J Inf Process Syst, Vol.10, No.4, pp.581~588, December 2014
http://dx.doi.org/10.3745/JIPS.01.0002

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

An Unified Representation of Context Knowledge Base for Mobile Context-Aware System

Jang-Seop Jeong* and Dae-Wook Bang*

Abstract—To facilitate the implementation of a wide variety of context-aware applications based on mobile devices, general-purpose context-aware framework that applications can use by calling is needed. The context-aware framework is a middleware that performs the sensing, reasoning, and retrieving based on the knowledge base. The knowledge base must systematically represent the information required on the behavior of the context-aware framework, such as context information and reasoning information. It must also provide functions for storage and retrieval. To date, previous research on the representation of the context information have been carried out, but studies on the unified representation of the knowledge base has seen little progress. This study defines the knowledge base as the unified context information, and proposes the UniOWL, which can do a good job of representing it. UniOWL is based on OWL and represents the information that is necessary for the operation of the context-aware framework. Therefore, UniOWL greatly facilitates the implementation of the knowledge base on a context-aware framework.

Keywords—Context-Awareness, Knowledge Base, Unified Context Information

1. INTRODUCTION

Currently, studies for implementing context-aware applications are underway worldwide in the mobile computing environment [1-4]. A context-aware system linking the real space to virtual space is to inform the real situation in virtual space, thereby providing user-centered intelligent service [5]. Typically, a context-aware system is constructed by using a middleware called the context-aware framework.

The context-aware framework is a middleware that performs the sensor input, reasoning, such as questions and answers based on the knowledge base. The knowledge base must systematically represent the information, such as context information, reasoning information, that is required for the behavior of the context-aware framework, and also provide functions for storage and retrieval.

To date, previous research on the representation of the context information continues to be carried out, but studies on the unified representation of the knowledge base has made little progress. This study proposes, UniOWL, which uniformly represents the knowledge base with the OWL syntax.

※ This study is the result of a study on the LINC Project, which was supported by the Ministry of Education.
Manuscript received October 14, 2013; first revision March 26, 2014; accepted May 21, 2014;
onlinefirst November 27, 2014.
Corresponding Author: Jang-Seop Jeong (hdca64@hanafos.com)
* Dept. of Computer Engineering, Keimyung University, Daegu 704-701, Korea. (hdca64@hanafos.com, dubang@kmu.ac.kr)

2. MOBILE CONTEXT-AWARE SYSTEM AND FRAMEWORK

Considering the context in the mobile computing environment, a two-tier architecture, which consists of a context server and mobile devices, is suitable for a context-aware computing system providing intelligent services [6]. The two-tier architecture of the context-aware system, as shown in Fig. 1, distributes the context-aware frameworks in mobile devices and the context server, and connects them via a wired or wireless network. Also, it shares and exchanges the sensed, environmental and reasoning information amongst the context-aware frameworks.

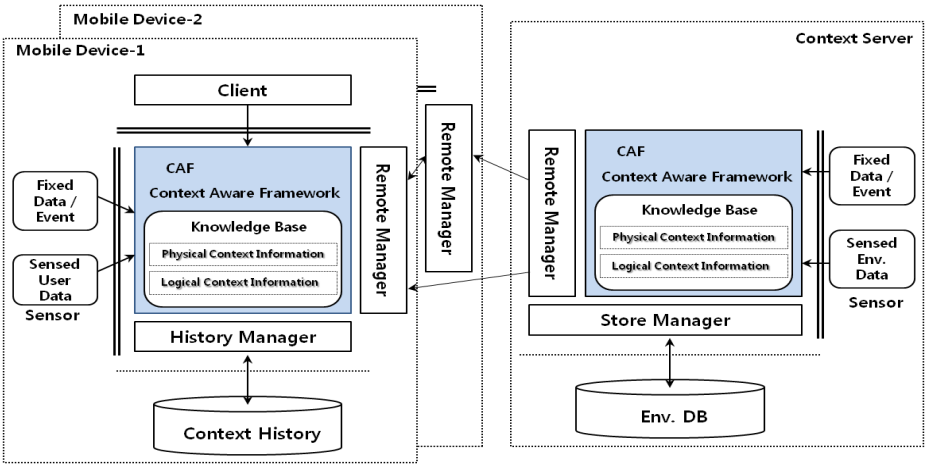


Fig. 1. Mobile context-aware systems architecture.

