U-TOPIA: A Ubiquitous Environment with a Wearable Platform, UFC and Its Security Infrastructure, pKASSO

Kyu Ho Park, Ki-Woong Park, Jupyung Lee, Jong-Woon Yoo, Seung-Ho Lim, and Hyun-Jin Choi

Computer Engineering Research Laboratory,
School of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology
{kpark,woongbak,jplee,jwyoo,shlim,hjchoi}@core.kaist.ac.kr

Abstract. U-TOPIA, introduced in this paper, is an advanced ubiquitous computing environment mainly focusing on a university campus. Research in U-TOPIA spans various components each of which is essential to realize U-TOPIA: from user device hardware/software, user interface, communication technology, indoor/outdoor testbed, middleware to practical applications and security infrastructure. We designed and implemented a wearable platform from the scratch and make use of it as a main user device inside U-TOPIA. In addition to this, as a new user interface, we developed a wireless gesture recognition device, called i-Throw to communicate with U-TOPIA in an intuitive manner. For data communication and location tracking in U-TOPIA, campus-wide indoor and outdoor testbed was deployed. To keep up with secure and dynamic U-TOPIA environment, a new security infrastructure,called pKASSO, and extensible middleware, μ-ware, was developed. Finally, as a practical application for U-TOPIA, we implemented a ubiquitous testbed room where multiple users interact with various ubiquitous devices or other users securely in a user-friendly manner. Integrating these components all together, we show that U-TOPIA can be a realistic role model to improve current paradigm of ubiquitous computing environment one step forward within a few years.

1 Introduction

Ubiquitous and pervasive computing have had a wide ranging influence in the ideas of how the future would look like. Ubiquitous computing, already described in the early 1990 by Mark Weiser, can be seen today in various forms [1,2,3,4]. In recent years, the rapid progress of ubiquitous and pervasive computing technology, fueled by either a government-led or a company-led efforts to encourage the research, has led to the emergence of various 'smart' places powered by ubiquitous computing technology, ranging from smart room and smart campus to smart city.

M. Mühlhäuser et al. (Eds.): AmI 2007 Workshops, CCIS 11, pp. 183–193, 2008.

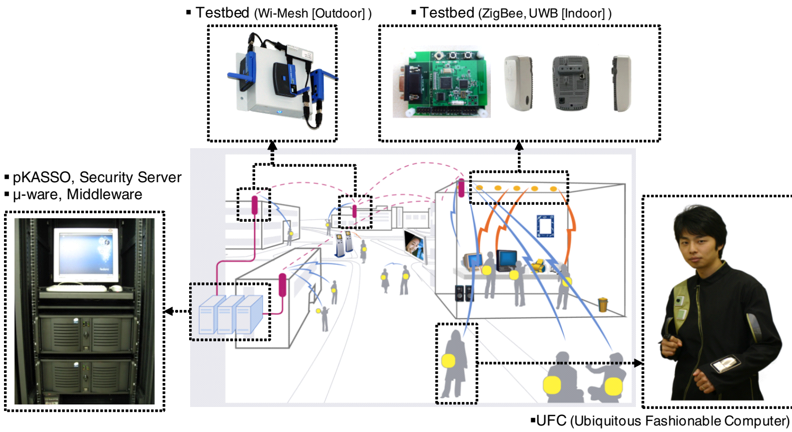

Fig. 1. Overall Architecture of U-TOPIA

In 2005, our team launched a government-funded project aimed at realizing a campus-wide advanced ubiquitous computing environment until the end of 2007. The ubiquitous computing environment was named as *U-TOPIA*, where 'U' stands for 'ubiquitous' and 'TOPIA' stands for 'place' in Greek. In order to realize U-TOPIA, we developed a wearable computer that allows people to exploit ubiquitous computing environment in a user-friendly manner and security infrastructure for secure ubiquitous services based on our testbed and middleware called μ-ware. The most important contribution of U-TOPIA can be summarized as follows;

- **UFC:** A wearable computer with modularity and extensibility is implemented based on ARM9 processor and embedded LINUX. Our hardware platform is called Ubiquitous Fashionable Computer (UFC), which is named based on our special emphasis on its wearability, aesthetic design and close interaction with ubiquitous environment.
- **i-Throw:** A new human computer interface called i-Throw is invented to throw/get information in one's UFC to the other UFC(s) or Objects when a throwing/getting action is taken by his hands.
- **μ-ware:** It is composed of light-weight service discovery protocol, distributed information sharing, context manager and instance service loader, all of which are useful to manage dynamic data and to develop new application that utilizes various ubiquitous resources.
- **pKASSO:** A new PKI-based security mechanism and security infrastructure, pKASSO, is devised for secure and seamless transaction between UFC and U-TOPIA where various devices frequently interact with each other or a surrounding infrastructure.

