NOTTV Mobacas Software

Technology Reports

New Service Merging Communications and Broadcasting—NOTTV—

Mobile Terminal Technology Supporting Mobacas Service —Mobacas Software Platform for Android—

*With the aim of facilitating the propagation of Mobacas-viewing terminals, NTT DOCOMO has developed a Mobacas software platform whose specifications and modules will be released for general use. This Mobacas software platform was developed so that the functions needed to enjoy the Mobacas service could be operated on the middle layer and application layer of the Android™*1 platform. This article introduces these functions and describes techniques for implementation.*

Communication Device Development Department

Motoyuki Uchida
Kazuma Nachi
Kohei Ichihara
Yuki Fukuyoshi

1. Introduction

On setting out to develop software for use with Mobacas-viewing terminals, it was assumed that both software specifications and the developed software would be released to other communications operators and broadcasters to promote the propagation of Mobacas-viewing terminals.

For this reason, NTT DOCOMO decided to develop a Mobacas software platform for general use independent of specialized hardware, drivers, engines or library*2. This article studies the functions to be placed on this Mobacas software platform and techniques for implementation.

It was found from this study that the target of our development work on the Mobacas software platform would be between the middle layer and the application layer above the driver layer. We also adopted a software configuration that would allow each broadcaster to develop original User Interface (UI) applications to differentiate their UI from others.

In this article, we describe the configuration of this software platform for achieving Mobacas viewing functions and the functions to be placed on this platform.

2. Software Configuration of Mobacas Software Platform

The Mobacas software platform runs on Android. The software configuration diagram is shown in **Figure 1**. This Mobacas software consists of an application layer that contain UI applications and a service platform and a middle layer that contain a middleware platform. This configuration facilitates interface compatibility so that existing Android functions can be effectively used.

*1 **Android™**: A software platform for smartphones and tablets consisting of an operating system, middleware and major applications. A trademark or registered trademark of Google Inc., United States.

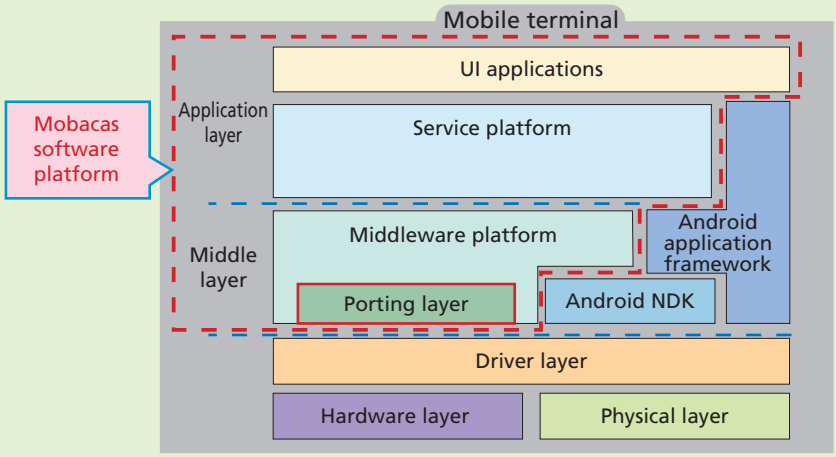


Figure 1 Software configuration

The above software is released for implementation to handset vendors as a Mobacas software platform consisting of an application layer and middle layer. This Mobacas software platform features a porting layer[*3] within the middleware platform to absorb differences in the driver, hardware and physical layers. This has the effect of making the software independent of terminal-dependent hardware, drivers and libraries and enables interface compatibility to be achieved. To implement Mobacas functions, a handset vendor need only make whatever modifications are needed to enable its terminal environment to accept the porting layer.

3. Mobacas Built-in Functions

Mobacas targets mobile terminals as a precondition and aims to combine broadcasting and communications so that users can enjoy content and services regardless of time and place. To achieve this usage format, the Mobacas terminal provides the user with the following functions as UI applications. These functions are based on ARIB[*4] technical reports [1] and standards [2] – [7].

