Ginga-J: The Procedural Middleware for the Brazilian Digital TV System

Guido Lemos de Souza Filho, Luiz Eduardo Cunha Leite, Carlos Eduardo Coelho Freire Batista

Digital Video Applications Lab
Department of Informatics
Federal University of Paraíba
Campus I – Cidade Universitária
Phone: +55 (83) 3216 7093 – Ext 26 (FAX)
Zip 58.035-000 – João Pessoa – PB – BRAZIL
{guido, leduardo, bidu}@lavid.ufpb.br

Abstract

The recent development of the research on digital terrestrial television in Brazil has led the country's government to state a series of premises in which the government shows to care not only for technology improvement, but also to use this development as a tool for ameliorating the Brazilian social context, in what concerns digital inclusion. These premises and necessities have generated some peculiarities in the development process, which directly influenced in the functionalities granted by the Brazilian's middleware choice. This paper, thus, seeks to explain all the architecture of the Java part – called Ginga-J – of the Ginga middleware, highlighting the new features, especially when confronting the Brazilian middleware with the other middlewares worldwide defined.

Keywords: Digital TV, middleware,

1. INTRODUCTION

Ginga is the name of the middleware specified for the International Standard for Digital Television - Terrestrial (ISDTV-T). ISDTV-T is currently being adopted as Brazilian's official system for Terrestrial Digital TV.

The applications to be executed over Ginga are classified into two categories depending upon the way they are written. Procedural applications are those written using the Java language and declarative applications are those written using the NCL language. Ginga application execution environments are similarly classified into two categories depending upon whether they process declarative or procedural applications, and are called Ginga-J [19] and Ginga-NCL [18], respectively.

Ginga-NCL is a logical subsystem of the Ginga middleware which is responsible for processing and presenting NCL documents. In addition, Ginga-J is the middleware's subsystem in charge of defining all the Java Application Program Interfaces (APIs), content and data formats, besides protocols up to the application level.

In comparison to the middleware systems conceived for the other Digital TV standards, some features of

Ginga are innovative. These features arise from the synergy resultant from the conjunction of two previous projects: FlexTV [1] and MAESTRO [2], respectively the procedural and declarative Brazilian reference middlewares conceived during the SBTVD (from the Portuguese Sistema Brasileiro de TV Digital – Brazilian Digital TV System) Project. This paper contains information about Ginga-J's implementation and functionalities.

The SBTVD project engaged researchers from various Brazilian universities and research centers. SBTVD activities involved the research of all parts that compose a Digital TV System and the development of some new characteristics to fit to the Brazilian context (requirements defined by the Brazilian government).

The main objective of the SBTVD was to give the government the necessary information to conduct a decision on the Brazilian open terrestrial Digital TV System. As a result of the SBTVD, a new standard, named International Standard for Digital Television Terrestrial – ISDTV-T, was created and adopted in Brazil. Ginga is, thus, the middleware for the ISDTV-T standard. Despite the fact that Ginga carries innovations, compatibility was considered requirement and the execution of applications designed to different middleware – those adherent to ITU's standards J.200 [4], J.201 [5], J.202 [6], and consequently to GEM [10] (Globally Executable MHP) definition – is supported.

2. A DIGITAL TV MIDDLEWARE

The advent of Digital TV has brought computational functionalities to the Television service, which is worldwide widespread. The new TV environment becomes interactive, since applications can be transmitted and executed alongside with TV programs.

The possibility of executing the same application on different devices, with different processing power and different manufacturers is solved by the adoption of a common middleware, which can be defined as a software layer to abstract the platform's hardware specific characteristics. That is the mechanism used by Digital TV, an environment which is composed by devices developed by different manufacturers. The middleware allows content providers to develop an application that will run (adapted or not) on all Digital TV receivers on a particular network.

2.1. COMPATIBILITY BETWEEN MIDDLEWARES

To allow the execution of MHP (Multimedia Home Platform) [3] applications on other Digital TV platforms, the DVB group proposed a unified specification for Digital TV System's middlewares, called GEM [10]. GEM includes MHP characteristics that are not related to specific characteristics of DVB system. This specification has been adopted by the Japanese (ISDB ARIB B.23 [20]) and American (ATSC ACAP [22] and OCAP [21]) middleware standards. Ginga, the middleware of the ISDTV-T – in its Java Part called Ginga-J – is compatible with GEM.

