

On the transport of NCL Applications over the Digital Radio Mondiale Broadcasting System for Ginga enabled receivers: Signaling and Definitions

RAFAEL DINIZ
Lab. TeleMídia, PUC-Rio
rafaeldiniz@telemidia.puc-rio.br

February 18, 2016

Abstract

Ginga is the middleware defined in ISDB-T International to provide support for running NCL applications in Digital TV receivers. It is adopted by the majority of Latin America countries, and is already installed in most TV sets sold in recent years in the region. Just like in Digital TV, Digital Radio provides the means to carry applications. This report considers the importance of interactive applications for TV and Radio Broadcasting in the region and the importance of using the same language and middleware for both Digital TV and Radio. It is also considered that Brazil is carrying trials with Digital Radio Mondiale standard in order to evaluate the best system to be used as reference for the Brazilian Digital Radio System. This report establishes the signaling and definitions on how to transport NCL applications over a Digital Radio Mondiale transmission, and how the application should be executed in the receiver.

Keywords. interactive applications; digital radio; Digital Radio Mondiale; Multi-media Object Transfer; middleware; NCL.

Contents

1 Introduction	2
2 On the transport of NCL applications over DRM	4
2.1 Auxiliary Data Stream	8
2.1.1 TimeBase message	10
2.1.2 EditingCommand message	11
2.1.3 SignLanguage message	12
3 Extensions to NCL	12
3.1 The Digital Radio Profile	12
3.2 Extra URI definitions	13
4 Extensions to the NCLua API	13
4.1 The <i>geolocation</i> class	13
5 Receiver Profile	14
5.1 Monomedias support	14
5.2 Screen Resolution	15

1 Introduction

Ginga is the middleware adopted by the Brazilian Terrestrial Digital TV System. The Brazilian DTV System, which is an evolution of the original Japanese ISDB-T, 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 BML¹-based technology, in case of the original ISDB-T, and Ginga, in case of ISDB-T International. A television set running Ginga is shown in Figure 1.

¹BML: Broadcast Markup Language



Figure 1: Standard TV running an application on Ginga middleware.

Ginga is defined in the ITU-T H.761 [13] and ABNT NBR 15606-2 [1] standards. By definition, Ginga is an IBB (Integrated Broadcast-Broadband) system, as specified in ITU-R BT.2267-5 [12]. 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.

Ginga is currently adopted by 14 countries. These Latin American countries are depicted in green in Figure 2.

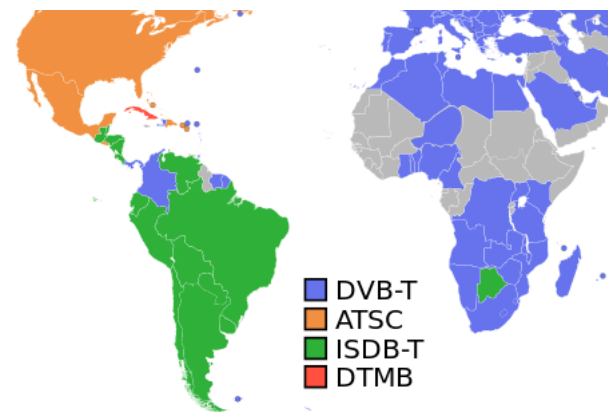


Figure 2: Countries painted in green are the ones where Ginga is used as DTV middleware.

This report describes the transport of NCL applications in the Digital Radio

Mondiale ² (DRM) system, i.e., it gives a set of definitions that allow for such transport. And it also discusses some adaptations to the middleware itself (Ginga) and its application language (NCL) that improve their support to digital radio-specific application requirements. These adaptations are defined as amendments to the 2014 ITU-T H.761 [13] text. The main purpose of this document is to serve as a reference for a possible standardization of the use of Ginga in the DRM system, and consequently to serve as a contribution to the Brazilian Digital Radio System specification ³.

