

PUC

ISSN 0103-9741

Monografias em Ciência da Computação

nº 05/2019

A Ginga-enabled Digital Radio Mondiale Broadcasting Chain: Signaling and Definitions

Rafael Diniz

Álan L. V. Guedes

Sergio Colcher

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

A Ginga-enabled Digital Radio Mondiale Broadcasting Chain: Signaling and Definitions

Rafael Diniz, Alan L. V. Guedes and Sergio Colcher

rafaeldiniz@telemidia.puc-rio.br, alan@telemidia.puc-rio.br, colcher@inf.puc-rio.br

Abstract. ISDB-T International standard is currently adopted by most Latin America countries. To support interactive applications in Digital TV receivers, ISDB-T defines the middleware Ginga. Similar to Digital TV, Digital Radio standards also provide the means to carry interactive applications; however, their specifications for interactive applications are usually more restricted than the ones used in Digital TV. Also, interactive applications for Digital TV and Digital Radio are usually incompatible. Motivated by such observations, this report considers the importance of interactive applications for both TV and Radio Broadcasting and the advantages of using the same middleware and languages specification for Digital TV and Radio. More specifically, it establishes the signaling and definitions on how to transport and execute Ginga-NCL and Ginga-HTML5 applications over DRM (Digital Radio Mondiale) transmission. Ministry of Science, Technology, Innovation and Communication of Brazil is carrying trials with Digital Radio Mondiale standard in order to define the reference model of the Brazilian Digital Radio System (Portuguese: Sistema Brasileiro de Rádio Digital - SBRD).

Keywords: Digital Radio; Digital Radio Mondiale; Middleware; Ginga; NCL

Resumo. O padrão internacional ISDB-T é atualmente adotado pela maioria dos países da América Latina. Para suportar aplicações interativas em receptores de TV digital, o ISDB-T define o middleware Ginga. Similar à TV Digital, padrões de Rádio Digital também fornecem os meios para transportar aplicativos interativos; no entanto, suas especificações para aplicações interativas são geralmente mais restritas da que as usadas na TV Digital. Além disso, aplicações interativas para TV Digital e Rádio Digital são geralmente incompatíveis. Motivado por tais observações, este relatório considera a importância de aplicações interativas para TV e Radiodifusão e as vantagens de usar a mesma especificação de middleware e linguagens para TV Digital e Rádio Digital. Mais especificamente, estabelece a sinalização e as definições de transporte e execução de aplicativos Ginga-NCL e Ginga-HTML5 sobre transmissão DRM (Digital Radio Mondiale). O Ministério da Ciência, Tecnologia, Inovações e Comunicações do Brasil está realizando ensaios com a norma Digital Radio Mondiale, a fim de definir a modelo de referência do Sistema Brasileiro de Rádio Digital (SBRD).

Palavras-chave: Rádio digital; Rádio Digital Mundial; Middleware; Ginga; NCL

In charge of publications:

PUC-Rio Departamento de Informática - Publicações

Rua Marquês de São Vicente, 225 - Gávea

22453-900 Rio de Janeiro RJ Brasil

Tel. +55 21 3527-1516 Fax: +55 21 3527-1530

E-mail: publicar@inf.puc-rio.br

Web site: <http://bib-di.inf.puc-rio.br/techreports/>

Table of Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | On the transport of Ginga applications over DRM | 2 |
| 2.1 | MOT protocol | 2 |
| 3 | Extensions to NCL | 6 |
| 3.1 | The Digital Radio Profile | 6 |
| 3.2 | Extra URI definitions | 7 |
| 4 | Extensions to the NCLua API | 7 |
| 4.1 | The <i>geolocation</i> class | 7 |
| 5 | Ginga Full Receiver Profile | 8 |
| 5.1 | Monomedia support | 8 |
| 6 | Conclusion | 9 |
| | References | 10 |
| | Annex A Auxiliary Data Message | 12 |

1 Introduction

In a Digital Television System, the middleware is the software layer that is part of the receiver and is responsible for receiving and executing (interactive) applications that can be sent by the broadcaster. It must be standardized by the Digital TV System [20] so that any receiver, from any manufacturer, is able to receive and interpret applications.

