

A Java-Based RFID Service Framework with Semantic Data Binding Between Real and Cyber Spaces

Kei Nakanishi, Makoto Setozaki, Jianhua Ma, and Runhe Huang

Faculty of Computer and Information Sciences, Hosei University, Tokyo 184-8584, Japan
{nakanishi, setozaki}@malab.k.hosei.ac.jp
{jianhua, rhuang}@k.hosei.ac.jp

Abstract. Recently there are getting more and more systems and applications using RFID. There is a necessity of a framework for them. This paper presents a Java-based framework that offers a set of general services to support various RFID systems for different purposes and application scenarios. The framework emphasizes on the semantic data binding for contextual information mapping between real and cyber spaces. The Java interface classes are provided to support general communications among a RFID, a reader and an application. Real and cyber spaces are synchronized via dynamic and real-time mapping from symbolic strings or IDs to the semantic XML data representations that are more effectively and efficiently useable by RFID applications. In this paper, the architecture and functional modules of the Java-based RFID service framework are explained in detail.

1 Introduction

The ubiquitous computing is aimed at supporting human's memory, judgment and action with pervasive computers and networks throughout physical environments [1]. RFID (Radio-Frequency-Identification) is enumerated as a technological element to achieve the ubiquitous computing. It is used as one of the means to acquire information in a real space. There are more and more studies on RFID related systems and developments, such as location-aware information technology [2], location tracking system [3], reminding system [4], human detection [5], etc. One of the main common features in developing an RFID-based system is the mapping of information on a real space to a cyber space.

There are two essential issues in developing RFID systems. The first is that a RFID operation method requires a library to operate RFID data and it is not easy to operate RFID data without a common interface API even if having the library in developing various RFID systems. It is usually a design requirement that the interface for reading and writing RFID data must be general though a communication means of reading and writing RFID data depends on its concrete type of RFID. It is also preferred that the interface and operation method can be used in different RFID systems by only switching the library. The second is the mapping between real and cyber spaces. A RFID keeps only symbolic code or ID data, i.e., a set of strings, that doesn't have much meaning if it is not associated with a real object or some real objects. Information in a real space in general has a special meaning, this framework is therefore

aimed at offering a general service to convert symbolic codes into semantic data so as to closely link real and cyber spaces.

In the rest of this paper, we first present several representative RFID based application scenarios, next summarize the common features of the RFID systems, and then propose and discuss a Java based framework. It consists of three basic parts: a RFID programming interface, the information mapping between real and cyber spaces using RFID and XML data binding, and intelligent application in a cyberspace. Finally, related work is discussed, conclusions are drawn, and future work is addressed.

2 Representative RFID System Scenarios

To capture common features of RFID systems, this section shows several RFID-based application scenarios: object finding, thing reminder, person location detection, and meeting member detection, respectively.

Object Finding System. This system is supposed to be able to find an object or some objects in a room or house where RFID tags with ID codes and readers are placed. Some RFID tags are put under the floors in a pre-designed layout. Such a tag is named *position tag* that keeps a symbolic number, which is associated with the coordinates of the tag stored in a database. In a room, a robot carrying a RFID reader and wireless LAN can move in a pre-defined course. The robot reads all RFID tags that are near the RFID reader. The obtained tag IDs are sent to the database via the wireless LAN and then used in a location analysis process as shown in Fig. 1. If a tag attached to an object is read, the four surrounding position tag's coordinates determine the object's position coordinate.

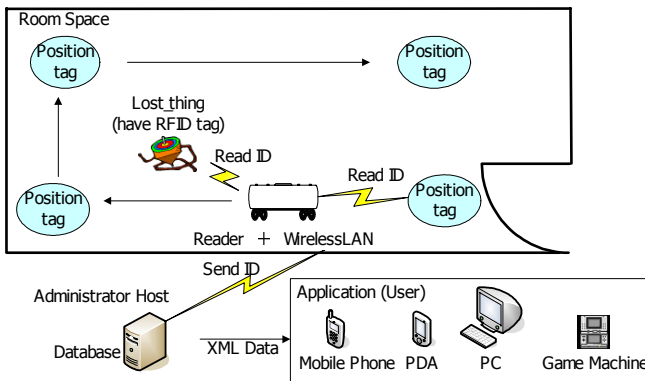


Fig. 1. A scenario of an object finding system

The process continues when the robot is moving and the obtained coordinate data is used in the application for drawing a map of the room on a GUI. All the related objects in the room are drawn on the map. To find a lost object, a user inputs the object name, and the system will inform the location of the object via some audio/visual interface.