- Real-time broadcasting viewing function
- Storage-based broadcasting viewing/compensatory functions
- Electronic Program Guide (EPG), Electronic Content Guide (ECG)
- Social Networking Service (SNS) linking function

The following describes how we implement the above functions on the Mobacas software platform.

3.1 Real-time Broadcasting Viewing Function

As the name implies, the real-time broadcasting viewing function enables the viewing of real-time broadcasting content (content that, among the various Mobacas services, is mainly intended for real-time viewing). The Mobacas broadcasting system is Integrated Services Digital Broadcasting Terrestrial for mobile multimedia (ISDB-Tmm)[*5], which, as an extension of the ISDB-T[*6] digital terrestrial television broadcasting system, can provide the same services as before such as digital terrestrial broadcasts and One Seg. The key features of Mobacas real-time broadcasting content are higher quality and the capability of linking with other functions, as described below.

1) High-quality Content

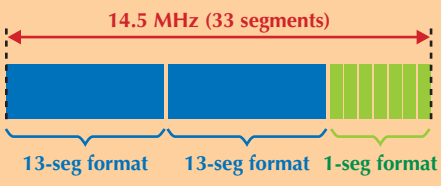
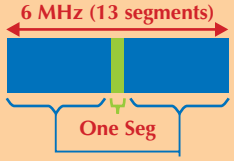
Mobacas adopts a high-quality coding method surpassing that of One Seg to provide crisp and clear images on terminals like smartphones and tablets having relatively large screens. **Table 1** compares the real-time broadcasting coding methods used by Mobacas and One Seg.

Additionally, a Mobacas-compatible terminal provided with an external output having a copyright-protection function in the manner of the High-Definition Multimedia Interface (HDMI)[*7] or Digital Living Network Alliance (DLNA)[*8] can connect to a larger external display for more enjoyable viewing of real-time broadcasting content.

There are many mobile terminals today that incorporate hardware for decoding video and audio content. For

*2 **Library**: A collection of high-versatility programs that can be reused/recycled.
*3 **Porting layer**: A layer designed for facilitating the migration of programs and applications to a different environment by giving them generality.
*4 **ARIB**: An organization subordinate to the MIC that sets standards for systems that use the radio spectrum in the fields of communications and broadcasting in Japan.
*5 **ISDB-Tmm**: A multimedia broadcasting standard for mobile terminals in Japan based on the ISDB-T (see *6) digital terrestrial broadcasting standard. It was formulated for the purpose of mobile reception by mobile phones and other mobile devices

Table 1 Comparison of real-time broadcasting coding methods

Service	Mobacas	One Seg
Method	ISDB-Tmm	ISDB-T "One-seg"
Image coding format	H.264/MPEG4 AVC	
Profile/Level	Main Level : 3	Baseline Level : 1.2
Pixel count	720×480 (525SD)	320×180 (QVGA)
No. of frames	30fps	15fps
Segment arrangement		

H.264/MPEG4 AVC : A moving-picture compression standard
Profile/Level : Profile indicates target application while Level indicates the maximum load that can be processed. These are used in combination to indicate performance.

this reason, we have designed the Mobacas software platform without a function for decoding so that it can be performed on the terminal side.

2) Linking with Other Functions

The real-time broadcasting viewing function is equipped with a browser based on the Broadcast Markup Language (BML)[*9], which makes it possible to superpose a display area for BML-based data broadcasts on the viewing area of the program being watched the same as in digital terrestrial television broadcasts.

Since linking with communication functions is a Mobacas precondition, the provision of interactive content using BML and communications is possible. Mobacas has also been designed to enable mutual linking between real-time broadcasting content and storage-based content. It would be easy, for example, to schedule the saving of storage-based content from real-time broadcasting content, and conversely, to link to the viewing of real-time broadcasting content from storage-based content.

Furthermore, as the Mobacas software platform supports the media scheme[*10] mechanism, it is possible to link with viewing functions, scheduling functions, etc. by having Mobacas-terminal applications like the browser and mail application support this mechanism.