Following the same trend of compatibility, The International Telecommunication Union (ITU) published a set of recommendations called ITU-T, which consists in: J.200, J.201 and J.202 (equivalent to GEM). These recommendations intend to harmonize the systems of Digital TV at different levels [10][3][7].

3. GINGA MIDDLEWARE ARCHITECTURE

Ginga definitions do not mandate a basic architecture. The architecture section of the Ginga definitions are defined as *an informative example of a possible architecture for the Ginga middleware.*

This section briefly introduces one reference for the Ginga Architecture, which is adherent to ITU J.200 recommendation [4], and specifies a set of common functionalities – Ginga-Core, which must support the Ginga declarative application environment (Ginga-NCL) and the Ginga procedural application environment (Ginga-J).

The universe of Ginga applications may be partitioned into a set of declarative applications and a set of procedural applications. A declarative application is an application whose initial entity is of a declarative content type. A procedural application is an application whose initial entity is of a procedural content type. A purely declarative application is one whose every entity is of a declarative content type. A purely procedural application is one whose every entity is of a procedural content type. A hybrid application is one whose entity set contains entities of both declarative and procedural content types. A Ginga application needs not to be purely declarative or procedural. In particular, declarative applications often make use of script content, which is procedural in nature. Furthermore, a declarative application may reference an embedded JavaTV Xlet. Similarly, a procedural application may reference declarative content, such as graphic content, or may construct and initiate the presentation of declarative content. Therefore, either type of Ginga application may make use of facilities of both declarative and procedural application environments.

Ginga-NCL is a logical subsystem of the Ginga System that processes NCL documents, and its adherent to ITU J.201 recommendation [J201]. A key component of Ginga-NCL is the declarative content decoding engine (NCL formatter). Other important modules are

the XHTML-based user agent, which includes a stylesheet (CSS) and ECMAScript interpreter, and the LUA engine, which is responsible for interpreting LUA scripts.

Ginga-J is a logical subsystem of the Ginga System that processes active object content, and its adherent to ITU J.202 recommendation [J202]. A key component of the procedural application environment is the procedural content execution engine, composed by a Java Virtual Machine.

Common content decoders serve both procedural and

declarative application needs for the decoding and presentation of common content types such as PNG, JPEG, MPEG and other formats. The Ginga-Core is composed by common content decoders and procedures to obtain contents transported in MPEG-2 Transport Streams and via the return channel. The Ginga-Core shall also support the conceptual display model as described in the Volume 1 of ISDTV Standard 06.

Figure 1: Ginga high level architecture , based on ITU J.200 [4]

The architecture and facilities of the Ginga Specification are intended to apply to broadcast systems and receivers for terrestrial (over-the-air) broadcast. In addition, the same architecture and

facilities may be applied to other transport systems (such as satellite or cable TV systems).

.

Figure 2: Ginga Common Core components [18] [19]

4. GINGA-J ARCHITECTURE

4.1. GINGA-J CONTEXT

Figure 3 shows the context in which the GINGA-J

software stack will execute. In this Figure, the GINGA-J software stack resides on the GINGA Host Device, which may be, for instance, a set-top box, a digital TV set, a cell phone etc.
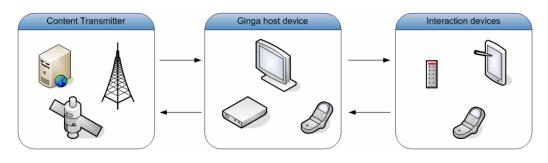
Figure 3: Ginga-J context [19]

The GINGA-J software has access to streams of video, audio, data, and other media assets. These streams are distributed over the air, received and processed by applications, which can present content to the viewers.

