On Controlling Digital TV Set-Top-Box by Mobile Devices via IP Network

Chi-Chia Lin and Ming-Syan Chen
Department of Electrical Engineering,
National Taiwan University
nuspran@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw

Abstract

Digital TV already rises. Unlike the traditional TV system, the digital TV architecture allows users to execute application programs in their set-top-boxes. One of the benefits is that users can communicate interactively with others by executing certain applications via some channels, e.g., The Internet. However, the only control device in recent years is the remote controller, which is limited to the transmission range. In this paper, we devise a system that allows mobile appliances, e.g., mobile phones and laptops, to control our set-top-boxes. Due to the issue of compatibilities, we adopt MHP (Multimedia Home Platform) as the middleware. The proposed system model is also compatible with the EPG (Electronic Program Guide) architecture. Two services are provided: (1) Viewing DTV contents online; (2) Remotely commanding the set-top-box to record a DTV service. Assuming that in the future when every house will be equipped with a "Personal Multimedia Server", this server will even process complex tasks such as transcoding and broadcasting.We believe that our system model is very feasible for future DTV systems and is applicable to advanced DTV applications.

Key Words: Digital Home, MHP, EPG

1 Introduction

Recently (2005) the digital TV architecture (DTV) has created many issues and application opportunities for communication technology. In comparison to the traditional analogue TV system, the DTV system provides not only better QoS of audio / video services, but also a fine environment for developing applications. Combining DTV network and IP network, we are able to develop interactive applications for users to interact with their digital content providers and even other DTV subscribers.

"Digital Home" [2] stands for a home network, which transmits digital data. It proposes solutions to manage audio and video equipments, home automation, and home

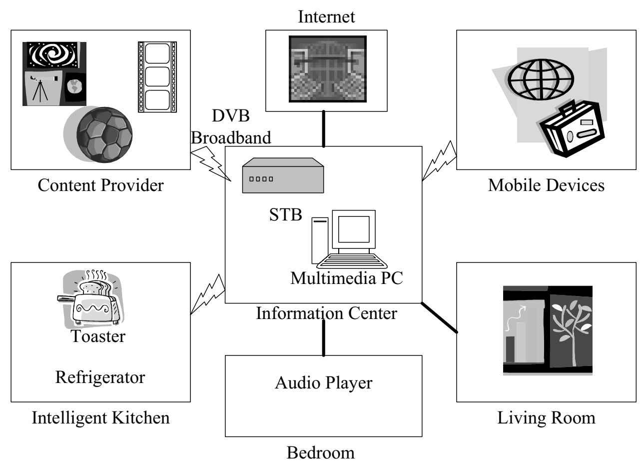

Figure 1. Digital Home

cinema using web gateways, wireless networks and common browser. Figure 1 illustrates the concept of digital home. In the future we believe that the user's living room will become personal entertainment center since most display and control devices are located in the user's living room. The DTV system is already a very mature system. It is based on many existing technologies such as MPEG-2 transport stream and boradcasting technologies on terristrials and satellites. Some DTV systems, e.g., DVB [3], also define their own middlewares to provide a fundamental environment for applications. MHP (Multimedia Home Platform) [4] is a middleware which is developed with the DVB project in Europe. MHP runs on the Java virtual machine, developers do not need to worry about the problems such as platform compatibility. The HAVi technology [6], which is partly adopted by MHP, also provides great functionalities to realize the digital home concept [15].

Interactivity is the main feature of MHP, where several communication mechanisms for TV enironment are also developed [18] [22]. Developing GUI for DTV applications also provides research opportunities [16] [20] [21].

Up to 2005, MHP is widely adopted in Europe, including Finland, Sweden, Denmark, Norway, Germany, and Spanish. Although MHP is not the only middleware for interactive services in the DTV, it is, at present, with the most potential as standard in Europe.

Since MHP defines Internet profile and provides many APIs. Interactive applications which access the Internet can be implemented much more easily. In the DVB standard, it also defines how to transmit DVB signals via IP network since the the internet bandwidth grows rapidly. This tells us that DVB network and IP network are integrating each other [19].

