An Extensible Digital Television Middleware Architecture Based on Hardware Abstraction Layer

ZHANG Hong-guang, ZHENG Shi-bao
Institute of Image Communication and Information Processing
Shanghai Jiao Tong University, Shanghai, 200030, P.R.China
(hgzhang, sbzheng)@cdtv.org.cn

Abstract

China is drafting the digital television (DTV) interactive standard itself aiming to make it more suitable to Chinese market. This paper presents a modular, layer-oriented and extensible middleware solution compatible with the future standard. Furthermore this paper discusses the concept of Hardware Abstraction Layer (HAL) that is the base of upper software. And a resource management model on HAL is also given, which serves to guide upper applications competing for limited resources in the set-top box such an embedded platform.

1. Introduction

With the development of DTV, set-top boxes (STBs) will become the home multimedia terminals which can provide a wide range of services such as digital television, data broadcast, Internet access and so on. In consequence, STB hardware and software system becomes more and more advanced but complicated at the same time. Compared with the hardware system, the software system is more important, because it must be flexible enough to adapt to newly emerging interactive services. In addition, adapting to the short life cycles of hardware platforms and writing applications independent of any platform are other requirements affecting the software architecture. Therefore it is desirable to have an independent software platform which enables various interactive services to run transparently across vendor-specific hardware platforms. Under this background DTV middleware emerges.

The major DTV broadcasting standard organizations such as DVB and ATSC have been working to develop middleware standards by defining a generic interface between interactive digital applications and the terminals. In 2001 Chinese Ministry of Information Industry (MII) founded DTV National Middleware Standard Group aiming at drafting Chinese middleware standard. Until now the standard named Interactive Media Platform (IMP) has nearly finished. Following closely with IMP, Media Beacon Tower (MBT) is a DTV middleware solution for Chinese market. MBT solution offers the data broadcast system for network operators, the application executing system in STBs and the toolkit for the development of interactive applications.

This paper focuses on the software architecture of MBT embedded in the STB. The rest is organized as follows. Section II briefly introduces the development of DTV in China. Section III presents an overview of the MBT middleware architecture and describes each subsystem. Section IV gives an in-depth description of the Hardware Abstraction Layer which is the base of the upper software. Meanwhile the resource management model is also discussed in the part. Section V summarizes the paper.

2. Digital Interactive Television in China

China has been researching and developing digital television broadcasting system since 1990s. On 1st of October 1999, China succeeded in broadcasting the live celebration of the 50th National Day through the HDTV system. In 2001, Nation Audio/Video/Multimedia System and Device Standard Committee (AVMSC) was founded. Directed by Ministry of Information Industry, the committee focuses on drafting the standards in the field of DTV. National Middleware Standard Group is one of the 7 members. Until now the middleware

0-7803-8603-5/04/$20.00 ©2004 IEEE.
standard has been submitted and will soon be open to the public. The digital television interactive standard has made reference to existing standards such as MHP, DASE etc. But some new ideas are also brought to IMP. In general, IMP can be described in three broad categories as Enhanced Service profile, T-Commerce profile and Multi-network Service profile. To be compliant with the trend of harmonization and interoperability, IMP also adopts Java environment as the procedural execution engine. Its JVM is based on J2ME CDC Foundation Profile. And IMP adopts all Java APIs in DAVIC and a few in MHP as DTV professional Java packages. But some functions have been modified. For example in the field of multimedia framework, IMP drafts the new media package referring to JMF. And we also draft new APIs in fields such as service selection and service information. As for the broadcast protocol, IMP accepts the DSM-CC Data Carousel, but adds the concept of stream and event to it just like in the Object Carousel. The standards involved in content and presentation are also defined in IMP. Since 2003, China has begun to pay much attention to promoting the industry of DTV, intending to popularize DTV in the cities before 2008. The standards such as IMP will drive the development greatly. The IMP prototype named MBT has been implemented so far. And several provincial cable television operators in China have decided to accept MBT solution.

3. Software Architecture Overview

The software embedded in the STB is modularized and layer-oriented as depicted in figure 1. Its architecture can be divided into 5 layers, namely Platform Driver Layer, Hardware Abstraction Layer, Function Layer, System Layer and Application Layer.

