PUBLICATION P1

G. Sivaraman, P. Cesar, and P. Vuorimaa. System software for digital television applications. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, August 22–25, 2001, pages 784–787. IEEE.

SYSTEM SOFTWARE FOR DIGITAL TELEVISION APPLICATIONS

Ganesh Sivaraman, Pablo Cesar, and Petri Vuorimaa
Telecommunications Software and Multimedia Laboratory
Helsinki University of Technology
P.O. Box 9700, FIN-02015 HUT, Finland
Fax: +358-9-451 5351
Ganesh.Sivaraman@hut.fi, pcesar@tml.hut.fi, and Petri.Vuorimaa@hut.fi

ABSTRACT

Interactive Television is fast becoming a necessity as it converges the popular web browsing and the standard television systems better. This paper discusses the underlying system - Operating system and Java Runtime Environment - for the Digital TV. A review of the needed system capabilities for Digital TV, a probable solution of the underlying system, and future improvisation of the system are dealt herewith.

1. INTRODUCTION

Television has mostly been a one-way communicating device, giving a set of pre-determined information to the viewers. However, this is fast changing, and will continue doing so in the future. Internet TV, which allows to access the Web and send/receive emails, and Interactive TV, which shows interactive programs demanded by the user, are just a few means to make the rather “dumb device” intelligent.

The Digital Video Broadcasting Project (DVB) [15], formed in 1993, is a consortium of broadcasters, manufacturers, network operators, and regulatory bodies in more than 30 countries worldwide. DVB provides a global standard for the delivery of digital television. The Multimedia Home Platform (MHP), a subproject of DVB, consists of a common Application Programming Interface (API) to allow for the creation and broadcast of interactive television applications that will run on any set-top box, integrated Digital TV receiver, and Multimedia PCs [15]. MHP specification has selected *Java technology* for its digital interactive television-broadcasting standard [15].

Digital TV not only brings along more channels with better audio and picture quality, but allows the possibility to provide various services and applications to the viewers in addition to the regular channels. Figure 1 shows the system support needed for digital television applications. Some of the potential applications/services for digital television are [16]:

Navigator [11] – the basic user interface, with which user can select channels, launch resident programs, and configure the
set-top box.
Electronic Program Guide (EPG) [11] – provides information about programs, which are days or even weeks ahead and helps the viewer to mark them for reminding or recording. Can be integrated with Navigator.
Channel Info Bar [11] – provides a brief information about the current programs and is viewable in a small pane, which is automatically activated with selection of channel or by using “info” button on remote control. Can be integrated with Navigator.
Enhanced Text TV [12] – an advanced version of the existing tele-text, which provides information about sports, news, and program information, that is divided into pages with text and graphics.
WWW-browser – used for browsing Internet, and
Interactive programs [10] – instead of using fax, emails, or telephone for betting or buying, viewers can do it by clicking on the interactive programs.

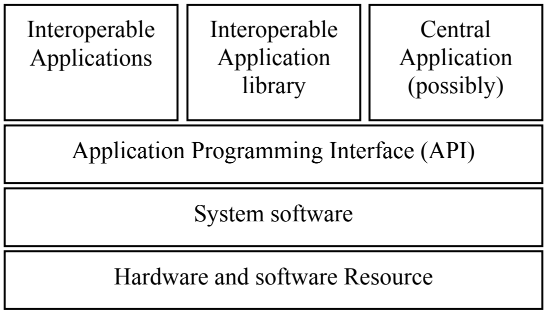
Interoperable Applications
Interoperable Application library
Central Application (possibly)
Application Programming Interface (API)
System software
Hardware and software Resource

Figure 1. DVB Technical issues Associated with MHP [5].

Digital TV being at an early stage has immense potential for research and development spanning from the hardware to the system software to the APIs. This paper discusses a prototype platform, comprising of hardware and the system software, which could probably provide an environment for digital TV applications. Most of the afore mentioned applications were executed on this environment to measure the processor load and the memory usage.

The paper is organized as follows. Section 2 gives a technical background about the capabilities of the system to support digital television applications. Section 3 explains a probable solution with test results of the system software for the digital television. Results and conclusions follow in section 4 and section 5, respectively.

2. TECHNICAL BACKGROUND

System or more specifically set-top box (STB) for Digital Television comprises of hardware, system software, and APIs that are needed for the applications during run-time. As seen in the Section 1, DVB MHP specification uses Java for the development of Digital TV applications. Hence, the system software should have the necessary support for executing Java applications.