The Brazilian Digital TV System ¹, which is an evolution of the Japanese ISDB-T, defines the middleware Ginga [21], and its declarative language, NCL (Nested Context Language) [3], to support interactive applications. It is currently adopted by 14 countries in Latin America (the ones in green in Figure 1) and is an ITU-T recommendation for IPTV systems [18]. Indeed, by definition, Ginga is an IBB (Integrated Broadcast-Broadband) system, as specified in ITU-R BT.2267-5 [17]. This means that the middleware permits that applications received via broadcast channel —e.g., broadcast radio applications— use the Internet as a return channel, whenever such support is available in the receiver. Ginga applications are written in NCL (Nested Context Language), which is a declarative, domain-specific language for the description of interactive multimedia presentations. Recently (2018) the Brazilian DTV Forum added HTML5 as a supported application to Ginga (together with NCL), as specified by ABNT 15606-10[1] named Ginga-HTML5, which is a strict subset of W3C HTML specification. Also added to the middleware of the Brazilian DTV in 2018 was the Ginga Common Core WebServices, which allows native applications of the digital receiver ecosystem (eg. Android, WebOS, Tizen) to interoperate with Ginga applications through a REST API, as defined by ABNT 15606-11[2].

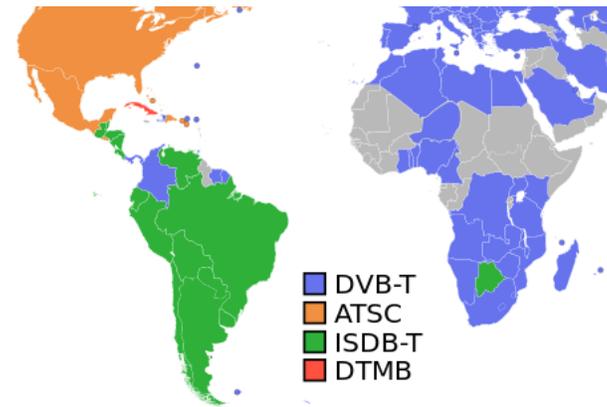


Figure 1: Countries painted in green are the ones where Ginga is used as DTV middleware.

Like in DTV systems, Digital Radio must also permit the transmission of applications. Radio broadcasters have very different coverage, content and audience, ranging from high power public Broadcasters, which cover the whole Amazon region, high power commercial broadcasters, educative and small community radios. Applications will be able to

¹In 2009, the Brazilian DTV System was rebranded as “ISDB-T International” and harmonized with the original ISDB-T. The main differences between the Japanese ISDB-T and ISDB-T International are the audio and video codecs, which were upgraded from MPEG 2 to MPEG 4, and the application middleware, which is a BML (Broadcast Markup Language)-based technology, in case of the original ISDB-T, and Ginga, in case of ISDB-T International. See <http://www.dibeg.org/techp/aribstd/harmonization.html> for more information.

target very different contexts, be it providing public services, traffic information, games, multimedia content related to the audio, advertisement and so on.

This report describes the transport of Ginga NCL and Ginga HTML applications in the Digital Radio Mondiale ² (DRM) system —i.e., it gives a set of definitions that allow for such transport— and discusses minor adaptations to Ginga and NCL that can improve their support to digital radio-specific application requirements. The proposed adaptations are defined as amendments to the 2014 ITU-T H.761 [18] text. Thus, this report may be used as a reference for a possible standardization of the use of Ginga in the DRM system. As a consequence, it is also a contribution to the Brazilian Digital Radio System specification ³.

The remainder of the report is organized as follows. Section 2 discusses proposed signalling extensions to support the transmission and correct interpretation of NCL applications using the DRM transmission structure. Section 3 proposes NCL extensions that are useful in Digital Radio scenarios. Section 4 brings extensions to the NCLua APIs ⁴. Section 5 presents receiver profiles that may be useful in the context of Digital Radio broadcasting. Finally, Section 6 discusses our main conclusions and future work. An annex A also is present in the end, with complimentary information.

2 On the transport of Ginga applications over DRM

This section describes how to multiplex Ginga NCL and Ginga HTML applications over DRM. The multiplex scheme used in DRM is described in the DRM System Standard [8], where also the modulation and channel coding, transmission structure and source coding are defined.

2.1 MOT protocol

The DRM multiplex defines a transmission scheme consisting of three logical channels: the *Main Service Channel* (MSC), the *Fast Access Channel* (FAC), and the *Service Description Channel* (SDC). MSC carries the data for all the DRM Streams in the DRM multiplex; it may contain between one and four DRM Streams, and each DRM Stream may be either audio or data (in this case, a DRM Stream can carry up to 4 sub-stream in Packet Mode). FAC provides information on the channel width and related parameters, and it also provides service selection information, allowing for fast scanning. Finally, SDC gives information on how to decode the MSC and provides descriptors defining the DRM Services within the multiplex.