![Figure1. MBT DTV middleware architecture](image)

3.1 Platform Driver Layer

Platform Driver Layer refers to STB hardware platform together with all device drivers and the operating system (OS). All hardware-related functions are accomplished by the device drivers which provide a suit of application program interfaces (APIs) to control the hardware such as Audio/Video decoder, tuner, etc. The OS is the core of runtime environment. Generally this layer is provided by the STB manufacturer. As a result there may be a lot of differences that mainly locate in three aspects—the function of the device driver, the form of the API and the type of the OS. The hardware platform’s diversity results in the complexity while implementing the middleware. So an adaptation layer must be put forward to absorb the uncertainty.

3.2 Hardware Abstraction Layer

Driven by the requirements above, a single layer named Hardware Abstraction Layer (HAL) is constructed in MBT. It manages to shield the influence upon the upper software because of the diversity of hardware platforms. In another word this layer is the base of middleware. Section III will give a description of the layer in detail.

3.3 Function Layer

Function Layer realizes DTV functions by supporting upper software based on HAL. It influences the middleware performance directly. But there is nothing related to this layer in open middleware standards. After all it is the middleware designers to decide what functions this layer should meet. So this part is the most private one among those middleware solutions. In MBT the Function Layer includes 4 subsystems, namely SI Engine, Graphic & AV Subsystem, Communication Protocol and Core Subsystem. SI Engine is responsible for monitoring and filtering the specific information in data stream to set up SI database. Graphic & AV Subsystem acts as the multimedia engine responsible for graphic library and Audio/Video playing. Communication Protocol Subsystem realizes the protocol stacks such as DSM-CC and TCP/IP. And Core subsystem consists of memory manager module, event manager module, file system etc.

3.4 System Layer

System Layer acts as the executing environment for upper applications. It is the core of the middleware system. In MBT this layer basically is composed of Application Program Interface (API) Packages, Java Virtual Machine (JVM) and Web Engine. API Packages provide APIs to develop DTV applications. It generally includes Java platform packages and a series of DTV professional packages such as graphic, multimedia and so on. JVM manages to interpret and execute Java applications. And it can support such functions as Java debug. And the Web Engine provides a familiar browser environment based on HTML and ECMAScript. The differences among those interactive DTV standards mainly gather on this layer.
3.5 Application Layer

Application Layer means the application software running on the DTV middleware, which can be divided into two kinds: Resident Application and Download Application. The first one usually contains the key applications resident in STB together with the middleware such as the Navigator that can be used to select services, applications and control the life cycle of applications etc. The second one contains the applications which can be downloaded to STBs such as Electronic Program Guide (EPG), interactive games etc.

4. Hardware Abstraction Layer in STB

As a complicated software system, the DTV middleware should be implemented on a STB hardware platform finally. So the DTV middleware must be highly independent of any specific hardware and OS platform. However, the middleware architecture itself is configurable: Enhanced Broadcast profile, Interactive Broadcast profile or Internet Access profile. On the other hand the STB hardware platform and OS differ in thousands of ways whatever on function or performance. So there must be an adaptation layer between the upper software and the underlying hardware platform. This paper names the layer as Hardware Abstraction Layer (HAL) which is abstracted through classifying the functions of STB platform by using the object-oriented method. The most important three concepts in HAL are resource modules, device context and resource management.

4.1 Resource Modules and Device Context

From the perspective of abstraction, the underlying hardware platform is divided into hardware and software entities that can both be used by application environment. It can be said that these entities are system resources making use of the STB device drivers to implement various services of DTV. Therefore these entities can also be called resource modules. Figure 2 lists some commonly used resource modules that are abstracted according to the function of interactive DTV. Of course, with the development of the hardware platform and interactive services, many other function modules may emerge.

Resource modules in HAL may differ on different hardware platforms. However, no matter what kind of the hardware platform is, its resource modules should have the following characters:
(1) Each resource module runs independently. It communicates with other modules or upper software through the event mechanism.
(2) The upper software may use the same resource module simultaneously in multi-tasks.
(3) Each resource module may be mapped into one or more device drivers generally such as AV decoder. It can also be software and hardware compound behavior such as section filtering and table assembling.

