A Ubiquitous Fashionable Computer with an i-Throw Device on a Location-based Service Environment

Jupyung Lee, Seung-Ho Lim, Jong-Woon Yoo, Ki-Woong Park,
Hyun-Jin Choi and Kyu Ho Park
Computer Engineering Research Laboratory
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
{jplee, shlim, jwyoo, woongbak, hjchoi}@core.kaist.ac.kr, kpark@ee.kaist.ac.kr

Abstract

The ubiquitous fashionable computer(UFC), introduced in this paper, is a wearable computer that allows people to exploit ubiquitous computing environment in a user-friendly manner. We present the design approach and philosophy of the UFC that is wearable, aesthetic, and intuitive . The UFC supports the interoperability of various communication interfaces among WLAN, Bluetooth and ZigBee devices. We developed a wireless gesture recognition device, called $i - Throw$, which is small enough to be worn on one's finger like a ring. The UFC, with the help of $i - Throw$, can control ubiquitous environment using an intuitive hand motion. To explain the practical use of the UFC platform and the user-friendly interaction with ubiquitous environment, we implemented a ubiquitous testbed where multiple UFC users interact with various ubiquitous devices or other UFC users. In addition, we implemented a practical application which makes it possible to exchange the various objects and control ubiquitous devices very easily.

1. Introduction

In recent years, wearable computing and ubiquitous computing system environment have realized due to the rapid progress of computing and communication technology. Wearable computing system can be broadly defined as mobile electronic devices that can be unobtrusively embedded as part of garment or accessory [1]. Unlike conventional mobile devices, it is always active and running without user's attention and gives services to a user with the support of ubiquitous environment. In the ubiquitous environment, using the wearable computer, we get necessary information anytime and anywhere with a minimal constraint.

In a wearable computer, the design of new paradigm has concentrated on the problems that arises when user has to put up with considerable inconvenience carrying around devices [2][3][4][5]. The success of wearable computer will heavily rely on good wearability, usability, aesthetic appearance, and social acceptance. In addition, not merely considering about these uncomfortable carrying issues, the reflection about the exploit of ubiquitous computing environment should be done. The emerging wearable ubiquitous computer should have various communication interfaces, such as WLAN, Bluetooth, and ZigBee communication to support various ubiquitous network environment.

In addition to that, a novel user interface should be considered to minimize user inconvenience while users access and utilize the system resource. The uncomfortable interface means that users should get training its function with burdensome input or output devices using wearable keyboard and mouse. In wearable computing, it is desired to develop comfortable and user friendly input devices for wearable computing. A novel user interface is required to be simple, easy and intuitive by recognizing human friendly gestures, activities, or senses. Intuitive interface can be described as the mechanism of controlling devices in a ubiquitous environment by human friendly gestures which everyone can easily accept and recognize. Human voice, eyes, and hand gesture are good means to realize input devices as user interfaces in wearable computer.

In this paper, we present the design approach and the philosophy of our wearable computing system that is wearable, aesthetic, and intuitive. In our wearable computer system, novel user interfaces are developed and integrated to help the intuitive use of it. Among possible user interfaces, in this paper, we focus on what we call $i - Throw$, which means intuitive input devices. This input device can recognize a user's hand gestures and direction, and allows the user to control ubiquitous devices with the gestures. Our wearable computer 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. Our UFC system is realized and tested with actual testbed environment that has various network interfaces, sensor nodes and ubiquitous components. Moreover, various applications are implemented and presented.

2. UFC System Architecture

The UFC system consists of portable wearable computer with various communication interfaces and user interfaces, and ubiquitous environment. The basic design concept of UFC wearable computer is modularity and extensibility.

2.1. UFC Platform

(a) UFC Platform

(b) i-Throw

(c) UFC Modules

Figure 1. UFC Platform Design and Implementation

Our UFC consists of several module parts: main module including CPU and Memory, communication modules including various communication interfaces, and user interface modules with I/O interfaces.

The implemented UFC platform is shown in Figure 1. 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, some wireless module can be worn as a form of a necklace. Moreover, as an extreme case, we made a ZigBee earring that has only a ZigBee transceiver and several LEDs, which is tiny enough to be worn as a form of a earring.