As aforementioned, DRM defines two type of services (DRM Services): *audio* and *data*. An audio service must be associated with an audio DRM Stream and can be optionally associated with data DRM Streams as *Program Associated Data* (PAD). A data service must be associated with a data DRM Stream.

²DRM is a digital radio broadcasting system standardised for all broadcasting frequencies

³Brazilian Digital Radio System specification, in Portuguese, Sistema Brasileiro de Rádio Digital, was established in March 30, 2010, but has no reference model defined until today. See: <http://www.abert.org.br/web/index.php/legistecnica/item/portaria-n-290-de-30-marco-de-2010>

⁴NCLua is the scripting language supported by Ginga. NCLua scripts can be called a from a NCL application.

In the case of data DRM streams, DRM multiplex provides a *Packet mode*, which defines a generalized way to deliver packetized data. The data stream can be associated by one or more *standalone data service* or *associated to an audio service* as PAD. The mapping between the DRM Services and the DRM Streams is defined by SDC entities. Figure 2 shows an example of multiplex configuration composed of four DRM Services.

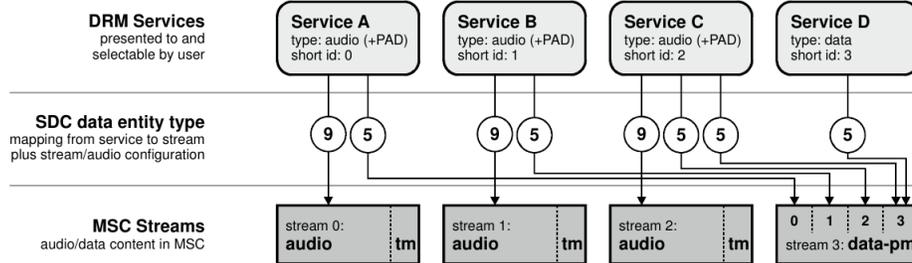


Figure 2: An example of a DRM multiplex configuration composed of three audio services (A-C) using the Program Associated Data and one standalone data service (D)

Data streams can carry interactive applications, to be executed by the receiver.

To signalize the transmission of Ginga applications as standalone data services⁵ in the DRM multiplex, Table 1 proposes a FAC parameter *Application identifier* with value 4—the first available “reserved for future definition” in ETSI TS 101 968 [12], which is the standard that contains the identifiers of DRM applications.

Table 1: FAC Application identifier value for Ginga application transmitted as standalone data service.

| FAC Service Parameter | Value |
|---|-------|
| Service Descriptor (Application identifier) | 4 |

To carry applications composed of a set of files, which is usually the case of an NCL application, DRM uses the DAB MOT protocol [7]. MOT (Multimedia Object Transfer) is a protocol that allows for the transmission of one or more files in a cyclic way (i.e., as a carousel). It is already used in standardized digital radio applications such as SlideShow [10] and Broadcast Website [9]. Similarly, the files that are part of an NCL application may also be carried as MOT objects.

MOT objects are segmented in DAB MSC data groups. Data groups are mapped directly to DRM data units. As detailed in Chapter 5.2 of ETSI TS 101 968 [12], DRM data units are subsequently split into packets that are transported by the DRM Packet mode protocol. To reference an NCL application, the *Application information* parameters, which identifies an application in the SDC and associates the application to a service, should be set to those presented in Table 2. The first two parameters, *Packet Mode Indicator* and *Data Unit Indicator*, are required by the MOT protocol; an application domain with value 0 indicates a DRM application, and the user application domain with the

⁵We assume that most applications will be transmitted as PAD of audio service, and not as standalone data service.

proposed value 0x0001 indicates an NCL application. Note that 0x0001 is the first available application identifier value for openly specified applications in ETSI TS 101 968 [12].

Table 2: SDC Application information parameter values for NCL applications transmitted using MOT protocol.

| Application information parameters | Value |
|------------------------------------|--------|
| Packet mode indicator | 1 |
| data unit indicator | 1 |
| application domain | 0 |
| user application identifier | 0x0001 |

Since Ginga expects that the NCL application files are organized into a directory tree, the MOT protocol’s Directory Mode must be used. The *Directory Mode* provides support for transmitting many files organized into a directory tree with the possibility of transmitting them in an interleaved way. The mandatory *DirectoryExtension* parameter is presented in Table 3. The *DirectoryExtension* parameter contains information that apply to the whole MOT transmission.