Thing Reminder System. This system is supposed to give you advice messages in a room/house entrance when you forget to take something with you. This system requires that there is a RFID reader at an entrance, all things for a user to bring are attached RFID tags, and the item IDs are input to the application in advance. When the user goes out, the reader at the entrance reads the user's identity tag and the items' tags. These IDs are sent to a database via wireless LAN and then used in a process as shown in Fig. 2.

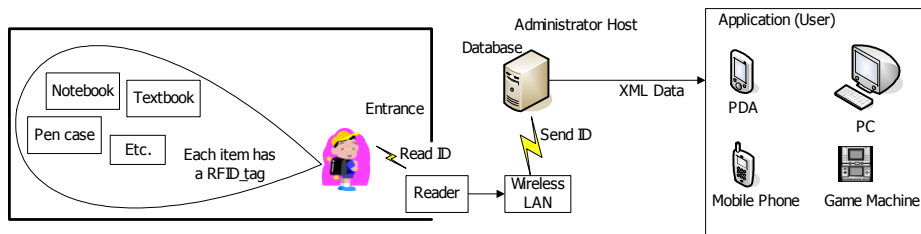


Fig. 2. A scenario of a thing reminder system

The obtained ID data is compared with the ID data in the database that the user input beforehand. If there is unmatched ID data that remains in the user input ID data set, the application reminds the user in some way such as a speech message of “you forget to take the desk pad”.

Person Location Detection System. This system can get a person's location by detecting RFIDs carried with the person. The system requires at least two RFID tags per person and some RFID readers. Two tags are attached to each bottom of the person's shoes. Some RFID readers are buried under the floor in special places such as an entrance, a kitchen, a toilet, etc. The location data in XML format of RFID readers is stored in a database. If the person steps over a RFID reader, the reader reads the person's RFID tags and sends the IDs to the database via a wireless LAN that is connected with the reader. The tag IDs and RFID reader's location are used to locate the person's position in the application as shown in Fig. 3.

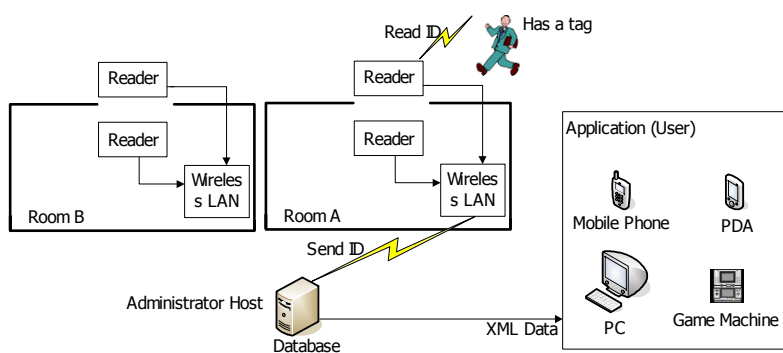


Fig. 3. A scenario of a person location detection system

In the application, there is a graphic image/icon that corresponds to a person’s tag ID. On the monitor, a house/building map is showed. Once a reader detects a person’s entering/leaving a room, an image/icon corresponding to the person’s tag ID is drawn in a certain position on the map. Watching the map, one can know who is in the room and where the room is in the house/building.

Meeting Member Detection System. This system is supposed to let a user know who is meeting whom by detecting RFID tags carried by the persons in a meeting. It requires that every one have a RFID tag and a reader with a wireless LAN. For example, a person A is going to meet person B and C. If A approaches to B, A’s reader reads B’s tag. After that, the tag’s data is sent to a database via the wireless LAN. In the database, the ID data is converted to XML-based data representation. In the application, a message generated from the XML data, such as “A is meeting with B”, is displayed on a monitor. If C is joining the meeting, with the similar processing, a message, such as “A is meeting with B and C”, is displayed on the monitors. A scenario of such a system is given in Fig. 4.

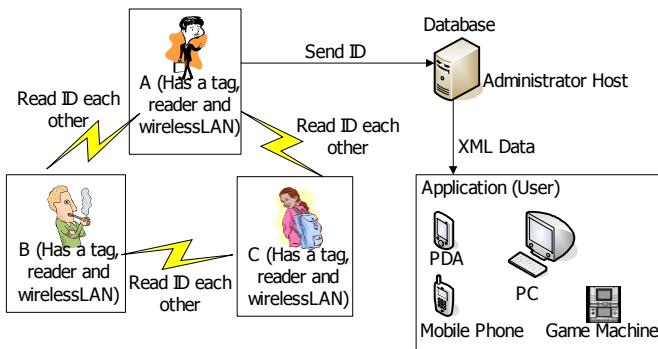


Fig. 4. A scenario of a meeting member detection system

3 RFID Service Framework

The four representative scenarios show some common features and process.

