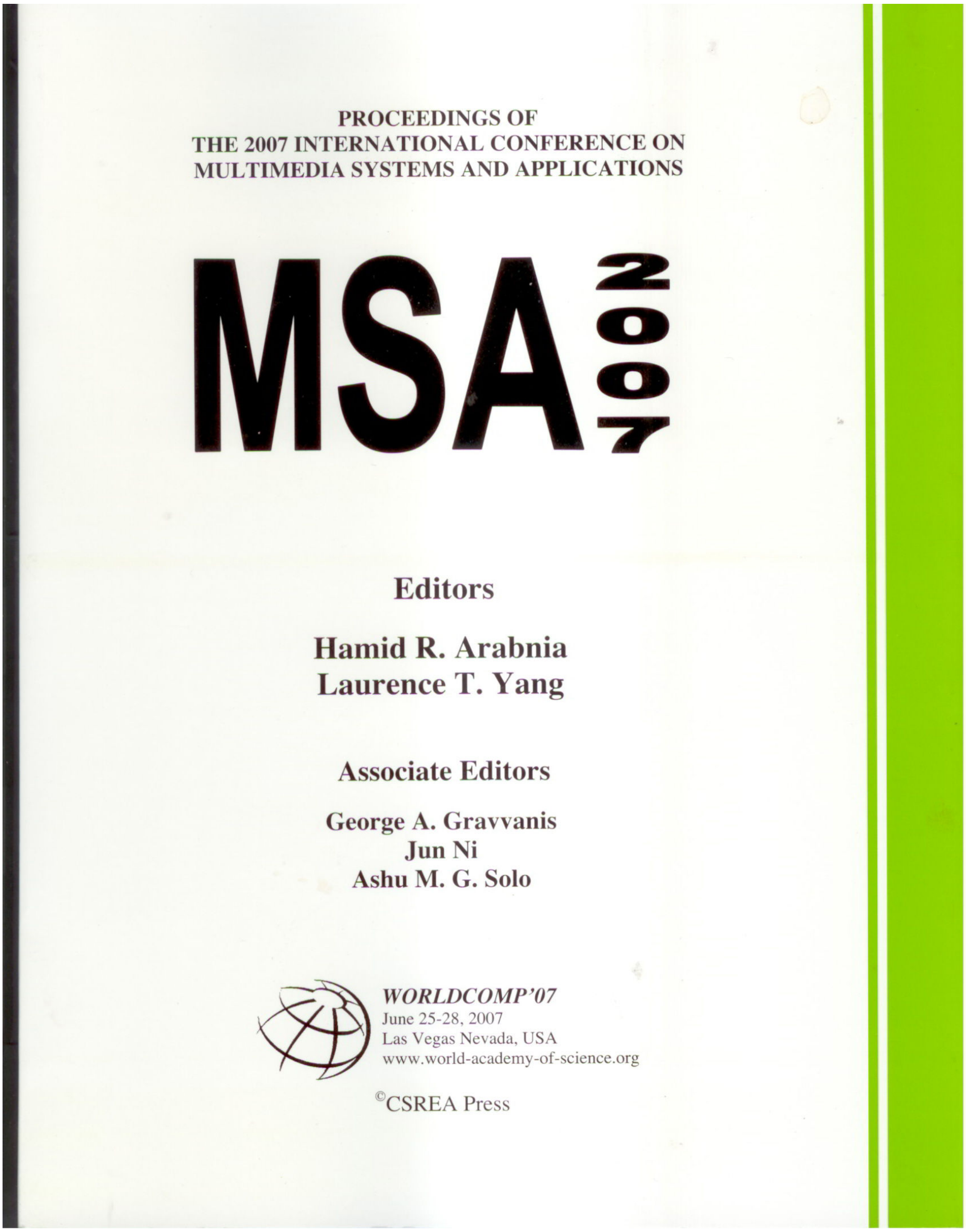

PROCEEDINGS OF
THE 2007 INTERNATIONAL CONFERENCE ON
MULTIMEDIA SYSTEMS AND APPLICATIONS

MSA 2007

Editors

Hamid R. Arabnia
Laurence T. Yang

Associate Editors

George A. Gravvanis
Jun Ni
Ashu M. G. Solo

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

Session: STREAMING

The Remote User Interaction Mechanism for Contents Sharing of Home Entertainment System