2 On the transport of NCL applications over DRM

The Digital Radio Mondiale (DRM) system specification [4] contains the modulation and channel coding, transmission structure, source coding and multiplex definitions.

The DRM multiplex definition establishes three channels: Main Service Channel (MSC), Fast Access Channel (FAC), and Service Description Channel (SDC). The MSC contains the streams, more precisely called MSC Streams, of the services. The FAC provides information on the channel width and related parameters, and it also provides service selection information to allow for fast scanning. Finally, the SDC gives information on how to decode the MSC and gives the attributes of the services within the multiplex.

From a service perspective, DRM defines two type of services: audio service and data service. An audio service must be associated with an audio stream and optionally can be associated to Program Associated Data (PAD) streams. A data service must be associated with a Data stream.

In an interactive radio context, the application is transmitted via a data stream, and must be associated with an audio service, as PAD, or associated with a standalone data service. The mapping between services and streams is defined by Service Descriptor Channel (SDC) entities. Figure 3 shows an example of multiplex configuration with four services.

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.mc.gov.br/portarias/25477-portaria-n-290-de-marco-de-2010>

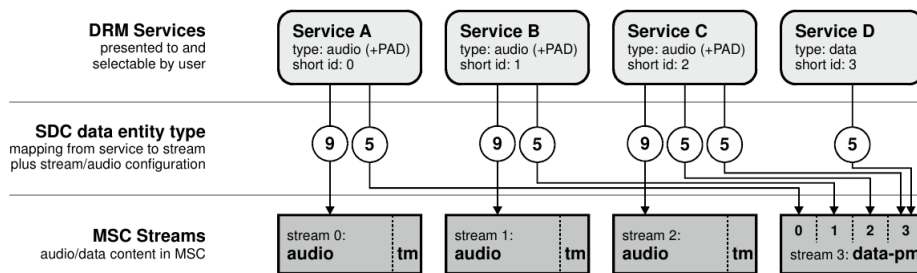


Figure 3: DRM multiplex configuration example with three services with Program Associated Data and one data service.

The DRM multiplex system provides a *Packet mode* for data streams, which defines a generalized way to deliver packetized data.

If an NCL application is broadcasted as standalone data service⁴, the proposed FAC parameter Application identifier is presented in Table 1. This value is the first available “reserved for future definition” in ETSI TS 101 968 [8], which is the standard that contains the identifiers of DRM applications.

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

FAC Service Parameter	Value
Service Descriptor (Application identifier)	4

To carry an application composed of a set of files, which is normally the case of an NCL application, DRM uses the DAB MOT protocol [3]. MOT (Multimedia Object Transfer) is a protocol that allows for the transmission of one or more files in a cyclic way. It is already used in standardized digital radio applications such as SlideShow [6] and Broadcast Website [5].

The files of an NCL application may be carried as MOT objects. MOT objects are segmented in DAB MSC data groups, which are mapped directly to DRM data units, that are subsequently split in packets that are transported by the DRM Packet mode protocol, as detailed in Chapter 5.2 of ETSI TS 101 968 [8]. To reference an NCL application, the Application information parameters, which identify 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 requirements of the MOT protocol; an application domain with value 0 indicates a DRM application, and the user application domain with the proposed value 0x0001 indicates an NCL application. Note that 0x0001 the first available application identifier value for openly specified applications in ETSI TS 101 968 [8].

⁴We assume that most applications will be transmitted as PAD of audio service, and not as standalone data service.

Table 2: SDC Application information parameter values for NCL applications.

