Creating Pervasive Services with Self-organizable Universal Boards

Naohiko Kohtake, Takuro Yonezawa, Ryo Ohsawa, Yuki Matsukura, Masayuki Iwai, Kazunori Takashio, Hideyuki Tokuda

Abstract

This paper proposes a novel way that allows users to create pervasive computing environments without experts. In this way, non-smart components which form everyday objects such as furniture and appliances commonly found in homes and offices are made smart in advance, not in the way to convert non-smart objects to smart ones by attaching computers, sensors, and devices. As our first prototype to realize this method, we have developed u-Texture, a self-organizable universal board as a building block that can change its own behavior autonomously through recognition of its location, its inclination, and surrounding environment by assembling them physically.

1. Introduction

In pervasive computing environment, computers, sensors, devices, and networks are embedded in or attached to non-smart objects and connected for creating context-aware pervasive services. Recently, many non-smart objects are converted into smart objects by various approaches, and a large number of smart homes [6], smart rooms [7], smart furniture [11], etc., are realized and they support human activities. However, it is difficult for users who are unfamiliar with computing technology, especially, pervasive computing technology, to create and maintain pervasive computing environments without experts. Moreover, most of these environments are handmade by users and the cost and time for building such an environment is a barrier to the development of appropriate pervasive services.

To solve this problem, our vision is to establish technologies which enable non-expert users to create a pervasive computing environment easily. With these technologies, anyone can create the environments anytime and anywhere, and obtain context-aware pervasive applications. As one of our approaches to establish these technologies, we have been developing non-smart objects' components as smart ones in advance, not in the way to convert existing non-smart objects to smart objects. In the case of past researches for realizing pervasive computing environments, objects such as tables, shelves, and drawers in homes and offices are not originally smart, namely most of them are enhanced their functions by attaching computers, sensors, and devices [2, 10]. On the contrary, our approach aims to enlarge functions by making components, such as boards, legs, and props of tables and shelves, smart in advance. As users assemble objects with those smart components, they alter their functions autonomously according to the shapes of how they are assembled, and then, objects work as smart objects. Those components are beforehand made to have functions to recognize the

Keio University, {nao,takuro,ryo,matsu,tailor,kaz,hxt}@ht.sfc.keio.ac.jp, http://www.ht.sfc.keio.ac.jp/˜nao/

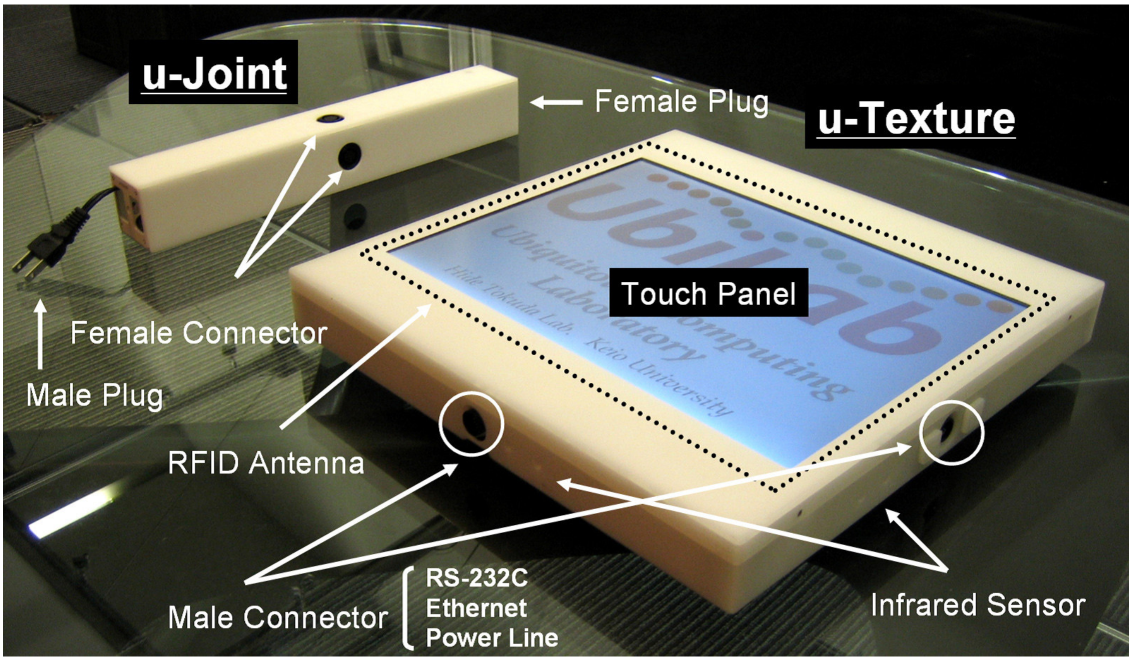

Figure 1. Appearance of a u-Texture (right) and a u-Joint (left)

surroundings and to realize the pervasive computing environments.

