAWSDL(Semantic Annotation WSDL for XML Schema), which is Semantic Web Services technique. SAWSDL provides two construct: modelReference and schemaMapping. A modelReference specifies the association between a WSDL or XML Schema component and a concept in some semantic data. Semantic annotations using modelReference help to discover substitutable operations. In addition, a schemaMapping solves the heterogeneity of I/O message structures and parameter types of operations. Consequently, even if the operation, which has same functionality of unavailable operation, has different I/O message structures and parameter types of unavailable operation. In order to recommend the operation has good quality, we define the quality of operation. It is a criteria that user can choose the best operation. In addition, we take a user weight for accommodation of user's requirements. The recommendation system prioritizes operation through calculating quality of operation based on user weight value. Therefore, operation by operation's priority recommends users. Our system supports substitutable operation discovery mechanism using Semantic Web Services technique, and substitutable operations list to satisfy user's requirement recommend in ubiquitous environment. Therefore, user can discover substitutable operation in substitution situation. In addition, substitutable operations to fit user requirement recommend user according to weight value from user.*

**Keywords:** operation recommend, service substitution, quality of service, ubiquitous computing

### 1. Introduction

Ubiquitous computing means that user can use any services of any devices in any network regardless of time and place. There are a number of services in ubiquitous computing

---

\* Corresponding Author

environment. Some of these services cannot communicate with other services, because they use a different communication protocol. If any user uses the combination service, it should be able to send and receive message between the services involved in the combination. In addition, if one of the participant services of composite service has to be replaced, it should be able to communicate between the alternative service and the front/back services.

In our prior study, Web Services on Universal Networks (WSUN) provides a dynamic combination method among the heterogeneous services. This method is based on interoperability between heterogeneous services. The interoperability is that communication is possible between heterogeneous services. In addition, WSUN defines that a minimum unit of the service combination is operation. This research uses the ontology to find the substitutable operation. Our ontology, ontology group, is grouping operation functionality. If two operations involve the ontology group, it means that they support same functionality. When the situation occur operation replacement, WSUN offers the alternative operation list involved same operation group of existing operation. This approach does not consider communication between operations. In other words, it causes a problem that alternative operation cannot be connected to the front/back operations. In order to address this problem, we suggest the enhanced approach considering the structure and type of I/O parameters. Furthermore, our approach provides user a ranked list of alternative operation. The related work about service substitution assumes that all levels of quality factors are given and did not propose a specific guideline to determine levels. The existing approaches of related quality of services 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 user requirements. To resolve confliction between quality factors, we propose a quality-driven operation composition methodology for ubiquitous applications. 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 wants to listen to the music 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 operation is composited with operation having same functionality and satisfying the user requirements. In this paper we propose an enhanced service substitution method of service brokering approach (universal service broker: US-Broker) in WSUN. 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. 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, and next

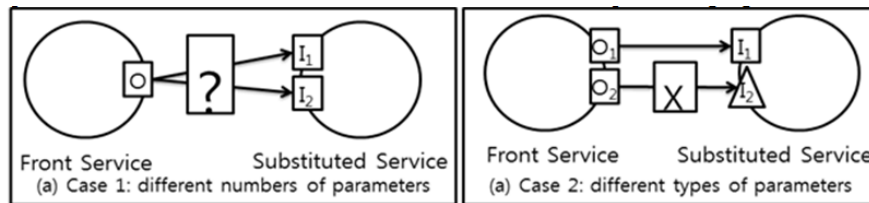
section represents the prototype system. Finally, we present the conclusions of this study and further research related to this study.

## 2. Related Work

This section introduces the Web Services on Universal Networks(WSUN), and related work about service substitution. Our composition method is based on WSUN.

### 2.1. Web Services on Universal Networks

WSUN supports service brokering mechanism (US-Broker) for interoperability and dynamic composition of heterogeneous ubiquitous services[7]. US-Broker translates all ubiquitous services (such as Bluetooth services, ZigBee services, Jini services, and *etc.*) to Web Services(called Virtual Web Services). This means service descriptions of all ubiquitous services are described in WSDL(Web Services Description Language) documents.



**Figure 1. Considering Cases during Substitution**

US-Broker also supports simple mechanism for substitution of service. The basic idea is classifying services depends on their functionality. The services, which have same functionality, belong to same ServiceGroup. One service can belong to one or more ServiceGroup, and ServiceGroup has also one or more services. If a service participating in combined services is not available or executable, US-Broker finds the substitutable services in same ServiceGroup with being substituted service. It is very simple and fast, but it does not guarantee substitute service is executable. The substituted service has to be able to communicate existing services are located at the front and the back of being substituted service.

