U-Interactive: A Middleware for Ubiquitous Fashionable Computer to Interact with the Ubiquitous Environment by Gestures

Gyudong Shim, SangKwon Moon, Yong Song, Jaesub Kim, and Kyu Ho Park

Computer Engineering Research Laboratory,
Department of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology
{gdshim,skmoon,ysong,jskim}@core.kaist.ac.kr, kpark@ee.kaist.ac.kr

Abstract. In this paper we present a system, called U-interactive, that provides spontaneous interactions between human and surrounding objects in heterogenous ubiquitous computing environments. Our U-interactive system introduces a virtual map, which contains interactive objects around a user in each ubiquitous environment. In the virtual map, each interactive object is tagged with geographic information and attributes to interact with. Each user can create interactive objects in the virtual map corresponding to physical objects. Also the scope of the map is automatically adjusted according to user's location (inside building or outdoors) by location services. U-interactive system runs on both mobile devices and infrastructures. U-interactive system provides inter-operability with communication methods, such as UbiSpace, UPnP, and Web services. We developed our U-interactive system upon a prototype of ubiquitous environment. U-interactive system contains interactive kiosk, printer, sensor networks, and users.

Key words: Middleware, Ubiquitous Fashionable Computer, HCI, *iThrow*, Spontaneous Interaction, Service discovery, File Sharing

1 Introduction

Many researches have been come out to realize ubiquitous computing environments in the real world. The common philosophy of the researches is to make a convenient life with surrounding computers. People in the ubiquitous environment access any information by multiple interfaces and displays. It is important to provide easy and comfortable human interfaces in the ubiquitous environments.

As to realize ubiquitous environments, our research project teams have developed testbed on KAIST campus called U-TOPIA and a wearable computers named Ubiquitous Fashionable Computer(UFC)[1]. In order to realize U-TOPIA, we are elaborating the campus-wide infrastructures in KAIST campus. U-TOPIA has been equipped with indoor and outdoor testbed for location

services which consist of ZigBee and UWB[11] sensor networks. In addition, Wireless Mesh networks in the campus extends WLAN coverage to the outdoor areas[8].

UFC has been developed as a wearable gadget and cloth as shown in Fig.1 for convenience and portability. In order to interact with the environments user-friendly, UFC has a gesture recognition device called *iThrow*[2]. *iThrow* is a ring typed intuitive interface device. *iThrow* has a 3D accelerometer and a 3D magneto-resistive sensor; thus it can recognize specific commands and two dimensional pointing direction of finger from user gestures.

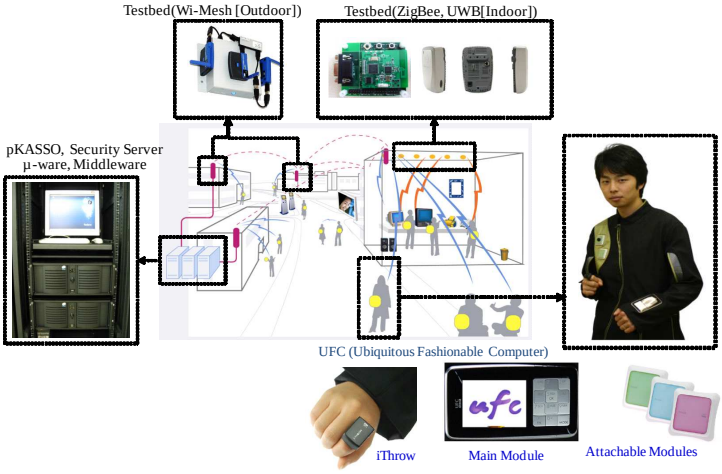


Fig. 1. U-TOPIA architecture and UFC components

A UFC user can select a target inside the testbed by finger pointing as shown in Fig.2-(a). Then the selected target is represented in the display of UFC. After the user selects a target in the U-TOPIA, the user can throw data such as a image file to the target as shown in Fig. 2-(c). Through intuitive gestures, users can interact with objects such as throwing a file, receiving a file, controlling devices, and querying some information of the target. For example, when a user throws a file to a printer, the printer will print the file.