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

Since NCL assumes that the application files are organized in a directory tree, the MOT protocol’s Directory Mode must be used. The Directory Mode provides support for transmitting many files organized in 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 ⁵
DirectoryIndex_parameter_data_field() {	
profile_id	8 bits
for (i=0;i<N;i++) {	
entry_point_byte	8bits
}	
}	

The DirectoryIndex parameter indicates to the Ginga middleware the application entry point for a given receiver profile, defined in the `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 [10] characters (using UTF-8 transformation format), being the comma (’,’) a reserved character. The entry point must follow one of the possible syntaxes expressed in Table 5. In the first syntax, one specifies the NCL file to be started, while in the second syntax, one specifies both the NCL file and a specific port to be started. Note that in the second syntax the file name and the port identifier must be separated by a comma (’,’). Furthermore, in both syntaxes the file name must be a relative path, i.e., one not starting with character ‘/’, e.g., “code/main.ncl”.

Table 5: Entry point syntax.

Entry point syntax	Description
{application_filename}.ncl	The middleware should start the NCL file <i>application_filename.ncl</i> .
{application_filename}.ncl,{InterfaceId}	The middleware should start port <i>InterfaceId</i> of the NCL file <i>application_filename.ncl</i> .

Other 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 [3].

The MOT protocol contains parameters which are 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. These two mandatory parameters are the `ContentType` and `ContentSubType`, whose possible values specified in Table 17 of ETSI TS 101 756 [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
<code>ContentType</code>	0
<code>ContentSubType</code>	0

The MOT protocol defines optional parameters for each file. These parameters go in the Header extension part of the protocol header and their use is specified in Table 7.

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 [7] (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 [7]. The middleware must support GZip decompression.

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 [3].

2.1 Auxiliary Data Stream

The MOT protocol provides most of the features required for the transport of NCL applications that will be executed by Ginga installed in client receivers. However, three important features are missing: (1) transmission and maintenance of independent time bases [14]; (2) transport of live editing commands [15]; 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. In DTV, for example, these data are transmitted using the Stream Events of DSM-CC [11].

The DRM standard already provides means to transmit the absolute clock, but

not an independent time base. 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, in a manner that is independent 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, and it is also a requirement of the Brazilian DTV standard [2].

The Auxiliary Data Stream (ADS) is a generic mean to transmit data that does not fit in the data carousel model of MOT protocol. ADS uses the Data Unit structure of the DRM system, and must be associated with an audio service or a data service of a NCL application. The ADS must not be transmitted as a standalone service, so there is no FAC identifier for it. The SDC parameters and ADS proposed user application identifier is shown in table 8.

Table 8: SDC Application information parameters values for the Auxiliary Data Stream.

Application information parameters	Value
Packet mode indicator	1
data unit indicator	1
application domain	0
user application identifier	0x0002

The ADS is composed by messages, in which the payload can be at most 8191 bytes, plus 2 bytes of header. ADS messages are transported by one DRM Data Unit.

An ADS message is composed by a 2 bytes header and by the message payload, as shown in table 9,

Table 9: ADS message syntax.

Field	Size	Possible Values ⁶
ADS message type	3 bits	See table 10.
ADS message size	13 bits	0 up to 8191 bytes
ADS message payload	N bytes	Dependent on message type.

The ADS message size indicates the size of the message payload, starting after the size field.

The ADS message type identifiers are shown in table 10.

Table 10: ADS message type identifiers.

ADS message type	Identifier
TimeBase	0
EditingCommand	1
SignLanguage	2
Reserved for future alocation	3 up to 7

2.1.1 TimeBase message

The ADS message payload for the TimeBase message type are shown in table 11.

The message has three fields: Status, DiscontinuityIndicator and the TimeBaseValue.

The TimeBaseValue is a 33 bit value, with a 48kHz cadence, driven by the audio that comes in the DRM 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 audio frame is played.

When transmitting an audio of, for example, 24kHz of sample rate, the internal time base counter should increment 2 units each audio sample, with 32kHz, each 2 audio samples implies in 3 units of time base increase, with 16kHz audio, 1 sample implies in 3 units of time base increase, and with a 48kHz audio, each sample played is equivalent to one unit increase in the TimeBaseValue, and so on.