Hyunju Lee
Department 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 automatically integrates and manages various UPnP AV devices that are connected to the home network. A large number of existing home entertainment systems target the support of the multimedia service that is based on home server as a media content provider. In this paper, to build the ubiquitous media home and to guarantee transparency of all UPnP AV devices, we propose the remote user interaction mechanism for contents sharing of home entertainment system which is based on UPnP AV architecture. It provides the remote AV media sharing and the remote UI sharing among all AV devices regardless of AV device뭩 location and media data뭩 playback types. The home entertainment system sufficiently can perform role of control center for AV electronic devices via the remote user interaction mechanism.*

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

1 Introduction

The home network has multiple heterogeneous devices with different capabilities, and they communicate with other home network devices using the home network middleware such as UPnP. The UPnP (Universal Plug and Play) [1-2] forum developed UPnP AV (Audio/Visual) architecture as the standard of multimedia system in home network to provide the media distribution services. The UPnP AV architecture provides the communication mechanism for AV devices to share AV media contents and interoperability among different kinds of multimedia appliances.

The home entertainment service뭩main purpose is that provides the dynamic multimedia service to users regardless of media content뭩 location and playback types. Most of the existing home entertainment systems provide very stable technologies, and have high quality performance. But, servers? role in these systems is very limited, which stores a large number of media contents and provides the media entertainment service to other AV devices as a content provider. Also, roles of each device in these systems are limited according to their main use purpose and properties, i.e., a device for contents storage, a device for content rendering, a device for various controls, etc. Moreover, most of the servers do not permit access or control from the remote client device in system side. And, components in client device provide limited performances and their roles are not clearly defined.

In this paper, we propose a remote user interaction mechanism for contents sharing of home entertainment system which is based on UPnP AV architecture. The remote user interaction mechanism provides interoperability of all UPnP AV devices including the server, which means that the home entertainment system guarantees transparency of all AV devices. Finally, it builds the ubiquitous media home. The contents sharing means that the home entertainment system can play various AV contents in local device, and automatically discovers remote AV devices (and browses and plays remote AV contents), and automatically recognizes remote server devices and shows its screen and execute applications in remote AV device. To provide remote contents sharing service, we composed two controllers in the home entertainment center as core technologies.

This paper is organized as follows. Section II, we describe the several related works. In section III, we describe the remote user interaction mechanism 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 aims to the high-quality multimedia service support to users. There are several well-known home entertainment technologies such as Cidero[3], VideoLan Client[4], Linpus Media Center[5], MythTV[6], etc.

Cidero is based on UPnP Control Point which supports various O/S platforms. It can browse and control the media contents by the media controller. This system provides multithreads and interaction among AV devices to control AV devices on the wire/wireless network, and it allows non-UPnP devices via the UPnP Bridge. But, in case of device discovery and media playback, the controller has random long delay time than the VideoLan Client and Linpus because it uses the play queue for the renderer. And, users have to confirm its media type when plays an AV content because it supports multi-renderer and each renderer supports different kinds of AV contents. Moreover, it is very difficult to improve system펼 generalization and extension because this system is just used to several commercialized products.

VideoLan Client is a media player which is based on an open source project. It supports various O/S platforms like the Cidero, and supports a large number of multimedia formats. This system can be compatible with Geexbox uShare, MediaTomb, Darwin server, VideoLAN server, etc. But, this project does not support the MediaRenderer which is a core component of UPnP AV architecture. Users have to directly connect with a media player in local when plays a media content. That means that the media service performance of system is influenced by performance of a connected media player because content control is not allowed by the MediaRenderer. Also, this system does not support the multimedia service for other devices except for PC.