The viewer may interact with the application through input and output interaction devices attached or associated with the GINGA host device. The GINGA host device will receive input from the viewers via interaction devices such as remote controls or keyboards. In response to the viewer input, the Ginga host device will present visual output as well as audio output using its own display and loudspeakers or using displays and loudspeakers of the interaction devices. A single device may have input and output capabilities simultaneously. An example of interaction device could be a PDA connected to the GINGA platform through a wireless network. Using such an interaction device, a viewer can send commands to the platform through the PDA's keyboard and the platform's applications can send visual content to be

presented in the PDA's screen. An interaction device may also have sound capture and playback capabilities.

Many viewers may interact with the GINGA platform at the same time. In this case, each viewer may have an interaction device and the platform must distinguish the commands sent by and to each device.

4.2. GINGA-J ARCHITECTURE OVERVIEW

The GINGA-J model distinguishes between hardware entities or resources, system software, and applications as depicted in Figure 4.

Native applications may be implemented using no standardized functionalities, provided by the Ginga host device Operating System or by the Ginga implementation. Naïve applications may also use Ginga-J standardized APIs. Broadcasted applications (Xlets) must use standardized APIs provided by the Ginga-J.
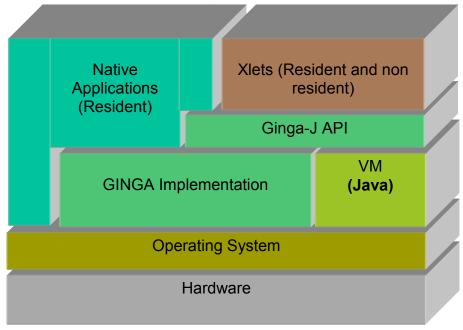

Figure 4: GINGA-J Architecture and execution environment [19]

In general, Ginga is unaware of any native applications. These include but are not limited to: closed captioning, conditional access (CA) system messages, receiver menus, and native electronic program guides.

Native applications may take precedence over Ginga applications. As an example, closed captioning and emergency messaging shall take precedence over the Ginga System.

5. GINGA-J SPECIFICATIONS

Ginga-J [19] is the procedural part of Ginga, the Brazilian's Digital TV middleware. Based on Java technologies (the Java Virtual Machine plus some APIs), it incorporates many innovations, but maintains compatibility with most of the current Digital TV middlewares, since it adheres to GEM [10].

Brazilian Digital TV was created considering digital convergence as a mandatory requirement, since the modulation standard employed by the Brazilian System is aimed to transmit the digital TV signal simultaneously to different devices such as High and Standard Definition fixed TV receivers, mobile and portable devices, like cell phones. Ginga-J specification also includes support for communication with devices using Bluetooth, Wi-Fi, Infra-red, Powerline, Ethernet or any network technology typically used in Home Area Networks. Applications can provide support for multi-user interaction, since the TV receiver can be accessed by different devices simultaneously.

5.1. GINGA-J API

Some important requirements identified on the Brazilian context did not meet any correspondent functionality on the established international middleware definitions. In order to fulfill the Brazilian specific requirements and simultaneously maintain international compatibility with the GEM APIs, Ginga is based on three sets of APIs called:

Green, Yellow and Blue (Figure 5). The Ginga Green APIs are the GEM compatible APIs. The Yellow APIs were extensions proposed to fulfill the Brazilian specific requirements that can be implemented through the use of a software adapter using Green APIs. The Blue APIs are the ones which are not software compatible with the GEM APIs. In such a way, applications that only make use of Green APIs can be executed in Ginga, MHP, OCAP [21], ACAP [22] and B.23 [20] middlewares. Applications that use Green and Yellow APIs can only be executed in MHP, ACAP, OCAP and B.23 if the necessary software adapter is transmitted and executed together with the application. Applications that use Blue APIs will only be executed in Ginga middleware environments.

