Design of Middleware for Interactive Data Services in the Terrestrial DMB

Gwangsoon Lee, KyuTae Yang, Kwang-Yong Kim, YoungKwon Hahm, Chunghyun Ahn, and Soo-In Lee

ABSTRACT—In this letter, we introduce a new middleware architecture and its generic application programming interface (API) (called the T-DMB MATE API) for terrestrial digital multimedia broadcasting (T-DMB). Middleware in T-DMB enables inter-operable applications to be downloaded from both broadcast and telecommunication networks in advance and to be executed in any type of T-DMB receiver. The middleware we introduce here is especially designed to support a proposed method for application provisions applied to a concept of application module appropriate to the service environment of T-DMB. We also verify the designed T-DMB MATE API through the implementation of the middleware and its application in a PC-based receiver.

Keywords—DMB, middleware, interactive data service, java, DAB, data broadcasting.

I. Introduction

Terrestrial digital multimedia broadcasting (T-DMB) is one of the applications that has emerged from the Eureka-147 DAB system [1]-[3]. Particularly in Korea, T-DMB focuses on the broadcasting of programs and their reception in various mobile environments. In addition to video services, T-DMB also accommodates basic data services such as Broadcast Website (BWS) and Slide-Show through various data transport protocols. As well as these basic data services, T-DMB is now ready to supply rich media services.

Recently, the scope of rich media services has been extended to work on generic application programming interfaces (APIs) enabling inter-operable applications to be downloadable from both broadcast and telecommunication networks and to be executable in any types of receivers. The standard for this purpose under the standardization of the Telecommunication Technology Association (TTA), the Korea telecommunications standardization body in information and communication technology (ICT) fields, called the T-DMB MATE (mobile application terminal environment) standard, defines the environment for running applications and APIs so that a variety of applications can access the hardware resources in any type of receiver.

In this letter, we introduce the T-DMB MATE architecture and focus on the proposed method for broadcasting T-DMB MATE contents effectively to T-DMB MATE receivers.

II. Interactive Data Services in T-DMB

1. Broadcasting Protocol Stack in T-DMB

Broadcasting protocol being supported in T-DMB is shown in Fig. 1. Since T-DMB is based on the Eureka-147 system, T-DMB accommodates audio and data services as well as video

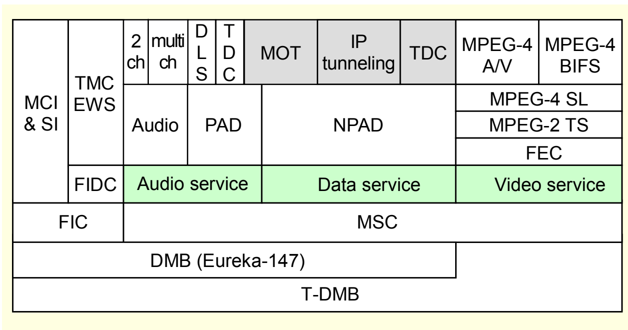

Fig. 1. The broadcasting protocol stack of T-DMB.

Manuscript received Feb. 21, 2006; revised June 26, 2006.

Gwangsoon Lee (phone: +82 42 860 1676, email: gslee@etri.re.kr), KyuTae Yang (email: ktyang@etri.re.kr), Kwang-Yong Kim (email: kimky@etri.re.kr), YoungKwon Hahm (email: ykhahm@etri.re.kr), Chunghyun Ahn (email: hyun@etri.re.kr), and Soo-In Lee (email: silee@etri.re.kr) are with Radio & Broadcasting Research Division, ETRI, Daejeon, Korea.

services including MPEG-2 and MPEG-4 technologies [1]-[3].

For data services, T-DMB supports a variety of transport protocols such as Multimedia Object Transfer (MOT), IP Tunneling, and Transparent Data Channel (TDC) [4], [5]. For the T-DMB MATE service, the MOT protocol is mainly used for the transmission of applications and the TDC protocol is used for the transmission of real-time streaming data.

