PROCEEDINGS OF
THE 2007 INTERNATIONAL CONFERENCE ON
PARALLEL AND DISTRIBUTED PROCESSING TECHNIQUES AND APPLICATIONS

PDPTA 2007

Volume II

Editor

Hamid R. Arabnia

Associate Editors

Sanjay Ahuja, Jesus Carretero
Ping-Tsai Chung, Jose D. Garcia
Kazuki Joe, Mario Nakamori, Chung Ng
Hiroaki Nishikawa, Zornitza Prodanoff
Christian Rehn, Ashu M. G. Solo

WORLDCOMP'07
June 25-28, 2007
Las Vegas Nevada, USA
www.world-academy-of-science.org

©CSREA Press

Session: PARALLEL AND DISTRIBUTED ALGORITHMS

Session: NOVEL APPLICATIONS AND SUPPORTING UTILITIES

An Interoperability Mechanism for Transparency of UPnP AV Devices

Hyunju Lee
Dept of Computer Science
Kyungpook National University
Daegu, Korea

Sangwook Kim
School of Electrical Engineering
and Computer Science
Kyungpook National University
Daegu, Korea

Abstract -*UPnP AV architecture provides interoperability among AV devices as a home network middleware technology. It automatically integrates and manages various AV devices that are connected to the home network. A large number of home entertainment systems based on UPnP AV architecture target the support of the multimedia service that is based on home server as a media content provider. In this paper, we propose an interoperability mechanism for transparency of UPnP AV devices to provide ubiquitous home. It provides two main services. There are the media streaming service for remote AV contents and the UI sharing service for remote device. Users can receive the high-quality multimedia (or device) management service regardless of media content (or device)퓔 location and playback types via t he proposed interoperability mechanism.*

Keywords: UPnP AV, media contents, remote AV contents, remote UI, home entertainment center.

1 Introduction

A large number of heterogeneous devices with different capabilities can be run on the home network. And they communicate with other network devices using the home network middleware such as UPnP, Jini, HAVi, etc. Specially, the UPnP (Universal Plug and Play) [1-2] forum developed UPnP AV (Audio/Visual) architecture provides the media distribution services in home network as the standard of multimedia system. It also provides the AV device communication mechanism to share AV media contents among different kinds of multimedia appliances.

Most of the existing home entertainment systems provide very stable technologies to provide the multimedia service to users regardless of AV device퓔 location and the media content퓔 playback types. And these systems have high-quality performance such as easy application and high-adaptation performance for related technologies. But, most of the servers in these systems provide very restricted service, which stores many media contents of different kinds and provides the media service to other AV devices as a content provider. Moreover, most of the servers do not permit the control from other client devices in system'side. And, the roles of components in client device are not clearly defined.

In this paper, we propose an interoperability mechanism for transparency of UPnP AV devices in order to provide the ubiquitous media home. The interoperability mechanism for all UPnP AV devices means that the home entertainment system guarantees transparency of all AV devices and it improves the usability to users. It includes the control service for a wide variety of applications and AV contents. If the interoperability mechanism is applied to the home entertainment system, the system can play various AV contents which exist in local device, and discovers remote AV devices (and browses and plays their AV contents) automatically, and shows its UI and execute their applications in remote device. To provide the interoperability of all AV devices, we composed the home entertainment center which contains two controllers as core technologies.

This paper is organized as follows. Section II, we describe the several related works. In section III, we describe the interoperability mechanism of AV devices including two services as a core technology. In section IV, we present implementation results. Finally, conclusions are given in the last section.

2 Related Works

The home entertainment service should be provided stably. There are several well-known home entertainment technologies such as Meedio[3], Showshifter[4], Linpus Media Center[5], MythTV[6], etc.

Meedio is based on MS-Windows, which is similar to Windows Media Center. This system is the open source software for HTPC (Home Theater PC), and provides full-screen graphic. It has very small-size code and runs stably. It provides consumers with a customizable interface to browse, manage, and access nearly any type of digital media or information using a remote control, touch-screen, or PDA. Its integrated media players, digital video recorder (DVR), photo slideshows, weather forecasts and many other features can be easily extended with additional plug-ins from the Meedio developers network. But, it is loaded to specific commercialized products such as Hewlett Packard PC or Sam Sung PC.