We consider two cases to achieve these goals (Figure. 1). The first case is the services, which are located the front and the back of being replaced service, and substituted service have different numbers of I/O parameters. The left case in Figure 1 shows this problem. The other case is there are inconsistent between two services (Figure. 1 (b)). The both cases can appear between substituted service and front or back service of substituted service. US-Broker cannot address two cases. Therefore, we enhanced US-Broker to cover these problems. To address these problems, we used the semantic technology is SAWSDL (Semantic Annotation of WSDL), because our service description is represented by WSDL standard. The next section described our detailed approach.

### 2.2. Service Substitution Approach

We represent two types of related work. The one is service substitution method, the other is related of quality of services.

Among the main challenges of [8], 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.

[9] 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. [11] 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.

[11] 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.

[12] defined the QoS for selection services are participated in the service composition. It considers execution price, execution duration, reputation, successful execution rate and availability.

[14] enhanced ISO/IEC9126 for ubiquitous computing. ISO/IEC9126 is standard about quality of services. [14] 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. [14] proposed a set of guidelines for service composition using user utility functions for QoS. However, they did not present a mechanism for resolving conflictions 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 user requirements. To resolve conflictions between quality factors, we propose a quality-driven operation composition methodology for ubiquitous applications.

### 3. Operation Recommendation Method

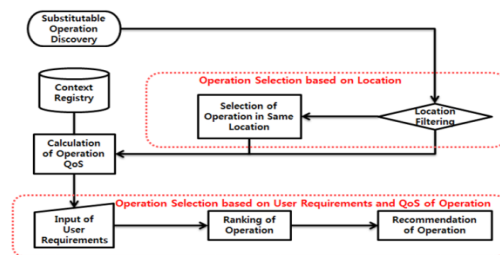


Figure 2. Operation Recommendation Process

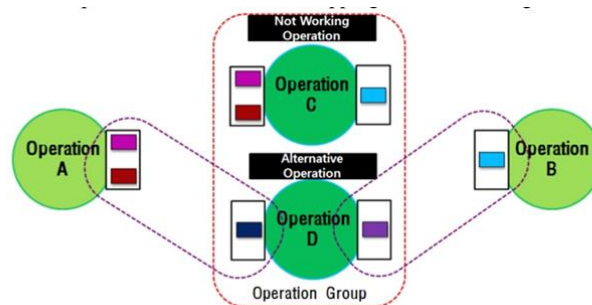
The following figure is shown an operation recommend process of our system. The first process is discovery of substitutable operation. The whether use or not of location filtering process is depending on positional relationship between the user and the operation. In the case of using location filtering process, user can choose the location in which they want to use the service. If the user determines the location, our system makes the target operation in the selection location. Conversely, if the user does not want to user location filtering process, the target operation is all operation through operation discovery process.

Next, an operation is selected to obtain the quality value, which value is stored in a registry context. Recommended candidate operations are stored in the registry operation context to derive the quality value might be used. For recommending operation to consider user requirements, our system requires the QoS weight from users. This weight is used as a tool for recommending operations.

The ranking operation process is composed into the following sub-processes. The first is formalization of quality of the operation, and next is T-Score calculation of quality of the operation. The final is the process of deriving the final value of quality of operation. Operation of the particular quality T-Score calculated to determine the ranking of the quality of the operation can be tailored to complement the problems. The final operations of quality values derived T-Score value of each operation input by the user by calculating a weighted user requirements required by the operation to the user is also recommended as a high priority now for the process.

### 3.1. Input/Output 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 3. Example of Heterogeneous I/O Message Structure and I/O Parameter Type between Operations**

The process to find substitutable services consists of four steps. The first step is making first candidate services. US-Broker extracts services from ServiceGroup of target service should be substituted, and makes candidate substitutable services list. Next, US-Broker

checks whether between the model reference of candidate service and existing service, which will connect to substituted service. This process is based on from Eq. 1 to Eq. 5.

$$\forall FSO.m \in O(FSO) \tag{1}$$

$$\forall BSI.m \in O(BSI) \tag{2}$$

$$(\forall CI.m \in O(CI)) \cap (\forall CO.m \in O(CO)) \tag{3}$$

$$O(CI) \in O(FSO) \tag{4}$$

$$O(CO) \in O(BSI) \tag{5}$$

*BSI.m.modelreference* of input parameter of existing services will be located back of substituted services.

