Smart Furniture:
Creating Smart Hot-Spots Everywhere

Hideyuki Tokuda
hxt@ht.sfc.keio.ac.jp
Graduate School of Media and Governance, Keio University
5322, Endo, Fujisawa, Kanagawa 252-8520, Japan

Abstract

Many researchers introduced special rooms equipped with sensors, devices, networks, and computers to demonstrate ubiquitous computing environment. However, the cost and time for building such a room is a barrier to the deployment of various ubiquitous applications.

We have developed Smart Furniture which instantaneously converts the legacy non-smart space into a Smart Hot-Spot which can provide the accessibility to the Internet, location-based context-aware services, service roaming, and personalization services. Since the Smart Furniture is equipped with networked computers, sensors and various I/O devices, it can provide various services in public space as well as in private space.

In this paper, the physical structure and middleware for the Smart Furniture are described. The prototype systems to realize Smart Hot-Spot such as a Personalized Message Board System, a Zero-stop check-out System, and a Mobile TV-phone System.

1 Introduction

This paper presents the Smart Furniture project which aims to realize improvised smart hot-spot computing environment. The goals of the project are (1) creating a physical structure called Smart Furniture which is composed of hardware such as computers, sensors, and appliances, and (2) developing software technologies for realizing smart hot-spots. By the smart hot-spot, we mean that not only the accessibility to the Internet, but also location-based services, context-aware services, service roaming services, and personalization services are provide. We have designed and implemented the Smart Furniture to acquire extendibility and reconfigurability for future development.

To realize ubiquitous computing environment[1], various researches introduced smart space testbeds such as Active Badge[2], Active Bat[3], EasyLiving[4], AwareHome[5], and Cooltown[6] projects. In many cases, special rooms equipped with sensors, devices and computers are used to detect human activity and provide services. We built Smart Space Laboratory (SSLab)[7] as a laboratory of ubiquitous computing, which is equipped with sensors, embedded devices and computers. To coordinate these devices, four layered networks such as a LAN plane, sensor plane, audio/visual plane, and lighting plane, were build. However, the cost and time for building such a smart room is a barrier to the deployment of the ubiquitous computing environment.

The Smart Furniture project, instead, aims to extend non-smart space to have smart functionality by simply putting Smart Furniture in place. Smart Furniture contains computers, sensors, devices, and appliances. Users can use Smart Furniture as a gateway to the cyber world, as a service operator, or as a service receiver. We have also designed and implemented middleware and application software for creating smart hot-spot services. The combination of the Smart Furniture's physical structure, hardware devices, and its software technology can transform the legacy space into smart space.

In the remainder of this paper, we present the definition of Smart Hot-Spot in Section 2. Section 3 describes the design of the Smart Furniture. Section 4 describes the architecture of middleware, called Wapplet for the Smart

Furniture. Section 5 describes the Smart Hot-Spot applications running on the Smart Furniture. Section 6 concludes the paper and mentions future work.

2 Smart Hot-Spot

We have developed the "Smart Hot-Spot" and "Smart Hot-Spot Networking" technologies by which we can take ubiquitous smart computing environment to outside of a laboratory.

2.1 Smart Spaces and their Limitations

Most of the previous researches built special rooms equipped with various sensors and devices. Middlewares to deal with sensors and devices and support applications are also developed and running on computers in these spaces. We call these spaces "Smart Space". Smart Spaces are characterized by following features; (1) identification and detection of activity of users (2) sensing users' and devices' location and (3) control and coordination of devices.

In these environment, computational power is not pervasively provided, but provided in the specific environment such as office and laboratory. Here, users are required to be specific space to be benefited from embedded computers. In addition, in the current approach to construct IT equipped space, constructors need to design and construct them from the scratch and costs are relatively high.

2.2 Smart Hot-Spot

We define the terminology "Smart Hot-Spot" as an extended hot-spot where features of the Smart Space are provided. Within the Smart Hot-Spot, various services are provided to users and devices. These services enable context awareness, device coordination and personalization. The Smart Hot-Spot does not just provide low level functionalities such as accessibility to a network. The Smart Hot-Spot provides functions independant of services.