In addition to this, for data communication and location tracking in U-TOPIA, campus-wide indoor and outdoor testbed was installed. These efforts are closely related with realizing U-TOPIA in our campus.

2 U-TOPIA Internal

Overall architecture of U-TOPIA is described in Fig. 1. U-TOPIA consists of various components each of which is essential to realize U-TOPIA: from wearable platform (UFC), user interface (i-Throw), middleware (μ-ware) and security server (pKASSO) to practical and secure applications, and indoor/outdoor testbed. We have tried to push each research field a step forward to meet the ambitious goal, realizing a campus-wide advanced ubiquitous computing environment.

2.1 Indoor and Outdoor Testbed

Since U-TOPIA aims for a campus-wide environment, it is essential to build a large-scale testbed inside which various services can be operated. Two important components of a target testbed are a communication infrastructure and a location-tracking infrastructure.

Communication infrastructure lays the groundwork for ubiquitous computing. Communication inside U-TOPIA is made possible by wireless mesh network that is installed inside our campus. Location-tracking infrastructure is necessary for a location-based service. In outdoor environment, we made use of widely-used GPS information to track the location of moving objects. In indoor environment, we basically used ZigBee signal strength-based location tracking mechanism. To do this, we installed enough number of ZigBee sensor nodes inside two selected buildings inside U-TOPIA. During the measurement, however, we found that the resolution of location sensing using this mechanism was not sufficient for our target application. Thus, we also utilized UWB-based location tracking device[13] whose typical accuracy is 6 inches(15cm). Due to the high cost of this solution, UWB-based location tracking device have installed in only two rooms inside U-TOPIA.

2.2 Middleware, μ-Ware

In U-TOPIA, we assume a situation where thousands of users move here and there, interact with each other or ubiquitous computing environment, share information with authorized other users, access to diverse devices for diverse purposes, run various location-based applications. In this situation, an extensible middleware framework is necessary to keep up with highly variable dynamic environment. We have been working with middleware team and they developed a extensible middleware, called μ-ware[11]. μ-ware is composed of light-weight service discovery protocol, distributed information sharing, context manager and instance service loader, all of which are useful to manage dynamic data and develop new application utilizing various ubiquitous resources.

2.3 Wearable Platform, UFC

In U-TOPIA, a hardware platform is necessary to provide a user with plentiful ubiquitous computing resources. The hardware platform should be light-weight,

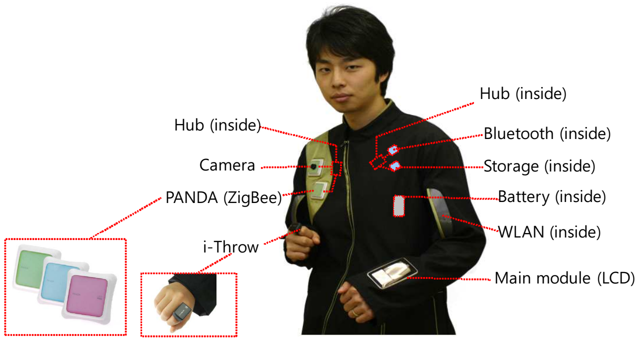

Fig. 2. UFC Platform Design and Implementation

easy to carry, easy to use and it should have aesthetic appearance and social acceptance. We chose to design and implement a wearable platform from the scratch and make use of it as a main user device inside U-TOPIA. Our wearable platform, in contrast to either laptop PC or handheld device, allows a user to carry the computing device in a comfortable and natural manner, because clothes have been already a essential component of our daily lives. Large surface area of clothes can be utilized for various purposes and thus, I/O interface does not have to be located in only a small-sized computing device. Moreover, the wearable platform makes it easier to measure and gather bio signal data such as temperature and heart rate by integrating body-attached sensors and computing devices upon a same clothes interface.