The mobile device in Fig. 1 is maintained by the context-aware framework, and executes the client that uses the knowledge base of the context-aware framework. The client accesses and utilizes the knowledge base of the framework with the APIs provided by the framework, and through the UI of mobile devices interacts with the user. The mobile device delivers user-related information, which is detected by its sensors in order to keep track of the current context, to the context-aware framework in real time.

The context-aware framework stores and manages physical context information in its knowledge base and by the various reasoners, generates logical context information from the physical information context. It also provides physical and logical context information to the client.

In addition to the context information, the knowledge base should include reasoning information, system information, and so on. In order to implement a context-aware framework, studies on the representation of the knowledge base should be carried out. This study takes into account all of the information that is stored in the knowledge base as the unified context information. As such, we are proposing the UniOWL syntax, which can do a good job of representing the unified context information.

3. UNIFIED CONTEXT INFORMATION

The term ‘context’ means the information characterized as the status of the entity that exists in the real world. ‘Entity’ means a human, place, or interaction between people and services.

However, this definition does not consider the application profile and users intentions. Considering these points, Dey [7] defines context information as all the information describing the context of a particular entity. In this case, an entity includes all of the objects that are related to the interaction between the user and the application.

Besides Dey's context information, the knowledge base of mobile context-aware framework should include reasoning information, system information, and so on. This study regards all of the information as being the unified context information. The unified context information consists of context information, reasoning information system information, and sensor information [8], as shown in Fig. 2.

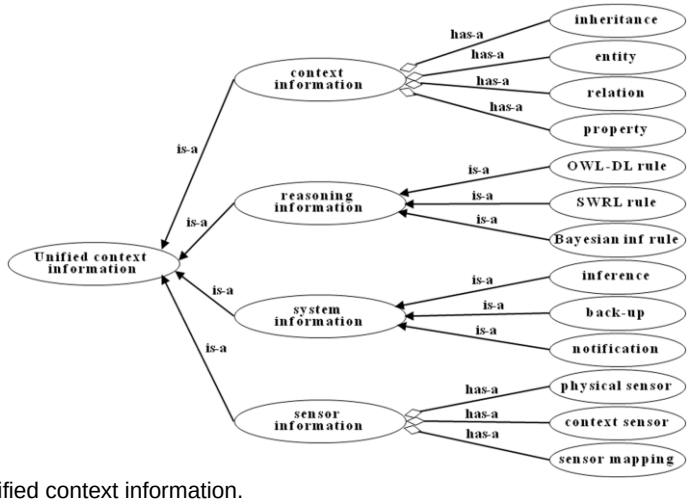


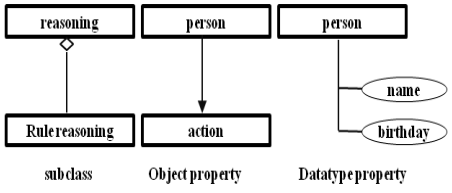
Fig. 2. The unified context information.

The reasoning information is the rule set for rule reasoning, the OWL-DL operation set of ontology reasoning, or the Bayesian network for probabilistic reasoning. The system information is the action information (e.g., reasoning action, notification action, and backup action) that defines the operation set that is to be executed, depending on the state of the context-aware framework. Finally, the sensor information consists of the system sensor specification, context sensor declaration, and the mapping information between the system and context sensor. The context sensor declaration is the OWL class name, which is defined in the context information. There are two types of mapping information. One is the information for one-to-one mapping using physical sensor values as the system sensor value, and the other is the information for mapping aggregating system sensor values.