Mobile devices such as PDAs, laptops, PocketPCs have become more and more popular. The ability of accessing the IP network has become the basic requirement of these mobile devices. For example, wireless adapters are built in laptops and mobile phones support GPRS or even 3G nowadays. Although the DTV system provides us so many services, currently the only way to control the set-top-box is the remote controller which has limited transmission range. Users carry mobile devices with them when they go out instead of their remote controllers. If these mobile devices could also control the set-top-box, users would be able to manipulate their set-top-boxes even if they are not at home. Since DTV network and IP network are integrating, it is possible for mobile devices to communicate with DTV network via IP network. For example, we are able to view EPG (Electronic Program Guide) [5] data on mobile phones, and send recording request to our PVR (personal video recorder) at home.

We have implemented a system to demonstrate how convenient it is to control our set-top-boxes at home by our mobile devices. Two services are provided. One is to view DTV contents online; the other is to send recording requests to the set-top-box. For future compatibility, our system model is also compatible with the EPG architecture.

EPG will be one of the most important features in the DTV system. Since there may be more and more DTV broadcasters in the future and DTV subscribers will have hundreds of services to choose. The traditional TV program guide which is printed on newspapers or posted on web sites will be infeasible for the DTV. DTV content providers may boradcast their EPGs via some channels, e.g., the Internet, to DTV subscribers. After receiving the EPG data, the EPG decoder and presenter in the user's set-top-box may present it in a fine EPG menu on the user's TV screen. Thus, for DTV system developers, they must take EPG into consideration.

We have not implemented any EPG applications yet since we would like to wait until an EPG format is adopted world-widely. Thus we have designed a system model which is friendly to the EPG architecture. Combining EPG, more services could be realized. For example, the EPG data could be pre-processed first and then transmitted to user's mobile devices. Users could decide what to do according to EPG, e.g., recording the baseball game tonight.

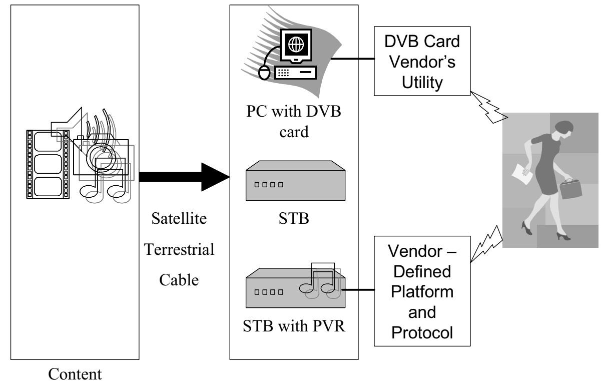

Figure 2. Model of Related Works

2 Related Works

So far there have been many DTV products for DVB which allows remote control by mobile devices. In recent years, two kinds of DTV products which receive DVB signals are popular: (1) A DVB set-top-box; (2) A PC with a DVB card. Vendors of these equipments usually provide their own applications and utilities which allow users to control their DVB equipments by mobile devices via some channel. Figure 2 shows the model of these related works.

A PC with a DVB card makes it very easy to allow remote control by mobile devices, since we do not need to worry about the computing resource and the ability to access the Internet. Most related system architecture works as what follows. A server runs in a PC which controls the DVB card and also processes user's requests. Besides, there are also set-top-boxes with PVRs which allow remote control, e.g., recording, by mobile devices. However, the platform and the communication protocols are both defined by vendors. There is no middleware presented in such systems either.

Perhaps the most mature related work is the TiVo [13]. The TiVo is essentially a set-top-box with PVR. The underlying hardware is ARM, and the operating system is Linux. It also has its own middleware for developers and it adopts java as the programming language. The TiVo system provides many convenient services. "Online Scheduling!" is the one which relates to our work. The TiVo.com has a central web site for "Online Scheduling!" users to login and edit their recording schedules by any devices with browsers, e.g., a laptop with Microsoft Internet Ex-

Proceedings of the Seventh IEEE International Symposium on Multimedia (ISM'05)
0-7695-2489-3/05 $20.00 © 2005 IEEE