The implemented UFC platform is shown in Fig. 2. UFC modules are distributed on a garment, considering the distribution of weight and aesthetic design. Moreover, each UFC module can be attached and detached easily on a garment, allowing users to construct one's own UFC platform. Since we utilized a standard USB protocol to communicate between the main module and various UFC modules, due to the hotswap capability of USB devices, each UFC module can be attached and detached while the system is running.

The success of wearable computer relies on not only wearability, but also the aesthetic appearance and social acceptance. We tried to find the solution to fulfill the requirements by repeating the prototyping bodystorming progresses. We defined the target users as young university students and drew design concepts by analyzing their activities in everyday life and fashion trend. In addition, we have made effort for each part of the UFC platform to look like familiar fashionable components: for example, an attachable/detachable module is comparable to a button of clothing and an i-Throw device is comparable to a ring. Also, PANDA can be worn as a form of a necklace.

2.4 User Interface, i-Throw

In U-TOPIA, since plentiful resources locate outside a user, rather than inside a user device, a brand-new user interface is required to manipulate various outside

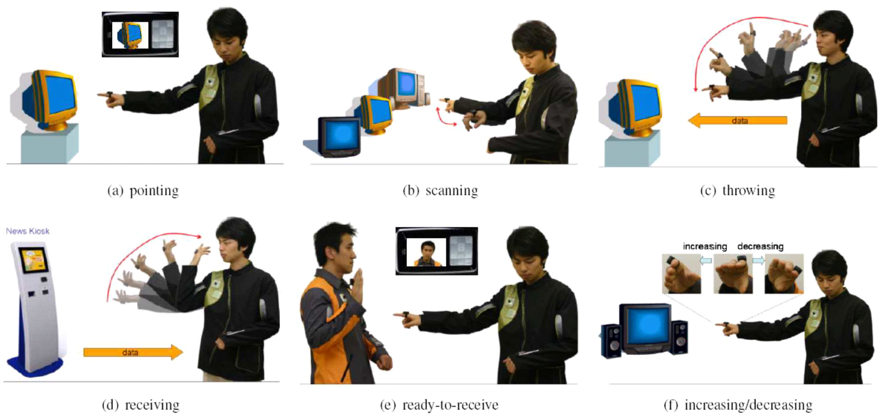

Fig. 3. Gesture Sets of i-Throw

resources in a user-friendly manner. To meet this requirement, we developed a wireless gesture recognition device, called i-Throw. Using this device, a user can express one's intention easily by using one's spatial movement and hand gesture. It is small enough to be worn on one's finger like a ring and has a three-axis accelerometer and a three-axis magneto-resistive sensor for recognizing a gesture and the direction of the finger. It also has a ZigBee transceiver for transmitting the recognized gesture information to the UFC platform.

The gesture recognition consists of two stages: *feature extraction* stage and *testing* stage. The feature extraction stage is a preprocessing stage to find reference features of each gesture. A feature f is represented by a four dimensional vector as follows:

$$f = (A_{THx}, A_{THy}, A_{THz}, T_H), \tag{1}$$

where A_{THx}, A_{THy} and A_{THz} are the acceleration thresholds of each axis and T_H is time duration threshold. In the feature extraction stage, we should find appropriate thresholds for each possible input gesture. Due to the limitation of space, we omit the detailed explanation of the feature extraction stage. In the testing stage, i-Throw device compares the output of the accelerometer with each reference feature for over T_H seconds. If one of the features is matched, then i-Throw transmits the recognized gesture to the UFC platform via ZigBee interface. The gesture recognition algorithm is designed to be simple enough to run on a micro-controller inside the i-Throw by extracting the minimum set of required features and using threshold-based simple features. We summarized and illustrated the gesture sets that the i-Throw recognizes in Fig. 3. Other possible gestures, scrolling up/down and canceling, are intentionally omitted here. Every time a UFC user points to a device, the UFC platform displays the selected target device upon its screen. This feedback information helps the UFC user find the correct target device. Similarly, a scanning gesture allows the user to investigate controllable devices inside the room. This scanning operation is