We designed an interactive system, called U-interactive, that provides spontaneous interaction between UFC users and U-TOPIA. U-interactive system provides followings.

- location detection from multiple sources
- target selection from the virtual map
- interaction method and command matching upon an interactive object
- communication between UFC and interactive objects

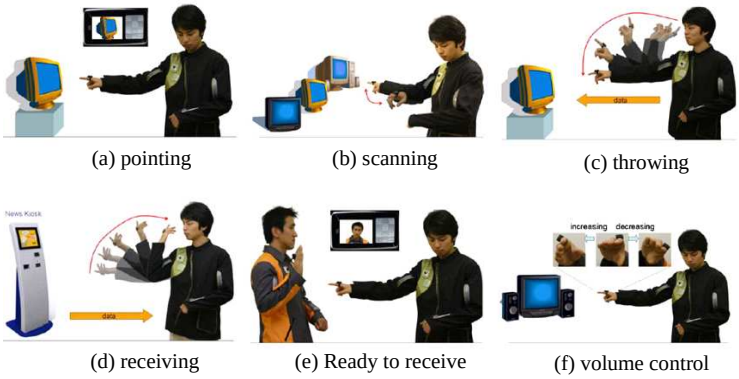
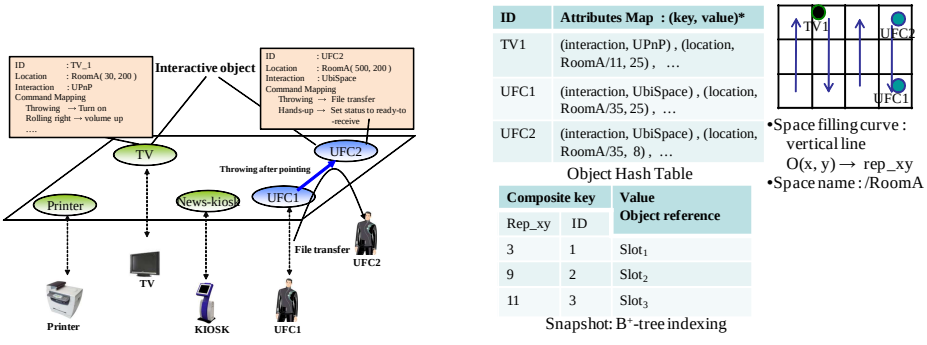


Fig. 2. Gesture operation sets of *iThrow*

U-interactive system introduces virtual maps and interactive objects. A *virtual map* contains interactive objects around a user in a specific ubiquitous environment. 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. Fig.3 shows the virtual map and interactive objects. In Fig.3, the UFC1 has information about UFC2's location, interaction method, and the command configuration that assigns throwing motion as transferring a file.



(a) concept (b) data structures

Fig. 3. Virtual map and interactive objects

To support such interactions at indoor areas and outdoor areas, we designed and implemented U-interactive system. U-interactive system is a customized and evolved incarnation of μ-ware for U-TOPIA where thousands of clients and hundreds of infrastructures coexist. U-interactive system aims spontaneous interaction via HCI, especially *iThrow*. U-interactive includes coordination of location service, automatic virtual map resolution, and interaction through multiple interfaces and commands.

The remainder of this paper is organized as follows: Section 2 presents related works. Section 3 describes U-TOPIA and U-interactive system architecture. Section 4 specifies interactive object management method, and Section 5 describes interaction methods such as UbiSpace and UPnP. Section 6 shows implementation status. Finally we conclude in Section 7.

2 Related Works

Gesture control has been studied by two categories, camera-based and movement sensor-based[5]. Juha at el. presented accelerometer-based gesture recognition method. They developed the gesture control system for design environment especially for TV, VCR, and lighting. Our *iThrow* device is similar to it in movement sensor-based, but our gesture sets are designed for adaptability and mobility of users. So UFC users can probe and choose any items in a given environment.