*FSO.m.modelreference* of output parameter of existing services will be located front of substituted service.

*CI.m.modelreference* of input parameter of candidate substitutable service.

*CO.m.modelreference* of output parameter of candidate substitutable service.

*O*: ontology representing relationship service and service parameters.

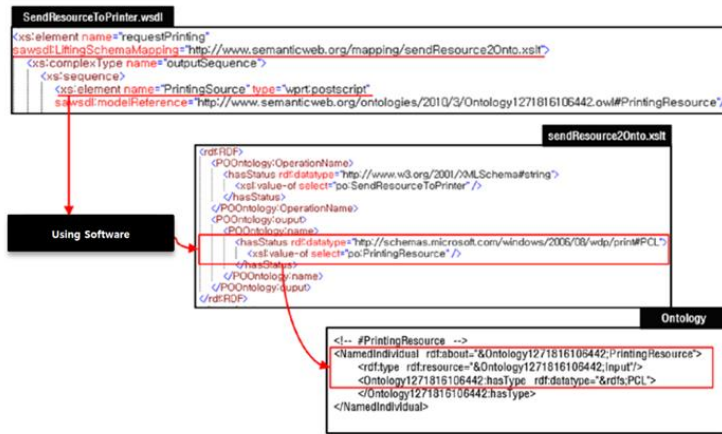


Figure 3. Lifting Schema Mapping Process

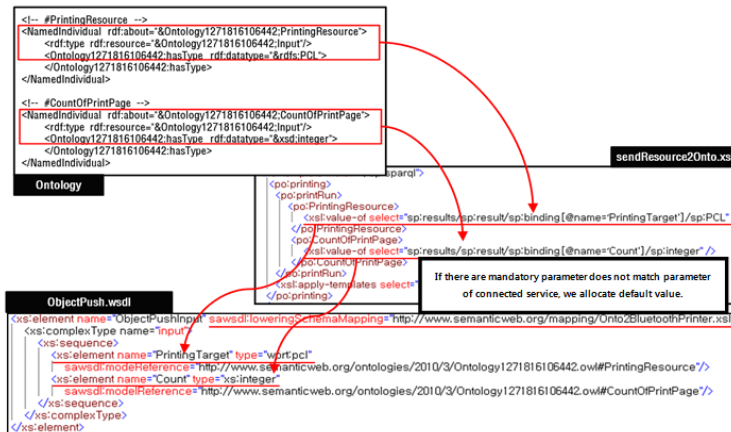


Figure 3. Structure Matching Process

According the location of services will connect to substitutedservice; candidate service satisfies equations from Eq. 1 to Eq. 4 or from Eq. 1 to Eq. 3 and Eq. 5. The third and last process is lifting and lowering process between different parameter types. Figure4 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.

This process is displayed in Figure 5. 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.

### 3.2. 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

$$q_{time}(op) = \frac{T_{process}(OP) + T_{mappint}(onto, op)}{MAX_{T_{process}(op)} + MAX_{T_{process}(onto, op)}} \times 100 \quad (6)$$

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

$$T_{mapping}(onto, op) = T_{lifting}(onto, o) + T_{lowering}(o, i), \quad (7)$$

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

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

$$q_{pr}(op) = \frac{F_{invoke}(op) + F_{invoke}(Soft)}{MAX_{F_{invoke}(op)} + MAX_{F_{invoke}(soft)}} \times 100 \quad (8)$$

- successful execution rate: The successful execution rate  $q_{rat}(op)$  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

$$q_{rat}(op) = \left(1 - \frac{N_{completed}(op)}{T}\right) \times 100 \quad (9)$$

where  $N_{completed}(op)$  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  $q_{fr}(op)$  is that how many the operation is invoked  $N_{selection}(op)$  by the operation requester. The expression is

$$q_{fr}(op) = \left(1 - \frac{N_{selection}(op)}{\sum_1^n N_i(op_{selection})}\right) \times 100(10)$$

where  $\sum_1^n N_i(op_{selection})$  is the total number of invoked operation in same operation group ontology  $onto$ .

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

$$Q = (q_{time}(op) \times W_{time}(op)) + (q_{pr}(op) \times W_{pr}(op)) + (q_{rat}(op) \times W_{rat}(op)) + (q_{fr}(op) \times W_{fr}(op))(11)$$

where  $W_{time}(op)$  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 1. Example of Candidate Operations**

QoS criteria	Candidate operation B	Candidate operation C	Weight
Execution time	0.15	0.5	4
Execution price	0	0.4	1
Successful execution rate	0.9	0.9	0
Use frequency	0.4	0.2	1

### 3.3. T-Score

Operation of the T-Score calculated quality replacement can be found for prioritized operations to compare the operation method. Operation of the particular quality T-Score calculated to determine the ranking of the quality of the operation can be tailored to complement the problems.