In communication modules, major theme is to implement coexistence and multimodal communication among WLAN, Bluetooth, and ZigBee RF communication. WLAN is used for network services such as internet, ftp, and e-mail, and Bluetooth is for wireless headset, mouse, and keyboard. The ZigBee is adapted for the location based service(LBS), which can be operated with ubiquitous sensor network systems. There are several studies on the interference and coexistence problems among communication devices that use the 2.4GHz ISM band. Up to now, many coexistence models were presented in terms of WLAN and others. In case of the interference between WLAN and Bluetooth, various problems were discussed in [12]. However, the coexistence problems between WLAN and ZigBee are not widely addressed yet. The study about coexistence problems between WLAN and ZigBee is very important because in a near future, ZigBee is expected to be widely applied in various communication environment, such as wireless sensor networks, personal area networks and body area networks. To minimize the interference between communication devices, we have designed and implemented the dynamic channel allocation algorithm in device driver and application layers that use communication interface devices. For that, we investigate the frame error rate(FER) for each communication device, and dynamically allocate the appropriate channel of ZigBee devices considering the channel status. A more detailed explanation is provided in [15].

Operating system running on UFC is GNU/embedded Linux with kernel 2.6. Linux 2.6 with ARM processor shows more deemed feasible performance in real time embedded system than lower versions. Efficient middleware platform is implemented with UFC to provide various helpful services with low overhead and power. The middleware interface can be implemented with standard Java execution environment called Java Native Interface(JNI)[10]. Useful functions which are covered by middleware include context management, service discovery, and local file sharing.

2.2. UFC Ubiquitous Environment

The interoperability of mobile devices in ubiquitous environment is an important issue because various network environments are deployed. The wireless sensor network based on IEEE 802.15.4[11] is one of the possible technologies to provide the location sensing and mobility to mobile users. The sensor nodes are installed in a regular manner for location sensing of users, and three types of servers are installed for the management : location management, network management, and application management.

The connection between servers and users could be done via multi-hop sensor nodes for low speed data transmission, and WLAN access point for high speed data transmission, respectively. The MAC layer of sensor node is implemented as ZigBee which supports IEEE 802.15.4. ZigBee network. For the location sensing, the Sensor node broadcasts beacons periodically, which uses 2.4GHz as a physical channel. The user who wears UFC wearable computer receives beacon signals from the ZigBee communication interface. When the user receives multiple beacons from the multiple sensor nodes, the users who have received the beacons can identify his location by calculating each Received Signal Strength Indicator(RSSI) value from each sensor node. For high speed data transmission service such as file transfer and multimedia services, UFC wearable computer can transmit or receive via WLAN network using WLAN access pointer, which also uses 2.4GHz as a physical channel.

3. Application Example: user-friendly interaction with ubiquitous environment using i-Throw

To explain the practical use of our UFC platform and the user-friendly interaction with ubiquitous environment, we have implemented a ubiquitous testbed where multiple UFC users interact with various ubiquitous devices or other UFC users. Figure 2 illustrates the concept of the ubiquitous testbed room . In addition, we have implemented a practical application that runs upon the UFC platform and the ubiquitous devices, 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

Figure 2. The Concept of the Ubiquitous Testbed Room

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. Efficient utilization of a small-sized display and intelligent mapping of various commands on the input buttons can partially solve this problem. However, such an approach usually makes it difficult to learn how to use the device, which degrades the usability of the UFC platform. One recent workshop underscored that usability is one of the primary challenges in a next-generation “smart” room, that is full of various ubiquitous devices[13].

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 gesture 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[18] 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.
- *service discovery platform*: For a UFC platform to exchange information with one ubiquitous device, the UFC platform should be able to obtain the interface through which the communication is made possible. The interface includes IP address, port number and simple properties of the device. We have been working with middleware team and they developed ubiquitous service discovery(USD) protocol as part of KUSP(KAIST Ubiquitous Service Platform)[16]. USD protocol was originally based on UPnP[17], which is widespread as a service discovery, and this protocol is simplified to avoid XML parsing overhead. In this study, the USD protocol and KUSP was used as a service discovery platform.
- *application that runs upon a UFC platform* infers a UFC user's intention based on a gesture, a target device and previous operations and conducts a corresponding operation.

Overall architecture is illustrated in Figure 3.

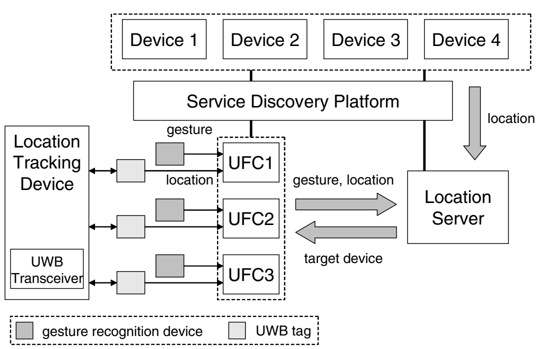


Figure 3. Overall architecture

Among these necessary components, in this paper, we focus on the gesture recognition device and the application that runs upon a UFC platform.