Table 3: Mandatory MOT DirectoryExtension parameter.

| Parameter Id | Parameter |
|---------------|----------------|
| 0x22 (100010) | DirectoryIndex |

The syntax of *DirectoryIndex* parameter’s data field is shown in Table 4.

Table 4: Syntax of DirectoryIndex parameter’s data field.

| Syntax | Size ⁶ |
|---|-------------------|
| DirectoryIndex_parameter_data_field() { | |
| profile_id | 8 bits |
| for (i=0;i<N;i++) { | |
| entry_point_byte | 8bits |
| } | |
| } | |

The *DirectoryIndex* parameter indicates the application entry point (*entry_point* field) for a given receiver profile (*profile_id* field). One may insert more than one *DirectoryIndex* parameter in order to signalize different entry points for distinct receiver profiles. The *entry_point* is composed of *entry_point_byte* fields, which must be ISO/IEC 10646 [15] characters (using UTF-8 transformation format), being the hash sign (‘#’) a reserved character. The entry point must follow one of the possible syntaxes expressed in Table 5. In the first syntax, one specifies a NCL or HTML application to be started, while in the second syntax, one specifies both the NCL file and a specific interface (<port>) to be started. Note that in the second syntax the file name and the port identifier must be separated by a hash sign (‘#’). In both syntaxes the file name must be a relative path, i.e., one not starting with character ‘/’, e.g., “code/main.ncl” or “code/index.html”.

Table 5: Entry point syntax.

| Entry point syntax | Description |
|--|--|
| {application_filename}.{ncl,html} | The middleware should use the NCL or HTML application as entry point according to file name extension. |
| {application_filename}.ncl#{InterfaceId} | The middleware should use port <i>InterfaceId</i> to start the NCL application <i>application_filename.ncl</i> . |

Another optional *DirectoryExtension* parameters that must be supported by the Ginga middleware running in the receiver are *SortedHeaderInformation*, *DefaultPermitOutdatedVersions*, and *DefaultExpiration*. Their semantics is the same specified in the MOT standard [7].

The MOT protocol contains parameters related to the individual files, transmitted by the MOT structures. All files transmitted over the MOT have, among other parameters, two mandatory parameters with file type information: *ContentType* and *ContentSubType*⁷. The Ginga middleware must ignore these values, as they do not specify all supported file types supported by Ginga. The recommended values for these fields are shown in Table 6.

Table 6: Recommended values for ContentType and ContentSubType.

| Field | Value |
|----------------|-------|
| ContentType | 0 |
| ContentSubType | 0 |

Moreover, the MOT protocol defines optional parameters for each file. These parameters are defined in the *Header extension* part of the protocol header and their use is specified in Table 7.

The *ContentName* parameter specifies a relative path for each file that composes an application (e.g., “media/pic.jpg”) and the *CompressionType* specifies compression algorithm used to compress the file, when this is transmitted compressed.

Other parameters can be optionally present in the Header extension. The optional parameters that must be correctly interpreted by the Ginga middleware are the following: *PermitOutdatedVersions*, *Expiration*, and *TriggerTime*. The semantics of these parameters is specified in the MOT standard [7].

⁷The possible values specified in Table 17 of ETSI TS 101 756 [11]

Table 7: Header extension parameters.

| Identifier | Parameter | Content |
|---------------|-----------------|---|
| 0x0C (001100) | ContentName | Contains the character set indicator, which must be set to ISO/IEC 10646 as specified in Table 19 of ETSI TS 101 756 [11] (value 1111b), and the file name of the content, which must use a relative path. This parameter is mandatory. |
| 0x11 (010001) | CompressionType | Must be used when a file is transmitted compressed. The only allowed compression is GZip (value 0x01) as specified in Table 18 of ETSI TS 101 756 [11]. The middleware must support GZip decompression. |

3 Extensions to NCL

The use of NCL in the radio context is similar to (and most time compatible with) its use in DTV or IPTV contexts. However, there are some differences that are explained in this section. Motivated by these differences, the remainder of this section propose an NCL profile to Digital Radio.