Quality value of Operation =

$$T(O,norTime) * W_{time} + T(O,norPrice) * W_{price} + T(O,norSuccess) * W_{success} + T(O,norSelect) * W_{select}$$

**Figure 3. Final Value of Quality about Operation**

$$T(O1,time) = \left[ \frac{O1.norTime-m}{s} \right] \times 10 + 50$$

$$s = \sqrt{\frac{(O1.norTime-m)^2 + (O2.norTime-m)^2 + \dots + (On.norTime-m)^2}{n}}$$

**m: average, n: number of operation**

**Figure 4. T-Score Calculation**



For example, operation A is the faster than operation B. This means that operation B requires the time of performing replacement more than operation B, but operation A uses more than 1000 times lower than a B operation. If the user determines the first priority is operation replacement time and the next priority is operation using counts, the operation A is best choice better than operation B. However, in user's perspective, the operation B is a good operation, because operation B has similar replacement time of operation A and it has much higher using time than operation A. Thus, the quality of a particular ranking the recommended build for this operation to exclude situations where the quality of this study were compared to previous operations to solve problems in standard scores (Standard score) [13] method of calculating T-Score was used.

#### 4. Our Systems

Figure 8 shows the structure of our systems. The substitution agent, core component, is composed of four sub-components: location filter, recommender, substitutable operation finder, and WSDL Parser.

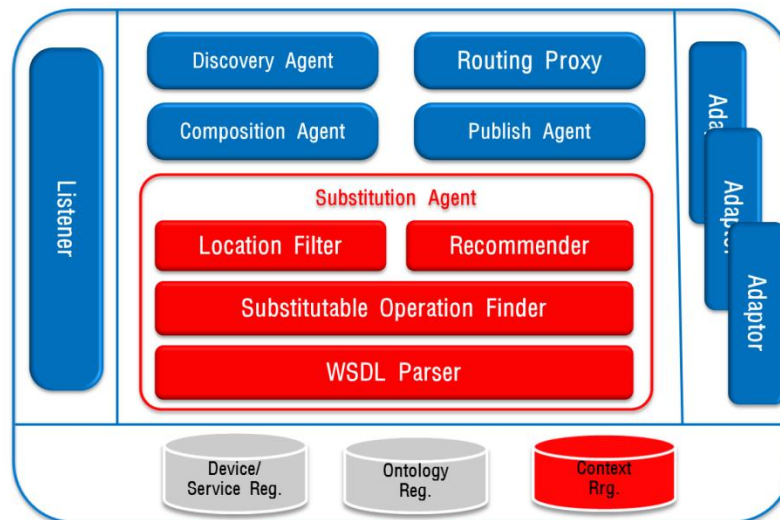


Figure 5. The Structure of our System

- **Location Filter**

The location filter supports operation filtering functionality based on the location selected by user. If the operation does not exist in the user-selected location, this operation is excluded from the list of the alternative operation.

- **Recommender**

It recommends appropriate operation to a user. This component requires the weight value about user requirements from user. Our system calculates the quality of operation value using pre-calculated weighting value. The pre-calculated weighting value is registered in the context registry.

- **Substitutable Operation Finder**

To find the alternative operation, we compare structure and type between input parameter of one operation and output parameter of another operation.

### ● WSDL Parser

This component supports the transport the identifying information and its analysis the structure of operation structure. In addition, it analyzes the information of SAWSDL annotation, and structure of I/O parameter.

## 5. Conclusion

When ubiquitous environment occurs 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.

In this paper, to solve these limitations, we support the substitution operation method based on SAWSDL mechanism. In addition, we define the quality factors of operation to find the operation, and our system uses the weight value from user. It can discover operation which is more suitable for user requirements.

Future research directions of this study are as follows. Currently, our system can support the selection of the operation has specific position, but it needs the location filtering process. This filtering process includes location information on street. The using of space and distance information is expected to be selected more substitutable operation.

## Acknowledgements

This research was supported by the Global resource utilization project in science and technology from operation of KOSEN of the National Research Foundation (NRF) funded by the Ministry of Science, ICT & Future Planning (N-14-NM-IR09).