With this approach, it will make possible for a user to realize the pervasive computing environment easily by assembling components physically, without knowledge of each computer, sensors, and devices or setting up them. Because they are getting smaller and cheaper day by day, it is considered as possible to pre-install them in most of the components in the future. In this paper, we introduce a u-Texture which is a prototype made as a smart component beforehand and has been developed for the purpose of realizing the above approach. The u-Texture is a self-organizable universal board, and assembling u-Textures realize smart objects corresponding to the assembled shapes.

In the process to develop a prototype of the first smart components, we focused on the ”board shape” which is the basic to form most furniture such as tables and shelves. Because we have regarded that, by assembling furniture with those smartlized components, each smart component would work accordingly to the assembled shape. As a result of that, each furniture turns into and works as a smart object. A connecting structure is required to assemble one u-Texture and another one in this prototype; the connections can be implemented horizontally and vertically. For that purpose, this prototype system is designed to use props able to support assembled u-Textures and also to connect them in both vertical and horizontal directions. The prop is called a u-Joint and, with the u-Joint, a u-Texture can be assembled in various shapes. Several smart furniture realized by the same u-Textures and their availabilities are also introduced.

2. u-Texture

Figure 1 shows an appearance of u-Texture and u-Joint. Basic three actions of u-Texture to be functional as a smart object are given as follows.

Figure 2. Smart Table (upper left), Smart Multi-display (upper right) and Smart Shelf (lower): u-Texture connects with other ones horizontally and vertically, and collaborates with each other by exchanging location and inclination information, commands, and data.

Recognition. As a user assembles u-Textures, those assembled u-Textures share information such as whether connected or not, directions of connection, IDs of adjoining u-Textures, and inclinations of themselves. With that information, each u-Texture recognizes its assembled shape and location on the assembled shape.

Adaptation. Available applications corresponding to the recognized smart object will be selected automatically among different applications pre-installed in each u-Texture. A user's input determines one application when there are several choices.

Cooperation. Once an application to be worked is determined, each u-Texture behaves autonomously and works together according to the smart object shape, and each location and inclination.

3. Example Applications

We have developed some example applications to identify the potential of u-Texture. The followings are several kinds of these applications which are shown in Figure 2.

Smart Table enables to exchange data of each u-Texture on its screen. It can be created by connecting u-Textures and setting it horizontally. When several u-Textures are connected, an arrow will indicate the direction of the other connected u-Texture on screens of each u-Texture. The data will be copied to the u-Texture connected to the direction of the arrow by dragging it to the arrow.

Smart Multi-display enables to display data from a u-Texture widely in cooperation with the connected u-Textures. It can be created by connecting u-Textures and setting it vertically. In the case of connecting u-Textures equally in length and breadth like two times two or three times three, a magnified picture of one designated u-Texture will be displayed. In other cases, longer length takes priority to be output of its magnified picture and un-showed parts also can be scrolled by dragging the picture.

Smart Shelf is a shelf that recognizes what is put on itself. It can be created by assembling u-Textures vertically and horizontally. It enables to browse information about a name, owner, ID and duration of times on the u-Texture of a thing which user put on u-Textures set up horizontally. The information can be checked on a display of another u-Texture connected vertically to the u-Texture on that a thing is placed. These u-Textures are working cooperatively. This system is used via network for finding on which Smart Shelf is placed with things attached RFID tags.

4. System Architecture

The prototype of the u-Texture is 320 mm square, 48 mm thick, and weights 4300 g, approximately the size and shape of a pizza box. u-Texture consists of the following devices with basic computer devices such as processor (Mobile Pentium processor, Intel) and hard disk drive.

Each of the four sides is implemented with RS-232C serial interfaces to examine and find out whether the u-Texture is connected to other u-Textures or not and, if so, in which direction; infrared sensors can tell if other u-Textures exist nearby. Two Dual Axis Accelerometers (ADXL202, Analog Devices) with outputs are processed by a microcomputer (H8/3664F 16MHx, Hitachi) to process their outputs are implemented as inclination sensor so the u-Texture can discover information about itself. The microcomputer dispatches the u-Texture ID and IDs that correspond to each of the four sides as data via each RS-232C serial interface and each infrared sensor. Also, by connecting to other u-Textures, the microcomputer receives data from those u-Textures. Moreover, the microcomputer outputs information about the area around the u-Textures and itself to the main computer by processing these data. Ethernet interfaces are implemented in each of four sides of a u-Texture and connected to the main computer through a USB hub. It is implemented with a touch panel (AST-140, DMC) as an input device for direct input by users and a display (SXGA 14.1 type) and a speaker as output devices. Moreover, u-Texture is built in with IEEE 802.11b wireless LAN interface and a RF-ID reader (13.56MHz, Sofel) to exchange data with surrounding environment. To make the u-Texture lighter, a board-type Li-ion battery (Slim60, BAYSUN) is used for the electric battery. It can work independently in an active condition for about one hour on average.