Ginga-J Green API set is composed by the Sun JavaTV [7], DAVIC [9], HAVi [8] and DVB [11][3] packages, all included on GEM framework specification. Ginga-J Yellow API set is composed by the JMF 2.1 API [12], which is necessary for the development of advanced applications, with sound capturing for instance; an extension for GEM's Presentation API, with functionalities to support the video stream specifications defined in the Ginga-J standard; an extension for GEM's return channel API, which allows the sending of asynchronous messages; and an extension to ISDB ARIB B.23's [20] Service Information API [23]. Ginga-J Blue API set is composed by a Device Integration API, which allows the digital TV receiver to communicate with any device with a compatible interface (Wired, as Ethernet or PLC; or Wireless Network, as Infrared or Bluetooth), which can be used as an input or output device; a Multi-User API, which uses the Device Integration API to allow multiple users to interact simultaneously with Digital TV applications; A NCL Bridge API, which allows the development of Java applications containing NCL applications.
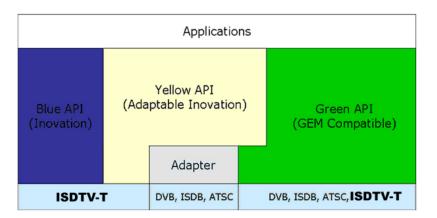

Figure 5: Ginga-J green, yellow and blue APIs [19]

6. GINGA-J REFERENCE IMPLEMENTATION

A Ginga-J reference implementation is being developed to validate the Ginga-J's specifications and APIs. That implementation now holds more than 800 thousands of source lines of code. The development of the reference implementation currently uses Intel® Kalaheo platforms as testbed.

The Ginga-J reference implementation was designed to be the procedural middleware for a universal Digital TV receiver, since it incorporates three important characteristics:

- o *It is multi-network* – Ginga-J specification [19] defines a Tuner API that is compatible with all networks currently used for Digital TV transmission (terrestrial, cable, satellite and IPTV).

- o *It is multi-system* – The ISDTV-T System adheres to ISDB ARIB B.10 [24] Service Information specification, so Ginga-J specification [19] defines a Service Information API compatible with ARIB B.23 [20]. However, Ginga-J reference implementation includes DVB, ATSC and ARIB Service Information table processing, using its Service Information API (with B.23 syntax) as the data output.

- o *It is application compatible* – Since Ginga-J specification [19] defines the Green API set (section 4.1), the Ginga-J reference implementation is able to run most of the applications designed to run on GEM compatible middlewares (adopted by systems such as DVB [11], ATSC [22], ISDB [20], Blu-ray [25], etc).

6.1. A COMPONENT BASED REFERENCE IMPLEMENTATION MIDDLEWARE

The Ginga-J reference implementation uses an approach based on software components. This approach facilitates the temporal functional evolution of the middleware, by allowing the incorporation of new functionalities through the addition of new components, and also allows the reuse of some of the components in others middleware implementations, for example, for PDAs, mobile phones, high-end digital TV receivers (with many resources, more expensive), low-end Digital TV receivers (with fewer resources, cheaper), etc. Those components can be gathered on different profiles, depending on some characteristics of the platform (see Figure 6).

The main architectural elements of Ginga-J architecture can be grouped by their functionalities using seven denominations:

- o Low level stream access components – contains the

elements responsible for accessing the transport streams, and for processing and demultiplexing them on the various elementary streams that constitute them.

- o Elementary Stream processing components – contains the elements responsible for processing the elementary streams, decoding and providing them to other components.

- o User Interface Components – contains the elements responsible for enabling user interaction through the presentation of audio-visual elements, as well as managing user generated events (that may involve external devices).

- o Communication Components – are the elements responsible for enabling the communication between the applications being executed over the middleware, as well as with other applications.

- o Management Components – elements responsible for the middleware management (context management, middleware update, etc) and for the applications management (lifecycle control, fault control, etc).

- o Persistence Components – elements responsible for storing persistent data.

- o Conditional Access – elements responsible for the security of the restrict access content broadcasted by content providers.

Each of these groups, are detailed on the subsections above, with their main components being described.

6.1.1. LOW LEVEL STREAM ACCESS

The Low Level Stream Access components are:

Tuner – The Tuner is the component responsible for the physical channel selection, as well as the selection of one of the transport streams, which are being transmitted over the selected channel.

Stream Information Server – The Stream Information Server is responsible for identifying which are the elementary streams present on the transport stream selected by the Tuner and to provide related information (program scheduling, audio and subtitle available options, etc) to the other middleware components. This component was developed to handle multiple types of information bit streams, from the major middleware standards, which makes Ginga-J independent of the transmission standard.

