Ubiquitous Smart Interaction Space

Sangchul Ahn, Donghoon Kang, Hyoung-gon Kim and Heedong Ko

Abstract—The context aware systems are often designed and implemented based on a specific scenario with predefined resources. In this paper, we describe a ubiquitous interactin space that supports rapid context-aware applications reflecting the smart interaction space with dynamic resources and provides a programmable interface in integrated development environment (IDE) in .NET framework.

***Index Terms*— Context aware systems, smart spaces, Universal Plug and Play, Web Service**

I. Introduction

One of the most distinguishing characteristics of a context-aware application is that the application is aware of its dynamic operating environment in physical space as well as the user. The physical space consists of many physical devices embedded with computing and internetworking capabilities that may be involved or released dynamically for contnext-aware applications. Here, we distinguish system context from user context as all those device states available for a context-aware application.

In order to develop context-aware application efficiently, we need a middleware support that collects and controls the system context efficiently. That is, the system provides the direct programming interface of its own pervasive system resources that are changing dynamically with user's interacting environment. Context-aware interaction manager (CAIM) [1] is being developed as a middleware to support for application programming of smart devices and sensors in smart interaction space..

II. Context-aware Interaction manager

In this paper, we describe CAIM - a service oriented architecture based middleware to manage the system context and to provide the unified programming interface. In service oriented architecture manner, each service provider has the description which represents its own capability. It is a strong point for supporting the dynamic discovery of resources and the reconfiguration of the system. Therefore, we have assumed every physical object in the environment has the self-description. Actually, we use the Universal Plug and Play (UPnP) [2] as the device platform.

The key features of CAIM are the following:

- Dynamic resource discovery: CAIM reflects the dynamic changes of system resources which are comparable with Universal Plug and Play devices. It discovers and binds whole upnp devices in domain. If the user wants to add a new system resource, it can be added simply as a upnp device.
- Maintaining the system context: When a new resource involves, CAIM monitors and caches each states of resource continuously. The contents of system context that CAIM has are which resources are available, what services that the resource has are, how the services can be invoked are, and what the states of the devices are.
- Application programming interface: CAIM uses the Web Service to provide the unified application programming interface. To support Web Services, it describes the WSDL (Web Service Description Language) [] definition according to the type of resource. When a new resource is discovered or removed, it updates the definitions dynamically. If, the user connects to CAIM through the general IDE which supports Web Service such as Microsoft Visual Studio .NET, it publishes the current available resource definitions. Then, the user can build context-aware applications alike the general software programming.
- Integrated Web Server and UDDI registry: CAIM communicates with other components (UPnP for resources and Web Service for applications) over http. It contains the simple web server. It is not required other web server or UDDI for using Web Service. The user can bind the resources by searching the UUID (Universal Unique IDentifier) or device type.

III. Implementation

We have developed CAIM based on Java. It is composed of three major components; Device Adapter, Interaction Manager and Web Service Interface. Device adapter detects and binds the UPnP compatible devices. When the devices are bound, it subscribes the events of devices. Finally, it also executes the actions from the applications. Interaction Manager stores the system context to repository, transfers the messages between Device Adapter and Web Service Interface, and schedules the tasks. Web Service Interface is a communication interface for

This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

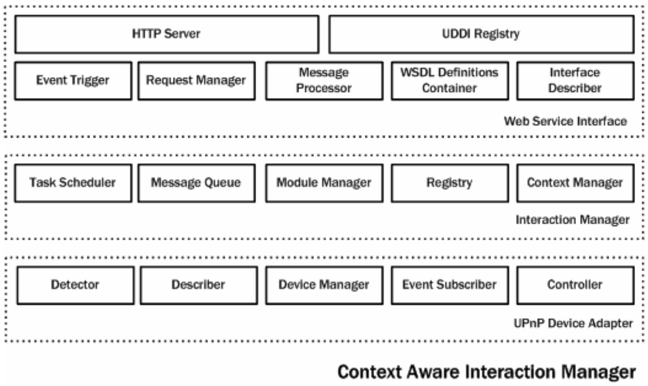
Sangchul Ahn is with the Imaging Media Research Center, Korea Institute of Science and Technology, Seoul, 136-751 KOREA. He is also with the Yonsei University, Seoul, 120-749 KOREA. (corresponding author to provide phone: +82-2-958-5637; fax: +82-2-958-5769; e-mail: prime@kist.re.kr).