Showshifter is an All-In-One media center which can play the media contents in PC. That means that it provides access to all forms of home entertainment through a single interface. It is connected with various applications such as TV, radio, juke-box, CD/DVD player and image viewer. Also, it provides wire/wireless network and the control service via the remote controller. But, this package is not open source project and this system cannot be run in other AV devices except for PC.

Linpus Media Center and MythTV are based on Linux desk-top. Linpus Media Center provides various embedded systems such as Intel X86, StrongARM and Xscale. It supports fast and easy installation, various storages and easy-to-use navigation. It contains a remote controller to control the media contents. However, it does not support other media formats except for MPEG format, and it also does not support the MediaRenderer. Moreover, this system cannot be run in other AV devices except for PC. In spite of its high-performances, it has need of too many requirements within the framework of hardware/software.

MythTV is a very famous homebrew project. It provides various and powerful services such as TV recording, video/music/DVD playback, weather, game emulation and news service. All functions are existed with plug-in form except for TV service. It supports various menu compositions and an easy-to-use user interface. MythTV (version 0.20) contains an UPnP server among UPnP AV architecture. But, the system뭩 size is considerable and it requires too many resources as much as it provides various functions and capabilities. For example, users have to install MySQL because it uses Database. To perform the fundamental function, a large number of packages have to be installed in advance. Moreover, users have to keep up the latest libraries because it is based on the latest development environment. And users have to update all packages before setting up. It does not support UPnP client modules except for UPnP server conclusively.

These technologies are very stable, and these are very useful to applications of related technologies. In spite of their excellent performance for AV content playback and control, server in these systems does not permit access and control from the remote other AV devices. Users only can receive the AV contents playback service from the server. Therefore, in order to support interoperability among AV devices including all servers in home network, the remote contents sharing service can be a core technology for ubiquitous home entertainment.

3 An Interoperability Mechanism

A large number of servers and AV devices can be existed in home network as multimedia appliances. Each AV device can become a server device or a control device or a media player device according to their performances and properties. Also, a device can become a server and client at the same time if it has high performance and flexibility.

Figure 1 shows the architecture of the UPnP AV-based home entertainment system.

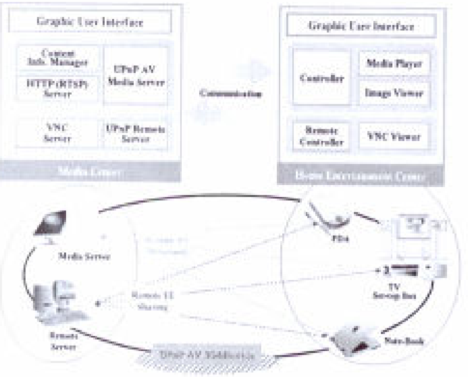

Fig. 1. The architecture of UPnP AV-based home entertainment system

The media center contains two servers (i.e., the AV MediaServer and the remote server) as a contents provider, and the home entertainment center contains two controllers (i.e., the controller and the remote controller) as a client in figure 1. Two controllers are systematically operated, and are independently managed. To provide the interoperability of AV devices, the mechanism contains two services. There are the media streaming service for remote AV contents and the UI sharing service for remote device. To share AV media contents which are included in server device, we applied the UPnP AV architecture and composed the integrated controller in the home entertainment center. To share the UI (User Interface) of a remote device and to control it in remote, we applied the remote UI sharing mechanism to the system via the remote controller.

3.1 The Media Streaming Service for Remote AV Contents

To compose the network and to build communication among AV devices, we used UPnP AV architecture. UPnP AV architecture consists of MediaServer, Control Point and MediaRenderer [7-9].