ISBN: 0-7659-1198-8/01 © 2001 IEEE

2.1 Hardware

There exist only “Terminal Specification”, meaning they specify the I/O capabilities, leaving the rest of the hardware specification for the vendor to define [3]. Hence, when it comes to the selection of the hardware for Digital TV, crucial deciding factors are the performance and memory usage. Some vendors provide a full-fledged desktop type of computing power with 333Mhz processor, 4GB of storage space, 64MB RAM for Digital Television [13]. Others provide a bare minimum computing power of 233Mhz, 16MB of disk-on chip/flash memory for storage, and 32 MB RAM [2].

Because of such varying features in the hardware, it is difficult for the application developers to foresee the computing power of the target system. Thus, a specification chalking out the minimal support for the system is needed [3], and Nordig II [8] specifies one such specification for the Scandinavian region.

2.2 System software

System software comprises of:

Operating system - An operating system is needed to perform all the low-level processes like memory management, scheduling, I/O operations and most of all to keep all of this as abstract entities from the application's perspective.

Java Runtime Environment (JRE) - In order to run Java applications, be it textual or graphical, Java Runtime Environment (JRE) [1] is required in the target system.

2.2.1 Operating system

For any system, selecting or designing the operating system depends entirely upon the hardware capabilities, and digital TV system is no exception to this.

Certain types of hardware for digital TV systems can have computing power with plenty of storage space and processing power. In such cases, desktop kind operating system can be used. On the contrary, there could be systems with minimal hardware support that have less processing power and storage space. In such cases, the operating system needs to undergo tailoring to comply with the hardware support.

2.2.2 JRE

JRE comprises of:

i) Class loader that loads the byte code from .class, .zip, or jar files.

ii) Java Virtual Machine (JVM) interpreter that processes the byte code to execute them.

iii) Class libraries (i.e., java.awt, java.io, etc) containing the implementation of the Java API's that Java byte code can use during run-time.

Implementation of the JRE depends upon the type of the operating system and hardware. For example, java.awt package in JRE for Linux with X-Windowing System (Operating System) for i386 (hardware) is usually implemented using Motif Toolkit on top of X Toolkit that in turn access the X libraries [7].

2.3 Application Programming Interface (API)

API is a built-in programmer's toolkit for requesting data objects or services resident on a particular operating system [6]. DVB Java platform includes Fundamental APIs, Presentation APIs, Data Access APIs, Service Information (SI) and Selection APIs, Common Infrastructure APIs, Security APIs, and Other APIs, as show in figure 2.

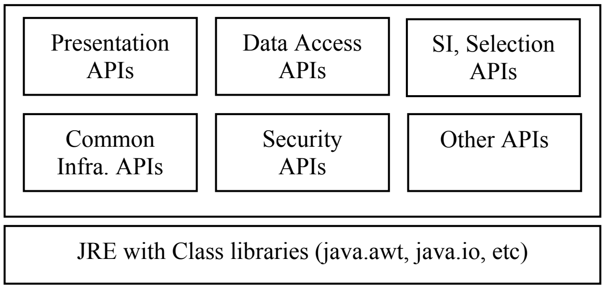

Figure 2. API stack for Digital TV applications.

DVB-MHP APIs are extended from the existing Java APIs, though certain methods from the Java APIs are not supposed to be used by inter-operable applications. Details of this are found in Fundamental APIs description [4].

3. IMPLEMENTATION

This section discusses the selected hardware – in terms of processor speed, memory storage and processing memory, etc. , and system software – operating system and JRE, for the prototype platform. Though, digital TV applications can utilize a wider suite of interfaces specified by the DVB-MHP to interact with the STB environment, we did the feasibility study of only those APIs needed by our research group.

3.1 Hardware

Our selection of the hardware has been based on the study made by Vuorimaa in his paper Digital Television Service Architecture [16]. His study shows that various services, stated in Section 1, need a bandwidth support of 22.1 Mbits/s and memory requirement of 4.4MB. With this as the basis for our study, we selected Intel based PCM5864 [9] from Advantech.

PCM 5864 has support for Pentium MMX and AMD K6, K6-II processor with speed up to 333Mhz, RAM of max. 128MB, CompactFlash card support, on-board Ethernet of 10/100 Mbps, 4Mbps IrDA, video display with 2MB, and Video-in/TV-out supporting NTSC and PAL systems.

The hardware configuration used in the prototype was 266Mhz K6-II processor, 32 MB RAM, and 16MB compact-flash memory.

ISBN: 0-7659-1198-8/01 © 2001 IEEE

3.2 System Software

Instead of focusing on the design and development of a brand new OS, it was prudent to opt for an already existing OS and to use it as a base for further improvisation.