3.2 i-Throw

In order to add a gesture interface to the UFC platform, we developed a wireless gesture recognition device, called *i-Throw*. This device is small enough to be worn on one's finger like a ring. It 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. Figure 1-(b) shows its appearance.

3.3 Recognition of Gesture

The gesture recognition performed by the i-Throw device 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 4 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. Table 1 shows the extracted features of various input gestures.

Table 1. Features of various gestures

Gestures	Features
Throwing	(2g+,X,2g+,150msec)
Increasing	(X, 0.5g+, X, 500msec)
Decreasing	(X, 0.5g-, X, 500msec)
scrolling up	(0.5g-, X, X, 500msec)
scrolling down	(0.5g+, X, X, 500msec)
selecting	(X, X, 0.5g-, 70msec) & (X, X, 1.5g+, 70msec)
scanning	(1.7g+, X, X, 100msec) & (X, 1.4g+, X, 100msec)

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 microcontroller 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 Figure 4. 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 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.

3.4 Detection of Target Device

Our target detection algorithm is based on Cone selection which is used in virtual computing environments[14]. 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 the location server, as mentioned in Section 3.1. 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 object sets which can be sent and received inside our ubiquitous testbed room as follows:

- *music* : mp3 format, music objects can be played either by a UFC platform or public speaker, called *u-speaker*
- *news* : html format, news objects can be obtained from *news kiosk*
- *photo* : jpg format, photo objects are generated when UFC users take picture

Next, we further 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 2.

Among these devices, the *news kiosk* automatically gathers recent news 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.

Table 2. Target Device Sets

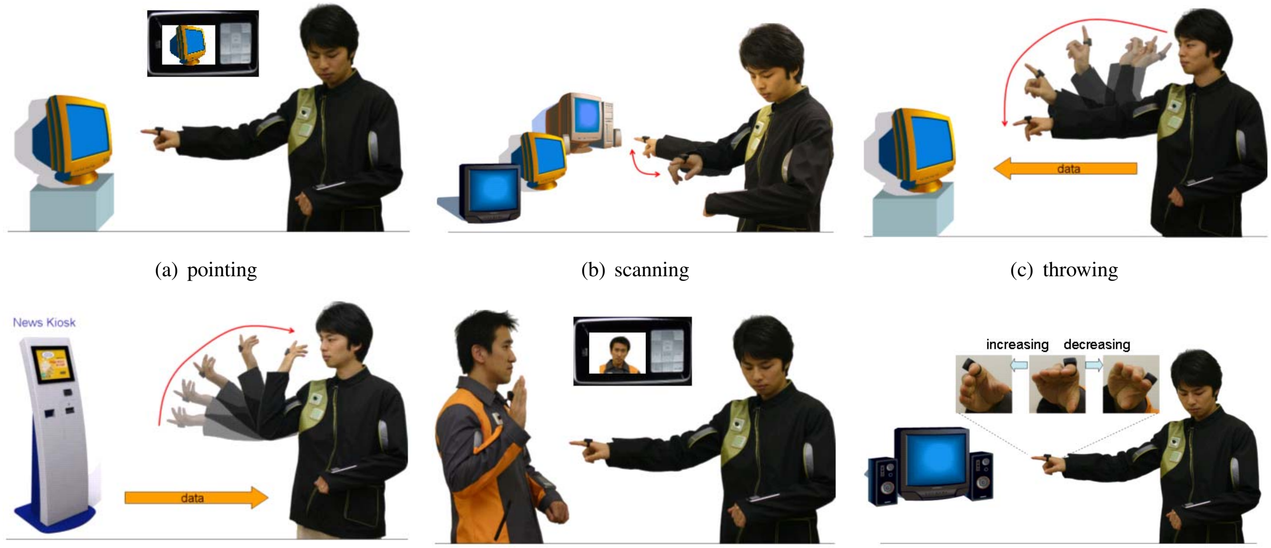
Target device	Supported objects	Flow of objects
u-display u-projector	news, photo	input, output
news kiosk	news	output
u-printer	news,photo	input
u-trash	news,photo,music	input
u-speaker	music	input, output
UFC	news,photo,music	input

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.