AV contents need a MediaRenderer to play itself because of its own playback characteristic. If Control Point and MediaRenderer are embedded together in the same device and two components are integrated, usability and management abilities will improve in user뭩 point of view because number of necessary devices can be reduced and two components can be simultaneously managed by using the same device. In this paper, we composed the integrated controller which is an integrated module of Control Point and MediaRenderer. It can be embedded in all AV devices which can play AV media contents and can perform role of the MediaRenderer.

Figure 2 shows the architecture of the controller and its processing flow with AV MediaServer.

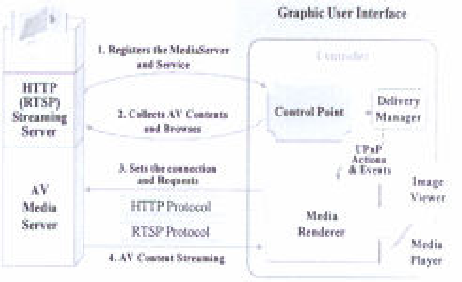

Fig. 2. The architecture of the controller and its processing flow with AV MediaServer

The MediaServer is used to locate AV contents and it includes a wide variety of devices. Its main purpose is to allow Control Points to enumerate (i.e., browse or search for) media content items that are available for users to render. The MediaServer contains three services. There are a content directory service, a connection manager service, and an optional AV transport service. It depends on supporting transfer protocols.

The MediaRenderer in the controller is used to render a media content received from the MediaServer on its local device and it includes a wide variety of devices. Its main feature is that it allows the Control Point to control how content is rendered (i.e., volume, brightness, etc). Additionally, it also allows the user to control the flow of the content. The MediaRenderer contains three services. There are a rendering control service, a connection manager service, and an optional AV transport service. It depends on transfer protocols which are supported by the server. The MediaRenderer is connected with a media player and an image viewer. The media player plays an audio/video content via the URI (Uniform Resource Identifier) information of media content which a user chooses by the Control Point on screen. At this time, the MediaRenderer transmits it to the media player. The image viewer plays image contents on the screen. It receives the playback information from the MediaRenderer like a media player. A user can view image contents with slide-show form. Media player and image viewer are joined with plug-in form as each independent component.

Control Point in the controller coordinates the operation of the MediaServer and MediaRenderer, usually in response to user interaction with the Control Point뭩 UI. Control Point in UPnP network is a controller capable of discovering and controlling other devices using UPnP뭩 various mechanisms. After device discovery, a Control Point retrieves a new device뭩 description and gets a list of associated services and retrieves service descriptions for the interesting service and invokes actions to control the service and subscribes to the service's event source.

The delivery manager performs the role transmitting various UPnP actions and events which are happened by user. It exists between Control Point and MediaRenderer.

Content processing from content choice to rendering is very complex and has various action scenarios in user뭩 point of view. Therefore, an efficient UI composition is very important. In this paper, we composed the GUI (Graphic User Interface) module for efficient UI composition as well as Control Point뭩 essential function execution. One of its important roles is that it integrates Control Point and MediaRenderer. The GUI module provides the dynamic and visual screen and processes various interactions that happen on screen by user, and delivers it to each device according to the interaction types. GUI module runs on top-level of the system as an interaction module with users, and it supports an easy-to-use interface. Users can easily add the related modules with plug-in form. It consists of five components. There are the plug-in interface, the display menu manager, the event handler, the skin manager and GUI libraries.

Figure 3 shows the architecture of the GUI module.