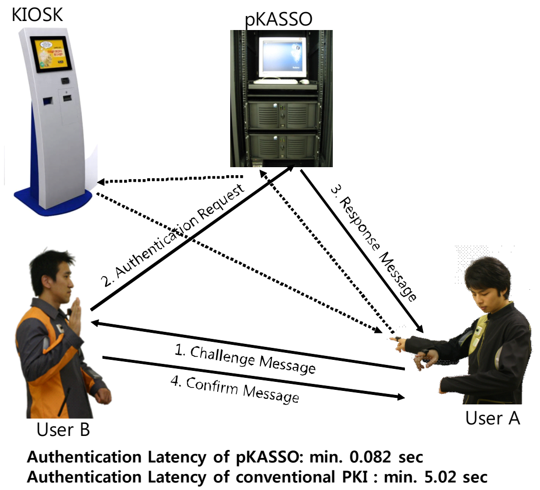

Fig. 4. Authentication flow description with pKASSO

similar to the operation of moving a mouse pointer across several icons in a typical PC desktop environment. 'Ready-to-receive' gesture is necessary for a UFC user to express one's intention to receive other UFC users' objects. When one user makes a pointing or scanning gesture, only limited users who makes the 'ready-to-receive' gesture can be selected.

2.5 Security Infrastructure, pKASSO

In order to provide a full-fledged security solution especially tailored for U-TOPIA, wherein numerous devices and sensors with severe resource-constraints interact with each other, we developed an computationally efficient PKI-based security infrastructure, pKASSO enhanced with single sign-on and delegation technology. It enables a cost-effective but uncompromisingly secure development of UFC. The delegation mechanism of pKASSO makes it possible to offloads complex cryptography operations from UFC to server-side so that it significantly improves authentication latency as well. According to the performance evaluation, the authentication latency (Avg. 0.082sec) is much shorter than a contact type smart card (Avg. 4.31sec) and a conventional PKI-based authentication latency (Avg. 5.01sec) [5]. Overall process of authentication based on pKASSO is illustrated in Fig.4.

1. User A sends a challenge message to user B who wants to communicate with user A.
2. User B generates an authentication request message (two symmetric key operations) and sends it to pKASSO. and it performs transactions for verification and authentication on behalf of user B.

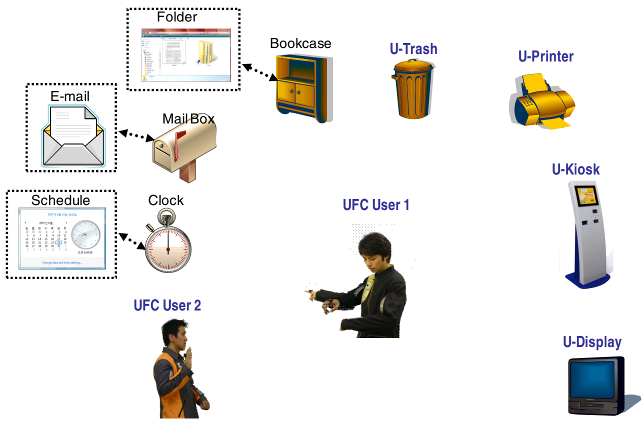

Fig. 5. The Concept of the Ubiquitous Testbed Room

3. pKASSO makes a response message and transmits it to user A.
4. The authentication is completed with the arrival of a confirming message from user B.

3 Target Application: User-Friendly Interaction with U-TOPIA

As mentioned in Section 2.4, we took 'user-friendly interaction with ubiquitous devices using i-Throw' as a target application. To execute this application, we have implemented a ubiquitous testbed room where multiple UFC users interact with various ubiquitous devices or other UFC users. Fig. 5 illustrates the concept of the ubiquitous testbed room which present a practical application that runs upon the UFC platform and the ubiquitous devices in testbed, which makes it possible to exchange the various objects and control ubiquitous devices very easily.

3.1 Motivation

Due to its small form factor, most portable devices, including our UFC platform, have only small-sized display and limited input devices. The UFC platform has 2.5" LCD display and 12 input buttons, which are definitely insufficient to monitor the status of a UFC main module and various peripheral modules, control the modules, and send a user's intention to the UFC platform.

This problem is exacerbated when a UFC user tries to control various ubiquitous devices using one's UFC platform: as the number of controllable ubiquitous devices increases, it becomes more inconvenient to find one among them and exchange information with it, due to the small-sized display and limited input devices of the UFC platform.

We attempt to resolve this problem by making full use of spatial resources inside the testbed room: given that various ubiquitous devices are spatially distributed inside the testbed room, a UFC user can express one's intention easily by using one's spatial movement and gesture. For example, let us assume that one UFC user takes a picture and intends to put it on a public display so that other people can see the picture he takes. From the perspective of the user, the most natural way of reflecting one's intention on the environment is pointing his finger at the public display and throwing one's picture at the public display. If this kind of a user-friendly spatial gesture interface is supported, the limitation of the I/O resources of the UFC platform can be overcome by fully utilizing abundant spatial resources.