Table 2 points out that some target devices do not support all kinds of objects: music objects can be supported by the device which is capable of plays the music such as u-speaker or UFCs and the news kiosk only supports the news object. Table 2 also shows that some devices only allow either an input channel or an output channel: a u-trash only opens an input channel, while a news kiosk only opens an output channel.

3.5 UFC Operation Sets

Until now, we have summarized the gesture sets, the object sets and the target device sets that our ubiquitous testbed supports. When a UFC user makes one gesture among the given gesture sets, the UFC platform should be able to decide which object and which operation needs to be processed. That decision depends on the type of a target device and the type of a recent operation. For example, if one UFC user takes a picture('taking a picture' operation) and made a 'throwing' gesture towards the u-printer, it is highly likely that the user intends to print the photo he has just took. On the other hand, if the user reads a news using his UFC terminal('reading a news' operation) and makes a 'throwing' gesture towards the u-printer, the user may in-



(a) pointing (b) scanning (c) throwing

(d) receiving (e) ready-to-receive (f) increasing/decreasing

Figure 4. Gesture Sets of i-Throw

tend to print the news he has just read rather than photos or other objects. These examples show that the UFC platform has to keep track of the recent operations, more specifically, recently selected objects. To do this, we define the followings: *mrso* indicates the most recently selected objects among various music, news and photo objects. *mrso_music*, *mrso_news* and *mrso_photo* indicate the most recently selected music, news and photo objects, respectively.

Finally, Table 3 summarizes the UFC operation sets which were actually utilized in our demonstration. In this table, '1)' indicates that the corresponding object depends on the status of the selected target device. The demonstration video clip is available in [19].

4. Conclusion

The ubiquitous fashionable computer(UFC), introduced in this paper, is a wearable computer that exploits ubiquitous computing environment. The success of wearable computer will heavily rely on good wearability, usability, aesthetic appearance, and social acceptance. Therefore, the external design of it should be aesthetic and comfortable to success its popularity and let more people use for their real lives. In addition, not merely considering about these uncomfortable carrying issues, the reflection the exploit of ubiquitous computing environment should be done. In the ubiquitous environment, wearable computing systems integrate various types of devices like communication interfaces to get services in any time and any where, and user interfaces to facilitate the computing system without any burdensomeness.

In this paper, we present the design approach and the philosophy of our wearable computing system that is wearable, aesthetic, and intuitive. The main features are as follows. It supports the coexistence and interoperability of various communication interfaces between WLAN, Bluetooth, and ZigBee devices that can be operated freely in any ubiquitous system environment. This can be done using dynamic channel allocation mechanism between communication devices that operates same ISM band. In addition, novel user interfaces are developed and integrated with the UFC to help the intuitive use of it in our wearable computer system. Among possible user interfaces, in this paper, we focus on what we call $i - Throw$, which means intuitive input devices. This input device can recognize human's hand gestures and direction, and control ubiquitous devices with the gestures. Our UFC system is realized and tested with real testbed environment that has various network interfaces, sensor nodes and ubiquitous components, and also many applications are implemented and presented.

References

[1] Oliver Amft, Michael Lauffer, Stijn Ossevoort, "Degisn of the QBIC wearable computing platform", *15th IEEE International Conference on Application-Specific Systems, Architectures and Processors*, Sep. 2004, pp. 398-410.

[2] S. Mann, "Wearable computing as means for personal empowerment.", *In Proc. 3rd Int. Conf. on Wearable Computing*, May 1998.

[3] A. Pentland, "Wearable intelligence", *Scientific American*, 276(1es1), Nov. 1998.