3.2 Storage-based Broadcasting and Compensatory Processing

Mobacas real-time broadcasting is analogous to digital terrestrial television broadcasting in that all users receive and watch a program at the same time from their terminals. Storage-based broadcasting, on the other hand, is a system whereby content is delivered by broadcast signals to a user's terminal and stored for later viewing at a more convenient time.

In unidirectional-communications type of broadcasting services, it is not possible for the terminal to send out resend requests to the broadcasting side, so to receive that portion of data that could not be initially obtained for some reason (such as terminal out-of-range), the terminal must wait until that portion is transmitted again over broadcast signals. This means the user may have to wait quite a long time before all desired data is received.

To therefore build up resistance to data loss, we have implemented error correction functions in layers other than the physical layer so that original data can be easily restored even for many occurrences of missing data. Moreover, for content that still has insufficient data

*6 **ISDB-T**: Japan's digital terrestrial broadcasting standard formulated for fixed-communications in the home as well as for mobile reception by mobile phones and other mobile devices.
*7 **HDMI**: A digital video/audio input-interface standard for digital home appliances. Includes a copyright protection function in addition to video/audio transmission functions.
*8 **DLNA**: An organization of manufacturers in the fields of information appliances, mobile terminals and PCs that promotes activities for standardization to ensure interconnection in the digital age and establish technical specifications. A DLNA function is one which conforms to standards defined by this organization. Home appliances supporting DLNA functions can be linked and used together.

even after error correction has been performed, we have implemented a content compensation function that enables missing portions of data to be downloaded via a communication network such as 3G, LTE or Wi-Fi®[*11] from a contents-compensation server deployed in the broadcasting system to supplement partially received content. Linking broadcasting and communications in this way results in a system that can deliver a large volume of content in an efficient manner. **Figure 2** shows the flow for obtaining storage-based data by the above system.

1) Delivering Storage-based Broadcasts

Storage-based broadcasting uses File Delivery over Unidirectional Transport (FLUTE)[*12] as a file transfer protocol. It performs Application Layer-Forward Error Correction (AL-FEC)[*13] to build up resistance to data losses and converts the data to Moving Picture Experts Group phase2 Transport Stream (MPEG2-TS)[*14] packets for delivery. We note here that content may consist of still pictures, HTML, Java script, etc. in addition to video; one item of content may consequently consist of multiple files.

The Mobacas software platform presented here supports a dedicated viewer for video playback and a storage-based-broadcast browser supporting HTML5[*15]. We can therefore envision the delivery of content combining video and HTML and providing links to communications content such as detailed information on the video being watched, and the delivery of newspaper or magazine content combining HTML, video and still images.

One drawback to achieving high-quality video in real-time broadcasts is that a certain amount of broadcast band must be occupied. This, however, is no longer an issue once storage-based content has been saved since no amount of broadcast band is needed to view that content. Video in the form of storage-based content can therefore be delivered at a higher level of quality compared with content delivered by real-time broadcasts. Coding of storage-based content is summarized in **Table 2**. At present, Mobacas supports only Spec2 coding specified in Part 3 of ARIB TR-B33 [1], but future support