3.2.1 Operating system

Operating system used was Linux. As the system has limited amount of storage and processing memory (RAM), it was not viable to use the standard Linux with X-Windows. Instead, it was stripped without X, but with the most essential modules such as network and framebuffer. Need for framebuffer is explained in Section 3.2.2. This reduced the size of operating system to approximately 12MB (though it will be stripped even further), which was storable in the compact-flash card. All the applications and the system software fitted well within the limits of RAM and compact flash, thus eliminating the need for any swap memory.

We used Debian Linux distribution, with 2.2.4 kernel version that has the support for the framebuffer.

3.2.2 JRE

As discussed in section 3.2.1, eliminating the X-Windows decreases the size to well within the limits. But then this raises the need for an alternative means for rendering graphical applications, such as the digital TV applications.

As mentioned in Section 2.2.2, implementation of JRE is dependent upon the operating system and hardware. Additionally, as Java based graphical applications use java.awt Graphical User Interface (GUI) toolkit, JRE needs some native graphics library for rendering. The proposed solution is based on Kaffe's implementation of JRE, which uses framebuffer for rendering [14].

Framebuffer represents the graphics hardware in such a way that the applications don't need to know anything about the low-level interface. Vesafb is the framebuffer driver for the Intel architecture, which supports VESA2.0 compliant graphics hardware. As Vesafb supports different graphical modes, a complete range of color depth and resolution are possible - 4bpp to 24bpp and 640x400 to 1600x1200.

3.3 Application Programming Interface

The research work carried out by our group focused on the development of GUI and applications for television, and hence only those were tested with the prototype platform. Service Information (SI) decoding, smart card readers, and other such applications are yet to be tested and integrated with the system.

Java applications for TV environment were developed based on the presentation APIs and the fundamental APIs of DVB MHP, which are specified in the DVB Java platform section. Fundamental APIs consists of all of the core Java classes (except Abstract Windowing Toolkit (AWT)), albeit certain methods from the standard Java class library are not used. Thus, to ensure that the applications are inter-operable with all systems, developers should avoid using those methods.

Presentation APIs are divided into *Graphical User Interface APIs and Streamed Media APIs*. *GUI APIs* are based on Java AWT class libraries. For the development of the TV user interface, *Home Audio/Video Interoperability (HAVi)* [4] class libraries are used. HAVi User-Interface allows applications written in Java to determine the user interface capabilities of its host display device, accept input from the user, draw to the screen, and play audio clips. HAVi User Interface augments Java "Lightweight User-Interface framework".

But as there is no reference implementation of HAVi, all applications have been developed using java.awt package, though adhering to certain extent to the norms of HAVi specifications.

Streamed APIs use javax.media and javax.media.protocol from *JMF* for presenting the broadcasting content (video with GUI).

Hence a JRE needs to have the necessary implementation to support the Fundamental and Presentation APIs of DVB-MHP specifications and Kaffe fulfills this requisite.

4. RESULTS

To quantify the overall performance of the system it is essential to know how much memory and processor time is consumed by each application. Linux utility, *free* shows memory usage (but in our case memory usage was in terms of RAM and *not swap* as mentioned in section 3.2.1) and *time* shows the processor load.

Time utility shows three timings; *i) user time* (i.e., higher the time, higher is Processor load), *ii) system time* (i.e., higher the time, higher is I/O load), and *iii) real time* (i.e., actual time elapsed).

Processor Load (%) = User Time / Real Time x 100

Table 1 and table 2 show the processor load and memory usage for the Ice hockey [10] demo and Navigator [11] respectively. As the processor was in idle state before launching the application, the load was unavailable, but presumably it was 0%. 18MB free memory was the memory available for running applications.

	Memory usage	Processor load
Before application starts	18MB free	
After application starts	3MB free	91%

Table 1. Memory usage and processor load for Ice hockey.

	Memory usage	Processor load
Before application starts	18MB free	
After application starts	7MB free	97%

Table 2. Memory usage and processor load for Navigator.

ISBN: 0-7659-1198-8/01 © 2001 IEEE

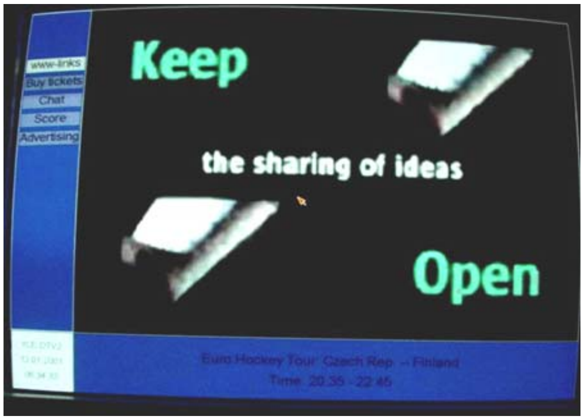

Figure 3. Page of *main screen of* Ice hockey demo.