Table 3. UFC Operation Sets

<table>
<tr><th rowspan="2">operation</th><th rowspan="2">selected object</th><th colspan="3">condition</th><th rowspan="2">change of mrso sets</th></tr>
<tr><th>target_device</th><th>gesture</th><th>etc.</th></tr>
<tr><td>increase volume</td><td>none</td><td>u-speaker</td><td>increasing</td><td>none</td><td>none</td></tr>
<tr><td>decrease volume</td><td>none</td><td>u-speaker</td><td>decreasing</td><td>none</td><td>none</td></tr>
<tr><td>delete a file</td><td>mrso</td><td>u-trash</td><td>throwing</td><td>none</td><td>mrso=none</td></tr>
<tr><td rowspan="4">send a file</td><td rowspan="3">mrso</td><td>u-display</td><td rowspan="4">throwing</td><td>mrso_type = photo or news</td><td>none</td></tr>
<tr><td>u-printer</td><td>mrso_type = photo or news</td><td>none</td></tr>
<tr><td>UFC</td><td>none</td><td>none</td></tr>
<tr><td>mrso_music</td><td>u-speaker</td><td>none</td><td>mrso_type=music
mrso=mrso_music</td></tr>
<tr><td rowspan="2">read news</td><td>1)</td><td>news kiosk</td><td>receiving</td><td>none</td><td>mrso_type=news
mrso=1)</td></tr>
<tr><td>other.mrso</td><td>none</td><td>ready-to-receive</td><td>other.mrso_type==news and
other.target_device==this and
other.gesture==throwing</td><td>mrso_type=news
mrso=other.mrso</td></tr>
<tr><td rowspan="2">listen to music</td><td>1)</td><td>u-speaker</td><td>receiving</td><td>none</td><td>mrso_type=music
mrso=1)</td></tr>
<tr><td>other.mrso</td><td>none</td><td>ready-to-receive</td><td>other.mrso_type==music and
other.target_device==this and
other.gesture==throwing</td><td>mrso_type=music
mrso=other.mrso</td></tr>
<tr><td rowspan="2">view a photo</td><td>1)</td><td>u-display</td><td>receiving</td><td>none</td><td>mrso_type=photo
mrso=1)</td></tr>
<tr><td>other.mrso</td><td>none</td><td>ready-to-receive</td><td>other.mrso_type==photo and
other.target_device==this and
other.gesture==throwing</td><td>mrso_type=photo
mrso=other.mrso</td></tr>
</table>

* '1)' indicates that the corresponding object depends on the status of the selected target device.

[4] T. Starner, "The challenges of wearable computing: Part 1.", *IEEE Micro*, July 2001.

[5] M. Weiser, "The computer for the 21st century.", *Scientific American*, 265(3):66-75, Sept. 1991.

[6] U. Anliker, and et al, "The WearARM: modular, high performance, low power computing platform Designed for Integration into Everyday Clothing", *In Proc. of 5th Int. Symposium on Wearable Computers*, 2001.

[7] Xybernaut Corp., Xybernaut wearable systems. Mobile Assistant wearable computer, www.xybernaut.com

[8] C. Narayanaswami et al., "IBM's Linux Watch: The Challenge of Miniaturization", *IEEE Computer*, vol. 35, no. 1, pp. 33-41, Jan 2002.

[9] J. Rantanen et al., "Smart clothing prototype for the arctic environment", *Personal and Ubiquitous Computing*, 2002.

[10] Sheng Liang, The Java Native Interface: Programmer's Guide and Specification, *Addison Wesley*, 1999.

[11] IEEE Standard 802, part 15.4, Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for Low Rate Wireless Personal Area Networks(WPANs), 2003.

[12] IEEE802.15.2 Specification, Coexistence of Wireless Personal Area Networks with Wireless Devices Operating in Unlicensed Frequency Bands, Aug. 28, 2003.

[13] M. Back, S. Lahlow, R. Ballagas, S. Letsithichai, M. Inagaki, K. Horikira and J. Huang, "Usable Ubiquitous Computing in Next-genration Conference Rooms: Design, Evaluation, and Architecture," *UbiComp 2006 Workshop*, Sep. 2006.

[14] J. Liang and M. Green, "JDCAD: A Highly Interactive 3D Modeling System," *Computers & Graphics,* 18(4), 1994, 499-506.

[15] M. Kang, J. Chong, H. Hyun, S. Kim, B. Jung and D. Sung, "Adaptive Interference-Aware Multi-Channel Clustering Algorithm in a ZigBee Network in the presence of WLAN Interference," *IEEE International Symposium on Wireless Pervasive Computing*, Puerto Rice, Feb. 2007.

[16] Y. Song, S. Moon, G. Shim and D. Park, "Mu-ware: A Middleware Framework for Wearable Computer and Ubiquitous Computing Environment," *Middleware Support for Pervasive Computing Workshop at the 5th Conference on Pervasive Computing & Communications(PerCom 2007)*, New York, Mar. 2007.

[17] UPnP Forum, UPnP Device Architecture 1.0, Version 1.0.1, Dec. 2003.

[18] Ubisense, http://www.ubisense.net

[19] KAIST UFC Project, http://core.kaist.ac.kr/UFC