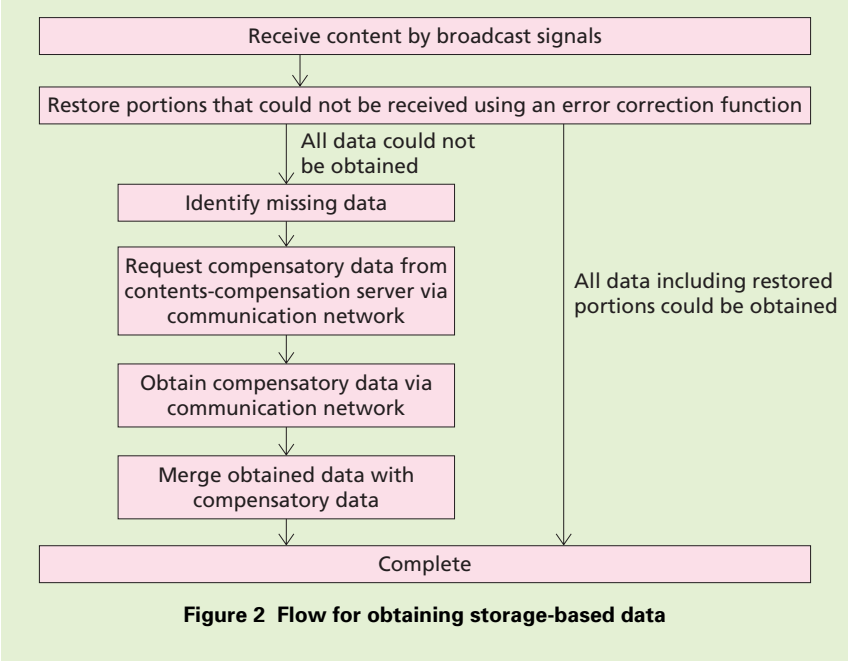


Figure 2 Flow for obtaining storage-based data

Table 2 Coding of storage-based content (Spec2)

	Video coding method	Audio coding method
Standard	H.264/MPEG4-AVC	MPEG-4 AAC HE-AAC v1 HE-AAC v2
Profile/Level	High/Level:3.1	
Pixel count	-1,280×720 (720p)	
No. of frames	-30fps	
Max. bit rate	10Mbit/s	
Notes	ISDB-Tmm standard includes specifications up to Full HD	

MPEG-4 AAC : An audio coding method featuring high compression efficiency
HE-AAC v1 : An audio coding method featuring low degradation of audio quality under low bit rates
HE-AAC v2 : An audio coding method featuring even less degradation of audio quality under low bit rates than HE-AAC v1

*9 **BML**: XML-based markup language for data broadcast.
*10 **Media scheme**: A mechanism for linking with and starting up applications such as broadcast viewing, content playback and scheduling from a browser, mail application, etc.
*11 **Wi-Fi®**: A registered trademark of the Wi-Fi Alliance.
*12 **FLUTE**: A unidirectional transmission protocol used in broadcast networks.
*13 **AL-FEC**: Generic name of an error correction code applied on the application layer.
*14 **MPEG2-TS**: An MPEG-2 system applied in broadcasting and communications.
*15 **HTML5**: An enhanced version of HTML formulated by WHATWG and W3C.

of Spec3 and Spec4 will enable users to view content at even higher levels of quality.

2) Receiving Storage-based Content

Content scheduling and receive processing is based on information contained in a previously delivered ECG or in transmission-control metadata within the broadcast TS at the time of content delivery.

Content scheduling can be performed by a function that enables the user to select desired content from the ECG or a function that automatically schedules content based on parameters specified in the ECG from the broadcaster.

Once content scheduling has been completed by the user or by automatic processing, the receiver will be activated just prior to the scheduled delivery time and storage of the content will commence at that time.

Received content is saved within the terminal itself or in external memory, and once saved, it is reconfigured into viewable content through AL-FEC error correction and Message Digest algorithm 5 (MD5)[*16] checking.

3) Content Compensatory Processing

After content delivery completes, that content will be checked for insufficiently received portions, and if found, they will be subjected to compensatory processing to produce complete, viewable content.

This processing consists of manual compensatory processing in which the user explicitly checks for insufficiently received, non-viewable content via a UI and automatic compensatory processing that performs this processing automatically based on delivery-related information.

Given that content compensatory processing can result in partial content downloading using the communication network, the possibility exists that a large amount of content compensating by a large number of users will occur. We have therefore designed content compensatory processing taking load on the communication network into consideration. An example of compensatory scheduling is shown in **Figure 3**.

