The Design and Implementation of a Location-Aware Service Bundle Manager in Smart Space Environments*

Minwoo Son, Soonyong Choi, Dongil Shin**, and Dongkyoo Shin

Department of Computer Science and Engineering, Sejong University,
98 Kunja-Dong, Kwangjin-Ku, Seoul 143-747, Korea
{minwoo15, artjian, dshin, shindk}@gce.sejong.ac.kr

Abstract. Ubiquitous computing as the integration of sensors, smart devices, and intelligent technologies to form a "smart space" environment relies on the development of both middleware and networking technologies. Several kinds of smart space middleware have been developed and OSGi (Open Service Gateway Initiative) was initiated mainly for interoperability support with service distribution among various middleware environments. In this paper, we analyze the weaknesses in the OSGi service platform such as a non-distributed framework, passive user management, device management, and service bundle management. Moreover, we propose that in a smart space environment, Location-aware SBM (Service Bundle Manager) is most capable of efficiently managing various service bundles. This paper concludes with the implementation results for the SBM.

1 Introduction

Ubiquitous Computing means that users can use computers naturally and conveniently, regardless of place and time [1]. It means that a computer existing anywhere can use specialized services, and change its contents according to place or time via sensing and tracking. Its ability to form a "smart space" environment depends on the availability of networks, services, sensors, wireless communication and smart middleware technologies [2]. Smart space offers people security, energy saving, convenience, and a better lifestyle.

A smart space is a living and office environment in which devices can be accessed and controlled either locally or remotely. For example, a user may monitor the status of home appliances status from the office and switch them on/off as needed. When someone approaches the front door, smart middleware turns on the porch light. In short, smart services support people.

By connecting smart devices and intelligent networks, a smart space allows the user to access information efficiently and to connect directly to a range of public and personal services (including banks, police, fire, and emergency responders).

* This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea. (0412-MI01-0416-0002).

** Correspondence author.

Convenience and efficiency are maximized by controlling information-communication equipment, digital audio and video devices, other existing electronic devices, smart sensors, etc.

Middleware for a smart space controls home appliances, facilitates interaction among electronics, and supports various services. A variety of middleware for smart spaces has been developed, including UPnP (Universal Plug and Play) [3, 4], Jini [5, 6], HAVi (Home Audio Video Interoperability) [7, 8], IEEE 1394 [9, 10], and PLC (Power Line Communication) [11]. The shortcomings of smart space middleware are a lack of interoperability and difficulty of distributing new middleware-specific services. OSGi (Open Service Gateway Initiative) has been developed to overcome these problems by enabling the easy deployment of services for local smart spaces [12, 13].

OSGi is gradually extending its influence to the smart space middleware market, and electronic devices based on OSGi are being used. And to control home and office electronic devices based on OSGi, Service Bundles based on OSGi have been developed and also available. Therefore a user's need for an efficient manager has suddenly increased by several service bundles.

OSGi Spec. version 3 offers many services. For example it includes Framework for a service bundle manager and event processing, Log Service for logging information and retrieving current or previously recorded log information in the OSGi Service Platform, and Device Access Service for an electronic home appliance manager. However, the OSGi Service Platform does not support updating, installing, or removing for the active life-cycle of service bundles, and will not automatically check-in a device's state, or update a device driver, or distributed framework. Therefore we suggest SBM (Service Bundle Manager) to solve these problems.

This paper is composed of six sections. Section 2 and Section 3 introduce OSGi and describe a smart space architecture based on OSGi. In Section 4, we propose SBM to efficiently manage many kinds of service bundles based on OSGi and describe the related implementation results in Section 5. Finally we conclude in Section 6.

2 Background

Many kinds of projects are currently in progress with a shared perspective of exploring a new research agenda. Some of these projects are Easy Living [14], Smart-Its [15], SSLab [16], the Aware Home [17], and iRoom [18]. Microsoft's "Easy Living" project is developing prototype architectures and intelligent technologies which include multiple sensor modalities combine, automatic or semi-automatic sensor calibration and model building, and on the like for smart space environments.

Users use many smart devices that include each other's middleware in smart space. Therefore we implement smart space middleware with OSGi of smart space environment because OSGi supports communication among several pieces of middleware.

OSGi was created in 1999 in order to define an open standard for the delivery of services in networked environments, (vehicles and homes, for example.) and was supposed to solve problems involving the interaction among several kinds of home network middleware and service distribution. The OSGi service platform is an intermediary between the external network environment and the smart space network environment.