The concept of virtual map is analogous to the augmented reality. But we focus on the interactions and efficient target selection mechanism in a specific scope. In Virtual Information Towers(VIT)[15], mobile users can interact with visible items on the advertising columns. VIT provides a metaphor to access information which is assigned to physical location. VIT does not provide the discovery of VIT-Directory which has a list of VITs. So VIT system can not provide seamless service and the user should manually set the environment's server. In addition, the user interacts with VIT by web-browser interface of the wearable device.

Tatsuo at el. developed *personal home server* to support spontaneous interaction in home computing environments [16]. The mobile device works as a personal home server which contains personal information and preferences to the services. They utilize inter-operable protocol SOAP for interaction methods. But U-interactive exploits UbiSpace for easy file transfer between devices and services. They aims personalized and secure interaction by RFID tags which store encryption keys. They designed and implemented the system for home networks rather than outdoor or large scale networks. Since the services are discovered by SSDP of UPnP, The system is not scalable for large number of services.

Personal Server[14] is a mobile device that contains personal data which can be accessed through surrounding displays and keyboards input systems such as public kiosk or PC monitor and keyboards. Personal Server contains an embedded web server in the mobile device. The discovery of the host is performed by Bluetooth and UPnP. Actual interactions between the Personal Server and the

host are performed by HTTP and SOAP[13] protocol. However Personal Server provides only limited interfaces from the infrastructures.

3 U-interactive System Architecture

U-interactive is a middleware component for ubiquitous interactive services which can be controlled with a gesture device- *iThrow*. The device of ubiquitous services consist of various devices such as kiosks and printers. The software on the devices are built on our middleware framework. The software architecture of our middleware is shown in Fig.4. Most of the interactive services are implemented as service bundles over Open Service Gateway initiative(OSGi) Platform[10]. In order to provide location based services, we proposed a ubiquitous context aware middleware framework for campus-wide infrastructures and UFC in μ-ware[3].

μ-ware is a middleware framework that adapts the runtime environment upon the location of the client. μ-ware provides JAVA based extensible and configurable runtime environment to the applications. μ-ware is composed of following components.

- KAIST Ubiquitous Service Platform(KUSP) : OSGi based runtime environments. This framework manages the life time of service bundles.
- USD Protocol: light-weight service discovery protocol. It can discover services across the network by multicast.
- UbiSpace: Coordination component like tuple space. It handles publish/ subscribe operation among UFCs.
- Instant Service Loader: it downloads a service bundle from the servers, after that it installs and executes the bundle.

By these components μ-ware not only gives an efficient computing environment to upper applications on UFC but helps the applications to exploit various ubiquitous service modules. [3]. In the previous work, the main target environments of μ-ware were indoor smart spaces. To be a foundational middleware framework for U-TOPIA, μ-ware has to be enhanced and customized in pursuit of UFC's energy efficiency and interactive functionality even for massive clients in outdoor areas. In this chapter we describe U-TOPIA environments, internal architecture of U-interactive, and coordination of location service.

3.1 U-interactive Infrastructures on the U-TOPIA

U-TOPIA equips infra structures such as location service, service discovery, and service provision servers. The servers are constructed with multi-level tree hierarchy for campus wide ubiquitous services. The entire space is divided by sections of the institute, building, floor of building, and room or hall. In order to provide information repository and support collaboration storage, we suggest virtual spaces which are constructed in a hierarchical tree. Each representative space holds the map and the service list of the region. Physical objects can be assigned by attributes of services. And they are referenced by a GUID (Global