3.1 The Digital Radio Profile

The Digital Radio (DR) NCL 3.1 profile is based on the ITU H.761 NCL 3.1 Enhanced DTV (EDTV) profile, with the following differences:

- *Transition* and *TransitionBase* modules of NCL are not defined in the DR profile. The `<transition>` and `<transitionBase>` elements are not defined, nor it is allowed to reference transition via `<property>` element;
- *clip* and *coords* attributes of the `<area>` element are not defined.
- the properties *transIn*, *transOut* and *plane* are not defined to be used in the element `<property>`.
- global variables in NCL are defined as special properties of a media with type equals to *application/x-ncl-settings* type. The following variables of the system group are not defined: *screenVideoSize*, *screenBackgroundSize*, *screenGraphicSize* and *screenGraphicSize(i)*. This is because different graphic planes are supported in DR profile.

- In the *si* group of local variables, new variables are introduced in the DR profile: *stationLabel*, *numberOfServices*, *channelFrequency*, *signalQuality*, and *serviceDecoding*. *stationLabel* contains the Broadcaster label. *numberOfServices* contains the number of services present in the received signal, *signalQuality* contains the Modulation Error Ratio (MER), in decibels (only positive values, 0 meaning no signal). And, *serviceDecoding* is a boolean value, *True* or *False*; being *True* when the Bitrate Error Rate (BER) is less than 10^{-4} , meaning Quasi Error Free (QEF) reception.
- the metadata group of local variables are not defined.

3.2 Extra URI definitions

The *dsm-cc:* and *ts:* URI schemes are not supported in the Ginga DR profile, and the scheme *drm:* defined in ETSI TS 103 270 [14], Table 9 (DRM parameter description) and Table 12 (Example of RadioDNS bearerURI construction for DRM) shall be supported.

4 Extensions to the NCLua API

NCLua is the set of APIs that supports the integration between NCL documents and the Lua scripting language. NCLua is standardized in [18, 3], and provides two main modules: *canvas*, which supports 2D drawing primitives; and *event*, which provides the event-based communication between Lua and NCL, and general event handling.

The Lua version 5.3 must be used to implement the NCLua player in the Ginga middleware for radio.

4.1 The *geolocation* class

Given the mobile nature of Digital Radio, location-aware applications play an important role in this context. Aware of such a feature, and due to the lacking of similar APIs in the DTV contest, this report proposes to extend the NCLua event API with a “geolocation” class. Geolocation events carry the global position, speed, and heading of the receiver—of course, if the receiver can resolve such queries.

An NCLua script requests the geolocation information of the receiver by posting an event of the form:

```
evt = { class = 'geolocation', [timeout=number] }
```

timeout is an optional field that, if present, determines the timeout (in seconds) for the query to be successfully completed.

An NCLua script receives the answer of geo-location request as event of the form:

```
evt = { class = 'geolocation',
        [latitude=number], [longitude=number],
        [accuracy=number], [speed=number], [heading=number] }
```

5 Ginga Full Receiver Profile

Recent tabletop receivers, media centers, TV with radio tuner, infotainment automotive receivers, and mobile phones with radio support can easily run the Ginga middleware and media decoders.

At least a touch screen or keypad with directional keys plus “ENTER” key is required as interface to the receiver. A “BACK” key is desired in a receiver with keypad.

No inferior or superior screen resolution and size are defined. It’s recommended for the receivers to have at least a 320x240 pixels color screen.

As the definition of receiver profiles depends on many industrial production aspects, just one receiver profile will be defined, leaving room for future definitions like supported resolutions for images and audio encoder features. This sole profile is named “Ginga Full Receiver Profile”, with *profile_id* in Table 4 equal to 1. A “Ginga Full Receiver Profile” compatible receiver shall be compatible with all the definitions contained in this document with regards to Ginga and NCL.

5.1 Monomedia support

The media types in Table 8 must be supported by a “Ginga Full Receiver Profile” compatible receiver.

Table 8: “Ginga Full Receiver Profile” supported media types.

| Category | Media Type | MIME Type | Extension(s) |
|-----------------|---|----------------------------|--------------|
| Image | PNG | image/png | png |
| | JPEG | image/jpeg | jpg, jpeg |
| | HEIF (HEVC codec) | image/heic | heif, heic |
| MPEG Audio | Supported encoder/profiles are: AAC Profile, Level 4, High Efficiency AAC Profile, Level 4, High Efficiency AAC Profile v2, Level 4, as defined in ISO/IEC 14496-3, and Extended HE AAC Profile, Level 4, as defined in ISO/IEC 23003-3, in both 960 and 1024 transform length. | audio/mp4 | mp4, mpeg4 |
| Vector Graphics | SVG Tiny 1.2 | image/svg+xml | svg, svgz |
| Voice synthesis | SSML 1.1 | application/ssml+xml | ssml |
| Text | Plain text | text/plain | txt |
| Application | ginga-NCL | application/x-ginga-NCL | ncl |
| | ginga-NCLua | application/x-ginga-NCLua | lua |
| | ginga-HTML5 | text/html | html |
| | ncl-settings | application/x-ncl-settings | - |
| | ncl-time | application/x-ncl-time | - |

Beyond the media types presented in Table 8, also Journaline [13] presentation shall be supported.