- A RFID reader reads a RFID tag.
- The ID is sent to a database.
- The database is used by an application.
- The application processes the current incoming data with comparisons of the data kept in the database.
- From the comparison results, the system can get conclusions, such as where is the object to be found, which object is forgotten, where a user is, who is meeting whom, or others.

As a result of the common features and the process, some commonly necessary components and their relations in a RFID system can be summarized in Fig. 5.

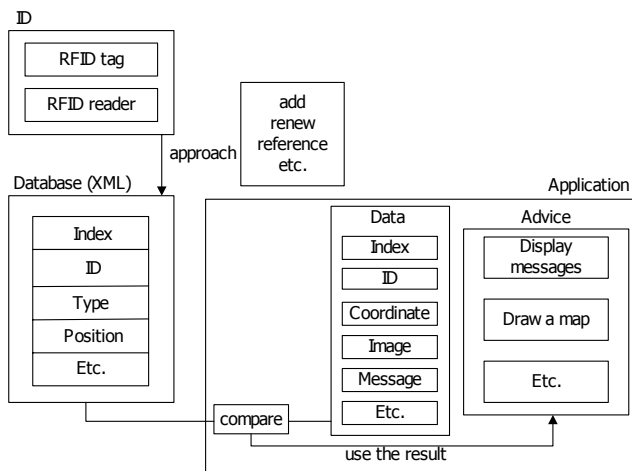


Fig. 5. Common components and a general structure of RFID systems

Further, the architecture of a general RFID service framework is resulted in to facilitate RFID service system developments and given in Fig. 6.

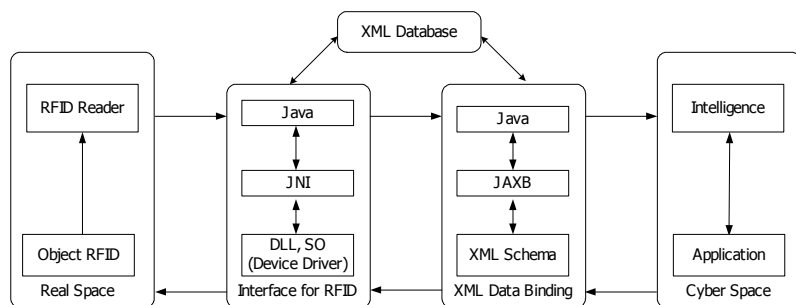


Fig. 6. A general RFID service framework

The RFID service framework is able to support a series of the processes. It mainly consists of two mechanisms, an interface API for RFIDs and XML data binding to bridge the real space and the cyber space. In the interface for RFID, the device drivers in DLL (dynamic link library) or SO (shared object) are necessary in operating specific RFID devices but can not directly be handled or accessed by or from Java. However, JNI (Java Native Interface) [6] can play the role of a bridge between a device driver in DLL or SO files and Java VM. Java VM can operate or access DLL and SO files through JNI. Thus, RFID service developers can easily operate RFID devices related data with using Java in an application layer. XML is used as a general data format in the system. To access and handle XML data in Java, JAXB (Java Architecture for XML Binding) [7] is imported to bind an XML schema to a representation in

Java code. As for application developments in Java, developers may only focus on how to use the data in their applications and do not need to know the details about how to get physical data and convert them into a correct format used in Java since these two steps are supported by this framework. Thus, a Java application processes information obtained from a real space as well as information on a cyber space.

4 Programming Interface for RFID

There are some common basic issues in the development of RFID-based systems. One is related to available methods of controlling the RFID reader and writer. The control methods for different kinds of RFID vary. Some kinds of RFID readers use RS-232 interface to connect to a computer. There is a case that a RFID is controlled directly by commands from neither the driver nor the library. Even if the driver is offered by a manufacturer, it is often impossible to control RFIDs at once because it is necessary to prepare the interface that connects a system and libraries. Next one is that RFIDs keep only symbolic ID codes. To operate the data from RFIDs, the codes should be closely related to their attached real objects. Another one is that a system should be general though a controller depends on kind of RFID used. Therefore, the mechanisms for accessing or operating any devices such as RFIDs and sensors to acquire information on a real space should be abstract. To enable such accesses to different kinds of RFIDs more easily, our framework adopts the XML technique. Though RFID has only symbolic number, it is possible to give each symbolic code a meaning and include it in a XML data file. Of course, for a XML file, it is impossible to record all data in a continuous time in a real space. Only those critical or turning points from which the changing trace of an object can be represented or recovered are recorded.