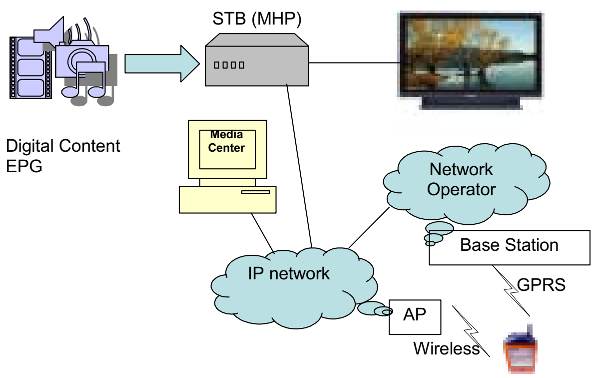

Figure 3. Remote DTV Controller System Model

plorer. After completing the recording request, the "Online Scheduling!" server will command the TiVo located in the user's home to perform the recording request. In comparison to the proposed system, we have three advantages: (1) Our system is constructed on MHP, which is a widely adopted middleware, but TiVo's middleware is not MHP; (2) Our system supports mobile phones which have no browsers; (3) Our system allows users to view DTV contents online. Note that we need an extra PC for the service of viewing DTV contents online since the transcoding process requires great computing power and storage.

3 System Architecture

The remote DTV controller system follows the Server / Client model. The server receives digital TV signals and user requests. It processes the signals and also responds to clients. In our system model, we assume that there will be a "Media Center" which has general computing power such as a PC. Figure 3 shows our system model.

Since the set-top-box is an embedded system, it has much limited computing power and storage in comparison to a desktop PC. Although MHP defines an internet profile which allows the set-top-box to connect to IP network, it can only sends and receives simple messages without complex processing. Thus, for complex tasks, we need another machine with powerful computing power and sufficient storage. In our system model, the "Media Center" plays such role. It is essentially a desktop PC. The media center processes the tasks such as audio / video transcoding, broadcasting, recording, data mining [17], searching and so on. These tasks all require great computing power and storage.

Out system model is compatible with EPG architecture. The EPG decoder and presentation engine can be implemented on MHP which runs in the set-top-box. The media center also has a copy of EPG for complex processing.

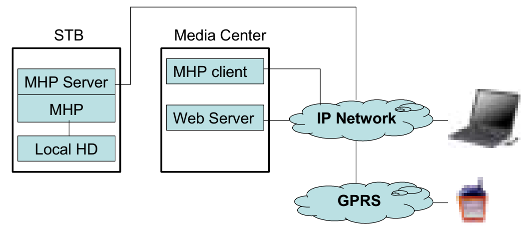

Figure 4. The Architecture of the Recording Sub-system

Our system provides two services: one is to view digital content online; the other is to record a program. Both services are archived by two sub-systems, respectively. Note that if the mobile device is a mobile phone, it cannot perform the viewing service due to limited display capability and low bandwidth.

We will discuss both sub-systems in the rest of this section.

3.1 Remote Recording Sub-System

Figure 4 shows the architecture of recording system.

3.1.1 MHP Server

The MHP server runs on the MHP. The server processes four tasks: (1) Listening to requests; (2) Tuning to a transport stream; (3) Playing a service; (4) Recording a service.

3.1.2 MHP Client for Mobile Phones

This program transmits commands entered by users directly to MHP server via GPRS. Note that this "MHP client" program never contacts the web server.

3.1.3 MHP Client for Mobile Devices with Browsers

Users who use laptops with browsers do not send commands to MHP server directly. The web server located in the Media Center plays the role as third party. Commands are received by the web server and the commands are transmitted to MHP server by a "MHP client" program embedded in web server. This scnerio is similar to the TiVo's system.

3.2 Live Broadcasting Sub-System

Figure 5 illustrates the architecture of live broadcasting system.

Proceedings of the Seventh IEEE International Symposium on Multimedia (ISM'05)
0-7695-2489-3/05 $20.00 © 2005 IEEE

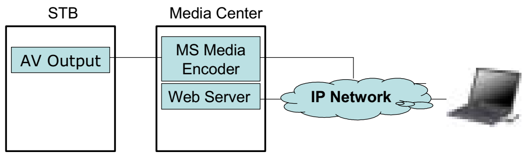

Figure 5. The Architecture of the Live Broadcasting Sub-system

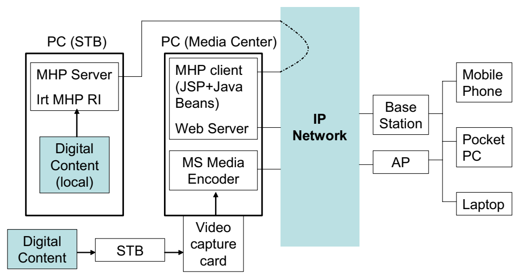

Figure 6. System Implementation

3.2.1 Transcoding and Boradcasting Server: Microsoft Media Encoder

Microsoft Media Encoder processes two tasks: one is to transcode DTV contents to video streaming data; the other is to broadcast it. The users may have to use Microsoft Media Player to receive and play the broadcasted streaming data.