Linpus Media Center is a Linux-based media platform, and it provides various embedded systems. It supports fast and easy installation, and also supports various storages and easy-to-use navigation. It contains a remote controller to control the media contents. But, Linpus does not support other media formats except for MPEG format. It does not support the MediaRenderer, and it also does not support the multimedia service to other devices except for PC like the VideoLan Client. In spite of its high performances, it has need of too many requirements within the framework of hardware/software.

MythTV is a homebrew project that is based on Linux desk-top. 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 a UPnP server among UPnP AV architecture. But, this system펼size is very heavy 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. Also, 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 considerable stable, and these are very useful to applications of related works. In spite of excellent performance of AV content playback and control, server in these systems does not permit access and control from the remote AV device as an AV contents provider. 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 of home entertainment center.

3 Remote User Interaction 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 performance and property. 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.

To provide the remote user interaction for contents sharing service, the AV MediaServer and the remote server run as the contents provider. 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. The remote contents sharing mechanism contains two mechanisms. There are the remote AV media sharing and the remote UI sharing mechanism. 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 system. 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.

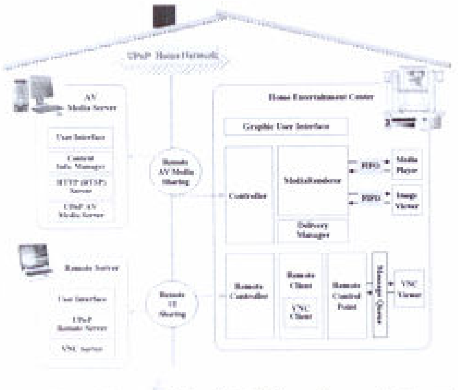

Fig. 1. The architecture of the UPnP AV-based home entertainment system

3.1 The Remote AV Media Sharing

AV contents need a MediaRenderer to play itself because of its own 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 user펼 point of view because number of necessary devices can be reduced and two components can be simultaneously managed by using the same device. UPnP AV architecture consists of MediaServer, Control Point and MediaRenderer [7-9]. 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 a processing flow between the AV MediaServer and Controller.

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) 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. Basically, UPnP AV architecture supports the AV media sharing mechanism between the server and client.

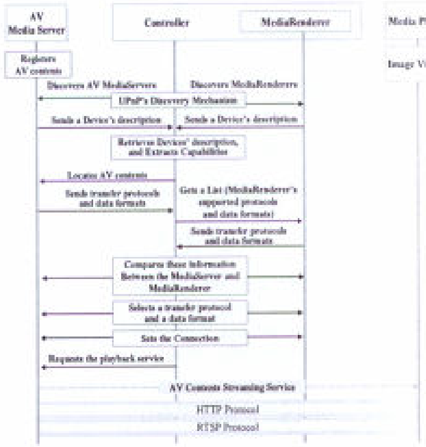
AV
Media Server
Controller
MediaRenderer
Media Player
Image Viewer
Registers
AV contents
Discovers AV MediaServers
Discovers MediaRenderers
UPnP's Discovery Mechanism
Sends a Device's description
Sends a Device's description
Retrieves Devices' description,
and Extracts Capabilities
Locates AV contents
Sends transfer protocols
and data formats
Gets a List (MediaRenderer's
supported protocols
and data formats)
Sends transfer protocols
and data formats
Compares these information
Between the MediaServer and
MediaRenderer
Selects a transfer protocol
and a data format
Sets the Connection
Requests the playback service
AV Contents Streaming Service
HTTP Protocol
RTSP Protocol

Fig. 2. The processing flow between the AV MediaServer and Controller

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 information of media content which a user chooses by the Control Point. 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 Point뭩UI. 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 device뭩description 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.

3.2 Graphic User Interface

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. One of its important roles is that it integrates Control Point and MediaRenderer. The GUI module provides a 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 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.