6 Conclusion

This report aims to contribute to the proper implementation of sound and multimedia broadcasting in Latin America region. The idea of a harmonized middleware for both Digital TV and Digital Radio in the region makes sense as we expect devices like TV sets, cellphones and automotive infotainment systems to have both Digital TV and Radio (ISDB-T and DRM) tuners in the same device.

References

- [1] ABNT NBR 15606-10. *Digital Terrestrial TV — Data Coding and Transmission Specification for Digital Broadcasting — Part 10: Ginga-HTML5 - Ginga HTML5 profile specification*. Brazilian National Standards Organization (ABNT), Brasília, DF, Brazil, October 2018.
- [2] ABNT NBR 15606-11. *Digital Terrestrial TV — Data Coding and Transmission Specification for Digital Broadcasting — Part 11: Ginga CC WebServices - Ginga Common Core WebServices specification*. Brazilian National Standards Organization (ABNT), Brasília, DF, Brazil, October 2018.
- [3] ABNT NBR 15606-2. *Digital Terrestrial TV — Data Coding and Transmission Specification for Digital Broadcasting — Part 2: Ginga-NCL for Fixed and Mobile Receivers: XML Application Language for Application Coding*. Brazilian National Standards Organization (ABNT), Brasília, DF, Brazil, March 2015.
- [4] ABNT NBR 15610-3. *Digital Terrestrial TV — Accessibility — Part 3: Brazilian Sign Language (LIBRAS)*. Brazilian National Standards Organization (ABNT), Brasília, DF, Brazil, March 2016.
- [5] ARAÚJO, T. M. U., FERREIRA, F. L. S., SILVA, D. A. N. S., LEMOS, F. H., NETO, G. P., OMAIA, D., FILHO, G. L. S., AND TAVARES, T. A. Automatic generation of Brazilian sign language windows for digital TV systems. *Journal of the Brazilian Computer Society* 19, 2 (Sept. 2012), 107–125. 00003.
- [6] ETSI EN 300 401. *Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers; Version 1.4.1*. European Telecommunications Standards Institute, Nice, France, June 2006.
- [7] ETSI EN 301 234. *Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) protocol; Version 2.1.1*. European Telecommunications Standards Institute, Nice, France, May 2006.
- [8] ETSI ES 201 980. *Digital Radio Mondiale (DRM); System Specification; Version 4.1.1*. European Telecommunications Standards Institute, Nice, France, January 2014.
- [9] ETSI TS 101 498. *Digital Audio Broadcasting (DAB); Broadcast website; Part 1: User application specification; Version 2.1.1*. European Telecommunications Standards Institute, Nice, France, January 2006.
- [10] ETSI TS 101 499. *Hybrid Digital Radio (DAB, DRM, RadioDNS); SlideShow; User Application Specification; Version 3.1.1*. European Telecommunications Standards Institute, Nice, France, January 2015.
- [11] ETSI TS 101 756. *Digital Audio Broadcasting (DAB); Registered Tables; Version 1.8.1*. European Telecommunications Standards Institute, Nice, France, December 2015.
- [12] ETSI TS 101 968. *Digital Radio Mondiale (DRM); Data applications directory; Version 1.3.1*. European Telecommunications Standards Institute, Nice, France, April 2009.