2. Structure of the T-DMB MATE Receiver

The structure of the T-DMB MATE receiver for interactive data services is shown in Fig. 2. Broadcast signal received by the RF tuner and de-modulator is divided into video and data streams, which are decoded by the respective decoders. The decoded video is played on the display through the frame buffer. The various data can be displayed, blending with the decoded video through the frame buffer. Further, the T-DMB MATE receiver can support an interface with a return channel such as a mobile communication network for interactive data services.

Applications use APIs defined by the MATE standard to access the actual resources of the receiver including the tuner, AV decoders, and the return channel interface. These resources are functional entities of the receiver and may be finally mapped onto the hardware of the receiver in some manner.

3. Requirements for Designing the T-DMB MATE

To provide data services efficiently in T-DMB, there are several points we should take into careful consideration. The first thing is an efficient use of data bandwidth in T-DMB. The effective channel bandwidth available for mobile broadcasting service in T-DMB is about 1.2 Mbps. So, the channel bandwidth allocated for one data service is expected to be less than around 256 kbps. However, compared with the bandwidth of other digital TV data broadcasting systems, this bandwidth is not sufficient to supply various data services with good quality. Under the conditions of such a channel environment, the user may wait to subscribe to some data services. Accordingly, minimization of the latency for launching data applications is essential in order to provide efficient data services. The second point is to minimize the resource usage of the MATE receiver, when we consider that the T-DMB is targeting mobile terminals such as cellular phones, PDAs, and so on. Such mobile terminals may have memory and CPU capacities too limited to perform high-level functionalities.

However, since the existing types of middleware for digital TV, such as MHP (multimedia home platform) and ACAP (advanced common application platform), do not meet these requirements, it is necessary to develop a new middleware appropriate to the T-DMB. In this letter, we provide solutions to meet the requirements described above.

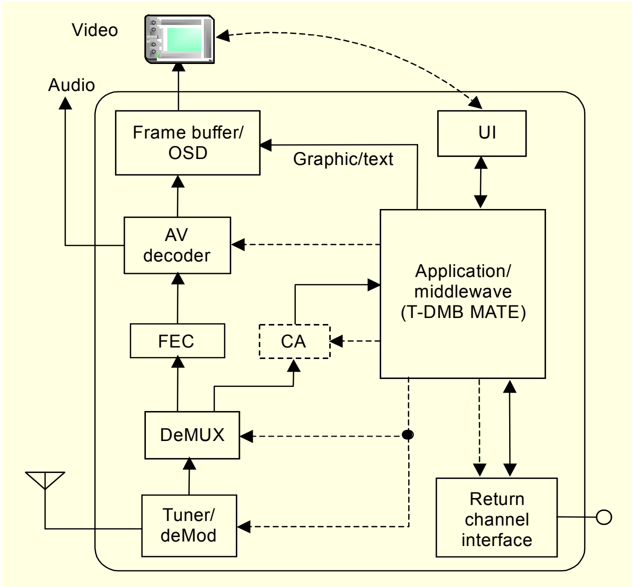

Fig. 2. The T-DMB receiver for the T-DMB MATE applications.

III. Novel Features of the T-DMB MATE

The T-DMB MATE's main functions are for application reception and its management. It provides high-level functionalities to applications. The architecture of the T-DMB

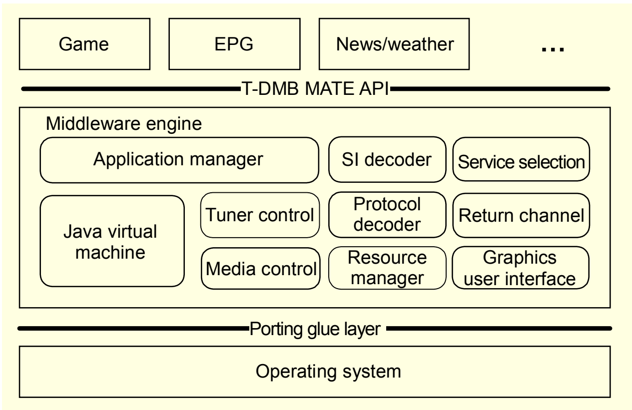

Fig. 3. The structure of the T-DMB MATE.