To prevent congestion in the

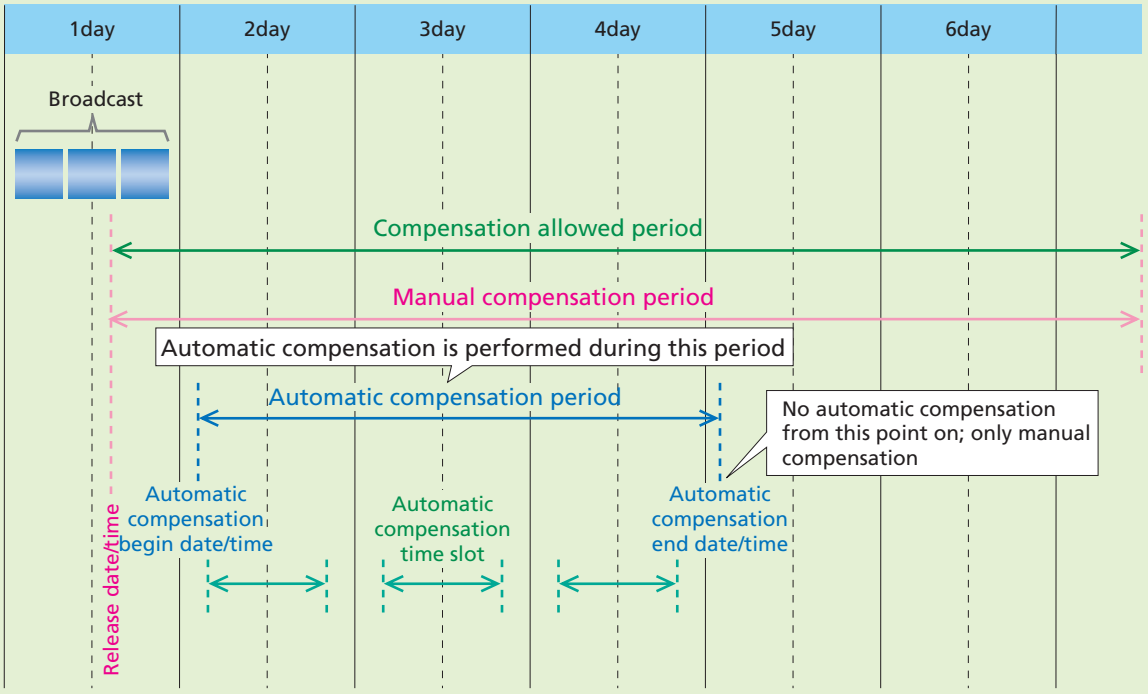


Figure 3 Example of compensatory scheduling

*16 **MD5**: A hash function (a one-way digest function) generally used with certificates and digital signatures. Since a unique value is assigned to input data, the detection of file tampering during transmission over communications networks (such as the Internet) is possible by comparing the MD5 value before and after transmission.

accessing of a contents-compensation server from terminals, compensation processing is performed at random times within the designated automatic compensation period and automatic compensation time slots.

3.3 EPG/ECG Functions

Mobacas includes an EPG function for presenting program information on real-time broadcasts and enabling program selection and an ECG function for presenting content information on storage-based broadcasts and enabling content selection.

The metadata information in the EPG/ECG enables the user to view, browse, search for and schedule programs. There is also transmission-control metadata for scheduling the receiving, saving and compensating of storage-based broadcasts.

Each of these types of metadata is delivered in the form of information elements as listed in **Table 3**.

By enabling these two types of metadata to be received and handled in an integrated manner, the user can seamlessly enjoy all types of content on the same screen without having to distinguish between real-time and storage-based broadcasts.

1) Metadata Transmission and Delivery

Mobacas provides metadata by both broadcasting and communications means. A terminal can always provide the user with up-to-date program information even if a connection by either of these means is disabled.

2) Obtaining Metadata by Broadcasting

Metadata transmission by broadcasting considers that the amount of bandwidth that can be used depends on the time slot. There is therefore a partial-reception layer for transmitting small amounts of metadata (Layer A) and a non-partial-reception layer for transmitting large amounts of metadata (Layer B).

Layer A transmits metadata by a 13-segment partial-reception layer. In particular, it transmits as needed metadata for programs to be delivered by real-time broadcasts or storage-based broadcasts within 72 hours from the present. Metadata obtained via Layer A also includes transmission-control metadata for storing B Layer metadata. Receiving B Layer transmission-control metadata from A Layer metadata in this way makes it possible to schedule the reception and storage of B Layer metadata.