A parser is needed to read the XML data. Though the parser for different programming languages has been distributed by some ventures, most of APIs that can read/parse XML are offered in Java such as JAXM, JAXB and JAXP. They are included in Java XML Pack distributed by Sun Microsystems. Taking Java and XML's advantages of platform independence and object-oriented model, we use Java as implementation language of the interface for RFID in this research.

A driver and its associated library distributed by a manufacturer are often written in native codes. However, Java VM cannot run the native codes directly. To make them run-able by Java VM, one approach is to use JNI that is the API used by Java programs to invoke the native code compiled from C/C++ codes. The JNI can generate a Java header to use the native code in C/C++. Figure 7 (in left) shows the relation between the JNI and C/C++.

A library might be written in not only C/C++ but also C#. Like Java, C# is a programming language that runs on a special environment named CLR (Common Language Runtime). The CLR is a managed code environment for the .NET Framework [8]. When a program is written in a managed code that runs on CLR, even if JNI is used, it is not run-able by Java VM. Though Java VM can use native codes with JNI, it cannot use managed codes. However, if the managed codes are wrapped with the native codes, it is able to use them with JNI [9, 10]. Figure 7 (in right) shows the relation between the JNI and C#.

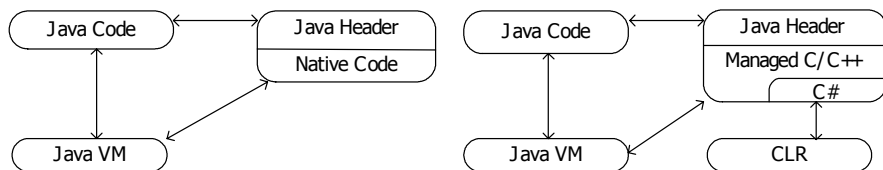


Fig. 7. The JNI as a bridge between Java and other languages

Thus, the interface based on Java is developed by using JNI, and a RFID ID code is read and send to the XML database. Its architecture is shown in Fig. 8. The interface is mainly to operate RFID. The next section explains how to map information from a real space to a cyber space with XML data binding.

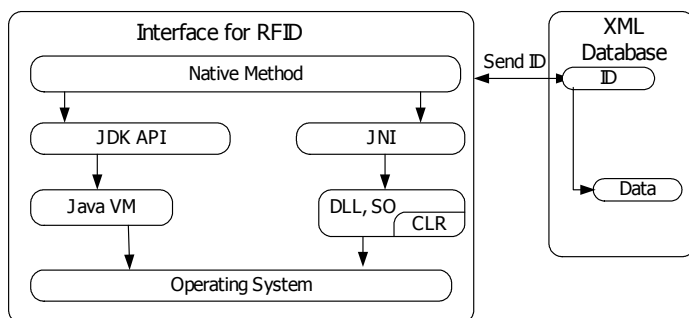


Fig. 8. The interface for RFID

5 Mapping Between Real and Cyber Spaces

Information obtained from a real space is about to be used by an application or some applications in a cyber space. However, information original from RFID readers is just a sequence of symbolic strings. Without attaching identical meanings, it is difficult to use such symbolic numbers by an application in a cyber space. To solve the problem, XML techniques play two important roles here. One is to use the XML scheme and attach some identical meaning to each symbolic code by adding necessary tags into the XML scheme. Another is to use the information represented in XML scheme by a Java application. In this case JAXB is required to parse XML structure and then generate a tree structure and further then map the tree into Java classes. Thus, information obtained from a real space is operational in a cyber space. The XML techniques link two kinds of spaces together and bind data from a real space to a cyber space, which makes two kinds of spaces reflective to each other. An example of the mapping between real and cyber spaces is shown in Fig. 9.