4. UniOWL – Unified Context Information Representation

UniOWL is the OWL syntax representation of the unified context information. OWL (Web Ontology Language) is designed for use by applications that need to process the content of information, instead of just presenting the information to people. OWL facilitates greater machine interpretability of Web content than what is supported by XML, RDF, and RDF Schema (RDF-S). It does so by providing additional vocabulary along with formal semantics. OWL has three increasingly expressive sublanguages: OWL Lite, OWL DL, and OWL Full [9].

The UniOWL of the context information, which is shown in Fig. 3, is represented as OWL syntax, such as Class, ObjectProperty, DatatypeProperty, and Individual. Similarly, the UniOWL of the reasoning information, which is shown in Fig. 4, is represented as OWL-DL, SWRL, or PR-OWL. All of which are based on OWL.



(a)

```
<owl:Class rdf:ID="Rulereasoning">
    <rdf:subClassOf rdf:resource="#reasoning">
</owl:Class>

<owl:ObjectProperty rdf:ID "has_action">
    rdfs:domain rdf:resource="#person" />
    rdfs:range rdf:resource="#action" />
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="name">
    rdfs:domain rdf:resource="#person"/>
    rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="birthday">
    rdfs:domain rdf:resource="#person"/>
    rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
```

(b)

Fig. 3. The UniOWL of the context information. (a) Graphic representation, (b) OWL syntax.

(working ∪ hiking ∪ synthesis) ->field_sports

(working_prescription∪hiking_prescription)- >prescription

¬mobile_device ->"Non_mobile_device"

(a)

```
<owl:Class rdf:ID="field sports">
    <owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#working" />
<owl:Class rdf:about="#hiking" />
<owl:Class rdf:about="#synthesis" />
    </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID="prescription">
    <owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#working prescription" />
<owl:Class rdf:about="#hiking prescription" />
    </owl:intersectionOf>
</owl:Class>
<owl:Class rdf:ID="Non_mobile device">
    <owl:complementOf rdf:resource="mobile device>
</owl:Class>
```

(b)

Fig. 4. The UniOWL of the OWL-DL reasoning information. (a) Operator representation, (b) OWL syntax.

The UniOWL of the sensor information, as shown in Fig. 5, is defined as system sensors and context sensors. The system sensor has attributes, such as input type, and the virtual sensor, which is a class name that is defined in the context information. Also, the sensor information has mapping properties between the system and context sensors.

The UniOWL of the system information is represented as an action state and lists reasoning behaviors per state, as shown in Fig. 6. The action state includes the reasoning action point ('onDemand', 'onSensing', 'onInput') as properties. The reasoning behavior is defined as a reasoning source, priority, and reasoning method (ontology reasoning: 'consistency', 'satisfiability', 'classification', realization; rule reasoning: 'all-rules', 'selected-rules', 'repeat-selected-rules').

```
<!-- define system sensor-->
<owl:Class rdf:about="#System Sensor"/>
<owl:Class rdf:about="#PhysicalSensor">
<rdfs:subClassOf rdf:resource="#SystemSensor"/>
</owl:Class>
<owl:Class rdf:about="#VirtualSensor">
<rdfs:subClassOf rdf:resource="#SystemSensor"/>
</owl:Class>

<owl:DatatypeProperty rdf:about="#isDriverName">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#SystemSensor"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#isSensorType">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#SystemSensor"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#isSensorMode">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#SystemSensor"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#isPeriod">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#SystemSensor"/>
<rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
```

```
<!-- define context sensor -->
<owl:Class rdf:about="#ContextSensor"/>

<owl:DatatypeProperty rdf:about="#isInstanceName">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#ContextSensor"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#isPropertyNames">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#ContextSensor"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<!-- define mapping information -->
<owl:DatatypeProperty rdf:about="#hasContextSensor">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#SystemSensor"/>
<rdfs:range rdf:resource="#ContextSensor"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#isMappingRules">
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#SystemSensor"/>
<rdfs:range rdf:resource="#xsd;string"/>
</owl:DatatypeProperty>
```