4 Implementation

In this chapter, we present the implementation of our system, including equipments, the platform, programs, and all software we used.

Figure 6 shows the implementation of our system.

4.1 The MHP Server

4.1.1 Irt MHP Reference Implementation

We use a PC which installs Irt MHP reference implementation [7] to simulate a set-top-box. The reference implementation supports several DVB-T cards, but currently none of these cards can receive the DVB-T signal in Taiwan due to incompatibility of bandwidth. The bandwidth of DVB-T signal in Taiwan is 6MHz but these cards only support 7MHz or 8MHz. The digital content (transport stream) was received by another DVB-T card which supports 6MHz and was recorded to local hard disk. The reference implementation uses the recorded transport stream as its digital content source. We implemented the MHP server under java Xlet architecture [14], which is the default interface of MHP application. We also adopted the recorder agent API implemented by Irt as our recorder.

4.1.2 The MHP Server Implementation

We implemented a server program which runs on the MHP. This server program follows the java Xlet architecture which is the standard interface defined for MHP applications. The server program creates a ServerSocket which binds to a port when it is launched by the user. This ServerSocket waits for TCP connections. When a TCP connection is established, the server continuously receives the data transmitted by the client and performs predefined actions until the client disconnects.

The MHP server is an application running on the MHP. Since applications are carried in transport stream in the DVB architecture, users must tune to a transport stream first and then are able to execute the application. If this application also tunes to the original transport stream that carries the same application will result in killing the application itself is the AIT table of the transport stream does not signal the application to launch continuously. Thus, the MHP server does not come with any transport stream. It is launched from the set-top-box's local storage. The MHP server processes three tasks. (1) Tuning (2) Listening to the Internet for client requests (3) Recording a program. Since it is launched from set-top-box's local storage, this application does not belong to any transport stream. Tuning to an existing transport stream will never killing itself.

4.1.3 Command Types

We define several commands for clients and each command has its corresponding action. When the MHP server receives the command, it performs the predefined action. If the MHP server receives an undefined instruction, it simply ignores.

The data structure of command is StringBuffer. All possible commands are StringBuffer "-1", "0", "1", "2", "3", "4", "5", "6", "7", "8", and "9". All transmitted commands except "-1" will be followed by a StringBuffer "\n" as termination of a command.

4.1.4 Recording Agent

We include the Irt's recording API and use their functions to record a transport stream. This is a temporary solution since not long ago the new MHP standard for recording was published. We are able to implement the recording service on MHP according the new standard. Note that current recording service is also compatible with the MHP.

4.2 The MHP Clients

We implemented two MHP client programs: one is for mobile phones; the other is for mobile devices with browsers.

4.2.1 The MHP Client for Mobile Phones

These mobile phones must have the ability to execute J2ME [8] programs and support GPRS (General Packet Radio Service). Clients have to download the MHP client program somehow by a transfer line or email to install the program first. The MHP client program requires the ability of accessing the Internet, clients may have to request GPRS service before they execute the MHP client program.

The program is developed and tested by java J2ME wireless toolkit. Whole program follows java MIDP MIDLet architecture [12], which is the most common program interface for mobile phones.

4.2.2 The MHP Client for Mobile Devices with Browsers

We assume that the mobile devices are laptops or PocketPCs, which have own browsers such as the Microsoft Internet Explorer. We also assume that they have the Microsoft Media Player, or other media player which supports WMV streaming data.

The MHP client for mobile devices with browsers is implemented by java. It is embedded in the web page by JSP (Java Servlet Page) framework. The MHP client is launched when a user browses the web site. Buttons that a user clicks are forwarded to the MHP client program process. The MHP client program process connects / disconnects to MHP server, or transmits commands to MHP server. Messages of the MHP server are received by function Receive_server_msg(), this function puts MHP server messages in a variable server_msg, which is a private variable of type String. The web server reads the string and displays it on the web page.