Demultiplexer – The Demultiplexer is the component responsible for providing the elementary streams present on the transport stream to the other middleware components. These elementary streams can be made available for any Ginga-J component, for an application

running over the middleware or to some hardware component (an audio or video decoder, for instance).

6.1.2 ELEMENTARY STREAM PROCESSING COMPONENTS

The components that are related with Elementary Stream (ES) processing are:

Media Processing Controller – The Media Processing Controller is the component responsible for controlling the processing of the multimedia elements and to provide them to the other middleware components. Through this API it is possible to select which audio and video stream must be decoded, which subtitle shall be used, to initiate and to stop the media decoding process, etc. The Media Processing Controller component contains specific media parsers and processors for each supported media type. The Media Processing Controller API is based on JMF 2.1 [12], which is backward compatible with JMF 1.0, currently used by GEM compliant middleware systems.

Data Stream Processor –The Data Stream Processor is the component responsible for accessing, processing and for providing to the other middleware components the elementary data stream such as: data carousel, IP packages transmitted through broadcast, etc. The Data Stream Processor is also responsible for notifying the other components of events such as the arrival of applications, synchronous and asynchronous DSM-CC events, etc.

6.1.3 USER INTERFACE COMPONENTS

Ginga-J reference implementation User interface components are:

Media Presentation Controller – The Media Presentation Controller is the component responsible for the presentation of the multimedia streams (audio, video, images, etc).

User Event Manager –The User Event Manager component is responsible for capturing the user generated events, such as the remote control key pressing, commands from an interaction device (like a mobile phone) and to pass those events to the registered applications and or middleware components.

Graphic Elements – The Graphic Elements component is composed by the main graphic elements used to develop an application. Those graphic elements include buttons, text boxes, etc.

6.1.4 COMMUNICATION COMPONENTS

The components developed for Ginga-J reference implementation that deal with communication are:

Interaction Channel – The Interaction Channel component is responsible for providing interfaces which can be used by other middleware components to access the interaction channel, which is a bidirectional data channel that can be used by the local applications to communicate

with remote applications. The interaction channel component was developed to support multiple kinds of networks (PTSN, Ethernet, GSM, WiMax etc) and it also includes some features which allow the scheduling of data to be sent at a specific time.

Inter-Application Communication – The APIs provided by the Inter-Application Communication component can be used to allow the applications running over Ginga-J to communicate with each other.

6.1.5 MANAGEMENT COMPONENTS

Application Manager – The Application Manager is a software element responsible for: loading, configuring, instantiating and executing the applications over Ginga-J; providing an API to control the application lifecycle; identifying and preventing application fault; and also to manage the resources utilization and the access control.

Middleware Manager – The Middleware Manager is the component responsible for updating the middleware code on execution time. It allows the middleware's components to be substituted for errors correction or due to functionality change or improvement. This component is also responsible for providing the information regarding the actual context of the Digital TV receiver. This context is defined on terms of capacity, usage and availability of CPU resources, memory, persistence devices, etc.

6.1.6 PERSISTENCE SERVICES COMPONENTS

The components dealing with Persistence services are:

Profile Manager – The main objective of the Profile Manager component is to provide, to all middleware components or to applications running over Ginga-J, a set of information defined by the Digital TV receiver user (user defined preferences). Those definitions can be related to parental control, general service authorization, etc.

Persistence – Component designed to allow an object to be stored after the process which created it is finished.

6.1.7 CONDITIONAL ACCESS COMPONENT

Conditional Access Module – The Conditional Access Module is the middleware module responsible to control the access to restricted content transmitted to the receiver. Those content can be received both from the broadcast channel as from the return channel (IPTV).

7. GINGA-J INNOVATIONS

When the Brazilian government led the researches on the development of the reference middleware for Brazilian Digital Television, it stated some important requirements to be fulfilled. Those requirements were mostly based on some particularities of Brazilian's social context. For instance, only 32.1 million of people have access to the internet, which represents 21% of the Brazilian population – Brazilian government then defined that Digital TV

should be a tool for digital inclusion, since that TV is present on 91% of the Brazilian households [13].