A Smart Hot-Spot is not provided by a building type infrastructure, but an equipment which is small and light enough to carry. It has the characteristics of extendibility and reconfigurability for future development. In the equipment, various input/output devices and computers are attached and connected via wired and wireless network. We call the equipment "Smart Furniture." User only need to place a set of Smart Furniture in any space, then the furnitures provide Smart Hot-Spot environment. Moreover, system software of the smart furniture connects lone neighborhood Smart Hot-Spots spontaneously and instantaneously to form a network. This fedration becomes ubiquitous computing environment.

2.3 Scenarios

The following scenarios show using Smart Furniture in our daily life. Various services are provided by the Smart Furniture placed on private and public spaces.

Smart Hot-Spot in Private Space

Ken bought Smart Furniture equipped with lamp and display for his home use. He placed it in his living room, which fitted to his room.

He had already had several home appliances such as TV and VCR with network connectivity. But he needed to control them separately even for simply watching TV. After having placed Smart Furniture in his room, an integrated user interface of applinaces appeared in the touch screen on the Smart Furniture. The middleware in the Smart Furniture coordinated home appliances.

Sensors equipped with the Smart Furniture detect Ken's presence and room's temperature. After controlling air conditioner several times, it learned his preference and automatically controlled when he entered the room.

Smart Hot-Spot in Public Space

Acco was going to participate in a meeting. Before leaving home, she checked the Smart Furniture showing weather forecast as colors of lamp. She decided to bring an umbrella. When she arrived at the station, she found public Smart Furniture showing train schedule. She learned that the train was delayed, and she called to a participant of the meeting that she was going to be late with her cell phone. She used a display and a camera on the Smart Furniture to communicate on TV-phone.

When she arrived at the station near the meeting place, she began to check the map with her PDA. Since its screen was too small, she used the display on the Smart Furniture as an output device of the PDA.

When she was going back home, she checked her room's temperature using Smart Furniture at the bus stop. Since she has her ID card, public Smart Furniture recognized her and coordinated with her private Smart Furniture in her home automatically. She controlled the air conditioner to maitain comforable temperature through the public Smart Furniture.

2.4 Challenges

The goal of the Smart Hot-Spot project is to develop next generation equipment that furnishes the legacy space with computational power extemporaneously and realizes spontaneous setting of ubiquitous computing environment. In the following, we will describe the challenges of the project in terms of physical and software aspects.

Physical Aspect

- **Improvisational Setting**: Since the main goal of the Smart Furniture is to improvise the Smart Hot-Spot, the system itself must be small and light enough to carry around. The installation procedure should also be simple. By simply placing it, we can transform a legacy space into a smart space.
- **User Friendliness**: User Friendliness must be accomplished in two aspects, the appearance of furniture and the interface of services. Smart Furniture should be placed especially in public place.
- **Extendibility and Reconfigurability**: Smart Furniture is a platform of various Smart Hot-Spot services. Therefore it must provide hardware level and software level extendibility and reconfigurability to run various applications.

Software Aspect

- **Context Awareness**: Smart Hot-Spot need to extract context information, including those other than location, using limited number of sensors.
- **Service Roaming**: There are two types of service roaming; intra-spot and inter-spots roaming. In intra-spot roaming, services are transfered between Smart Hot-Spot and devices surrounding it. In inter-spot roaming, services provided by one Smart Hot-Spot can be resumed in other Smart Hot-Spot.
- **Smart Hot-Spot Networking**: Heterogeneous Smart Hot-Spots need to be connected, to support federated smart services.
- **Personalization and Privacy Protection**: Some applications such as Personalized Message Board use user information to change their contents according to user's preference data stored in a user's personal device. Here, privacy issues arise. Malicious spot could disclose these information. We have to balance user's privacy with application's usability.

3 Smart Furniture

Smart Furniture is a platform for systems to realize Smart Hot-Spot. By simply placing the Smart Furniture in public or private space, we can turn legacy spaces into Smart Hot-spots. Smart Furniture is equipped with networked computer, various I/O devices and sensors. In the following, we will describe the physical structure of the Smart Furniture.