For the UFC platform to support such a practical and secure interface, the following components are necessary:

– **gesture recognition device** recognizes the target device that a UFC user is pointing at and the gesture such as 'throwing' and 'receiving'.
– **location tracking device** keeps track of a UFC user's location. This is necessary because finding the target device that a UFC user is pointing at is dependent on the absolute location of the UFC user. We utilized UWB-based location tracking device[13] whose typical accuracy is 6 inches(15cm).
– **location server** gathers and manages the location information of both UFC users and ubiquitous devices. When one UFC user points at a specific device, the recognized gesture information is sent to the location server and it finally decides what the target device is.
– **Mutual Authentication and secure key distribution**: In order to provide secure and seamless transaction between UFC and U-TOPIA where various devices frequently interact with each other or a surrounding infrastructure, pKASSO is deployed.
– **Application that runs upon UFC platform** infers a UFC user's intention based on a gesture, a target device and previous operations and conducts a corresponding operation.

3.2 Detection of Target Device

Our target detection algorithm is based on Cone selection which is used in virtual computing environments[9]. A cone is cast from i-Throw and a set of devices that intersect with it are chosen. Additionally, we have modified that typical cone selection algorithm to vary the area of the cone adaptively to improve the overall target detection accuracy. To do this, the orientation of i-Throw and the position of the UFC user and devices should be known. The location information is gathered and managed by *virtual map* of μ-ware. The *virtual map* contains interactive objects around a user in a specific ubiquitous environment and supports several services as follows:

– **Interactive Object Registration:** For fast spatial queries, the repository of *virtual map* maintains a current snapshot of the interactive objects.

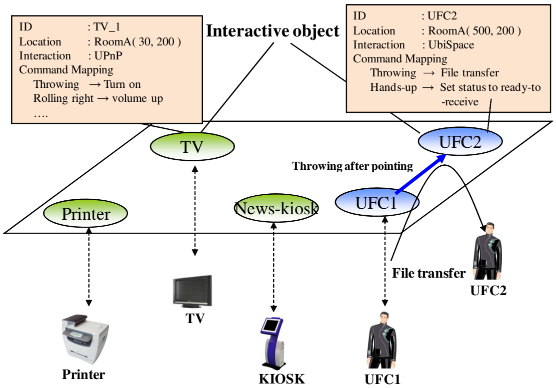

Fig. 6. Virtual map and interactive objects

- **Interactive Object Discovery:** When UFC detects logical address change, the virtual map repository is resolved by logical address key.
- **Target Selection in a Virtual Map:** UFC can see the target in the given boundary space with objects with priorities.

The scope of the virtual map is automatically adjusted upon user's location(inside building or outdoors). In the virtual map, each interactive object is tagged with geographic information and attributes to interact with. An interactive object is an abstraction unit of physical object on the virtual map. Interactive object contains location, and service attributes to interact with. Figure6 shows the virtual map and interactive objects. In Figure 6, the UFC1 has information about UFC2's location, interaction method, and the command configuration that assigns throwing motion as transferring a file. And the orientation of i-Throw can be obtained by combining the accelerometer and magnetic sensor outputs. The accelerometer is used for tilt compensation. By using the orientation and position information, target detection can be performed properly.

We define the target device sets which consist of possible target devices in the ubiquitous testbed and the characteristics of each one, which are summarized in Table 1. Among these devices, the *news kiosk* automatically gathers recent news

Table 1. Target Device Sets

Target device	Supported objects	Flow of objects
U-Display	News, Photo	Input, Output
U-Kiosk	News	Output
U-printer	News,Photo	Input
U-trash	News,Photo,music	Input
Clock, Mail, Bookcase	Supported Application	Input, Output
UFC	news,photo,music	Input, Output

from internet web sites and displays each one, that is refreshed every 10 seconds. When one UFC user sees the interesting news upon the news kiosk, he or she can obtain the news by making 'receiving' gesture towards the news kiosk. The *u-trash* functions as a symbol of 'deleting a file'. Similarly to the natural way of using an actual trash, throwing something that is useless anymore at the trash, if one UFC user makes a throwing gesture towards the u-trash, the current object will be deleted automatically. The u-trash device is different from other devices in that it is not an electric device; it acts only as a marker standing for a particular operation and thus actual operation is not conducted inside it. This example gives us insight as to how to fully utilize spatial resources inside the room. If various markers whose symbolic meaning can be easily interpreted are added inside the room and each corresponding operation is efficiently conducted, the spatial gesture interface allows UFC users to conduct various operations in a user-friendly manner.