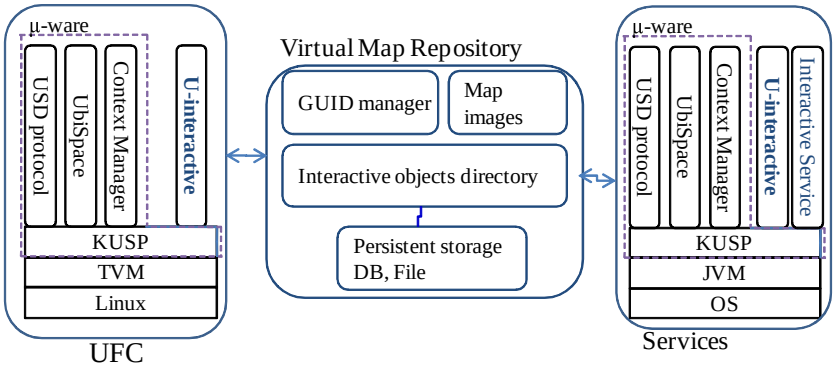


Fig. 4. U-interactive system architecture

unique identification). For the management reasons, the GUID manager exists in the root of hierarchy tree. Objects directory contains a table of GUID, name, type, service reference, and information to represent image files, video files, and HTML files. The temporarily available services and location of users are separated from static information in the object directory. File repository provides a storage for the system and makes it possible to share files between UFCs.

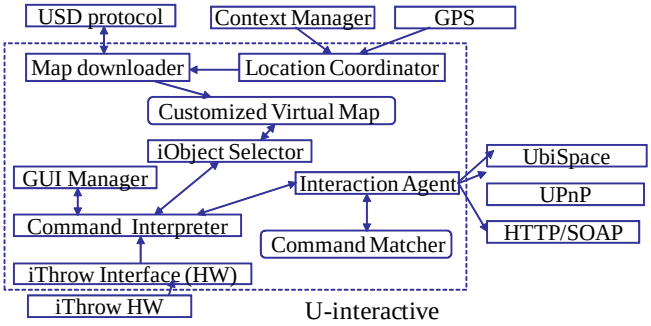


Fig. 5. U-interactive internal architecture: U-interactive cooperates *iThrow*, GPS and infrastructure services - UbiSpace, Context Manager, an USD protocol.

3.2 Internal architecture of U-interactive

U-interactive internal architecture is shown in Fig.5. U-interactive utilizes foundational services such as Context Manager, UbiSpace, USD protocol of μ-ware.

Main roles of U-interactive are virtual map management and interaction between UFC and interactive objects. The virtual map is automatically downloaded from the map server by location coordinator when the logical address of UFC is changed. Command Interpreter takes the command and pointing directions from *iThrow* hardware. Command Interpreter forwards the command to *iObject* Selector or Interaction Agent by the type of *ithrow* command. Interaction Agent takes the interaction by the interactive object type. Interaction Agent decides a proper interface to the interactive object from command matcher which has *ithrow* command to iObject service mapping.

3.3 Coordination of Location Service

Location is described by longitude, latitude, altitude, and logical address. Logical address consists of organization, building, floor, room number in hierarchical order, for example 'KAIST/E3-2/3F/Room3217', 'KAIST/Section1/'. UFC users can resolve their locations by multiple sources such as GPS, ZigBee sensor network, and UWB sensor networks depending on the place. When UFC users are located in an indoor area such as a room or a floor, their locations are resolved from UWB or ZigBee sensor networks. But if they are located in outdoor areas such as campus road or square, their locations are detected basically by GPS on UFC. Because GPS has inaccuracy in densely surrounded buildings, ZigBee beacons of the testbed is used as a hint for more accurate location. The beacon of Zigbee sensor network holds the longitude, latitude, and logical place description. The beacon signal is proximity guarantee since the receiver is within a transmission range from the landmark. If there are more than two beacons from different positions, Received Signal Strength Indication(RSSI) is used on the triangulation. U-interactive coordinates the multiple location sources adaptively based on predefined source priority(UWB>ZigBee>GPS). When GPS is the only available location source, it is required reverse geocoding to translate GPS location to the logical address.

4 Management of Interactive Object

U-interactive assigns physical objects to service and information to construct smart environments. For instance, electrical devices can contains control service as UPnP service and bus stop can have bus time schedule information. Complex building objects can construct logical service hierarchy. As the preceding step to interactions with smart object in the ubiquitous environments, U-interactive assigns GUID(Global Unique Identification) for each objects. Object are grouped by their information and service source type. Geographical information can be handled by organizations' administrator and useful information can be added by users later. *GUID Manager* in our system handles issuing of identifications. GUID has prefix of geographic information and sequential serial number for services. GUID could be URI or bar code or RFID depends on the object. Each UFC has unique PANDA ID(16 bytes) for recognition and authentication.