3.1 Physical Structure

We have designed and developed the physical structure of Smart Furniture with Uchida Yoko Corporation, an office and IT equipment manufacturer. There are currently three types of Smart Furniture; a pole type, a lamp type, and a mirror type.

Pole Type Smart Furniture

At the first stage of Smart Furniture project, we developed Pole Type Smart Furniture. Pole type is mainly focused on the use in public space, such as train stations, bus stop or campuses. In such places, smart furniture provides public service to non-specific person with various devices. Thus, Pole Type Smart Furniture was designed as an extendable pole, to which various devices and sensors can be attached.

Pole type Smart Furniture is composed of mainly three parts; a base, a pole, and a case. The photos of each parts are shown in Figure 1. To maintain the extendibility and reconfigurability of Smart Furniture, each part is designed with a reusable modular material, and we can compose each part to adapt to various services.

In the pole type Smart Furniture, we have attached and installed four pluggable devices. On top of the Smart Furniture, there is an IrDA reader for user detection. On the pole, from the above, there is a LCD display with touch panel, an RFID reader, and a speaker.

Figure 1: Pole Type Smart Furniture

Lamp Type Smart Furniture

Lamp type Smart Furniture shown in Figure 2 is designed for home use. At home, there are various information acquired by sensors or devices. This type is designed to map various information to specific colors.

The lamp type Smart Furniture is cylindrical lamp which consist of 6 color light in serial. Each color light contains red, green and blue LED and we can assign its color from PC through RS-232C so that we can set each color. The height of lamp type Smart Furniture is 1450 mm and diameter of the lamp is 70 mm. The length of each color lamp is 200 mm so that 1200 mm of lamp are used to show encoded information.

Figure 2: Lamp Type Smart Furniture

Mirror Type Smart Furniture

Mirror type Smart Furniture shown in Figure 3 was designed to coordinate with other Smart Furniture. Mirror type Smart Furniture has an iPAQ installed inside. We can see LCD and use stylus through a half mirror. When LCD is turned off, the Smart Furniture looks just like a mirror. Linux is installed in the iPAQ and it has wireless LAN(IEEE 802.11b) connection. This type of Smart Furniture can be used as a personal reminder or a controller for various appliances at home.

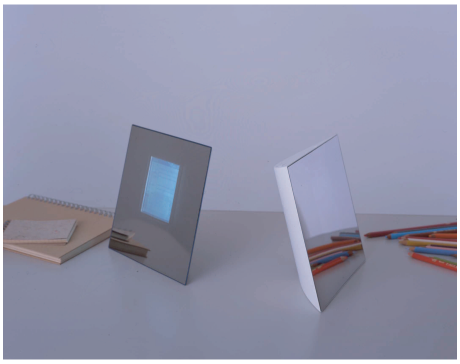

Figure 3: Mirror Type Smart Furniture

4 Middleware

Smart Hot-Spots need to federate heterogeneous devices. For this purpose, there are following requirements.

Smart Hot-Spots need to identify the status of devices and the environment surrounding a user of Smart Hot-Spot. Given that the devices are correctly identified, Smart Hot-Spots need to coordinate these devices. Simplest requirement for device coordination is creating a data path between two different devices. The status of devices and the environment can change when a user of wearable network moves. Thus, Smart Hot-Spots need to adapt to the identified status.

In the following section, we will describe the design of a middleware called Wapplet Framework, which we developed to meet the above mentioned requirements.

4.1 Wapplet Framework

A Wapplet framework[8] shown in Figure 4 is a distributed application framework for service roaming. This is originally developed for wearable computing environment. In such an environment we assume that user carry his own wearable computer and he use this device as a commander in order to improvise an application combining available services from the environment. Naturally he carries his identification information, application preferences and schedule information within the wearable device. In this Smart Hot-Spot network environment, we also assume that user carry his own device like PDA, cellphone or wearable computer, and Smart Hot-Spot play a roll of various kinds of service provider. Therefore, this wearable computing application framework can be applied as a middleware for our Hot-Spot networking environment. However, this is not all of the middleware, but we are engaged in the development of ad-hoc wireless sensor network technologies.