4.3 The Web Server

We use Java SDK 1.4.2_07. The web server is constructed by Apache Tomcat 5.5 [1], and the web page is written by JSP. We use the JavaBEANS [9] to embed the MHP client program which is described in section 4.2.2 to the web page. A Microsoft Media Player is also embedded in the web page.

```
Line 1: <%@ page import = "MHP.MHPclient" %>
Line 2: <jsp:useBean id="client" scope="session" class="MHP.MHPclient" />
Line 3: <jsp:setProperty name="client" property="*" />
```

Figure 7. JSP JavaBeans Code to Embed the MHP Client Program to the Web Page

```
Line 1: <object CLASSID="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95"
        ID="MediaPlayer" width="320" height="320">
Line 2: <param name="ShowControls" value="true">
Line 3: <param name="FileName"
         value="<%=request.getParameter("channel")%>">
Line 4: <param name="ShowStatusBar" value="true">
Line 5: <param name="AutoStart" value="true">
Line 6: <param name="ClickToPlay" value="true">
Line 7: </object>
```

Figure 8. The HTML Code to Embed a Microsoft Media Player

4.3.1 To Embed a java program to web page

We import the class of the MHP client by the JSP codes in figure 7.

Note that the "session" in line 2 means that different connections to the web server are treated as individual clients.

4.3.2 To Embed a Microsoft Media Player to web page

We embed a Microsoft Media Player by the HTML codes in figure 8.

At line 3 the “request” item is to get the value of variable “channel”. The variable “channel” is a local variable which defined in a form. We create several buttons which assign different values to the variable “channel”.

Note that this player reads an asx file which defines a MMS (Microsoft Media Server protocol) URL. The player will try to connect to the URL and play the video stream. The codes in figure 9 is an example of what .asx file contains.

4.3.3 The Message Bar

We use the JSP codes in figure 10 to display the message received by MHP client on web page.

Note that client is an instance of the MHP client class, and the function getmsg() returns a variable of type String. This variable contains the message received from the MHP server.

```
<asx version= "3.0">
<TITLE>test</TITLE>
<ENTRY>
<REF HREF="mms://127.0.0.1:9189"/>
</ENTRY>
</asx>
```

Figure 9. The Content of the asx File

```
<%=client.getmsg()%>
```

Figure 10. The JSP Code to Display Messages from the MHP Server

4.4 Microsoft Media Encoder 9 series

In order to prodive the service of viewing DTV contents online, the Microsoft Media Encoder [11] is adopted as our transcoding and broadcasting server. Since the Microsoft Media Encoder supports only common video capture cards and sound cards instead of any DVB card, the DTV content is received by another set-top-box, which has audio and video outputs. We connect the audio and video outputs to a Creative Sound Blaster PCI card and a Kworld Video Capture Card, respectively. Thus the Microsoft Media Encoder is able to receive both DTV audio and video signals. The Microsoft Media Encoder transcodes the audio and video data to streaming data and broadcasts them to the Internet via a pre-selected port. Clients (Microsoft Media Player) connects to the port and receives the streaming data by a MMS URL defined in an asx file.

5 Experimental Results

We test our system by two kinds of mobile devices. One is a mobile phone, and the other is a laptop.

We first launch Irt MHP reference implementation to simulate a set-top-box. We register our MHP server class files in application.cfg, and then we are able to launch our MHP server by pressing the “Arbor MHP Server” button on the screen. After launching our MHP server, the server continuously listens to a port for requests from clients.

We also start up the tomcat at port 8080. Thus the MHP server and the web server are both ready.

5.1 Experimental results of mobile phones

The Mobile phone which we test is SonyEricsson K700i. We first launch our MHP client program. This program connects to the MHP server by a pre-defined IP address when it launches. When the connection is established, we are able to edit the text field and transmit the text to the MHP server.

Figure 11. The MHP Server Stops Recording a Service

We send four strings to MHP server. “1” commands MHP server to tune to a transport stream. And then “3” commands th MHP server to play a service (we have three options “1”, “2” and “3” at this stage). “4” commands it to record a service. Finally “5” commands it to stop recording. Figure 11 shows the screenshot when the MHP server received the command "stop recording".

Finally we launch the “Recording Playback” application written by Irt to play the recorded service.

5.2 Experimental results of laptops

The laptop which we test is IBM X21. The operating system is the Microsoft Windows XP Professional with Service Pack 2.

We first launch Microsoft Internet Explorer to connect to our web server. Figure 12 shows the user interface of our system.