Donghoon Kang is with the Imaging Media Research Center, Korea Institute of Science and Technology, Seoul, 136-751 KOREA (e-mail: chohgk@kist.re.kr).

Hyoung-gon Kim is with the Imaging Media Research Center, Korea Institute of Science and Technology, Seoul, 136-751 KOREA (e-mail: chohgk@kist.re.kr).

Heedong Ko is with the Imaging Media Research Center, Korea Institute of Science and Technology, Seoul, 136-751 KOREA. (e-mail: ko@kist.re.kr).

applications. It generates the WSDL definitions for the registered devices. And it processes the input/output messages to interact with applications. UDDI Registry provides the search and binding mechanisms for the WSDL definitions.



IV. SMART INTERACTION SPACE

The smart interaction space consists of UPnP-based smart floor, media wall, and smart ceiling. UPnP smart floor is implemented using FSR (Force Sensing Resistor) array and can provide context of user activity as well as tracking of users without attaching sensor to the body. Ubiquitous media wall is a modified implementation of CAVE-like environment [4] with front projection system. Microphone array and speaker array is also included for sound-based interaction service purposes. Smart ceiling consists of UPnP LED lighting system for responsive illuminations, active video camera array with pan-tilt-zoom capabilities.

A. Smart Floor

Fig.1 shows an overview of our smart location tracking system. Using a GPIO(General Purpose Input Output) function of the Single Board Computer with Intel PXA 255 CPU, the FSR sensor array is scanned to get processed pressure data of a sensor in real-time. The size of an experimental room space is about 7.2m x 6.6m with each block size is 0.60m x 0.6m. This embedded board is working as a UPnP device, and can be controlled through UPnP Control Point.

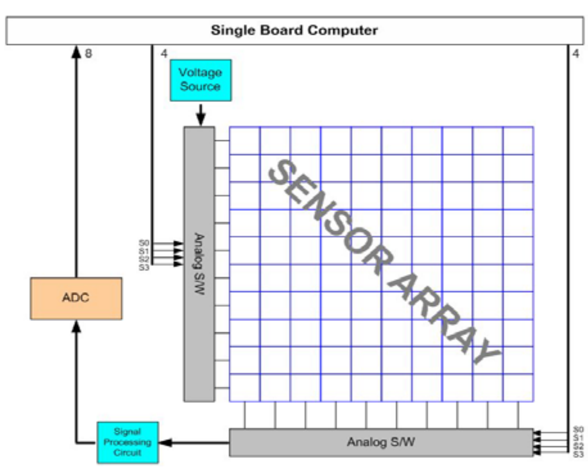


Fig. 1. UPnP Smart Floor

The FSR (Force Sensing Resistor) is made of PTF (Polymer Thick Film) device, and exhibits decreasing resistance value while increasing the applied force to the active surface. Smart floor systems have been implemented in numerous research efforts [5][6][7][8]

B. Media wall

Fig. 2 shows the conceptual view of the media wall. The media wall displays the media space that can be controlled by smart interaction devices in the room..

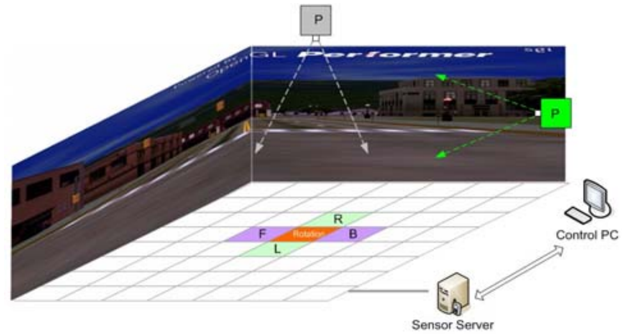


Fig. 2. Conceptual view of the ubiquitous media wall

Display Wall Projection. The 4 side of the room are display wall. Since the aspect ratio of the wall is different from the video of the projector, 2 projectors are used for each wall side. Software edge-blending and image warping technology are used for this purpose.

Floor Projection: The surface of the smart floor can be used as screen and may mark the floor with interaction cues in Fig 3. The center of the ceiling is equipped with active Pan-tilt-zoom projector with UPnP video render capability. This can be used for various user interaction markings.

C. Smart Ceiling

Fig. 3 shows the smart ceiling comprising UPnP LED lighting modules, projector for the floor video, and active camera array. 10 modules of high power (1W x 10) LEDs from Luxon are used as illumination of the smart space using UPnP device. It can be remote controlled trough UPnP control device.