Fig. 5. The UniOWL of the sensor information.

```
<!-- define action of context-aware framework -->
<owl:Class rdf:about="#StateAction"/>
<owl:Class rdf:about="#ReadyStateAction">
<rdfs:subClassOf rdf:resource="#StateAction"/>
</owl:Class>
<owl:Class rdf:about="&uniowl;RunStateAction">
<rdfs:subClassOf rdf:resource="#StateAction"/>
</owl:Class>
<owl:Class rdf:about="&uniowl;StopStateAction">
<rdfs:subClassOf rdf:resource="#StateAction"/>
</owl:Class>

<!-- define information of reasoning service -->
<owl:Class rdf:about="#ReasoningService"/>
<owl:Class rdf:about="#OntologyReasoningService">
<rdfs:subClassOf rdf:resource="#ReasoningService"/>
</owl:Class>
<owl:Class rdf:about="#RulesReasoningService">
<rdfs:subClassOf rdf:resource="#ReasoningService"/>
</owl:Class>
<owl:Class rdf:about="#ProbabilityReasoningService">
<rdfs:subClassOf rdf:resource="#ReasoningService"/>
</owl:Class>
<owl:Class rdf:about="#RankReasoningService">
<rdfs:subClassOf rdf:resource="#ReasoningService"/>
```

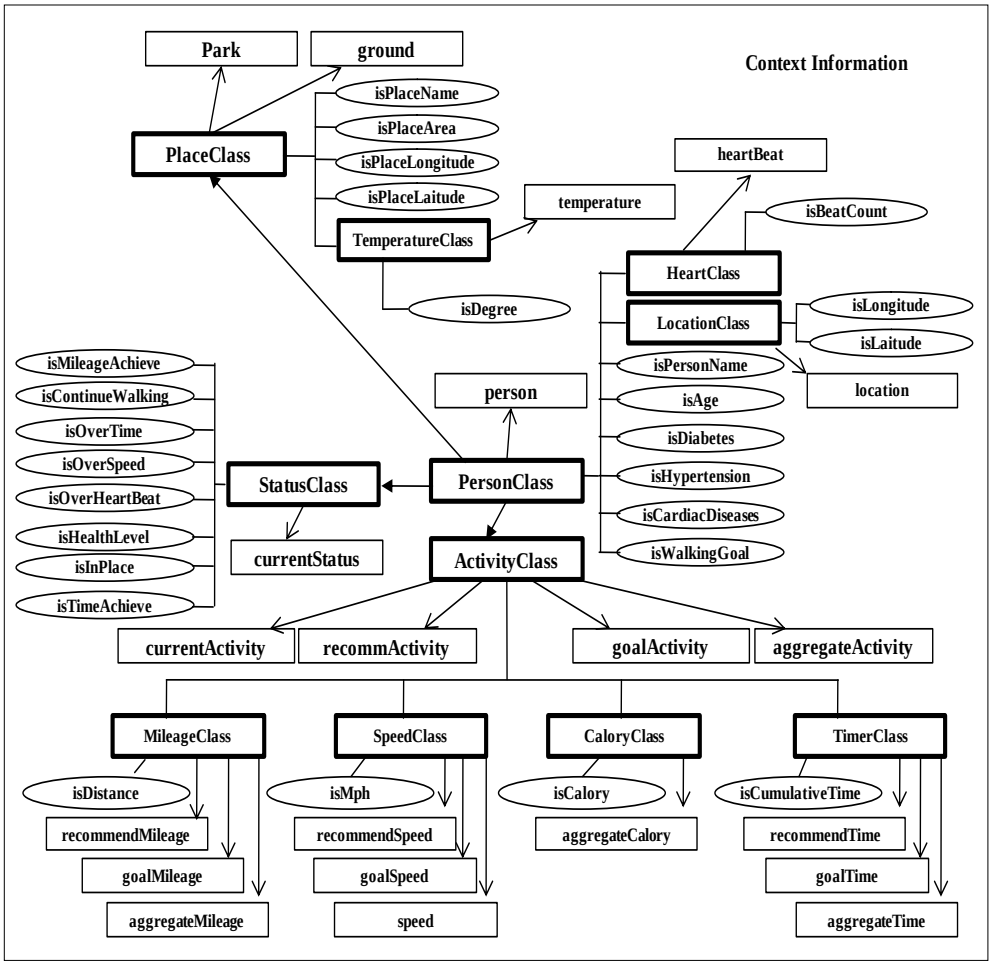
```
<!-- define property of action -->
<owl:DatatypeProperty rdf:about="#isActionPoint">
<rdf:type
rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#StateAction"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<!-- define property of reasoning service -->
<owl:ObjectProperty
rdf:about="#hasReasoningReference">
<rdfs:domain rdf:resource="#ReasoningService"/>
<rdfs:range rdf:resource="#ServiceSource"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#isSensorType">
<rdf:type
 rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#ReasoningService"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#isPriority">
<rdf:type
 rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="#ReasoningService"/>
```