The OSGi service platform is divided into two parts: the OSGi Service Framework and OSGi Service. Figure 1 shows the OSGi Architecture. The OSGi framework supports service registry, life-cycle management of service bundles, etc. and includes registry service, persistent, data storage and life-cycle management for a servicein an extendable Java runtime environment. The OSGi framework supports an execution environment for services. An OSGi service is defined by a Java Interface. OSGi services include HTTP, Logging, and Device Access Service. A service is implemented as part of a bundle. A bundle is the smallest unit of management for the framework and, the framework manages its installing, uninstalling, resolving, stopping, starting, and active life-cycle. A bundle consists of java class code files and additional resources. In addition, the framework has Manifest File which is Meta information for the bundle or a bundle's install, start, and stop information. Several OSGi services may be included on a bundle, and form a standard unit of distribution and management.

Recently, research into the OSGi service platform suggests that a user in a smart space environment can turn appliances on and off. In other words, a smart space based on OSGi service platform supported a solid infrastructure so that projects could focus on unifying the smart space with smart phone and other smart applications [13].

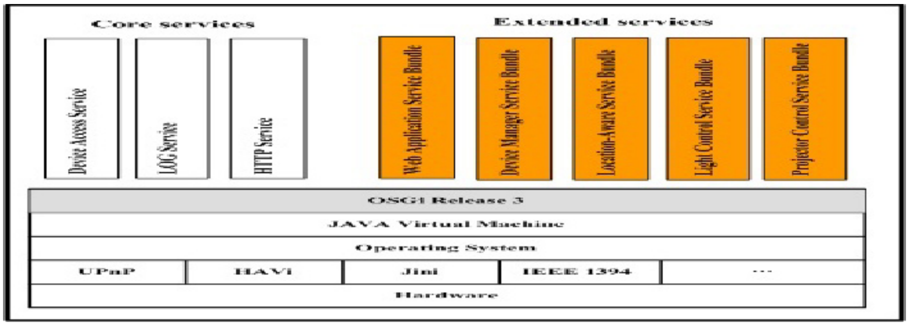

Fig. 1. OSGi Architecture

3 Smart Space Network Architecture Design Based on OSGi

OSGi, developed to resolve interoperability among pieces of equipment based on different middleware, may be activated in a smart space Network. Figure 3 is the design for a Smart Space Service Network Architecture to control devices that do not offer OSGi Device Drivers. When senses a new device, the Support Registry, the OSGi HTTP Service, is designed to support the control attachment of existing devices. For example, when DTV is a plug-in, it will confirm whether it is registered with the Service Registry and will support service or download drivers if necessary. To control devices based on other middleware (Jini, HAVi, UPnP, etc.), each device connects through service bundles that are driver modules based on other middleware and are implemented as Java packages. In Figure 2, the OSGi Transformation step checks that other device drivers for different middleware supports which kind of services. The service bundle is registered in the OSGi Service Registry step. Also, when other devices require communication, the bundle supports export services.

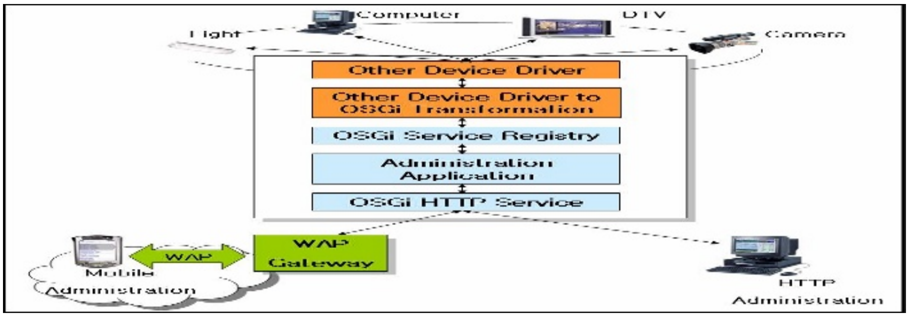

Fig. 2. Transform device based on other middleware into service based on OSGi in Smart Space Network Architecture

4 SBM (Service Bundle Manager) Design

4.1 Smart Space Gateway Architecture Design Based on SBM

Several service bundles will be used by a user, as a home network is widely used. Therefore users will need easier and more efficient manager service bundles. The OSGi service platform includes weaknesses for the management of service bundles. We compensated for the OSGi platform’s passive service element, user management, device manager component and non-distribution, etc. and designed SBM to efficiently manage several service bundles such as a Light Control Service and a Location-aware Service.