## References

- [1] A. A. Abbasi and M. Hussain, "A QoS Enhancement Framework for Ubiquitous Network Environments", *IJAST*, vol.43, (2012), pp.37-48.
- [2] A. A. Abbasi and A. Florentina, "An Analysis of QoS specific Coherence Issues in Distributed Networks", *IJAST*, vol.49,(2012), pp.73-82.
- [3] B.S. Jun, "SOA, What & How", *WowBooks*, (2008).
- [4] C. Brodersen, S.Bødker and C.NylandstedKlokmoose, "Ubiquitous Substitution", *Human Computer Interaction 07*, (2007), pp. 179-192.
- [5] D.Chappel, "Introducing SCA", In *Proceeding of the CHAPPELL & ASSOCIATES*, (2007).
- [6] G. Zayaraz and P. Kalamegam, "A test framework based on CPN model for functional testing of Web Service Composition", *IJAST* vol.53, (2013).
- [7] H.J.Yim, Y.Y. Hwang and K.C. Lee, "Task-based Services Composition for Ubiquitous Computing", *SocialCom2011*, (2011).
- [8] L.Zeng and B.Benatallah, "QoS-Aware Middleware for Web Services Composition", *IEEE Transaction on Software engineering*, vol.30,(2004), pp. 311-327.
- [9] N.Ibrahim, F.Le Mouel and Frenot, "Semantic Service Substitution in Pervasive Environments", *International Journal of Services/Economics and Management*,(2010).
- [10] OASIS, "Web Services Quality Factors v1.0," OASIS Working Draft, 07 June,(2010).
- [11] S.H. Oh, S.D. Kim and S.Y. Ryu, "UCQM: A Quality Model for Practical Evaluation of Ubiquitous Computing Systems", *Journal of KIISE* vol.34, no.4, (2007), pp.342-358.
- [12] *Software Engineering-Product Quality-part 2: Internal Metrics*. ISO/IEC TR 9126-2, (2003)
- [13] Wikipedia, "Standard score", [http://en.wikipedia.org/wiki/Standard\\_score](http://en.wikipedia.org/wiki/Standard_score).
- [14] Wikipedia, "Quality of Service", [http://en.wikipedia.org/wiki/Quality\\_of\\_service](http://en.wikipedia.org/wiki/Quality_of_service)
- [15] J. Farrel and H. Lausen, "Semantic Annotations for WSDL and XML Schema", *W3C Recommendation*, 2007.08, (2007).

- [16] X.Wang, T.Vitvar, M.Kerrigan, and I.Toma, "A QoS aware selection model for semantic web services," In 4th International Conference on Service Oriented Computing (ICSOC 2006), (2006), pp. 390-401.
- [17] Y. Taher, D.Benslimane, M.C.Fauvet and Z.Maamar, "Towards an Approach for Web Services Substitution", IEEE IDEAS06, (2006), pp.166-173.

## Authors



**Yun-Young Hwang**, She received B.E., M.E., and Ph.D. degrees in computer engineering from ChungnamNational University in 2002, 2004, and 2011, respectively. In 2012 she joined KISTI(Korean Institute of Science and Technology Information), Korea where she is currently a researcher in Dept. of Overseas Information.

Her current areas include Big-Data Processing, Cloud-Computing, Ubiquitous Computing and Long-Term Preservation of Electronic Records.She has published over 30 technical articles in various journals and conferences.



**Jungsun Yoon**, She received B.S., M.S. degrees in computer science from Korea Advanced Institute of Science and Technology in 1991, and 1993 respectively. From 1993 to 2000 she worked for KRISS(Korea Research Institute of Standards and Science) as a senior researcher. In 2000 she joined KISTI(Korean Institute of Science and Technology Information), where she is currently a principal researcher in Dept. of Overseas Information.

Her current areas of interest include information service, usability, and HCI.



**Kyu-Chul Lee**, He received B.E., M.E., and Ph.D. degrees in Computer Engineering from Seoul National University in 1984, 1986, and 1996, respectively. In 1994 he worked as a visiting researcher at the IBM Almaden Research Center, San Jose, California. From 1995 to 1996, he worked as a Visiting Professor at the CASE Center at Syracuse University, Syracuse, New York. He is currently a Professor in the Department of Computer Engineering at Chungnam National University, Daejeon, Korea.

His current areas include Multimedia Database System, Hypermedia Systems, Object-Oriented Systems, and Digital Libraries. He has published over 100 technical articles in various journals and conferences. He is a member of ACM, IEEE Computer Society, and Korea Information Science Society.