Software is adopted in accordance with the u-Texture's main processing, recognition, adaptation and cooperation. The same and unique middleware are working on all the u-Textures, and they exchange data regarding connections and inclinations explained in the second section and also deal with processing for cooperation among connected u-Textures. This prototype has been developed to store same database on each u-Texture in advance, and it is possible to update the database through a wireless LAN and to access to the database on the network by each u-Texture.

The main purpose of the u-Joint is to support the assembled u-Textures and to enable connections of u-Textures vertically and horizontally. As for the inside configuration, connected u-Joints connect the power supply line and supply power source to connected u-Texture. RS-232C serial and Ethernet lines are set up to distribute data to the four sides of the u-Texture via the USB hub.

5. Related Work

There are a number of researches aiming at establishing technologies which enable to create pervasive computing environments easily without experts have been implemented. Impromptu [2] enables us to rapidly build pervasive computing environments with autonomous cooperation of consumer appliances, such as computing devices, PDAs and digital cameras, and appliances which are made smart by Smart-Its [5] technology. The advantage of u-Texture is that it can recognize assembled forms and select executable context aware applications autonomously.

Researches for creating pervasive computing environments by altering functions of systems pre-built-in with computers and sensors have been implemented in various approaches, such as researches in changing functions by contacting [10][4][8] or by assembling [9][3][1] systems. As compared with these systems, the u-Texture can recognize assembled shapes and its position in the shapes autonomously, so that u-Texture can select several executable applications with recognized information. Moreover, the u-Texture can exchange data with peer-to-peer communications, so that basically it can work cooperatively without a server.

6. Conclusions

We proposed a novel way that aims to enlarge functions by making non-smart objects' components smart in advance for creating pervasive computing environments easily. u-Texture enables to change its own behavior autonomously by recognizing its location, it inclination, and surrounding environment. Our proposed way contributes to widespread many kinds of pervasive services rapidly.

7. Acknowledgement

This research has been conducted as part of Ubila Project supported by Ministry of Internal Affairs and Communications, Japan. We would like to express our appreciation to Keio Ubila Project team and Uchida Yoko Co.,Ltd. for supporting this research.

References

[1] David Anderson, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S. Yedidia. Tangible interaction + graphical interpretation: a new approach to 3d modeling. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 393–402. ACM Press/Addison-Wesley Publishing Co., 2000.

[2] Michael Beigl, Tobias Zimmer, Albert Krohn, Christian Decker, and Philip Robinson. Creating Ad-hoc Pervasive Computing Environments. In *Pervasive 2004 Advances in Pervasive Computing*, pages 377–381, Vienna, Austria, April 2004.

[3] Matthew G. Gorbet, Maggie Orth, and Hiroshi Ishii. Triangles: tangible interface for manipulation and exploration of digital information topography. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 49–56. ACM Press/Addison-Wesley Publishing Co., 1998.

[4] Ken Hinckley. Distributed and local sensing techniques for face-to-face collaboration. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, volume 36, pages 81–84, 2003.

[5] Lars Erik Holmquist, Friedemann Mattern, Bernt Schiele, Petteri Alahuhta, Michael Beigl, and Hans-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. *Lecture Notes in Computer Science*, pages 116–122, 2001.

[6] Cory D. Kidd, Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth D. Mynatt, Thad Starner, and Wendy Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Cooperative Buildings*, pages 191–198, 1999.

[7] Christian Muller-Tomfelde Shin'ichi Konomi Norbert Streitz, Peter Tandler. Roomware: Towards the next generation of human-computer interaction based on an integrated design of real and virtual worlds. In J. Carroll, editor, *Human-Computer Interaction in the New Millenium*, pages 553–578. 2001.

[8] Jun Rekimoto, Brygg Ullmer, and Haruo Oba. Datatiles: a modular platform for mixed physical and graphical interactions. In *CHI*, pages 269–276, 2001.

[9] Hideyuki Suzuki and Hiroshi Kato. Interaction level support for collaborative learning algoblock an open programming language. In *CSCL '95: The first international conference on Computer support for collaborative learning*, pages 349–355, 1995.

[10] Peter Tandler, Thorsten Prante, Christian Muller-Tomfelde, Norbert A. Streitz, and Ralf Steinmetz. Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In *UIST*, pages 11–20, 2001.

[11] Hideyuki Tokuda, Kazunori Takashio, Jin Nakazawa, Kenta Matsumiya, Masaki Ito, and Masato Saito. Sf2: Smart furniture for creating ubiquitous applications. *IEEE Proceedings of International Workshop on Cyberspace Technologies and Societies (IWCTS2004)*, pages 423–429, 1 2004.