We view the DTV content by pressing the “Play” button. After viewing the DTV contents, we press the buttons on the right-hand-side to transmit commands to the MHP server. We press “connect” to connect to the MHP server first, and then press “1” to command it to tune to a transport stream. We press “3” to command it to play a service and then press “4” to made it record the service. Finally we press “5” to command it to stop recording.

The following procedures: Launching Irt’s “Recording playback” application and playing the recorded service;

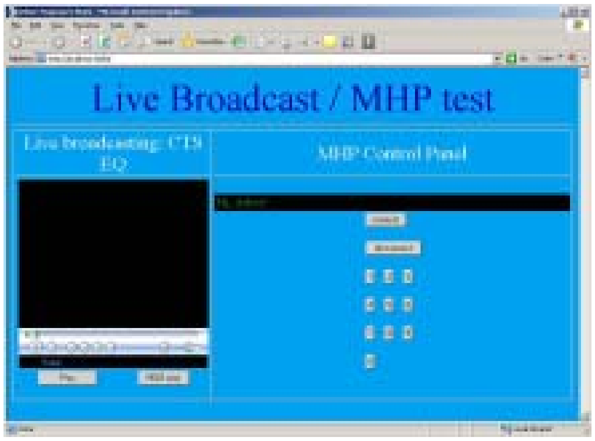

Figure 12. The User Interface of Our System

	Number of class loaded	Number of KB loaded	Permanent space capacity	Time for load / unload
Run Irt MHP RI	2320	2193.9	8192	0.76
Run MHP server	2420	2279.1	8448	0.9
Connection established	2421	2279.8	8448	0.9
Tune to TS	2486	2332.9	8704	0.7
Play a service	2575	2402.4	8960	0.64
Record	2649	2460	9216	0.64

Figure 13. The Memory Statistics of the JVM

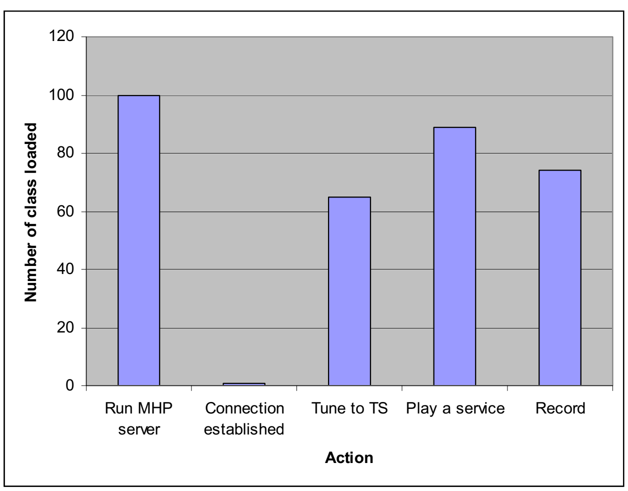

Figure 14. Number of Classes Loaded by Each Action

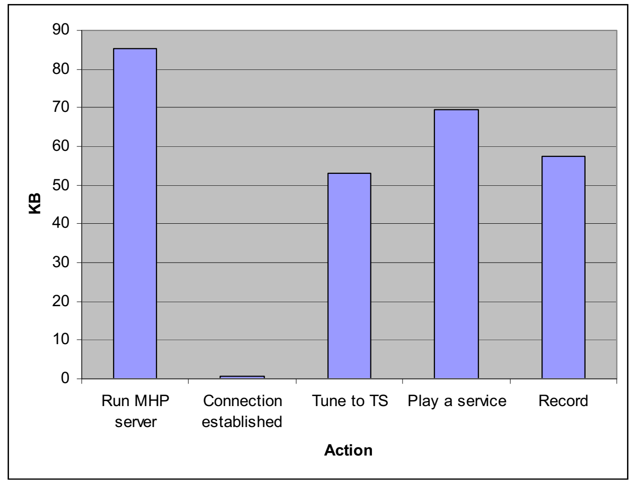

Figure 15. Number of KB Loaded by each Action

are the same as what we describe in section 5.1.

5.3 Memory consumption

In this section we describe the memory consumption of the MHP server. Since the set-top-box is an embedded system, the memory capacity is limited. Developers are not able to allow their applications to consume too much memory. We use the "jvmstat 3.0" [10] to monitor the memory capacity and consumption of the JVM. Table 13 shows the memory statistics of the JVM when we perform each action.