To operate a set of the data represented in XML in a cyber space is to make them being easily used by Java programs in a related RFID system or so called application. To different applications, data has different meanings. If an application is regarded as an object, data modeled in the XML scheme is a kind of object-oriented data. Both

Java and XML are based on object-oriented models. So the structure of XML can be easily mapped to the class of Java. An API that maps a XML model to a Java model is provided in the data-binding tool, JAXB, which enables to map the data obtained from RFID to XML and pass the XML to a cyber space. Consequently, it is achieved to map information from a real space to a cyber space.

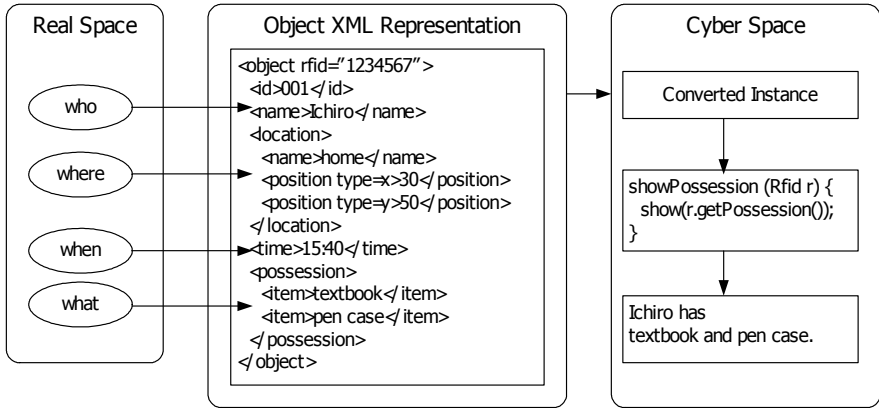


Fig. 9. An example of the mapping between real and cyber spaces

6 Intelligent Application in Cyber Space

A mechanism with which information on a real space is mapped to a cyber space is achieved by binding RFID and XML data. An application on a cyber space is a system that operates data acquired from a real space to offer some specified services. It is a fact that even if only critical data about objects from a real space is recorded, there is still much more data than it is necessary. To a specific application, how to dig out the data the application really needs is an important step. So called data mining or intelligent data processing is one of the important steps, which should be included as one part of the application. Besides, the application should have a specific algorithm or mechanism that uses or handles required data. For instance, in the application of finding a child's toy, the coordinate data is a kind of required data and can be obtained by such a RFID system. But how the location of the child toy is detected using this coordinate data is depended on an algorithm that the application adopts. The system does not limit the usage of the data.

It is worthwhile to emphasize that real and cyber spaces are synchronized by dynamically and real-time binding between RFID and XML data. With such synchronization between the two spaces, auto-login to a network might be possible. For instance, when a user moves from one room to another, the application changes the location data and the workgroup data in the XML at the same time. Even if a user does logout the current network, he/she can automatically login to the next network in another room as shown in Fig. 10.

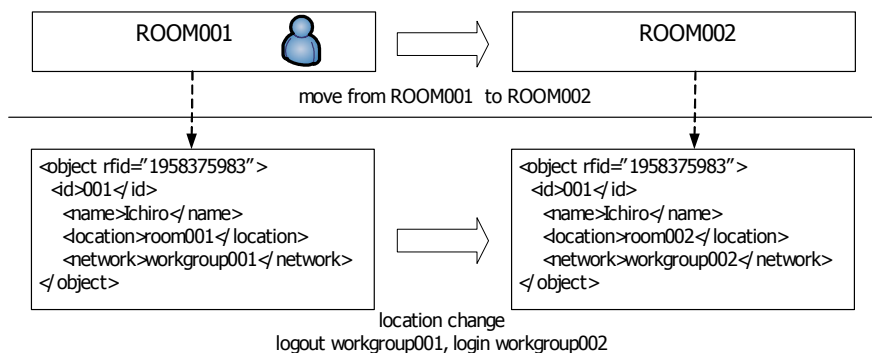


Fig. 10. The synchronization between a real space and a cyber space

7 Related Work

Mapping between real and cyber spaces by XML data binding is used in Robotic Communication Terminals [11]. The system focuses on barrier and barrier-free information on facilities at the National Institute of Information and Communications Technology. It converts object information on the real space obtained from the camera into XML, and does the mapping to a cyber space. Interactive Block System [12] also uses XML to map information between a real space and a cyber space. To support a variety of multimedia contents flexibly, the data is dynamically read. Compared to Robotic Communication Terminals and Interactive Block System, our research provides a common service framework for various RFID systems and Java applications. Semantic web and grid computing [13] use XML as metadata. The metadata technologies developed for the semantic web are applicable and relevant. XML data binding between real and cyber worlds needs to automate the management of the metadata. Our framework offers the service to manage the metadata by synchronizations between a real space and a cyberspace represented in XML. The applications greatly depend on situations and are hard to be developed one-by-one without a common framework. The Java Context Awareness Framework (JCAF) [14] is aimed at solving the problem that programmers develop context-aware applications.