HAL helps us ignore the differences of hardware platform while designing the DTV middleware system at the beginning. And it will be easy to make out porting solutions on different platforms. To the implementation, each resource module is an encapsulated entity that supports the upper software through APIs. As a result the functions of these modules can be described through defining APIs named Software Poring Interfaces (SPIs). Once the developers implement these SPIs the middleware can be ported to any specific platform. As for the definition of SPI there is no corresponding standard. It is the developers of the middleware system that draft SPIs according to the function and extensibility. The TV Linux Alliance has already tried to draft this kind of standard.

Nowadays with the development of hardware, some high-end STB platforms can support multi-channel streams’ decoding and displaying. For example, ATI Xilion220 is capable of decoding up to two high definition video streams, or up to eight standard definition video streams. And two separate display paths drive two display devices, with different content on each display. Therefore there may be multi resource modules corresponding to the same kind of device driver. To distinguish the same resource modules the concept of device context is introduced. The device context can be defined as a runtime environment in which the applications get presented. Generally in STB platform it means the data pipeline including tuner, demod, descramble, demux, A/V decoder and A/V output etc. So any STB platform has at least one device context. The user should point out which resource module in which device context. And it should be guaranteed that such a device context is set up before being used.

4.2 Resource Management (RM) model

HAL makes the middleware system independent of diversiform hardware platforms and OS. Consequently the middleware only interacts with resource modules in HAL. It does not care about the implementation of each module. However, there must be a logic channel between HAL and the upper software for...
communication. On the other hand, to STB the embedded system, the resource is so limited that there must be a method to allocate resource for multi-tasks runtime environment. What is the most important is that HAL may be different in resource modules on STB platforms from innumerous STB hardware manufacturers. So there must an adaptive porting solution for platforms ranging from high-end to low-end. Therefore it is necessary to set up a resource management model on HAL.

This paper gives a resource management model based on request-proxy architecture. The idea comes from the technology of CORBA. The client who wants to access the resource modules sends request to proxy. The model includes three parts: Resource Client (RC), Resource Server (RS) and Resource Proxy (RP). RC refers to the upper software that wants to utilize the resource modules to realize some specific functions. RS is responsible for controlling all the resource modules. RP interacts with RS to determine whether to give RC the right to access HAL modules. Figure 3 illustrates the relations among them.

![Resource management model](image)

MBT implements the model in a pure software module named Resource Manager. The working process is described as follow. When RC wants to acquire the right to access any module, it will send a request to RP that keeps the request and interview with RS. The request may include the device context, the module name and something else. Then RS will query inner records and return corresponding result. This model not only constructs a logic communication channel and resource management solution but also offers a kind of resource register mechanism that makes the porting solution more adaptive. When the resource manager is initialized, all the resource modules in the HAL will register themselves to RS. After that RS will record the number, capacity and state of each module. The middleware can extract the information such as the performance and function from STB platforms through accessing the inner record in RS. During the runtime environment if the platform can’t meet the need of upper clients the resource manager may let the users know friendly. In this way the porting solution will be independent of specific platforms. In MBT, resource manager works in the form of APIs. Any operation that tries to access resource modules must call these APIs according to specific order.

5. Summary

The middleware is the basic software platform for interactive DTV. Following closely to Chinese middleware standard IMP, MBT is a good solution for Chinese DTV market. Based on the HAL the modular and layer-oriented software system MBT is highly independent of any specific hardware and OS platform. The proposed resource manager serves as the bridge between HAL and upper software. Until now we have implemented MBT on some platforms such as ST 5518, ATI Xillion220/225 and Samsung S5H2000X01.

Acknowledgements

The presented DTV middleware summarizes the work of the whole MBT project team sponsored by Shanghai High Definition Digital Technology Innovation Center. We specifically would like to thank Liao Hong-tao, Sun Hai-sheng and all other team member for their tremendous contributions to the project.

References