When the audio frames are not aligned to super frame boundaries, which is the case for the xHE-AAC codec, the first audio sample to be considered must be the sample from the first audio frame as if no *bit reservoir* mechanism was being applied. The first sample to be considered must be based in the medium number of audio frames per super frame, which is constant, or in the *bit reservoir* information. When using the HE-AAC codec, with which the audio frames are aligned to the super frames boundaries, no extra step is needed.

The Status field, which can be Running or Paused, means that the time base is being incremented (Running), or Paused, in a given TimeBaseValue. When the time base is running and a message containing a status equal to paused is transmitted, the receiver must stop to increment the time base in the indicated TimeBaseValue and keeps the time base paused. When the time base is paused and a message containing a status equal to Running is sent, the receiver must start to increase the time base.

A message containing the DiscontinuityIndicator set to Yes means that a leap in time 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 ADS TimeBase message. 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 its greatest value⁷ and wrap around, or when the DiscontinuityIndicator is set.

Table 11: ADS TimeBase message syntax.

ADS 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

From a 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="192000tbv"/>
```

The time base period (time to wrap around) is the following:

$$TBPeriod = ((2^{33}/48000Hz)/(3600s/h))/(24h/day) = 2days, 1h, 42min.$$

In order to convert a time base value to seconds and milliseconds, the following formulas apply:

$$TB_seconds = TBV/48000Hz$$

$$TB_milliseconds = ((TBV * 1000)/48000Hz) - (TB_seconds * 1000)$$

2.1.2 EditingCommand message

The ADS message payload for the EditingCommand message type are shown in table 12. The syntax of the EditingCommand is composed of the fields EventId, DoItNow, TimeBaseValue, CommandTag and the command payload.

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 carries the command type and CommandPayload contains the command payload itself, just like specified in ITU H.761, but the time values in the standard should be read as TBV (Time Base Value) instead of NPT (Normal Play Time).

⁷A broadcaster must avoid time base value to wrap around, restarting the time base each radio program

Table 12: ADS EditingCommand message syntax.

ADS 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 (ADS message size minus 8 bytes)

2.1.3 SignLanguage message

The support for sign language messages for hearing impaired people is based in the Brazilian DTV standard ABNT NBR 15610-3 [2], and is transported in ADS messages with minor differences to the DTV standard, called LibrasTV ⁸.

The fields eventNPT should be understood as eventTBV, with the time base semantics specified in this text.

The StreamEventDescriptor should be used as ADS message payload, with the same syntax, except for the removal of the descriptorTag and descriptorLenght fields. The correct length should be written in the ADS message size field respecting the ADS message format.

In libras_content_type field of the sign language message, the mode 0x01 can not be used, as video transmission over DRM takes too much bitrate.

All the DTV specific fields should be ignored by the receiver.

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.

mandar isso para o Maritã

3 Extensions to NCL

The use of NCL in the radio context is similar then in DTV or IPTV, and most of times compatible. But there are some differences that are explained in this section.

3.1 The Digital Radio Profile

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

⁸Libras is the Brazilian sign language

- The Transition and TransitionBase modules of NCL are not defined in the DR profile. The elements <transition> and <transitionBase> are not defined, nor it's allowed to reference transition via <property> element.
- The 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 the application/x-ncl-settings type media. The following variables of the system group are not defined: screenVideoSize, screenBackgroundSize, screenGraphicSize and screenGraphicSize(i). This is because no 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 label of the Broadcaster, 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 contains 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 DR profile, and the scheme *drm:* defined in ETSI TS 103 270 [9], Table 9 (DRM parameter description) and Table 12 (Example of RadioDNS bearerURI construction for DRM) should be supported.