During the development of the reference procedural middleware for the Brazilian Digital TV System, many studies about the major Digital TV middleware solutions adopted worldwide were conducted, and, since the majority of the specifications are based on GEM [10] and J.202 [6] specifications, it was clear that some requirements would not be accomplished, since Europe's context (which guided GEM development) is very different from Brazilian's [13] (Figure 6).
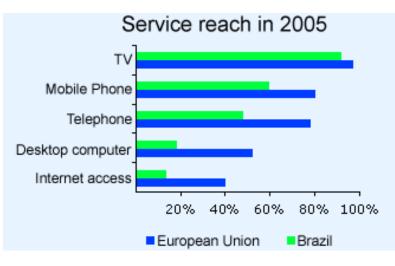

Figure 6: Reach of some communication services in Brazil versus in European Union [13][14].

Ginga-J innovative functionalities, provided by its APIs, allow the development of advanced applications, exploring the integration with other devices, such as mobile phones, PDAs, etc. That integration was motivated by another number: Brazil currently holds 79.5 millions of mobile phones **Erro! Fonte de referência não encontrada.**.

A mobile phone can be used as a return channel for the TV environment, used as a remote control, used as an interaction device (to answer polls individually, for instance), etc. Since those functionalities are all implemented using common protocols such as Bluetooth, USB, Wi-Fi, Ginga is compatible with many devices.

```
public void userEventReceived(UserEvent event) {
    if (event instanceof HRemoteUserEvent) {
        System.out.println("HRemoteUserEvent");
        ...
    } else {
        System.out.println("UserEvent");
        ...
    }
}
```

Figure 7: Example of source code using Ginga-J multi-user API

The source code seen on Figure 7 shows how an application, using Ginga-J APIs, detects whether a user event was generated by a user attached to an external device or not.

A pratical example of an application using the advanced resources provided by Ginga-J is the Virtual Cheering [15], which is an application that allows viewers of an event to share their acoustic space, simulating a stadium or an arena environment. The user audio input can be done with any device, a mobile phone with Bluetooth, for instance.

Another Ginga-J innovation to be highlighted is the existence of unbounded applications, which are applications that have a lifecycle not attached to a TV show, which can be also persistently saved. One of the uses for this feature is to distribute educational interactive

applications which can be saved by teachers and students and used on their classes.

Ginga-J's features will have an enormous reach, since the Brazilian's Communication Minister stated based on government researches that at the end of the transition to the digital model (the deadline for the total transition is 2017), there will be 80 million of digital TV receivers in Brazil [16].

8. CONCLUSION AND FUTURE WORKS

Ginga middleware aims to become a middleware specification for multiple purpose devices. The devices are going to be grouped in profiles, according to their characteristics and features (like the existence of a display, the processor capacity, the bandwidth of the return channel, etc), and each profile will have an associated set of Ginga's components.
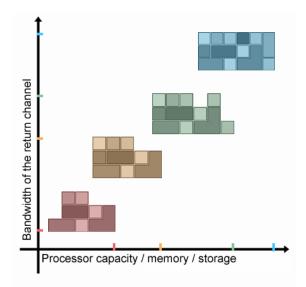

Figure 8: Example of four profiles for Ginga considering devices with different characteristics

Some innovations are being expanded toward convergences with other digital devices. The Digital TV receiver (attached to the TV or in a form of a set-top box) is a strong candidate to become the media hub for the upcoming smart home, providing content (multimedia documents, applications or both) to other devices. Ginga is ready for this trend already, and some improvements are being studied and developed.

REFERENCES

[1] LEITE, L. E. C., et al. "*FlexTV – Uma Proposta de Arquitetura de Middleware para o Sistema Brasileiro de TV Digital* (FlexTV – a Middleware Architecture Proposal for the Brazilian Digital TV System)". In *Revista de Engenharia de Computação e Sistemas Digitais*, v. 2, pp 29-50, 2005.

[2] SOARES, L. F. G. . "MAESTRO: The Declarative Middleware Proposal for the SBTVD". In *Proceedings of the 4th European Interactive TV Conference*, Athens, 2006.

[3] MORRIS, S., SMITH-CHAIGNEAU, A. "Interactive TV Standards – A Guide to MHP, OCAP and JavaTV". ISBN-13 978-0-240-80666-2. Elsevier, Focal Press, 2005.