4.1 Interactive Object Registration

We designed virtual map data structures as shown in Fig.3-(b). For fast spatial queries the repository maintains a current snapshot of the interactive objects. The location of interactive object is converted one dimension representative value by vertical line space filling curve. The snapshot is built by B^+-index whose key is assigned by the converted value and object id. Detail interactive object information is maintained in an attribute map hash table. When a new object is registered, a new element is inserted in the object hash table and the location is inserted the snapshot.

Our services on the KUSP platform can register themselves in the current level of their location repository. Since we expect most of services are ported in OSGi compatible bundle, they can register and deregister by bundle start/stop phase. As a representative location of the single KUSP platform, *Context Manager* has default location of the platform. Default location is assigned if the service doesn't specify the location. Context manager takes charge of the service registration of other service bundles. Holding only information objects such as building information are managed by a central administrative way(such as institute administrative building) or fully decentralized way(each home appliances by user administration). Geographic objects such as building, statue and pond are stored in the Map server conjunction with Geographic Information System.

4.2 Interactive Object Discovery

When UFC detects logical address change, the virtual map repository is resolved by logical address key lookup in USD protocol[4]. After the infrastructures are spread out over the same network and the number of users' discovery requests increases, the scalability of entire system become worse because of indirect multicast search algorithm of UPnP[12]. Therefore we add caching mechanism of service discovery results based on the repository of UFC and infrastructures. The global infrastructure topology of an institute is maintained by a hierarchical structure. The virtual map repository stores topology of the infrastructures in a global view, the discovery of repository from UFC can be resolved readily by a close repository. When UFC users move around the logical place, they can adapt their location and intention by instant downloading of information of the space such as addresses of map servers, available file server.

4.3 Target Selection in a Virtual Map

Basically U-interactive sets the space boundary by logical address scope to UFC. UFC can see the target in the given boundary space with objects with priorities. The public and hot services are ranked high priority by usage count and temporal and moving objects are ranked low priority by reduction of movement. This priority helps user to select clustered targets in narrow directions. UFC user can select a target object in user friendly manner by the scanning gesture of *iThrow*. Yoo et al, proposed target selection algorithm which is ray based minimum

angle difference and adaptive angle placement of targets [2]. We adopt ray based minimum difference angle target selection of [2].

When a user in the smart room alone, most of object are stationary and he keep stationary for a second or more than an hour. Then the objects position and status are static information. So as to reduce repeated location and object query to the space repository, UFC selects the target object in own repository on the stationary state. To simplify target selection operation, the space objects are modeled as two dimensional points. the origin is the location of UFC and each object calculated in polar coordination system. If the UFC location is (x_1, y_1) and the target is (x_2, y_2) in cartesian, the polar coordination is given by

$$(r, \theta) = (\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tan^{-1}(\frac{y_2 - y_1}{x_2 - x_1})). \quad (1)$$

As long as the UFC is stationary with a given boundary D_{update}, the calculated polar coordination positions are reused to reduce complex calculation of *sqrt* and tan^{-1} operation in the UFC. If a new target scanning is performed at the same position, target object is selected by lookup of given angle difference of pointing direction in the list of objects sorted by angle and distance.

5 Interaction Methods

U-interactive interacts with interactive objects by multiple interface methods to various services. The interaction methods are dependant on the interactive object interfaces. UPnP devices are interacted by UPnP protocol and Ubiquitous interactive services are communicated by UbiSpace. For the web services, U-interactive handles commands by SOAP.