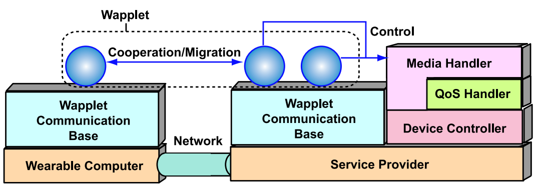

Figure 4: Wapplet Framework System Architecutre

Wapplet Framework consists of three components: Wapplet as application, Mission Mechanism as a control mechanism of Wapplet and Service Providers as the runtime of Wapplet on devices. Figure5 illustrates the outline of these components and the relationship between them.

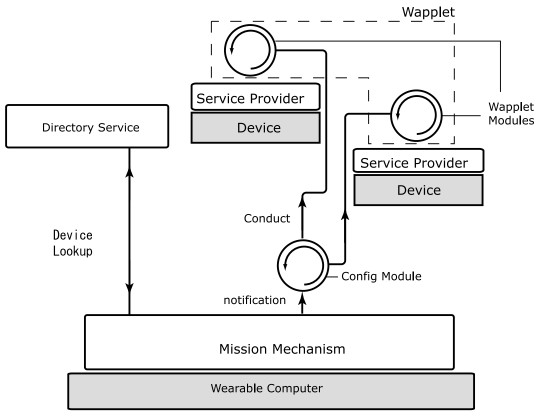

Figure 5: Software Organization

4.2 Wapplet

In Wapplet Framework, Wapplet uses multiple devices to accomplish a user's task. Hence, these devices should be integrated and their function should be provided to the application. To realize this, we adopt a distributed application method to build Wapplet in which, one application is separated into several modules, named Wapplet module. They are scattered on the devices that are required for the task. Each module is responsible for the task which should be performed using the device where the module is located on.

Furthermore, in order to achieve adaptation, we have made the module as migratable object. When adaptation becomes needed, a module should move from unavailable device to an alternative while preserving the state of the module.

Wapplet module can access only one data type which is defined in classes of devices(See 1). Hence, one Wapplet module can handle several devices which belong to an identical class.

Adaptation

We can divide changes of service availability into two cases. Wapplet Framework can handle both cases.

- a case in which an alternative service is available
  In this case, another alternative service can be found although a service becomes unavailable. A Wapplet module is able to start using the new service.

- a case without an alternative service
  In this case, no alternative can be found. A Wapplet module attempts media translation to compensate the unavailable service. If the media translation cannot be done properly, the task was using the service is guaranteed to be suspended correctly.

Mission Mechanism decides which way to adopt. The decision and adaptation are hidden from Wapplet programmers. Therefore, it is possible to build Wapplet applications without considerating the changes of service availability in detail.

4.3 Service Provider

Service Providers are executed on each device to provide Wapplet modules with both their execution environment and an interface to access a function of the device. Service Providers are classified from the viewpoint of providing an alternative service to Wapplet modules. Based on the top-level media-type of MIME definition, Service Providers are categorized by media types as shown in 1.

Table 1: Device Classes

class of devices		example
media type	I/O	
video	output	display
video	input	video camera
image	output	display, printer
image	input	digital still camera
sound	output	speaker, headphone
sound	input	microphone
text	output	display, printer
text	input	keyboard, scanner

For example, a speaker and a headphone are considered to provide an identical interface, sound output. Therefore a Wapplet module can use these devices alternately through one common interface.

4.4 Mission Mechanism

Mission Mechanism is a middleware on the wearable computer and it has two purposes: controlling a Wapplet and obtaining information about a device availability. For the first purpose, Mission Mechanism executes Configuration module for each Wapplet to recognize required Service Providers and to notify the changes of Service Provider. A Configuration module has functions to control Wapplet modules and to register a list of Service Providers to Mission Mechanism. To realize this, Mission Mechanism should provide an execution environment for Configuration module. Configuration module can be automatically generated from a description of Wapplet which is written by a Wapplet programmer.