The "jvmstat 3.0" computes the number of classes loaded in each action and figure 14 shows the result. Figure 15 shows the number of Kbytes loaded in each action. We see that running the MHP server consumes the most Kbytes of memories since it loads the most classes. We do not know exactly which classes the JVM loads but we are sure that these classes are necessary for starting up the Irt MHP implementation.

We also see that except running the MHP server, no actions consume more than 70 Kbytes of memory. This shows that the proposed system is very applicable for embedded systems.

6 Conclusion

In this paper, we propose a system to provide two major services over DTV: (1) Viewing DTV contents online and (2) Recording DTV contents. The system supports most mobile devices including mobile phones and laptops.

Furthermore, it also provides a very friendly user interface for laptop users.

The recording service of the proposed system is constructed on MHP. The client program for mobile phones is written under the J2ME framework. The communication channel is IP network. Web server is constructed by Apache Tomcat. The web page is written by JSP. Programming language and underlying technology are all existing and popular standards, showing the proposed system has great applicability. The system model which is compatible with the EPG architecture also has great flexibility for the future developments .

With the additional PC as the media center, two services are provided as example applications of the proposed system model. The media center processes the tasks which have great complexity, such as transcoding and broadcasting.

In the future we will try to solve the compatibility problem between the DVB card and the reference implementation. We will also focus on EPG and related applications. We will adopt a mature EPG data format and use our media center as an EPG processing center. The user behavior will be recorded in a local database and data mining will be conducted to find some interesting rules, which provide great benefit to those who develop commercial applications. For example, it tells a DTV content provider what TV program is popular, or what TV program the DTV subscriber prefers.

As regards the remote control by mobile devices, it is more convenient for users to control their DTVs. Moreover, since information can be transmitted to these mobile devices, interactivity will be realized more easily. Gaming, video editing, recording and banking can be done by pressing a few buttons on user's mobile device. If the DTV equipment which located in the living room became an information center, users would be able to use their mobile devices to control not only the DTV equipment but also devices in the kitchen.

Acknowledgements

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE and NSC 93-2219-E-002-031.

References

[1] Apache - tomcat. http://java.sun.com/products/midp/index.jsp.
[2] Digital homeconcept. http://www.digital-home-concept.com/en/index.htm.
[3] Dvb. http://www.dvb.org.
[4] Dvb - mhp. http://www.mhp.org.
[5] Epg. http://www.satellite.se/ordlistaen.html.
[6] Havi. http://www.havi.org.
[7] Institut fuer rundfunktechnik. http://www.irt.de.
[8] Java 2, micro edition. http://java.sun.com/j2me/index.jsp.
[9] Javabeans. http://java.sun.com/products/javabeans/.
[10] jvmstat 3.0. http://java.sun.com/performance/jvmstat/.
[11] Microsoft media encoder. http://www.microsoft.com/windows/windowsmedia/9series/encoder/default.aspx.
[12] Mobile information device profile. http://java.sun.com/products/midp/index.jsp.
[13] Tivo.com. http://www.tivo.com.
[14] The xlets. http://java.sun.com/developer/ technicalArticles/javatv/apiintro/index.html.
[15] R. Baler, C. Gran, A. Scheller, and A. Zisowsky. Multimedia middleware for the future home. In *M3W: Proceedings of the 2001 international workshop on Multimedia middleware*, pages 48–51, New York, NY, USA, 2001. ACM Press.
[16] K. Chorianopoulos and D. Spinellis. User interface development for interactive television: Extending a commercial DTV platform to the virtual channel API. *Computers & Graphics*, 28(2):157–166, Apr. 2004.
[17] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA, 2001.
[18] J. Jones. Dvb-mhp/java tv data transport mechanisms. In *CRPITS '02: Proceedings of the Fortieth International Confernece on Tools Pacific*, pages 115–121, Darlinghurst, Australia, Australia, 2002. Australian Computer Society, Inc.
[19] W. K. L. Belleguiea. Interactive dtv framework spread over broadcast and communication networks. page 198, 2001.
[20] C. Pablo and V. Petri. A graphical user interface framework for digital television.
[21] C. Peng and P. Vuorimaa. A digital television navigator. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 429–431, New York, NY, USA, 2000. ACM Press.
[22] R. Pulles and P. Sasno. A set top box combining mhp and mpeg-4 interactivity. In *EUSAI '04: Proceedings of the 2nd European Union symposium on Ambient intelligence*, pages 31–34, New York, NY, USA, 2004. ACM Press.