- [13] ETSI TS 102 979. *Digital Audio Broadcasting (DAB); Journaline; User application specification; Version 1.1.1*. European Telecommunications Standards Institute, Nice, France, June 2008.
- [14] ETSI TS 103 270. *RadioDNS Hybrid Radio; Hybrid lookup for radio services; Version 1.2.1*. European Telecommunications Standards Institute, Nice, France, September 2015.
- [15] ISO/IEC 10646. *Information technology – Universal Coded Character Set (UCS)*. International Organization for Standardization, Geneva, Switzerland, June 2014.
- [16] ISO/IEC 13818-6. *Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC*. International Organization for Standardization, Geneva, Switzerland, June 1998.
- [17] ITU-R REPORT BT.2267-5. *Integrated broadcast-broadband systems*. International Telecommunication Union (ITU), Geneva, Switzerland, July 2015.
- [18] ITU-T RECOMMENDATION H.761. *Nested Context Language (NCL) and Ginga-NCL*. ITU Telecommunication Standardization Sector, Geneva, Switzerland, November 2014.
- [19] MORENO, M. F., DE RESENDE COSTA, R. M., AND SOARES, L. F. G. Interleaved time bases in hypermedia synchronization. *IEEE MultiMedia Magazine* 22, 4 (November 2015), 68–78.
- [20] MORRIS, S., AND SMITH-CHAIGNEAU, A. *Interactive TV Standards: A Guide to MHP, OCAP, and JavaTV*, 1 edition ed. Focal Press, Amsterdam ; Boston, Apr. 2005. 00268.
- [21] SOARES, L. F. G., MORENO, M. F., DE SALLES SOARES NETO, C., AND MORENO, M. F. Ginga-NCL: Declarative middleware for multimedia IPTV services. *IEEE Communications Magazine* 48, 6 (June 2010), 74–81.
- [22] SOARES, L. F. G., RODRIGUES, R. F., COSTA, R. R., AND MORENO, M. F. Nested Context Language 3.0 Part 9 – NCL Live Editing Commands, 2006.

Annex A Auxiliary Data Message

The MOT protocol provides most of the features required for transporting NCL applications to be executed by Ginga-ready client receivers. However, there are three important missing features:

1. Transmission and maintenance of independent time bases [19];
2. Transport of live editing commands [22]; and
3. Transport of gloss symbols of sign languages, which are used to present sign language symbols in the client receiver, for hearing impaired users.

The DRM standard already provides means to transmit an absolute clock, but it does not support independent time bases. An independent time base is especially important if one wishes to achieve a fine synchronization between the main audio content and events in the application, independently from the absolute clock.

Ginga editing commands provide support for the live adaptation of a running application, such as changing some application structures, start and stop medias, etc.

Finally, the transport of gloss symbols permits the synthesis of sign language symbols in the receiver, which is especially important for hearing impaired users [5], and it is also a requirement of the Brazilian DTV standard [4].

In DTV, the above data are transmitted using the Stream Events of DSM-CC [16]. However, DSM-CC is not available in Digital Radio scenario, since it is huge overhead for the available bandwidth.

In order to address the transmission of such features, new types of messages must be defined to provide a generic mean to transmit data that does not fit in the data carousel model of the MOT protocol. Auxiliary Data Message (ADM) is defined in this subsection for this purpose.

ADM is carried in the same DRM data stream where NCL applications are carried, but using different *Data group type* on the *MSC data group*, also used by the MOT protocol, and defined in the DAB standard [6]. *Data group type* has a 4 bit size (0 up to 15).

MOT protocol already uses the 4-bit *Data group type* values 3 (MOT header), 4 (MOT body), 5 (scrambled MOT body and CA parameters), 6 (uncompressed MOT directory) and 7 (compressed MOT directory). DAB standard defines *Data group type* values 0 (General data) and 1 (CA messages).

The ADM types use different *Data group type* values. They are listed in table 9.

Table 9: *Data group type* values for ADM types.

| <i>Data group type</i> value | ADM type |
|------------------------------|------------------------|
| 10 | TimeBase message |
| 11 | EditingCommand message |
| 12 | SignLanguage message |

The *MSC data group* structure is defined in Chapter 5.3.3 of DAB standard [6]. In order to minimize the overhead, as the *MSC data group* has a variable header length, it's recommend to use *Extension flag* equal to 0, *Segment flag* set to 0 and also *User access flag* 0. In order to improve robustness, CRC flag equal to 1. The other header fields, which are

Continuity index and *Repetition index* must follow the standard, apart from *Data group type*, which must follow values in table 9. In such configuration, *MSC data group* will have a 2 byte header in the beginning, and a 2 byte CRC field in the end. The payload part of *MSC data group*, called *MSC data group data field* occupies all the rest of the data structure.

Considering the *MSC data group* is mapped directly to a *DRM data unit*, and the header size plus CRC size is 4 bytes, and the maximum payload size carried by a single *MSC data group* is 8191 bytes, an ADM is limited to 8187 bytes.