Mission Mechanism looks up Service Providers that are needed to execute the Wapplet. If availability of a Service Provider has changed, Mission Mechanism notifies the Wapplet module which is executed on the Service Provider to migrate the module.

Configuration Module

Configuration module is a part of Wapplet, and it is executed on Mission Mechanism. As described before, it is generated from the description of Wapplet requirements and capability to allow Mission Mechanism to obtain Wapplet information. Wapplet programmers can write a description for Configuration module in XML[9] document which is based on a DTD we have designed.

Furthermore, programmers can write information about adaptation in the document, such as a module structure of the Wapplet and information about Serivce Provider needed for execution.

Configuration module can communicate with Wapplet modules and conduct them based on rules of adaptation written in itself. Mission Mechanism can detect changes of environment, such as service availability and user's state. In this paper, we are not concerned about the user's state but adaptation to service availability.

Regarding adaptation to service availability, Configuration module needs to maintain the following information.

- Location of Wapplet modules
  In order for Wapplet modules to communicate with

each other, Configuration module should maintain their location.

- Service Provider availability
  Configuration module should obtain changes of Service Provider availability for adaptation. Accordingly, Configuration module needs to receive the notification about Service Provider availability from Mission mechanism. When availability has been changed, Configuration module conducts migration of Wapplet modules.

Service Provider lookup

Mission Mechanism uses a directory service to recognize Service Providers that can be used in the current environment. The directory service can maintain Service Providers in a particular area , and it replies to a look up request from Mission Mechanism. The Area that is maintained by the Service provider is decided by the administrator of the system or network. However, we assume a directory service exists in each room or floor.

Each Service Provider should register its media type as an attribute to the Directory Service in its proximity. Due to this registration, Mission Mechanism can look up Service Provider by media type. Therefore, the look up does not depend on a device name.

5 Smart Hot-Spot Applications

In this section, we will discuss several applications that we developed using Smart Furniture, and issues related to each of them. Only the last example application, mobile TV-phone uses Wapplet framework for implementation, since only this application requires service roaming and coordination of inter-spots services.

5.1 Personalized Message Board

We developed Personalized Message Board(PMB) application which displays commercial messages on to a Smart Furniture. Message publishers register their messages to a Smart Furniture with the interests of their target users. Users carry a portable device which contain their interest. When users approach the Smart Furniture, it selects and displays the message appropriate for the user.

Smart Furniture detects users with the signal strength of their portable device, and passes the event to PMB. PMB then communicates with the portable device, and obtains the users interests.

When developing an application such as PMB which uses user information, privacy issues arise. Malicious Smart Furniture could disclose these information. One way to approach this issue, is to check within the user's portable device which information to pass to Smart Furniture.

5.2 Zero-stop Check-out

We developed an active authentication system intended for use in libraries and supermarkets[10], and installed it into the Smart Furniture as shown in Figure 6. Each user has a client device with wireless LAN and books with RFID tags. When the user pass through in front of the Smart Furniture, it detects MAC address of user's client device and books' RFID tags and automatically authenticate them.

The application needs to correctly bind objects to users. When there is only a single user in the sensing area, All the objects also in the sensing area may be bound to the user. However, this becomes difficult when there are multiple users and multiple objects. One approach is to tag objects with data writable tags, and make the users mark their ID to the objects.

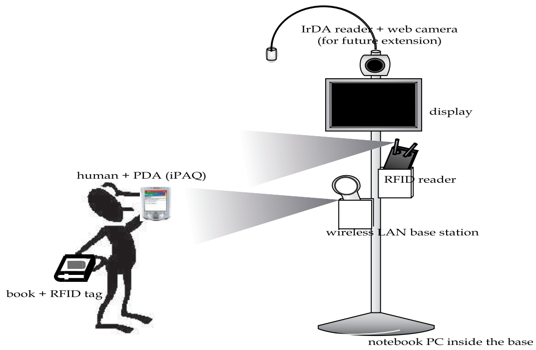

Figure 6: Active Authentication for Library

5.3 Mobile TV-phone

We developed a mobile TV-phone application using Wapplet framework. Figure 7 shows the architecture of the system. We assume that users carry a portable TV-phone device, with which they can talk to and look at a video of the person on the other side. We also assume that it is not equipped with a camera itself and can not send video images. When a user talking with such a TV-phone approaches Smart Furniture, display and camera on Smart Furniture is enabled. The video and camera modules move to the Smart Furniture and establish TV-phone connection.