Layer B transmits metadata by other than a 13-segment partial-reception layer. Specifically, it transmits metadata for programs to be delivered by real-time broadcasts or storage-based broadcasts within three to eight days from the present day. It transmits this metadata as a single item of storage-based content, which means that

Table 3 Metadata information elements

Metadata Type	Information element	Description
EPG/ECG metadata	Program information element	Program information (title, synopsis, genre, credit list)
	Group information element	Information on series or packages combining multiple items of program information
	Program location element	Broadcast time, broadcast period, location within terminal
	Service information element	Service information (service name/channel)
	Segment information element	Scene information within the program demarcated by time
	Segment group information element	Information combining multiple segments
	License information element	Information on program use (license period, format, acquisition)
	Purchase information element	Program-related charging information
	Coupon information element	Coupon description (coupon name, discount rate, valid period)
Transmission-control metadata	UserServiceDescription	CRID information corresponding to EPG/ECG metadata
	SessionDescription	Information related to program receiving
	AssociatedDeliveryProcedureDescription	Information related to compensation (compensating server URI, compensation period)

CRID : Content Reference ID

any missing data can be compensated and restored.

3) Obtaining Metadata by Communications

Metadata transmission by communications is performed with HTTP over TCP/IP. This method enables metadata to be obtained from the EPG/ECG server in the following sequence:

- Initial application startup
  Obtain two-days-worth of metadata.
- All subsequent application startups
  If eight-days-worth of metadata has not yet been received, obtain needed metadata in one-day units until eight-days-worth has been received, and update the most recent 48-hours-worth of metadata.

3.4 SNS Linking Function

The Mobacas software platform incorporates a function that enables users to share information on programs or content that they are currently viewing or reading by linking with a SNS like Twitter[*17] or Facebook[*18]. Here, by displaying a Twitter or Facebook screen on the same screen as the program being watched, the user can display and read program-related comments made by other viewers. The user can also post program-related comments and share information with other users after logging into a personal account. Furthermore, depending on the type of program being broadcast, we can envision a scenario in which a program host introduces a comment posted by a user over Twitter during the broadcast. In short, we can expect users to be able to share information not just with each other but also with hosts of live programs. Heightening the sense of live participation in this way can make a program even more enjoyable to watch compared to existing TV broadcasting formats.

4. Conclusion

In this article, we described the development of a Mobacas software platform for Mobacas-viewing terminals with a focus on built-in functions. This Mobacas software platform makes it possible for handset vendors to develop Mobacas terminals by simply modifying the porting layer. Looking forward, we plan to study the addition of more functions and improvements to this Mobacas software platform and to develop technologies for achieving such enhancements.

REFERENCES

[1] ARIB TR-B33: "Optional Guidelines for Terrestrial Mobile Multimedia Broadcasting by Transmission System Based on Connected Segments," Feb. 2012.
[2] ARIB STD-B25: "Conditional Access System Specifications for Digital Broadcasting," Mar. 2011.
[3] ARIB STD-B32: "Video Coding, Audio Coding and Multiplexing Specifications for Digital Broadcasting," Mar. 2011.
[4] ARIB STD-B38: "Coding, Transmission and Storage Specification for Broadcasting System Based on Home Servers," Mar. 2011.
[5] ARIB STD-B45: "Content Download System for Digital Broadcasting," Mar. 2011.
[6] ARIB STD-B46: "Transmission System for Terrestrial Mobile Multimedia Broadcasting Based on Connected Segment Transmission," Mar. 2011.
[7] ARIB STD-B53: "Receiver for Connected Segments for Terrestrial Mobile Multimedia Broadcasting (Desirable Specifications)," Mar. 2011.

*17 **Twitter**: A registered trademark of Twitter Inc. in the United States and other countries.
*18 **Facebook**: A trademark or registered trademark of Facebook, Inc.