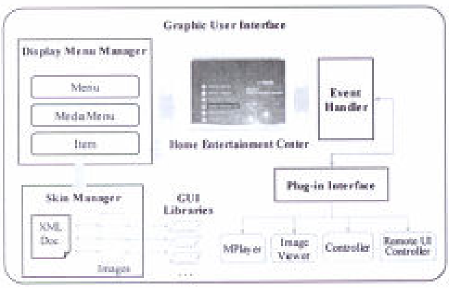

Fig. 3. The architecture of the GUI module

Display menu manager composes a menu that is related with lists of the remote device and AV contents. It creates a directory after classification into each item (i.e., Video, Audio, Images and UPnP AV/remote device) via the delivered information from each server. There are Menu, MediaMenu and Item as its sub modules. Menu composes an initial main screen 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 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 classify 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 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 user뮐 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 AV 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 content뮐 playback is terminated. To support user-friendly interface, we adopted various graphic themes in GUI module.

3.3 The Remote UI Sharing

Remote UI(User Interface)[10-13] is an interface mechanism which allows access from the local device to applications of remote device. 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.

The remote UI sharing mechanism consists of the remote device(server) and client. The remote device provides the remote accessible service, and client receives this service from the server and transmits event messages to the server. The proposed remote controller is included in the client. 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 device.

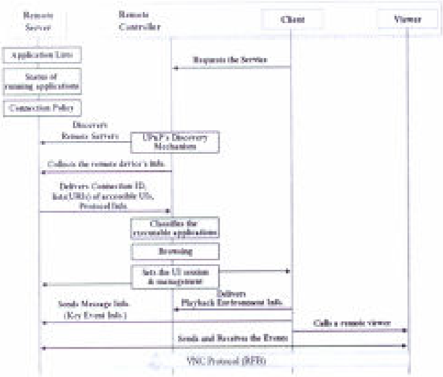
Remote Server
Remote Controller
Client
Viewer
Application Lists
Status of running applications
Connection Policy
Requests the Service
Discovers Remote Servers
UPnP's Discovery Mechanism
Collects the remote device's Info.
Delivers Connection ID, lists URI(s) of accessible UIs, Protocol Info.
Classifies the executable applications
Browsing
Sets the UI session & management
Delivers Playback Environment Info.
Sends Message Info. (Key Event Info.)
Calls a remote viewer
Sends and Receives the Events
VNC Protocol (RFB)

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

The remote device(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 device 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 executable application and UI, and contains 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 device 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 server뭩 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 device by Control Point. For this process, client module delivers important information (i.e., display screen size, keyboard type, etc.) to the Control Point. The remote client shares full screen of remote device.

The remote controller뭩 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 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 remote UI sharing and control service aims to thin client. There are several well-known thin client technologies such as Remote Desktop Protocol (RDP) of Microsoft, XRT(Extended Device Remote Transfer Protocol) of Intel, and Virtual Network Computing (VNC) of RealVNC. VNC[14] is an open architecture which supports a very simple structure than other technologies, and it is very useful in network implementation for the remote 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 remote UI sharing service as well as the essential home entertainment service at the same time.

4 Implmentation Results

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

For implementation of the home entertainment center, we used the UPnP library based on Fedora Core 5 of Linux platform. UPnP AV architecture module and remote user interaction 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 the GUI module. The GUI module is written in Python language including several packages and it was implemented by various libraries. 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[15], another is the UPnP AV MediaServer[16]. For AV content playback, we used MPlayer 1.0pre8 which supports high quality playback service.

Figure 5 shows a screenshot of the remote AV media sharing service.

To share various remote AV media 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 figure 5, two AV media servers run on PC which is based on MS-Windows, and the home entertainment center runs on set-top box which is based on embedded Linux. 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 UDNs of UPnP devices and lists for device name. The home entertainment center shows visual and dynamic screen compositions according to media types and user뭩menu choice.