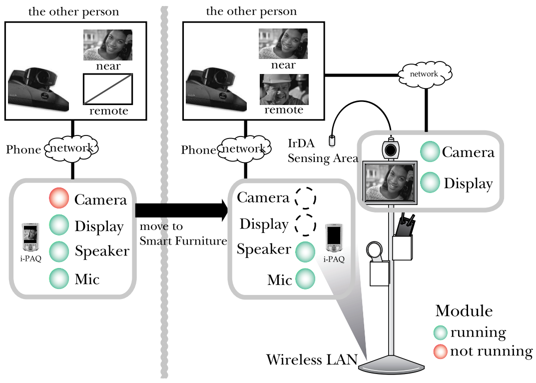

Figure 7: Mobile TV-phone

6 Conclusion

In this paper, we have described the design and implementation of Smart Furniture which can create a Smart Hot-Spot in the legacy non-smart space environment extemporaneously. There are three types of Smart Furniture; a pole type, a lamp type, and a mirror type. Users can use Smart Furniture as a gateway to the cyber world, as a service operator, or as a service receiver. By creating such Smart Hot-Spots, Internet access services, location-based services, context-aware services, service roaming services, and personalization services are provided.

We are planning to extend the functionality of Smart Hot-Spots by adding inter Smart Hot-Spot connectivity. Service roaming across Smart Hot-Spots will be provided. We are currently designing a table type, which will include advanced functions such as personalized access control. Malicious spot could disclose these information. We must balance user's privacy with it's usability.

Acknowledgement

We thank all members of the Tokuda Lab. and Uchida Yoko Corporation for their collaborative work on "Smart Furniture".

References

[1] M. Weiser, "The Computer for the 21st century", *Scientific American*, vol. 265, no. 3, pp. 66–75, Sept. 1991.

[2] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system", in *ACM Transactions on Information Systems, vol 10*, January 1992, pp. 91–102.

[3] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office", in *IEEE Personnel Communications*, October 1997, pp. 4(5):42–47.

[4] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "Easyliving: Technologies for intelligent environments", in *Handheld and Ubiquitous Computing*, 2000.

[5] K. Cory, R. Orr, G. Abowd, C. Atkeson, I. Essa, B. MacIntyre, E. Mynatt, T. Starner, and W. Newstetter, "The aware home: A living laboratory for ubiquitous computing research", in *International Workshop on Cooperative Buildings*, 1999.

[6] Hewlett-Packard Company, "CoolTown", http://cooltown.hp.com/cooltownhome/.

[7] T. Okoshi, S. Wakayama, Y. Sugita, S. Aoki, T. Iwamoto, J. Nakazawa, T. Nagata, D. Furusaka, M. Iwai, A. Kusumoto, N. Harashima, J. Yura, N. Nishio, Y. Tobe, Y. Ikeda, and H. Tokuda, "Smart space laboratory project: Toward the next generation computing environment", in *International Workshop on Networked Appliances*, 2001.

[8] M. Murase, T. Iwamoto, T. Nagata, N. Nishio, and H. Tokuda, "Implementation and evaluation of wapplet framework", in *Proceedings of International Workshop on Networked Appliances*, 2002, pp. 275–284.

[9] World Wide Web Consortium, "Extensible markup language(XML)", http://www.wc3.org/XML/.

[10] Kenta Matsumiya, Soko Aoki, Masana Murase, and Hideyuki Tokuda, "Zero-stop authentication: Sensor-based real-time authentication system", in *International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA 2003)*, Feb. 2003.