</owl:Class> **<!-- define relation between action and reasoning service -->** <owl:ObjectProperty rdf:about="#hasServiceList"> <rdf:type rdf:resource="#StateAction"/> <rdfs:domain rdf:resource="#ReasoningService"/> <rdfs:range rdf:resource="&xsd;string"/> </owl:ObjectProperty>	<rdfs:range rdf:resource="&xsd;integer"/> </owl:DatatypeProperty> <owl:DatatypeProperty rdf:about="#isDetailedMethod"> <rdf:type rdf:resource="&owl;FunctionalProperty"/> <rdfs:domain rdf:resource="#ReasoningService"/> <rdfs:range rdf:resource="&xsd;integer"/> </owl:DatatypeProperty>

Fig. 6. The UniOWL of the system information.

5. CASE STUDY

The proposed UniOWL can represent the unified context information of various application areas, such as a context monitoring service, a self-management service, and a recommendation service. In this study, the unified context information for walking exercise management, which is a kind of self-management service, was implemented and analyzed.



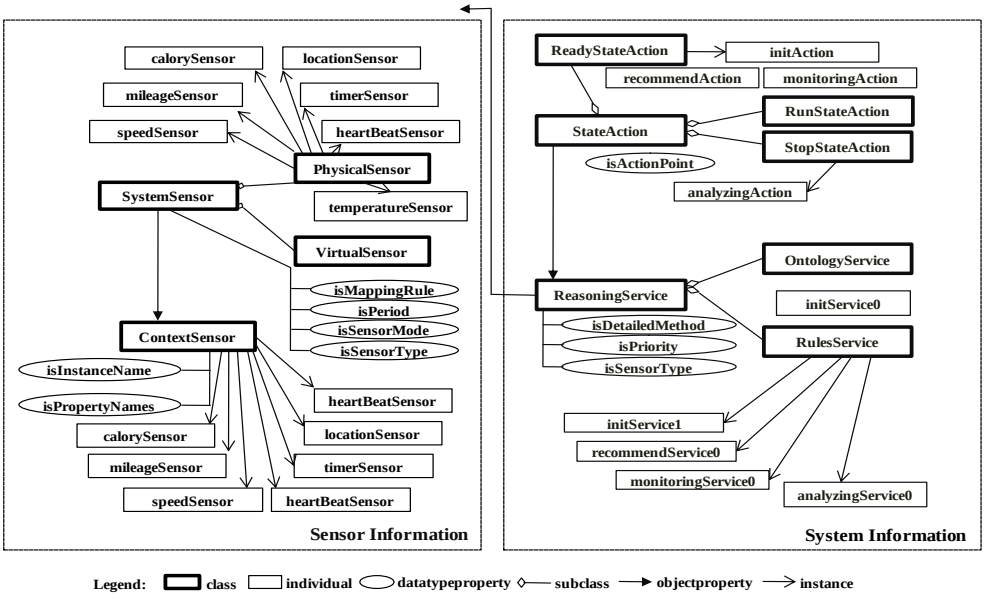


Fig. 7. The UniOWL of the walking exercise management.