MATE is shown in Fig. 3. An operating system (OS) supports the middleware engine placed below the upper MATE API. The protocol decoder in the middleware engine receives and decodes data transmitted through various broadcasting protocols such as MOT, TDC, and IP tunneling as shown in Fig. 1. The application manager is a major entity in the T-DMB MATE, and it supports a variety of functions such as application reception, application storage, application execution, and so on. This application manager especially deals with the proposed technique for effective application provision, which will be clearly described in the following section. The resource manager controls the usage right of system resources being used by applications. In brief, it has two functions: to manage the usage right and to solve a resource collision.

1. Application Provision Using Module Concepts

In order to provide the first solution for meeting the requirements described in the previous section, we present a method for providing applications comprised of codes and data by module units, a so-called application module, which is a unit to be intentionally grouped by a contents provider. As shown in Fig. 4, since an application is composed of several application modules managed by a module database, codes and data made up of applications can be transmitted and stored in the MATE receiver by module units.

Modules can be provided to be commonly used by many applications. In this case, the sharing of modules by many applications provides a memory saving effect on the receiver and a bandwidth saving effect from the viewpoint of transmission. Moreover, the applications can be easily modified by managing the application with the module unit. In addition, the modules can be transmitted via various bearers as well as broadcasting. Accordingly, the proposed method can provide applications in flexible ways, such as in a telecommunication network.

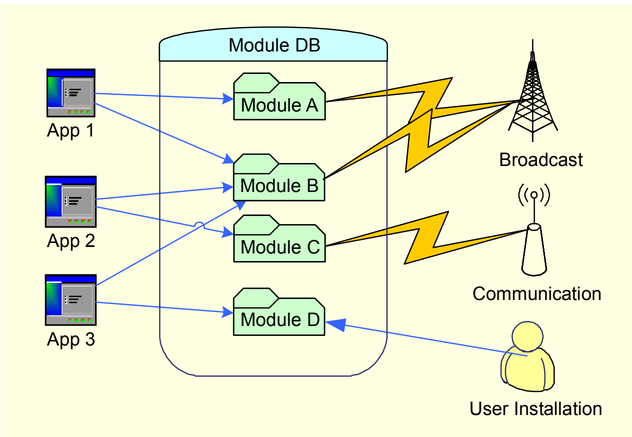

Fig. 4. Application provision by the proposed module concept.

An application module is identified by a module ID instead of a uniform resource locator (URL) since the module ID can be independent of the network's location. Therefore, the application modules can be transmitted via various networks and protocols. Each application module consists of body and metadata for its additional information, which may be compressed by a format like ZIP.

The application module's ID is defined as following:

[!]<*domain_name*>/<*module_specific_name*>, (1)

wherein "!" means an existence of electronic signature, *domain_name* is an Internet domain name of a contents provider that supplies application modules, and *module_specific_name* denotes a name for distinguishing an application module of the contents provider.

2. Application Signaling Messages in the T-DMB MATE

The T-DMB MATE defines five application signaling messages for supplying information about application modules to be downloaded by schedule and the like. The messages are shown in Fig. 5, illustrating their relationship with services. These messages are also transmitted by MOT protocol whose service component is signaled by a user application type defined by fast information channel (FIC) [1].

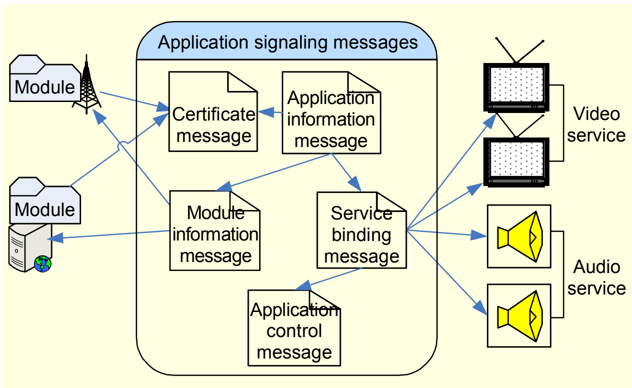

Fig. 5. Application signaling messages for application provision.