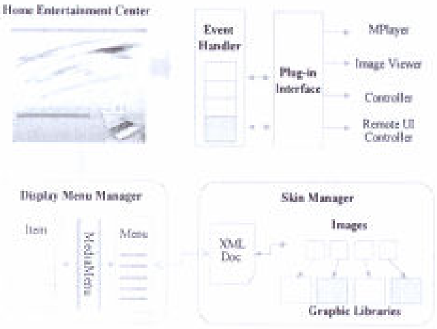

Fig. 3. The architecture of the GUI module

Display menu manager co mposes a menu that is related with lists of the remote server and AV contents. It creates a directory after classification into each item (i.e., Video, Audio, Images and UPnP AV/remote server) via the delivered information from each server. There are Menu, MediaMenu and Item as its sub modules. Menu composes an initial main-menu after loading the defined menu. MediaMenu connects the item with the menu which is browsed on the screen. It has the location information of each Plug-in module. MediaMenu creates a directory and sets the properties of each item after creating a menu. Item composes the sub item according to the property of the selected menu by the user. It classifies the sub item (i.e., Video, Audio, Image, Directory and Playlist) according to the kinds and properties of AV contents. Each item is connected with different kinds of images. Skin manager shows a visual screen to user via the display device. It handles a skin and image files. A skin consists of a set of XML files which is .fxd format, and image files are used in skin composition. We used .png and .jpg as image file format. GUI libraries consist of graphic libraries for visual and dynamic screen composition. For example, to provide various screen composition such as window templates and popup boxes, we used SDL (Simple DirectMedia Layer) cross-platform multimedia library which supports various operation system platforms (and languages).

Event handler handles user퀠 various input events. It can handle keyboard and remote controller keys. Plug-in interface connects a local application or an UPnP device with an item of the display menu manager. It runs with program codes that can perform activation, communication and close. For performance improvement of the whole system and stable running between the UPnP module and the GUI module, real-time message switching and effective data movement between modules are needed. Moreover, to combine various modules relationally, the effective interface is needed. To minimize random delay time which is happened while running modules, we deleted needless messages and action calls in UPnP source code. Plug-in interface processes four plug-in modules. For local audio/video/images content playback, two plug-in modules are connected with two local players (i.e., a media player and an image viewer). For remote contents sharing, two plug-in modules are connected with two controllers. All plug-in modules run according to user event.

For the visual and dynamic screen composition, the controller provides various display types. Service contents are displayed with menu form. AV devices and media contents are displayed with image item form. Each image item is different according to the media format such as audio, video, images, and etc. If a user chooses a media content, system and playback control are converted to the rendering device until the media content퀠 playback is terminated.

3.2 The UI Sharing Service for Remote Device

Remote UI(User Interface) [10-13] is an interface mechanism which allows access from another local device to the remote device퀠 applications. It is a useful mechanism in home automation, security system, home entertainment system, etc. because it provides interoperability and transparency of all AV devices. In this paper, we applied the remote UI sharing mechanism to the remote device and home entertainment center in local device via the remote controller. The remote controller automatically recognizes a remote device and shows the screen of the remote device and shares its UI on the home network. AV client device can access to remote device through the network that is connected by a unique home IP address regardless of network connection location. For example, a user can run application of PC in library through TV of living room, and can display control menus on all AV devices and can remotely control all media contents in remote device. Also, a user can directly edit AV contents? properties such as display quality, playback order, content characteristic, etc.

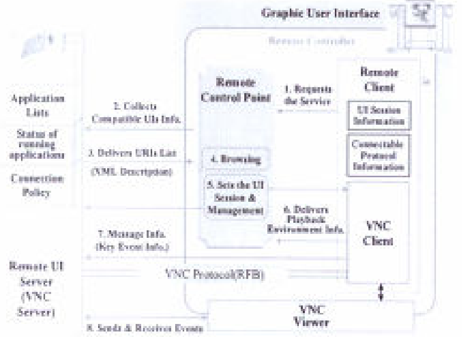
Graphic User Interface
Remote Controller
Application Lists
Status of running applications
Connection Policy
Remote UI Server (VNC Server)
2. Collects Compatible UIs Info.
3. Delivers URIs List (XML Description)
7. Message Info. (Key Event Info.)
VNC Protocol(RFB)
8. Sends & Receives Events
Remote Control Point
4. Browsing
5. Sets the UI Session & Management
1. Requests the Service
6. Delivers Playback Environment Info.
Remote Client
UI Session Information
Connectable Protocol Information
VNC Client
VNC Viewer

Fig. 4. The system architecture for remote UI sharing service and its processing flow

This mechanism consists of the remote server and client. The remote server provides the remote accessible service, and client receives this service from the server and transmits the event messages to the server. The proposed remote controller is included in the home entertainment center as a client module. It is an integrated module of remote client, Control Point and a remote viewer in client side.