[4] ITU. "ITU-T Recommendation J.200: Worldwide common core – Application environment for digital interactive television services", 2001.

[5] ITU. "ITU-T Recommendation J.201: Harmonization of declarative content format for interactive television applications", 2004.

[6] ITU. "ITU-T Recommendation J.202: Harmonization of procedural content formats for interactive TV applications", 2003.

[7] Sun Microsystems. "Sun JavaTV: Java Technology in Digital TV". Available at: http://java.sun.com/products/javatv/. Accessed on June, 2006.

[8] HAVi. "HAVi Level 2 Graphical User-Interface - Specification of the Home Audio/Video Interoperability (HAVi) Architecture". HAVi, Inc.

2001. Available on: <http://www.havi.org>. Accessed on November, 2006.

[9] DAVIC. "DAVIC 1.4 Part 2 – DAVIC Specification Reference Models and Scenarios", 1998. Available at <http://www.davic.org>. Accessed on November, 2006.

[10] ETSI. "TS 102 819 V1.3.1: Digital Video Broadcasting (DVB) Globally Executable MHP version 1.0.2 (GEM 1.0.2)". ETSI Standard, 2005.

[11] ETSI. "TS 102 812 V1.2.1: Digital Video Broadcasting (DVB) Multimedia Home Platform (MHP) Specification 1.1.1". ETSI Standard, 2003

[12] Sun Microsystems. "Java Media Framework API (JMF)". Available at <http://java.sun.com/products/java-media/jmf/index.jsp>. Accessed on November, 2006.

[13] IBGE. "*IBGE – Pesquisa Nacional por Amostra de Domicílios* (Brazilian Institute of Geography and Statistics – National Research by Home Sampling)". Available at <http://www.ibge.gov.br/home/estatistica/populac ao/trabalhoerendimento/pnad2005/tabsintese.shtm >. Accessed on November, 2006.

[14] Teleco. "*Por que a Internet tem penetração menor que o Celular?* (Why does Internet has less reach than Mobile Telephony?)". Technical Report. Available at <http://www.teleco.com.br/comentario/com175.as p>. Accessed on October, 2006.

[15] TAVARES, T.A., et al. "Sharing virtual acoustic spaces over interactive TV programs – presenting Virtual Cheering application". In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '04)*, Vol. 3. ISBN: 0-7803-8603-5. Taipei, Taiwan, 2004.

[16] MAZENOTTI, P. "*TV digital deve aumentar em 80 milhões número de aparelhos no país, diz ministro* (Digital TV shall increase number of TV sets to 80 millions on the country (Brazil), says minister)". RadioBrás – Agência Brasil. Available at <http://www.agenciabrasil.gov.br/noticias/2006/0 7/06/materia.2006-07-06.4998754189/view>. Accessed on October, 2006.

[17] ISDTV-T Forum. "Volume 1 of ISDTV-T Standard 06". ISDTV-T Forum Draft, December, 2006.

[18] ISDTV-T Forum. "Volume 2 of ISDTV-T Standard 06". ISDTV-T Forum Draft, December, 2006.

[19] ISDTV-T Forum. "Volume 4 of ISDTV-T Standard 06". ISDTV-T Forum Draft, December, 2006.

[20] ARIB. "ARIB STD-B23 Version 1.1: Application Execution Engine Platform for Digital Broadcasting (English Translation)". ARIB Standard, 2004.

[21] CableLabs. "OpenCable Application Platform Specification – OCAP 1.0 Profile". Cable Television Laboratories, Inc, 2005.

[22] ATSC. "ATSC Standard: Advanced Common Application Platform (ACAP)". ATSC Standard, 2005.

[23] ISDTV-T Forum. "ISDTV-T Standard 03". ISDTV-T Forum Draft, 2006

[24] ARIB. "ARIB STD-B10 "Service Information for Digital Broadcasting System". ARIB Standard, 2004.

[25] Blu-ray Disc Association. "Blu-ray Disc Association – Experience Blu". Available at < http://www.blu-raydisc.com/>. Accessed on December, 2006.