Figures from table 1 and table 2 clearly show that both applications were processor intensive and hence degrading the performance. This was primarily due to the pure Java decoders used for video streams.

Figure 3 and figure 4 show the screen-shot of Ice hockey demo and Navigator respectively. In figure 3, available options are seen in the left side of the screen. Figure 4 shows the *channel guide* that helps the viewer to browse the present and following program of all the channels/services available. Left upper part is a video stream.

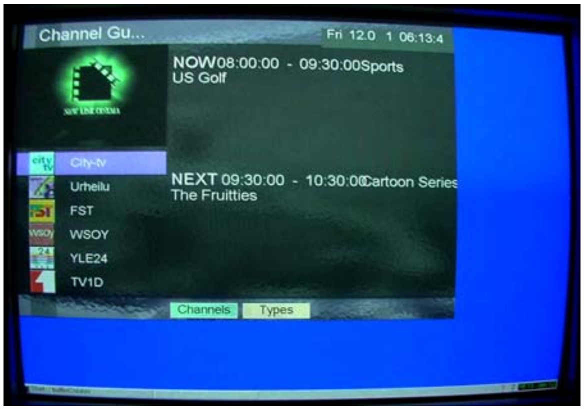

Figure 4. Page of *channel guide* in Navigator.

5. CONCLUSIONS

This paper examined system software for digital television applications. Firstly, we discussed the different hardware possibilities ranging from a full-fledged desktop to minimal hardware support. Due to the selection of Java by DVB MHP, we talked about the necessity of JRE and the possible solution of using Kaffe for Linux for this purpose. Finally, the built system was tested with Java applications for digital TV developed in the Future TV research project.

Due to the lack of any readily available DVB MHP API reference implementation, all the applications developed were an extension of java.awt package. But in reality the applications will have to use HAVi components, which is an extension of java.awt package. For test purposes, work is being carried out for providing a reference HAVi API in our project.

The DVB MHP specification states that the transport stream from the broadcasters will have MPEG2 video streams. Presently, pure Java JMF version drastically affects the performance (i.e. higher processor load), as the decoding of video stream was done in Java, instead of using native decoders. The probable solution is to have native software or MPEG hardware for decoding, which will enhance the overall performance.

REFERENCES

[1] Arnold K., Gosling K., and Holmes D. *Java programming Language Third Edition,* June 2000.
[2] Cool Logic Inc. *Internet Ready 7200 Technical Specifications.* http://www.coollogic.com/iready_specs.asp
[3] Datta T. *Digital Convergence: The Set Top Box and its Digital Interface,* White paper, Enthink, July 1998
[4] DVB Project office. *DVB: Multimedia Home Platform,* Revision 15, February 2000.
[5] Evain J. P. *The Multimedia Home Platform - an overview*, EBU Technical Review, Spring 1998.
[6] Jacklin M. *The Multimedia Home Platform: on the critical path to convergence*, IBC Show Daily, September 1997
[7] Motif FAQ's http://www.faqs.org/faqs/motif-faq/part1/
[8] NorDig information central, *Nordig II Version 0.9*, June 8, 2000 http://www.nordig.org/
[9] PCM5864 Specification http://www.advantech.com/products/PCM-5864L.asp
[10] Peng C., Vuorimaa P. *Development of Java User Interface for Digital Television*, The 8th Int. Conference in Central Europe on Computer Graphics, Visualization, and Interactive Digital Media, WSCG'2000, Czech Republic, Feb. 7-10, 2000.
[11] Peng C., Vuorimaa P. *A Digital Television Navigator*, ACM Multimedia 2000, March 20, 2000.
[12]Peng C., Vuorimaa P. *Digital Teletext Browser,* The 9th Intl. Conference in Central Europe on Computer Graphics, Visualization, and Computer Vision, February 5 – 9 2001
[13] Sigma Designs Inc. *Real Magic STB2000 Specifications*. http://www.sigmadesigns.com/products/stb2000_features_specifications.htm
[14] Sivaraman G., Vuorimaa P. *Compact Windowing System for Mobile devices*, Mobile Multimedia Systems and Applications 2000, Delft, The Netherlands, November, 2000.
[15] Stilgic J. *DVB Selects Java Technology For Its Digital Television Broadcasting Standard*, Java Press Release, November 1999 http://java.sun.com/pr/1999/11/pr991122-01.html
[16] Vuorimaa P. *Digital television service architecture,* in Proc. IEEE International Conference on Multimedia and Expo, ICME2000, New York City, NY, USA, July 30 - Aug. 2, 2000, pp. 1411-1414.

ISBN: 0-7659-1198-8/01 © 2001 IEEE