Figure 4 shows the architecture of the remote controller its processing flow with remote server.

The remote server is a virtual device which supports an accessible UI in remote, and it has lists of accessible (or executable) applications and status of running application. The remote server can be embedded in an AV device that UPnP AV MediaServer is running. Or, it can be embedded in an independent AV device separately. It has a similar function with UPnP AV MediaServer. That means that the server delivers the URI list of executable application to the client. At this time, the URI list is XML schema format which consists of metadata. It classifies the executable applications and UI, and contains the technical information of application and each UI in protocol side. The server establishes connection with the remote client via the connection policy. If the remote client requests the service, the server must permit the remote accessible service.

The main purpose of remote Control Point is that discovers the remote server and shows the UI list on the client and sets the connection between the server and client. After setting the connection between two devices, the session is preserved by the remote sharing protocol. Various input and event messages for remote UI are processed in client module. To browse server뭘 UI, a remote viewer is needed. We used the VNC viewer. Remote client in the remote controller is an important module which shows the user interface and accessible applications or executes it. It sets and manages the UI session which is connected with the remote server by Control Point. For this process, client module delivers the important information (i.e., display screen size, keyboard type, etc.) to the Control Point. The remote client shares full screen of remote server.

The remote controller뭘 processing step is as follows. The remote controller detects all UPnP remote devices that are running on the network, and it matches an UI description (XML form) with remote client capabilities and establishes sessions. The remote controller collects the information of remote accessible servers, lists of accessible UIs, remote accessible clients and protocol information. UI session is begun with reference to the collected UIs and protocol properties. The remote controller can execute applications in the shared UI after sending a key event as well as can send a simple message to the remote device. Also, it can receive an important key input from other network devices. This means that the remote controller can act as a proxy for network device to run applications in remote. Using the remote controller, the user is not dependent on a specific device and can frequently cross various devices. The remote controller classifies executable applications via the URI which can be related with other devices in home network. If connection is completed, users can access all applications of server device from the remote devices without additional application.

The UI sharing mechanism aims to thin client. There are several well-known thin client technologies such as Remote Desktop Protocol (RDP) [14] of Microsoft, XRT(Extended Device Remote Transfer Protocol) [15] of Intel, and Virtual Network Computing (VNC) of RealVNC. VNC [16] is an open architecture which supports a very simple structure than other technologies, and it is very useful in network implementation for the UI sharing service because it has an independent characteristic to platform such as Windows, Linux, Mac OS, and etc. It consists of two components. There are a server and a viewer. A server runs on the remote accessible computer, and a viewer runs on the local computer. In this paper, for thin client protocol, we used the VNC based on RFB(Remote Frame Buffer) protocol which is very simple and easy to be implemented on consumer electronics devices. That means that it improves usability of AV devices through loading the controller in AV device to construct ubiquitous environment in home. In this paper, we run a VNC server with the AV MediaServer, and run a VNC viewer with the remote controller in the home entertainment center. We created a special UI that is separated with UI of the home entertainment center. That means that users can execute the independent UI sharing service as well as the essential home entertainment service at the same time. We used FIFO mechanism for the fundamental message communication between components, and we created the message queue for the message delivery between components.

4 Implmentation Results

We organized two implementation results. One is a stable operation result of the AV media server and the integrated controller, another is a stable operation result of the remote server and the remote controller in the home entertainment center.