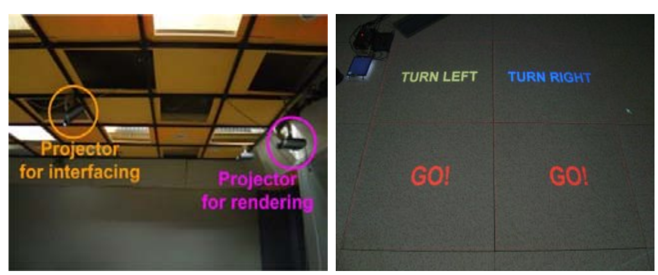


Fig. 3. Actual figure of smart ceiling and projected image for interaction

Active video camera array

8 active cameras array with pan-tilt-zoom capability video cameras are connected through USB2.0. It can be used for the

various multi-view vision systems, and currently used mainly for the real-time visual hull with dynamic view frustum.

D. UPnP Device Interface Design

Single board computers with PXA255 CPU are used for the UPnP device implementation. The CPU is based on the Intel XScale micro-architecture, and provides 85 GPIO pins for generating and capturing of application-specific input and output signals. Each pin can be programmed as either an input or output using the GPIO Pin Direction Register (GPDR). When programmed as an output, the pin can be set high by writing to the GPIO Pin Output Set Register (GPSR) and cleared low by writing to the GPIO Pin Output Clear Register (GPCR). The set and clear registers can be written to regardless of whether the pin is configured as an input or an output. Fig. 4 shows the implementation of the smart floor using this board.

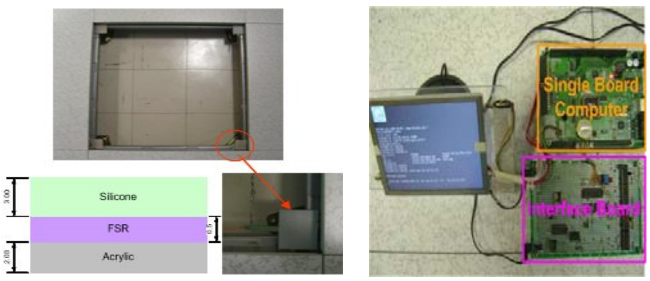


Fig. 4. UPnP device interface for the smart floor.

V. Conclusion and Future Work

We believe an infrastructure support by CAIM provides adequate support for resource awareness and unified application programming interface to facilitate rapid context-aware application development. Currently, .NET framework is supported as an integrated development environment for CAIM. We have made the CAIM kernel available for download at http://caim.kist.re.kr for those interested in context-aware application prototyping in UPnP device environment..

The room size smart interaction space is being extended to a connected corridor and a number of office and meeting rooms with UPnP cameras, displays and crickets. With the extended interaction space with location tracking and media display and input devices, the user may develop novel applications that are unique to ubiquitous interaction space that bridges the gap between virtual and physical space..

References

[1] Donghoon Kang, Sangchul Ahn, Heedong Ko, Weduke Cho, and Youngtack Park, “Context Awareness for ubiquitous Computing System”, Journal of Korea Intelligent Information Systems Society 2004-Vol.1, 2004.

[2] Universal Plug and Play Device Architecture Version 1.0 Available: http://www.upnp.org/download/UPnPDA10_20000613.htm

[3] Jerry R. Hobbs and Feng Pan, “An ontology of time for the semantic web”, ACM Transactions on Asian Language Information Processing (TALIP) Volume 3, 2004, pp. 66-85.

[4] Carolina Cruz-Neira, “Projection-based Virtual Reality : The CAVE and its Applications to Computational Science”, PhD Thesis, University of Illinois at Chicago, 1995.

[5] Paradiso, Joseph, Craig Abler, Kai-yuh Hsiao, and Matthew Reynolds, “The Magic Carpet: Physical Sensing for Immersive Environments.” In Late-Breaking / Shot Demonstrations of CHI’97, pp. 277-278, 1997.

[6] Pinkston, Kerkhoff, and McQuilken, “The U.T. Touch-Sensitive Dance Floor and MIDI Controller,”

[7] M. Addlesee, A. Jones, F. Livesey, and F. Samaria, “The ORL Active Floor.” IEEE Personal Communications, pp. 35-41, Oct. 1997.

[8] J. Robert and Gregory D. Abowd, “The Smart Floor: A Mechanism for Natural User Identification and Tracking,” CHI 2000, 2000