8 Conclusions and Future Work

This research, as a part of UbiKids Project [15], is focused on making a general RFID service framework that assists the developments of various RFID based applications in which acquired information/data by RFIDs from a real space is used and processed. In the proposed framework, XML is used as a unified format to represent semantic data kept in a database that stores information about or from a real space. Such format makes the developments of Java based applications easier. Moreover, it can avoid the repeat work in developing RFID applications by providing common required services in any RFID based applications.

RFID is one kind of the important devices used to achieve the ubiquitous computing. Other devices such as the GPS, sensors, microphones, and cameras also play important roles in acquiring information from a real space. Moreover, these devices can be used to get other types of contextual information while RFIDs can keep only ID numbers. The interface that can control RFID as well as these devices is greatly in need and it will be added to our future work.

References

- [1] S. Yamada and E. Kamioka "An Overview of Researches on Ubiquitous Computing Networks", *NII Journal No.5*, pp. 41-47, 2003.
- [2] Kenpei Morishima, Takahiro Konno and Syuhei Watanabe "Location Information Technology by RFID", *NTT COMWARE TECHNOLOGY*, pp.12-15, 2004.
- [3] K. Okuda, S. Yeh, C. Wu, K. Chang, and H. Chu, "The GETA Sandals: A Footprint Location Tracking System", Proceedings in *Location- and Context-Awareness*, May 2005.
- [4] G. Borriello, W. Brunette, M. Hall, C. Hartung and C. Tangney, "Reminding About Tagged Objects Using Passive RFIDs", Proceedings in *UbiComp 2004*, September 2004.
- [5] K. P. Fishkin, B. Jiang, M. Philipose and S. Roy, "I Sense a Disturbance in the Force: Unobtrusive Detection of Interactions with RFID-tagged Objects", Proceedings in *UbiComp 2004: Ubiquitous Computing*, pp. 269-282, September 2004.
- [6] Calvin Austin and Monica Pawlan, "JNI Technology", *Advanced Programming for the Java 2 Platform*, pp.207-230, November 1999.
- [7] JAXB, <http://java.sun.com/xml/jaxb/>.
- [8] MSDN .NET, <http://msdn.microsoft.com/netframework/>.
- [9] J. Bishop, R. N. Horspool, B. Worrall, "Experience in Integrating Java with C# and .NET", <http://www.cs.up.ac.za/cs/jbishop/Homepage/Pubs/Pubs2002/Bishop-Integrating-1.pdf>.
- [10] J. Bishop, R. N. Horspool, B. Worrall, "Experience with Integrating Java with New Technologies: C#, XML and Web Services", www.cs.uvic.ca/~nigelh/Publications/ccpe03.pdf.
- [11] H. Fujiyoshi, M. Okada, T. Komura, I. Yairi, L. K. Kayama, H. Yoshimizu, "Robotic Communication Terminals as a Mobility Support System for Elderly and Disabled People", In *The 18th Annual Conf. of the Japanese Society for Artificial Intelligence*, 2004.
- [12] Y. Ito, Y. Kitamura, H. Kikuchi, K. Watanabe and K. Ito, "Development of Interactive Multimedia Contents Technology That Unites Real Space and Virtual Space in Seamless - Interactive Block System -", <http://www.ipa.go.jp/SPC/repo/rt/03fy-pro/mito/15-757d.pdf>.
- [13] Carole Goble and David De Roure, "Semantic Web and Grid Computing", in *Real World Semantic Web Applications*, vol. 92, *Frontiers in Artificial Intelligence and Applications*, V. Kashyap and L. Shklar, Eds.: IOS Press, 2002.
- [14] Jakob E. Bardram, "The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications", Proceedings in *Pervasive Computing*, pp.98-115, May 2005.
- [15] Jianhua Ma, L. T. Yang, B. O. Apduhan, R. Huang, L. Barolli, M. Takizawa, "Towards a Smart World and Ubiquitous Intelligence: A Walkthrough from Smart Things to Smart Hyperspaces and UbiKids", *Int'l Journal of Pervasive Comp. & Comm.*, 1(1), Mar. 2005.