A.1 TimeBase message

TimeBase messages are used when there is a need the synchronize an audio content with events in an application.

Table 10 shows the payload of ADM for the TimeBase type. A TimeBase message has three fields: *Status*, *DiscontinuityIndicator* and *TimeBaseValue*.

Table 10: ADM TimeBase syntax.

| TimeBase field | Possible values | Size |
|-------------------------|-------------------------|---------|
| Status | 0 = Running, 1 = Paused | 1 bit |
| DiscontinuityIndicator | 0 = No, 1 = Yes | 1 bit |
| Reserved for Future Use | - | 5 bits |
| TimeBaseValue | Value | 33 bits |

The *Status* field indicates if the time base is running (value equals to 0) or paused (value equals to 1). When a time base is running, the receiver must constantly increment its current value. If a time base is running and a message containing a status equal to *paused* is transmitted, the receiver must stop incrementing the time base in the indicated *TimeBaseValue* and keep the time base paused. When the time base is paused and a message containing a status equal to *Running* is received, the receiver must start increasing the time base.

The current value of a time value—i.e. the *TimeBaseValue* field—is a 33 bit value that must be updated at audio super frame cadence (each super frame sums 1000 to the *TimeBaseValue*, for improved time resolution). The *TimeBaseValue* update is driven by the receiving super frames. When the last packet that completes a DRM Data Unit containing a TimeBase message arrives in a given DRM super frame, the *TimeBaseValue* indicates the moment when the first audio sample, from the first complete audio frame present in the super frame is played.

A message containing the *DiscontinuityIndicator* set to *Yes* means that a leap in time has occurred, and no application events must be triggered when a leap occurs. No time leaps should occur without *DiscontinuityIndicator* set.

The receiver must compensate internal clock drifts or reception drop-outs with the values that comes in the ADM TimeBase. If there are small differences between the internal receiver time base value and the time base value received in TimeBase messages, the receiver must use elastic time compensation and not rewind the internal time base value. The only case where a time base value goes back is when it reaches it greatest

value ⁸ and wrap around, or when the DiscontinuityIndicator is set.

From an NCL application, one must reference a time base value using a time base value followed by “tbv” suffix, like the example below:

```
<area id="anchor" first="5000tbv"/>
```

A.2 EditingCommand message

Table 11 shows the payload of ADM for the EditingCommand type. The syntax of the EditingCommand is composed the field *EventId*, *DoItNow*, *TimeBaseValue*, *CommandTag* and *CommandPayload*. *EventId* is a unique event identifier for the command transmission.

DoItNow flag indicates if the command should be executed immediately at arrival in the receiver or not.

TimeBaseValue contains the time base value the receiver should execute the command if the *DoItNow* flag is set to 0.

CommandTag indicates the command type and *CommandPayload* contains the command payload itself. The *CommandPayload* content must follow the same rules ITU H.761; the only difference is that the time values should be interpreted as TBV (Time Base Value) instead of NPTs (Normal Play Time).

Table 11: ADM EditingCommand syntax.

| EditingCommand field | Possible values | Size |
|-------------------------|-----------------------------|---------|
| EventId | Event Identifier | 16 bits |
| DoItNow | 0 = No, 1 = Yes | 1 bit |
| Reserved for Future Use | - | 6 bits |
| TimeBaseValue | Moment to execute the event | 33 bits |
| CommandTag | The command identifier | 8 bits |
| CommandPayload | The command payload | N bytes |

A.3 SignLanguage message

The proposed support for sign language messages is based on the Brazilian DTV standard ABNT NBR 15610-3 [4], named LibrasTV ⁹.

When a radio receiver receives sign language messages in a data stream associated with the active service, and if the sign language playback option is enabled in the native receiver software, the sign language must be synthesized in the screen.

In DRM, the sign language messages are transported in ADM with minor differences to the DTV standard:

- the *eventNPT* fields should be understood as *eventTBV*, with the time base semantics specified in this text;
- the *StreamEventDescriptor* should be used as ADM payload, with the same syntax, except for the removal of the *descriptorTag* and *descriptorLength* fields;

⁸A broadcaster must avoid time base value to wrap around, restarting the time base each radio program

⁹Libras is the Brazilian sign language

- in the *libras_content_type* field of the sign language message, the mode 0x01 cannot be used, as video transmission over DRM takes too much bitrate;
- all the DTV specific fields should be ignored by the receiver.