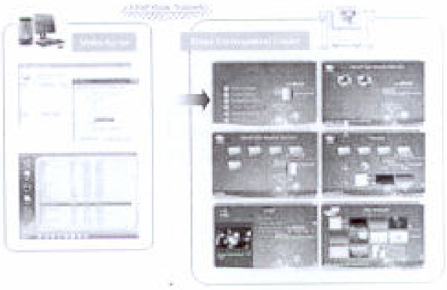

Fig. 5. A screenshot of the remote AV media sharing

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

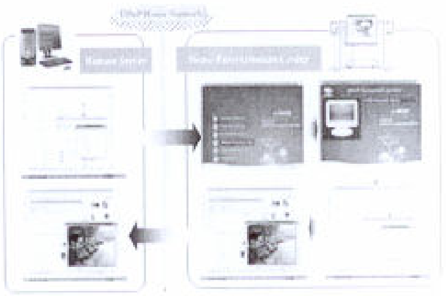

Fig. 6. A screenshot of the remote UI sharing service

To access the UI of a remote server device, the home entertainment center has to receive a list of the remote device by choosing the Share Remote UI menu in main screen of system. If connection between a remote device and the home entertainment center is successfully completed, remote server device뷥 UI is shared through the home entertainment center뷥 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.

In figure 6, the remote server device runs on PC which is based on MS-Windows. First of all, a user has to execute the remote server, and then, the home entertainment center discovers a remote server and shows server device뷥 UI. Whenever 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. In figue 6, a local user executed an image viewer of remote server device.

Table 1 shows support items of the home entertainment center as its performance. The remote user interaction for contents sharing was applied to the home entertainment center.

Table 1. Support items of the home entertainment center

Requirement	Support Items
Base Platform	Fedora Core 5 of Embedded Linux
Components	Control Point, MediaRenderer, Remote Control Point, Remote Client
Remote User Interaction	Fully Supported (Remote AV Media Sharing, Remote UI Sharing)
Network	Supported (Wire/wireless)
Mobile Devices	Restrictively Supported (Note-Book, PDA)
Video Format	MPEG-1/2/4, avi, wmv, Divx, asf, ts, etc.
Audio Format	MPEG-1(mp3), wma, ogg, ra, etc.
Independent management of each module	Fully Supported
Integration Operation of Each Component	Supported (The Controller and the Remote Controller)
Interoperability and transparency of AV Devices	Supported (by the remote controller)

5 Conclusions

In this paper, we proposed the remote user interaction mechanism for contents sharing of home entertainment system which is based on UPnP AV architecture. The home entertainment system stably runs on UPnP network. The remote user interaction mechanism provides the remote contents sharing service as well as the fundamental multimedia entertainment service. To support the remote AV media sharing service, we composed the integrated controller that can discover remote AV devices and can browse and play the AV contents. To support the remote UI sharing service, we composed the remote controller that can recognize remote devices and can show the UI screen of a specific device. All components of two controllers are systematically operated, and are independently managed. Finally, users can play AV contents in local AV device or remote AV device at any time, and can use the remote AV device including the server, and can control applications of the remote AV device. Also, an UPnP-based home AV network can be easily constructed without additional installation of other devices or components via the home entertainment center.

In the future, in case multiple MediaRenderers run in UPnP network, we will apply the media switching technique among MediaRenderers and we will improve user interaction processing technique of the remote media content according to playback environment of various renderers. And, to extend the entertainment service, we will add several entertainment module to the home entertainment center such as games, internet TV, and etc.

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, 밐ome Entertainment Automation Using UPnP AV Architecture and Technology,? http://www.go-embedded.com/UPnP%20White%20Paper.pdf

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

[3] Cidero, UPnP Media Controller, http://www.cidero.com

[4] Linpus, http://www.linpus.com.tw

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

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

[7] B. Oesterdieckhoff, C. Loeser, I. Jahnich, R. Glaschick, 밒ntegrative 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, 밐ome 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, 밢PnP AV Architecture V0.83,? June 2002, http://www.upnp.org/standardizeddcps/documents/UPnPAvArchtiecture0.83.pdf

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

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

[16] 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