The walking exercise management sets the exercise goal by considering the user's health condition, weather condition, and the outdoor exercise facility at the start of the exercise. While the user is exercising, it notifies them of their exercise level or of any health concerns. Finally, it analyzes and shows the results of the exercise after the workout is completed. Fig. 7 shows the UniOWL syntax that represents the unified context information for the walking exercise management.

6. CONCLUSION

This study defined the context information stored in a knowledge base as the unified context information and proposed UniOWL for representing the unified context information. UniOWL makes it possible to consistently complete the knowledge base that was developed with a client application.

In conclusion, it facilitates the development of mobile context-aware applications and extends context-aware application areas. It does so by providing the syntax that manages the activity state and its state transition, and by dynamically configuring the sensing information.

REFERENCES

[1] J. Gu and G. Chen, "Design of physical and logical context aware middleware," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 5, no. 1, pp. 113-130, 2012.

[2] C. Zhu, C. Guo, J. Wang, and T. T. Tay, "Towards scalability issue in ontology-based context-aware systems," in *Proceedings of the International Conference on Software and Computer Applications (ICSCA 2012)*, Singapore, 2012, pp. 127-131.

[3] P. D. Costa, L. F. Pires, M. van Sinderen, and T. Broens, "Controlling services in a mobile context-aware infrastructure," in *Proceedings of the 2nd Workshop on Context Awareness for Proactive Systems (CAPS2006)*, Kassel, Germany, 2006, pp. 153-166.
[4] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J. H. Hong, and A. K. Dey, "Factors influencing quality of experience of commonly used mobile applications," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 48-56, 2012.
[5] H. Liberman, T. Selker, "Out of context: computer systems that adapts to, and learn from, context," *IBM Systems Journal*, vol. 39, no. 3-4, pp. 617-632, 2000.
[6] J. Jeong and D. Bang, "A context-aware system supporting distributed processing and multi-reasoning," in *Proceedings of Korea Computer Congress*, vol. 39, no. 1D, pp. 91-93, 2012.
[7] A. K. Dey, "Understanding and using context," *Personal Ubiquitous Computer*, vol. 5, no. 1, pp. 4-7, 2001.
[8] L. S. Rosenthal and A. K. Dey, "Towards maximizing the accuracy of human-labeled sensor data," in *Proceedings of the 15th International Conference on Intelligent User Interfaces*, Hong Kong, China, 2010, pp. 259-268.
[9] M. Miraoui, C. Tadj, C. ben Amar, "Context modeling and context-aware service adaptation for pervasive computing systems," *International Journal of Computer & Information Science & Engineering*, vol. 2, no. 3, pp. 148-157, 2008.

Jang-Seop Jeong

He in 2009, the National Graduate School of Software Engineering, Kumoh National Institute of Technology and graduated, in 2012, the Keimyung University graduate school Ph.D. courses were completed computer engineering. Who is interested in context-aware system, distributed processing system, mobile computing, ubiquitous computing, u-healthcare system.

Dae-Wook Bang

He received the M.S. degree in Computer Science from KAIST in 1982 and a Ph.D. degree in Computer Engineering from Seoul National University in 1995. He has been a professor at Keimyung University since 1986. His research interests are in the area of Embedded Software, Healthcare IT, Mobile Computing and Distributed System Software. They include topics such as Context Awareness, Android-based Devices, DICOM/PACS and Mobile Agent.