To provide transparency of UPnP AV devices, we have implemented a home entertainment center. We used the UPnP library based on Fedora Core 5 of Linux platform. UPnP AV architecture module and remote UI module are designed by C/C++ language, communication module among UPnP devices is implemented by the UPnP library. For effective interoperability among UPnP modules, we made a share library and it is used in GUI module. The GUI module is written in Python language including several packages such as numeric, twisted, imaging, game, mmpython, etc., and it was implemented by various libraries such as SDL, smpeg, libdvdcss, libdvdread, etc. To support user-friendly interface, we adopted various graphic themes in GUI module. And, we used two UPnP AV MediaServers which are based on MS-Windows XP. One is included in Intel UPnP tool [17], another is the UPnP AV MediaServer [18]. For AV content playback, we used MPlayer 1.0pre8 which supports high quality playback service.

Figure 5 shows a screenshot of the media streaming service for remote AV contents.

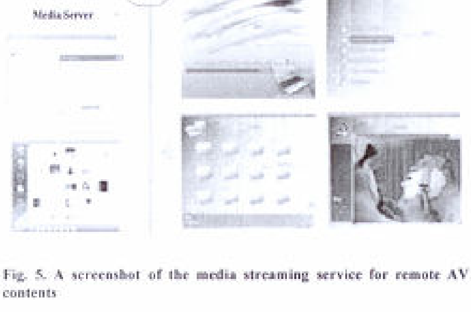

Fig. 5. A screenshot of the media streaming service for remote AV contents

A user has to execute an AV media server at first, and then, a user can register a large number of AV contents which are different kinds of media types. In this paper, an AV media server runs on MS-Windows based PC and the home entertainment center runs on embedded Linux-based set-top box. If a user chooses the Share Remote AV menu of main screen, servers is shown with the icon form after the controller brings lists of the UPnP server device. GUI module manages UDN of UPnP devices and lists for device name. The home entertainment center shows visual and dynamic screen compositions according to media types and user뀔choice menu.

Figure 6 shows a screenshot of the UI sharing service for remote device.

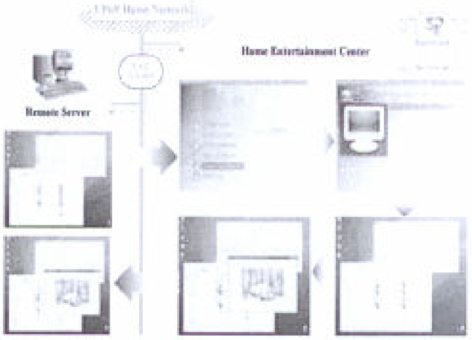

Fig. 6. A screenshot of the UI sharing service for remote device

To access a remote server뀔 UI, the home entertainment center has to receive a list of the remote server by choosing the Share Remote UI menu in main screen. If connection between a remote server and the home entertainment center is successfully completed, server뀔 UI is shared through the home entertainment center뀔 main screen, and a user can use applications in the server. The home entertainment center was designed by method which shares full screen of server in client.

The remote server runs on MS-Windows based PC in figure 6. First of all, a user has to execute the remote server, and then, the home entertainment center discovers a remote server and shows server뀔 UI. If a user chooses a remote server, connection between two devices is set. If connection is successfully completed, a user can confirm a UI of the server and can control its application in main screen of the home entertainment center.

Table 1 shows each RSS (Resident Set Size: Real memory size of the process in Kilobytes) when three systems are running on each system. It is two experimental results.

We have run the Home Entertainment Center (HEC) on embedded Linux which is based on Intel-Pentium 4 processor and supports 2.8GHz CPU and 512KB Cache Memory. In case of A, controller뀔RSS is a memory size of when a media player plays a video content (.MP4 format) which can be run during an hour, and remote controller (including remote viewer)뀔RSS is a memory size of when an application (MS-Word) of the remote device is run by the remote viewer. In case of B, controller뀔 RSS is a memory size of when a media player plays an audio content (.mp3 format) which can be run during four minutes, and remote controller (including remote viewer)뀔 RSS is a memory size of when an application (an image viewer) of the remote device is run by the remote viewer. The home entertainment center does not need many resources than related researches, and have high performance than them.