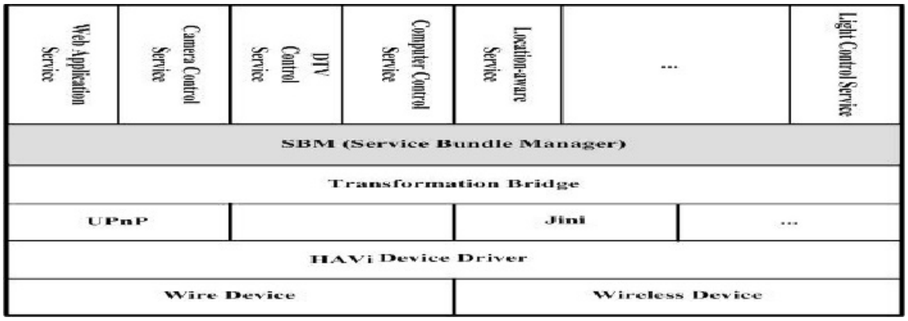

Fig. 3. Smart Space Gateway Architecture

Figure 3 shows the SBM-based Smart Space Gateway Architecture. The Device Driver and Wired/Wireless Device in the lower part decide the connection system among devices and certainly need standardization. Because the operating system uses programs like WinCE, embedded Linux, and real-time OS, it has less need of standardization. Connection systems for devices include wireless devices such as Wireless LAN, RFID (Radio Frequency Identification) [19, 20], Bluetooth [21, 22], and some of the wired devices consist of USB, IEEE 1394, and Ethernet. If a device physically connects to a smart space network, it connects the new device to middleware such as UPnP, HAVi and Jini, which automatically reconstruct the smart space network.

Transformation Bridge supports communication between middleware. When OSGi decides on supportable middleware, the home gateway uses the appropriate Transformation Bridge.

Like the OS in a computer, Windows, Linux and Max decide on applications for the computer system, SBM based on OSGi, which is a home gateway in a smart space network, supports home network services, when SBM connects devices inside or outside of the smart space. It is used to control service bundles such as the Web Application Service, Camera Control Service, and the Device Manager Service. SBM solves weaknesses in the service platform for OSGi Spec. version 3, such as passive service, User Management, Device Management and non-distribution.

4.2 SBM Architecture Design

Figure 4 shows SBM architecture. To control home appliances, a user uses two connection systems.

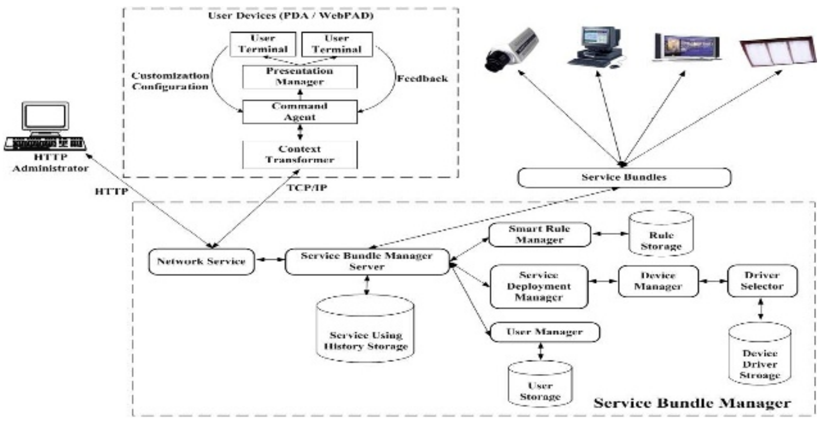

Fig. 4. Service Bundle Manager Architecture

The first method makes it possible for a user to control a service bundle in SBM, after the user is authenticated through a web browser.

The second method is a HIML (Human Interface Markup Language)[23] document that transmits using mobile devices such as PDA or Web PAD using a Network Service to Service Bundle Manager Server approach. A HIML document is stored to Service Using History Storage and the document is analyzed.

The HIML document pattern is made according to the data form of the HIML document to divide electronic devices, as seen in Figure 5, into image devices and sound devices; the DataSet is shown here.

To control electronic devices through using a web or mobile device, it accepts data on access privileges by a user's device in User Manager through a user ID if users approach. After the Device Manager receives an electronic device ID and device function services, it finds an appropriate driver through the device. Finally, users can control devices by service bundles.