5.1 UbiSpace

UbiSpace handles specific commands and file sharing of interactions. In order to reduce the burden of application development in the data sharing, we choose UbiSpace as a collaborative data sharing service for file and java objects. UbiSpace provides publish/subscribe event handling system. Furthermore it has repositories of file sharing in the infrastructures. Each object contains a key to identify and it is indexed by B^+-tree for fast look-up. Each client can subscribe data for specified the key. UbiSpace derives tuple spaces such as TSpaces[7] and JavaSpaces[9] in central temporal and time decoupling aspects, but it simplifies object matching by name key, and supports file publish/subscription operation for file sharing between clients. The only basic operations are as shown table 1 for limitation of the space.

To support basic simple file transfer, we integrated same API while UbiSpace inspects the object whether it is a file. The source file is uploaded the file repository of server and after that subscribers download the file later by *subscribe* or *read* operation. Because of large file transfer overhead, UbiSpace exploits network file system to clients. When a UFC joins the UbiSpace server,

long insert(String key, java.io.Serializable obj)
Object take (String key)
Object read (String key)
ITuple delete (long itemID)
long publish(java.lang.String key, java.io.Serializable obj)
long subscribe(java.lang.String key, EventHandler callback)
void unsubscribe (long seqNumber)

Table 1. UbiSpace basic APIs

it automatically mounts the file system on a given private directory. Large file operations can be exploited by native distributed file system operations. Most services are accessible anyone in the place but some administrative information and restrictive services should be protected from unauthorized access. By using publish/subscribe data access control of UbiSpace, it can release the burden of access control list management on the applications.

6 Implementation and Experiment

The testbed and terminal team of our project implemented the UFC show room. We implemented the indoor U-interactive prototype on the showroom. There are UWB and ZigBee sensor networks for location service. The location is accurate on limited region up to $15cm$ resolution. The outdoor virtual spaces are under development but they would be installed in months.

As U-services examples, U-Desktop and U-Print service in the room are implemented by OSGi bundles. The U-Desktop service displays multimedia file on public kiosk by subscription of UFC users in the space. The publish and subscription of the images and movie files are performed by UbiSpace *publish*, *subscribe* operations both of U-service and UFC. U-print service provides UFC with customized content generation by UFC's throwing image files.

The U-Kiosk and U-Print services are implemented as service bundles of OSGi. To exploit Windows native services and applications, the service bundles connect dynamic loadable libraries by JNI interface to display and print a file by *Execute* function of Windows MFC. As an example codes, the U-Print service can be easily and concisely implemented as follows.

```
int printSubscriptionID = us.subscribe(this.FILE_KEY + serviceID,
new EventHandler() {
    public void todo(String key, Serializable obj) {
        if (obj instanceof File) {
            File file = (File)obj;
            // JNI interface for Windows Print service
            uservicеAdaptor.print_process(file);
        }
    }} );
}
```

The gesture operation sets are shown in Fig.2. File sharing operation (c),(d) in Fig.2 is performed by UbiSpace publish/subscribe API also. The rest operations - pointing, scanning, volume control are performed by publish/subscribe of *IThrowCommand* object which is published by the UFC.

The performance of UbiSpace is experimented by latency comparison with T-Space. The latency between the server and the client(PC) is examined when there are 8000 tuples in the server. The tuple object size is 1KBytes. Fig.6. shows that our UbiSpace outperforms T-Space for the best case 111% in the take operation. The reason of fast response comes from efficient tuple indexing by B^+-tree and setting no TCP delay on the socket. The tuples are indexed by a composite key which consists of the tuple name, the tuple field size, and fields.

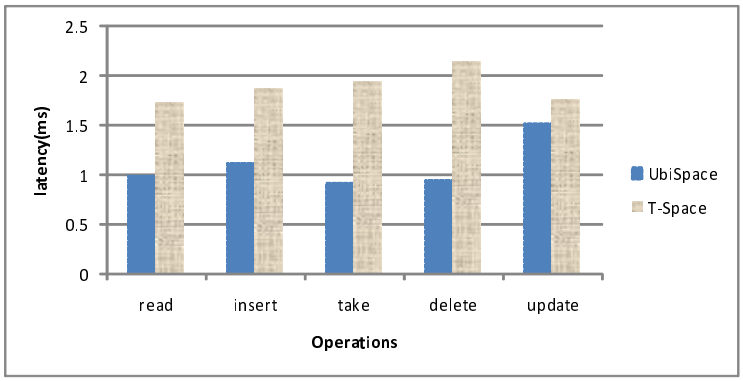


Fig. 6. UbiSpace latency comparison with T-Space.