4 Conclusion

U-TOPIA, introduced in this paper, is a campus-wide advanced ubiquitous computing environment. We summarized five essential components to realize U-TOPIA paradigm in our campus: indoor/outdoor testbed, middleware (μ-ware), wearable platform (UFC), user interface (i-Throw), security infrastructure (pKASSO). Since 2005, we have tried to push each research field a step forward to meet the ambitious goal, realizing a campus-wide advanced ubiquitous computing environment. Specifically, we designed and implemented a wearable platform from the scratch and make use of it as a main user device inside U-TOPIA. In addition to this, as a new user interface, we developed a wireless gesture recognition device, called i-Throw. To keep up with secure and dynamic U-TOPIA environment, a new security infrastructure,called pKASSO, and extensible middleware, μ-ware, was developed.

In addition to this, for data communication and location tracking in U-TOPIA, campus-wide indoor and outdoor testbed was installed. Finally, as a practical application for U-TOPIA, we implemented a ubiquitous testbed room where multiple users interact with various ubiquitous devices or other users in a user-friendly manner. All these efforts are closely related with realizing U-TOPIA in our campus. We believe that U-TOPIA can be a realistic role model to improve current paradigm of ubiquitous computing environment within a few years. As a future work, we will attempt to apply our indoor application to outdoor testbed, so that a user can extract suitable information from outdoor components, such as building, billboard, large display and vehicles.

References

1. Mann, S.: Wearable computing as means for personal empowerment. In: Proc. 3rd Int. Conf. on Wearable Computing (May 1998)
2. Pentland, A.: Wearable intelligence. Proc. Scientific American 1, 276 (1998)

3. Starner, T.: The challenges of wearable computing: Part 1. In: Proc. IEEE Micro, July 2001, pp. 44–52 (2001)
4. Weiser, M.: The computer for the 21st century. In: Proc. Scientific American, September 1991, vol. 3, pp. 66–75 (1991)
5. Park, K.W., Seok, H.C., Park, K.H.: pKASSO: Towards Seamless Authentication providing Non-Repudiation on Resource-Constrained Devices. In: Proc. 21st IEEE Symposium on Pervasive Computing and Ad Hoc Communications, May 2007, vol. 2, pp. 105–112 (2007)
6. Lee, J., Lim, S.H., Yoo, J.W., Park, K.W., Choi, H.J., Park, K.H.: A Ubiquitous Fashionable Computer with an i-Throw Device on a Location-Based Service Environment. In: Proc. 21st IEEE Symposium on Pervasive Computing and Ad Hoc Communications, May 2007, vol. 2, pp. 59–65 (2007)
7. IEEE802.15.2 Specification, Coexistence of Wireless Personal Area Networks with Wireless Devices Operating in Unlicensed Frequency Bands (August 28, 2003)
8. Back, M., Lahlow, S., Ballagas, R., Letsithichai, S., Inagaki, M., Horikira, K., Huang, J.: Usable Ubiquitous Computing in Next-genration Conference Rooms: Design, Evaluation, and Architecture. In: Proc. UbiComp 2006 Workshop, September (2006)
9. Liang, J., Green, M.: JDCAD: A Highly Interactive 3D Modeling System. Proc. Computers & Graphics 4, 499–506 (1994)
10. Kang, M., Chong, J., Hyun, H., Kim, S., Jung, B., Sung, D.: Adaptive Interference-Aware Multi-Channel Clustering Algorithm in a ZigBee Network in the presence of WLAN Interference. In: Proc. IEEE International Symposium on Wireless Pervasive Computing, Puerto Rice (February 2007)
11. Song, Y., Moon, S., Shim, G., Park, D.: μ-ware: A Middleware Framework for Wearable Computer and Ubiquitous Computing Environment. In: Proc. 5th Conference on Pervasive Computing & Communications(PerCom 2007) (March 2007)
12. UPnP Forum, UPnP Device Architecture 1.0, Version 1.0.1 (December 2003)
13. Ubisense, `http://www.ubisense.net`
14. KAIST UFC Project, `http://core.kaist.ac.kr/UFC`