When each service bundle starts, the Service Bundle Manager Server checks the Smart Rule Manager. The Smart Rule Manager then checks Rule Storage, which includes start-rule lists of service bundles. If the service bundle's start-rule exists, Smart Rule Manager sends Service Bundle Manager Server service bundle's start-rule. The Service Bundle Manager Server controls the service bundle through the service bundle start-rule. Rule Storage includes theses that support auto-aware system and are managed by the user (rule list installs, remove, modify, etc.).

Table 1. Storage Implements

Storage	**Implements**
Service Using History Storage	· stores HIML, which mobile device sends, to control device · makes each device table · connects HIML documents with Context Awareness data
User Storage	· stores users' personal information such as id, name, age, career, etc. · according to user id, allows certified user to access device control. · supports fitting device services to recognized user.
Device Driver Storage	· saves driver of each device. · driver always uses the latest version.
Rule Storage	· includes intelligent rule. · rules are inserted, removed, modified by user and auto-modify.

Table 2. Service Bundle Management Implements

Service	**Implements**
Network Service	· connects SBM through Web, TCP/IP · supports service that controls service bundle.
Service Bundle Manager Server	· manages execution life-cycle of service bundle (install, start, stop, resume, uninstall). · user manages service bundle through network. · collects service bundle's information · sends state information of service bundle to Administrator on schedule.
Service Deployment Manager	· links driver with fitting service bundle.
Device Manager	· manages each device's driver and service (addition/insert/remove). · periodically updates driver and stores driver in driver storage.
Driver Selector	· selects best suited to a driver service which user wants.
User Manager	· manages user's personal information (addition/insert/remove). · user level for limiting device control.
Smart Rule Manager	· checks service state of service bundles · sends Service Bundle Manager Server information of Rule Storage

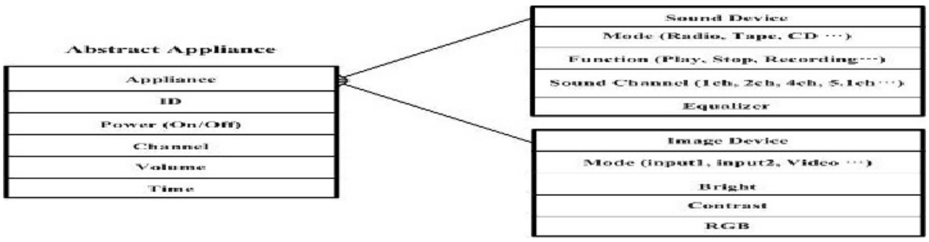
Abstract Appliance
Appliance
ID
Power (On/Off)
Channel
Volume
Time
Sound Device
Mode (Radio, Tape, CD ···)
Function (Play, Stop, Recording···)
Sound Channel (1ch, 2ch, 4ch, 5.1ch···)
Equalizer
Image Device
Mode (input1, input2, Video ···)
Bright
Contrast
RGB

Fig. 5. DataSet

5 Implementation of Service Bundles Based on SBM

This paper utilized an OSGi Release 3 compliant environment.

We suggest a scenario to test SBM in smart space. After SBM recognizes a person's location in the office through a Location-aware service bundle, and sends awareness-information to the Smart Rule Manager. Then the Service Bundle Manager Server turns light on and off through Light Control service bundle.

5.1 Light Control Service Bundle

Figure 6 shows the Light Control service bundle hierarchy. The Light Control service bundle turns the lights of homes and offices on and off. It can turn the light on and off automatically. SBM detects a user's location through a Location-aware service bundle. The Smart Rule Manager supports several services according to the user's location. For example, SBM automatically turns a light on or off from information provided by the Smart Rule Manager.

LightControl class controls which light goes on or off. After TimeZone class checks time, according to time (AM/PM), but only Light turn off at AM and turn on at PM automatically and at the same time Light Control Service Bundle is controlled light on/off by user.

SBM manages the transmission of data between Light Control Service and Client through the Client class to control the Light's Channel during the Light Control Service Bundle's run-time.