7 Conclusions

U-interactive system make it possible for UFC users to interact with interactive objects in ubiquitous environments by intuitive gestures. We propose new concepts, *virtual map* and *interactive objects*. The virtual map scope is automatically resolved by location of a user to be convenient. U-interactive system provides interoperability by choosing a interaction method adaptively, such as UPnP, SOAP, and UbiSpace. U-interactive provides learnable and customizable mechanism to adapt various interactive object types. The standard of command sets is required to be general and learnable interfaces for *iThrow*. It is crucial for the system to tag the location on any physical objects. We are plan to integrate multiple HCIs in U-interactive such as speech recognition and small keypads.

References

1. Jupyung Lee, Seung-Ho Lim, Jong-Woon Yoo, Ki-Woong Park, Hyun-Jin Choi and Kyu Ho Park, A Ubiquitous Fashionable Computer with an i-Throw Device on a Location-based Service Environment, PCAC-07
2. J.W. Yoo, Y.W. JEONG, Y. SONG, J.P. LEE, S.H. LIM, K.W.PARK, K.H. PARK, *iThrow*: A New gesture-based wearable input device with target selection algorithm, ICMLC 2007(to be appeared)
3. Y.Song, S.K.Moon, G.D.Shim, D.Y.Park, "A Middleware Framework for Wearable Computer and Ubiquitous Computing Environment", *PercomW'07*, pp. 455-460
4. SangKwon Moon, Jaesub Kim, and Daeyeon Park, USD Protocol: Ubiquitous Service Discovery Protocol on Infrastructure-based architecture for Ubiquitous Fashionable Computer, MUE 2007, pp. 779-784
5. Kela, J., Korpipaa, P., Mantyjarvi, J., Kallio, S., Savino, G., Jozzo, L., and Marca, D. 2006. Accelerometer-based gesture control for a design environment. Personal Ubiquitous Comput. 10, 5 (Jul. 2006), 285-299.
6. Ki-Woong Park, S.S. Lim, H.C. Seok, and K.H. Park, "Ultra-Low-Power Security Card, PANDA, for PKI-based Authentication and Ubiquitous Services,Proceedings of Confer-ence on Next Generation Computing, November 2006, pp.367-373
7. Tobin J. Lehman and Alex Cozzi and Yuhong Xiong and Jonathan Gottschalk and Venu Vasudevan and Sean Landis and Pace Davis and Bruce Khavar and Paul Bowman, Hitting the distributed computing sweet spot with TSpaces, Computing Networks. 2001, pp. 457-472
8. KAIST UFC Project, http://core.kaist.ac.kr/UFC
9. Eric Freeman and Ken Arnold and Susanne Hupfer, JavaSpaces Principles, Patterns, and Practice, 1999
10. OSGi alliance, http://www.osgi.org
11. UbiSense, http://www.ubisense.net
12. UPnP Forum, http://www.upnp.org
13. SOAP:Simple Object Access Protocol,http://www.w3c.org/2002/ws
14. Roy Want, Trevor Pering, Gunner Danneels, Muthu Kumar1, Murali Sundar, John Light, The Personal Server: Changing the Way We Think about Ubiquitous Computing, UbiComp 2002.
15. Leonhardi, A., Kubach, U., Rothermel, K., Fritz, A. : Virtual Information Towers - A Metaphor for Intuitive, Location-Aware Information Access in a Mobile Environment, ISWC 1999.
16. Nakajima, T. and Satoh, A software infrastructure for supporting spontaneous and personalized interaction in home computing environments. Personal Ubiquitous Comput. 10, 6 (Sep. 2006), 379-391