Table 1 Each RSS(KiB) when three systems are running on each system

System / Process Name		A			B		
		Intel Tool	Real VNC	Proposed HEC	Intel Tool	Real VNC	Proposed HEC
Graphic User Interface		-	-	30,276	-	-	28,176
Controller	Control Point	22,244	-	23,356	21,193	-	24,301
	Media-Renderer	44,640	-	12,664	34,564	-	8,635
Remote Controller	Control Point	-	-	23,672	-	-	28,652
	Client	-	-	632	-	-	1,009
Remote Viewer		-	15,458	12,316	-	23,020	19,510

5 Conclusions

In this paper, we proposed an interoperability mechanism among UPnP AV devices for ubiquitous home. It provides the remote contents sharing service as a main service. To support the media streaming service for remote AV contents, we composed the integrated controller that can discover remote AV devices and can browse and play remote AV contents. To support the UI sharing service for remote device, we composed the remote controller that can recognize remote device and can show the screen of the remote device. All components are systematically operated, and are independently managed. Users can play AV contents in the local or remote AV device at any time, and can control the applications of the remote AV device. In addition, an UPnP-based home AV network can be easily constructed without additional installation of other devices or components.

In the future, we will apply the media switching technique and the adaptation technique among MediaRenderers. And, we will add several entertainment services to the home entertainment center to extend the range of the entertainment service.

6 Acknowledgment

This research was supported by the MIC of Korea, under the ITRC support program supervised by the IITA(IITA-2006-C1090-0603-0026).

7 References

[1] E. F. Steinfeld, 및ome Entertainment Automation Using UPnP AV Architecture and Technology,? http://www.go-embedded.com/UPnP%20White%20Paper.pdf

[2] DLNA, 및igital Living Network Alliances및Overview and Vision White Paper,? 2006, http://www.dlna.org/en/industry/about/dlna_white_paper_2006.pdf

[3] Meedio, http://meedio.com

[4] Showshifter, http://www.showshifter.de

[5] VideoLanClient, http://www.videolan.org

[6] MythTV, http://www.mythtv.org

[7] B. Oesterdieckhoff, C. Loeser, I. Jahnich, R. Glaschick, "Integrative approach of Web services and universal plug and play within an AV scenario, 3rd IEEE International Conference on Industrial Informatics, p123-128, Aug. 2005

[8] G. Jun, 및ome media center and media clients for multi-room audio and video applications,? 2005 Consumer Communications and Networking Conference, p257-260, Jan. 2005

[9] UPnP Forum, 밝PnP AV Architecture V0.83,? June 2002, http://www.upnp.org/standardizeddcps/documents/UPnPAvArchtiecture0.83.pdf

[10] M. Wischy, 밝PnPTM Remote UI,? UPnP Forum, http://www.upnp.org/events/documents/1300_Wischy_RemoteUIUpdate.pps

[11] UPnP Forum, 밨emoteUIServerDevice V1.0,? Sept. 2004, http://www.upnp.org/standardizeddcps/documents/RemoteUIServerDevice1.0.pdf

[12] S. R. Bard, 밊ntel enables sharing of digital content on the home local area network,? Oct. 2004, http://www.hometoys.com/htinews/oct04/articles/intel/bard.htm

[13] UPnP Forum, 밨emoteUIClientDevice V1.0,? Sept. 2004, http://www.upnp.org/standardizeddcps/documents/RemoteUIClientDevice1.0.pdf

[14] RDP(Remote Desktop Protocol), Microsoft, http://msdn2.microsoft.com/en-us/library/aa383015.aspx

[15] XRT(Extended Device Remote Transfer Protocol), Intel, http://cache-www.intel.com/cd/00/00/23/41/234192_234192.pdf

[16] T. Richardson, 밫he RFB Protocol V3.8,? Technical Document, RealVNC Ltd., http://www.realvnc.com/docs/rfbproto.pdf

[17] Intel, http://www.intel.com

[18] H. Lee and S. Kim, 밫he u-MUSE System: An Integrated UPnP AV Home Entertainment System supporting RUI Service and Device Mobility,? Proceedings of the 2006 International Conference on Hybrid Information Technology, vol. 2, p712-717, Nov. 2006