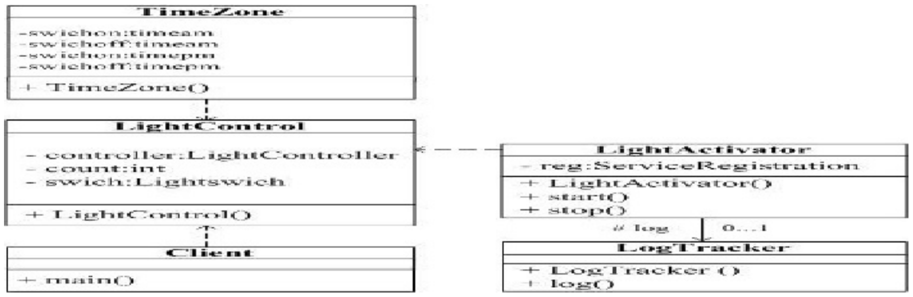

Fig. 6. Light Control Service Bundle hierarchy

5.2 Location-Aware Service Bundle

The Activator and Event Broker components approach the SBM framework, as shown in Figure 7. An Activator component, by interaction with the SBM framework, provides information on the Location-aware service bundle's start and stop from the administration interface of the framework. The Related Client in the Location-aware service bundle Event records in Event Broker. If Location-aware service bundle receives an Event, Event Broker sends an event object to the client. According to this method, a Location-aware service bundle shares Camera Handler information and Position Recognition information with other service bundles (ex. Light Control service bundle) at the same time by multi application in the SBM framework.

Location-aware service views a user's location in home and office in real-time through the application.

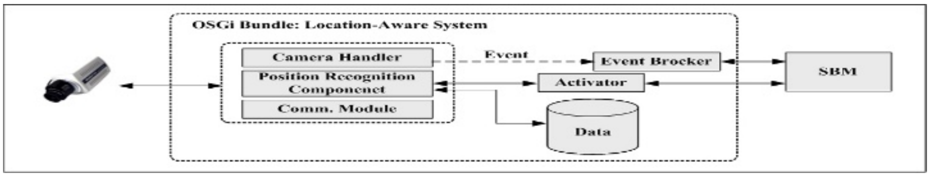
OSGi Bundle: Location-Aware System
Camera Handler
Position Recognition Componenet
Comm. Module
Event
Event Brocker
Activator
Data
SBM

Fig. 7. Location-aware Service Bundle Architecture

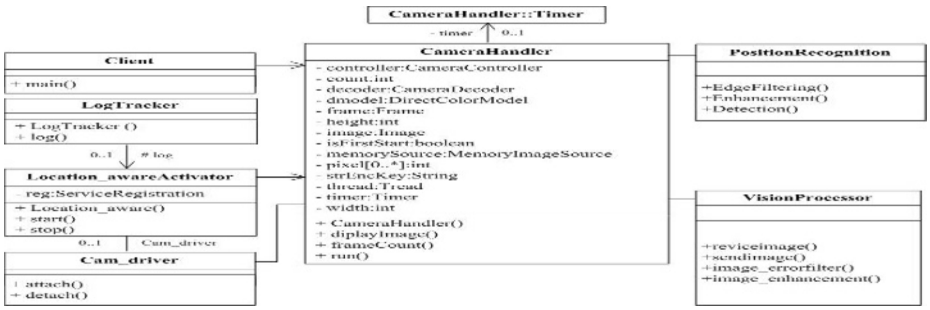

Fig. 8. Location-aware Service Bundle Hierarchy

Figure 8 shows each class' relation in Location-aware service bundle. Location_awareActivator class controls a bundle's states, such as start, install and stop. If calling start(), a service bundle scarcely starts when through CameraHandler, PositionRecognition and VisionProcessor classes perform. After CameraHandler class checks which camera attaches or detaches through Cam_driver, the service bundle receives the user's location information from PositionRecognition and VisionProcessor classes.

After Location-aware service bundle compare target picture, which is nobody in smart space environment, with real-time picture, the PositionRecognition and VisionProcessor class recognizes user's location in smart space.

LogTracker Class processes the recording of events and errors. The log() method logs a message with an exception associated with a specific service. The class starts as soon as the service bundle starts and if the service bundle stops, the class calls the log.close() method to stop. Figure 9 shows recognition of the user's location in smart space environment during the Location-aware service bundle's run-time. Then SBM turns light on through Light Control service bundle on the user's location.

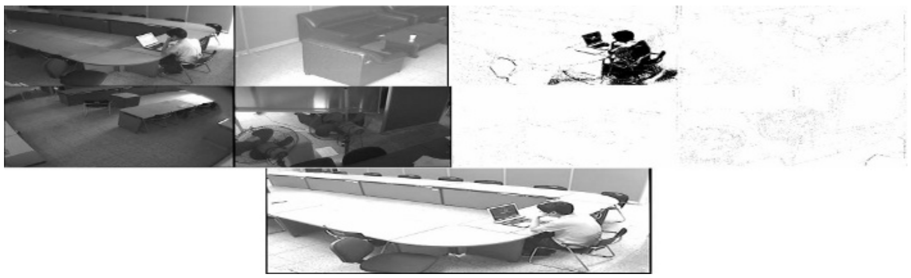

Fig.9. View Location-aware Service Bundle in Smart Space

6 Conclusion

This paper proposes SBM, which efficiently manages several service bundles based on OSGi and describes the execution of service bundles in an SBM environment.