A. Application Information Message (AIM)

The AIM includes the following kinds of information about applications available at present or in the future:

- Application ID, version and format
- Profile and version of executable receiver
- Definition of dependent relationship between applications
- Application modules constituting an application
- Icon, name, and description for designating applications
- Electronic signature for protecting the AIM

B. Module Information Message (MIM)

The MIM contains information about the application modules constituting an application defined in the AIM. It means that application modules referred to in the AIM should be shown in the related MIM. The MIM includes ID, version, size of the application modules, URL indicating a location to download the application modules, and a time schedule to download the application modules at each URL. Further, the URL has no limitation, that is, application modules can be downloaded via any protocol channels including broadcasting or telecommunication networks.

C. Service Binding Message (SBM)

The SBM allows applications to be related with each service such as video, audio, and so on. With this SBM and an ACM to be described in the following section, we can control the execution time of an appropriate application when a service is chosen by a user, and we can notify an application related to other services as well as an application that should be executed at present. Therefore, when the user selects some services, the user can select and download specific applications after checking the application list available in all services. For example, if the user is aware of an application in advance that supplies additional information about a program that is expected to broadcast on Saturday afternoon, the user can download application modules constituting the application in advance at any time and enjoy the service by executing the application at the time when serviced.

D. Application Control Message (ACM)

The ACM transmits a control signal for executing and terminating applications related with a specific service designated by the SBM. With this ACM, applications can be triggered and terminated during the active service. Because the ACM refers to binding tags between the service and applications defined in the SBM, the size of the ACM is so small that we can transmit it frequently.

IV. Implementation and Experimental Results

Besides the theoretical evaluation of the T-DMB MATE standard, we have verified it by implementing a PC-based T-DMB MATE receiver. We have also implemented a data server and several interactive applications for the data service based on the T-DMB MATE. The application introduced in this letter is about a data service to supply the additional information of a music program, such as a music list being played in the current TV program, information about singers, and so on. In order to verify the designed T-DMB MATE, we configured each signaling message introduced in the previous sections. In particular, we controlled this application bound with a video service by using SBM and its execution by using the ACM. Figure 6 shows the image which resulted from providing the MATE application and signaling messages as well as T-DMB video service to a receiver platform. We have confirmed that the application modules can be downloaded by scheduling and the applications could be executed when we selected the video service bound with the MATE application. The trigger time of execution also was controlled by the ACM. After execution of the application, the user would be able to enjoy the additional information related to the music video and also download to play audio clip files from the web server though the TCP/IP return channel.

Fig. 6. MATE application bound with a T-DMB video service.

V. Conclusion

In this letter, we introduced the T-DMB MATE, which is a middleware platform for application in T-DMB. The T-DMB MATE was especially designed to support the proposed method for the application provision and signaling so as to meet the service environment of the T-DMB. We also verified the designed MATE API through the implementation of the middleware and its application in a PC-based receiver. In the future, we will implement the T-DMB MATE in various T-DMB receivers, such as cellular phones, stand-alone receivers, and so on.

References

[1] ETSI EN 300 401 v1.3.3, *Radio Broadcasting Systems*, *Digital Audio Broadcasting (DAB) to Mobile, Portable, and Fixed Receivers,* Sept. 2001.

[2] G.S. Lee, B.J. Bae, Y.K. Hahm, and S.I. Lee, "A Novel Method for Inserting an MPEG-TS into Ensemble in a DMB Transmission System," *ETRI J.*, vol. 26, no. 6, Dec. 2004, pp. 653-656.

[3] ETSI TS 102 428 v1.1.1, *Digital Audio Broadcasting (DAB); DMB Video Service; User Application Specification*, June 2005.

[4] ETSI EN 301 234 v1.2.1, *Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) Protocol*, Oct. 1999.

[5] ETSI EN 101 759 v1.1.1, *Digital Audio Broadcasting (DAB); DAB Data Broadcasting Transparent Data Channel*, Sept. 2000.