SBM, which solves the OSGi service platform's weaknesses, such as User Management and Device Management, permits certified users to control each device and automatically designs a service for each device. After a user enters SBM using a web service and mobile device for the control of a device, SBM controls the sending of the device's service information, which analyses access privileges through User Manager and Device Manager, to the server. We did research on service bundles in a smart space system and on a manager for home appliances' control and user's location awareness service. SBM updates service bundles automatically and efficiently manages service bundles by managing a user's authorization and by controlling each device.

Future work will be done on a study of SBM to manage home appliances and service bundles, after extending its services such as context awareness, authenticated security and distribution. Smart Rule Manager parts and security still leave something to be desired.

References

1. Schulzrinne. H., Xiaotao. Wu, Sidiroglou. S., Berger. S.: Ubiquitous computing in home networks, Communications Magazine, IEEE, Vol. 41, Issue. 11, (2003) 128 - 135
2. George Alyfantis, Stathes Hadjiefthymiades, Lazaros Merakos: A Smart Spaces System for Pervasive Computing, EDBT 2004 Workshops, Vol. 3268, (2004) 375-384
3. UPnP Specification v1.0 homepage, http://www.upnp.org

4. Dong-Sung Kim, Jae-Min Lee, Wook Hyun Kwon, In Kwan Yuh: Design and implementation of home network systems using UPnP middleware for networked appliances, Consumer Electronics, IEEE Transactions on , Vol. 48, Issue. 4, (2002) 963 - 972
5. Jini Specification v1.0 homepage, http://www.jini.org
6. Landis. S., Vasudevan. V.: Reaching out to the cell phone with Jini, System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on, (2002) 3821-3830
7. HAVi Specification v1.1 homepage, http://www.havi.org
8. Lea. R., Gibbs. S., Dara-Abrams. A., Eytchison. E.: Networking home entertainment devices with HAVi, Computer, Vol. 33, Issue. 9, (2000) 35-43
9. IEEE 1394 Specification v1.0 homepage, http://www.1394ta.org
10. Nakagawa. M., Honggang Zhang, Sato. H.: Ubiquitous homelinks based on IEEE 1394 and ultra wideband solutions, Communications Magazine, IEEE, Vol. 41, Issue. 4, (2003) 74 - 82
11. Ferreira. H.C., Grove. H.M., Hooijen. O., Han Vinck, A.J.: Power line communications: an overview, AFRICON, IEEE AFRICON 4th, Vol. 2, (1996) 558 - 563
12. OSGi Specification v. 3.0, March.2003 homepage, http://www.osgi.org
13. Choonhwa Lee, Nordstedt, D., Helal, S.: Enabling smart spaces with OSGi, Pervasive Computing, IEEE, Vol 2, Issue 3, (2003) 89-94
14. Berners-Lee, T., Answers of Young People, http://www.w3.org/People/Berners-Lee/Kids
15. The Aware Home, http://www.cc.gatec.edu/fce/ahri
16. BrumittB., Meyers, B., Krumm, J., Kern, A. and Winograd T.: The interactive workspaces project: experiences with ubiquitous computing rooms", IEEE Pervasive Computing, Vol. 1, Issue. 2, (2002) 67-74
17. The Smart-Its Project, htpp://www.smart-its.org/
18. Smart Space Laboratory, http://www.ht.sfc.keio.ac.jp/SSLab
19. Radio Frequency Identification (RFID) homepage, http://www.aimglobal.org/technologies/rfid
20. Want. R.: Enabling ubiquitous sensing with RFID, Computer, Vol. 37, Issue. 4, (2004) 84-86
21. Bluetooth homepage, http://www.bluetooth.com
22. Kwang Yeol Lee, Jea Weon Choi: Remote-controlled home automation system via Bluetooth home network, SICE 2003 Annual Conference, Vol. 3, (2003) 2824-2829
23. Gunhee Kim, Dongkyoo Shin, Dongil Shin: Design of a Middleware and HIML(Human Interface Markup Language) for Context Aware Services in a Ubiquitous Computing Environment, EUC 2004, (2004) 682-691