4 Extensions to the NCLua API

The NCLua is a set of APIs that add support to 2D drawing and general event handling to Lua, which is the standard scripting language of Ginga-NCL. NCLua is standardized in [13, 1].

The Lua version 5.3 must be used to implement the NCLua player in the Ginga middleware for radio, as opposed to Lua 5.1 defined in the ITU H.761, but compatibility to Lua 5.1 must be retained.

4.1 The *geolocation* class

The mobile nature of radio reception implies that the access by an application to geo-location data is important. This report presents an extension to NCLua, called

NCLua for Digital Radio, which provides an API that allows for applications to access geo-location information of the receiver. In practice, a new event class, called “geolocation”, is added to the NCLua Event API. And events of this class may carry the global position, speed, and heading of the receiver—of course, if the receiver can resolve such queries.

An NCLua application request the geolocation information by posting an event in the form:

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

The *timeout* field is optional and if present, must contain a number which determines the timeout (in seconds) for the query to be successfully completed.

An NCLua application receives the response of geo-location request in an event of the form:

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

Tem que detalhar a sintaxe e semântica de cada um dos campos e dizer o que acontece em caso de erro. Eu posso fazer isso. Para quando você precisa disso?

5 Receiver Profile

Recent tabletop receivers, media centers, TV with radio tuner, infotainment automotive receivers and mobile phones with radio support will easily be able to 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.

As the definition of receiver profiles depends on many industrial production aspects, just one receiver profile will be defined, leaving room for future definitions. This sole profile is the “Full Profile”, with *profile_id* in table 4 equal to 1. A “Full Profile” compatible receiver must be compatible with all the definitions contained in the document with regards to the Ginga support.

5.1 Monomedias support

The media types presented in table 13 must be supported by a “Full Profile” compatible receiver.

Table 13: “Full Profile” supported media types.

Category	Media Type	MIME Type	Extension(s)
Image	PNG	image/png	png
	JPEG	image/jpeg	jpg, jpeg
	BPG (HEVC Intra)	image/x-bpg	bpg
Audio	MPEG-4 AAC	audio/mp4	mp4, mpeg4
	Opus	audio/opus	opus
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
	ncl-settings	application/x- ncl-settings	-
	ncl-time	application/x- ncl-time	-

Beyond the media types presented in table 13, other interesting media types for radio context that could be supported in future profile definitions are Journaline and a high compression video codec.

Citar as normas de todas as mídia e do Journaline... e citar direitinho os perfis que devem ser suportados do AAC, JPEG, SVG, etc...

5.2 Screen Resolution

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.

References

- [1] 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.
- [2] ABNT NBR 15610-3. *Digital Terrestrial TV — Accessibility — Part 3: Brazilian Sign Language (LIBRAS)*. Brazilian National Standards Organization (ABNT), Brasília, DF, Brazil, March 2016.

- [3] 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.
- [4] ETSI ES 201 980. *Digital Radio Mondiale (DRM); System Specification; Version 4.1.1*. European Telecommunications Standards Institute, Nice, France, January 2014.
- [5] 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.
- [6] 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.
- [7] ETSI TS 101 756. *Digital Audio Broadcasting (DAB); Registered Tables; Version 1.8.1*. European Telecommunications Standards Institute, Nice, France, December 2015.
- [8] ETSI TS 101 968. *Digital Radio Mondiale (DRM); Data applications directory; Version 1.3.1*. European Telecommunications Standards Institute, Nice, France, April 2009.
- [9] ETSI TS 103 270. *RadioDNS Hybrid Radio; Hybrid lookup for radio services; Version 1.2.1*. European Telecommunications Standards Institute, Nice, France, September 2015.
- [10] ISO/IEC 10646. *Information technology – Universal Coded Character Set (UCS)*. International Organization for Standardization, Geneva, Switzerland, June 2014.
- [11] 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.
- [12] ITU-R REPORT BT.2267-5. *Integrated broadcast-broadband systems*. International Telecommunication Union (ITU), Geneva, Switzerland, July 2015.
- [13] ITU-T RECOMMENDATION H.761. *Nested Context Language (NCL) and Ginga-NCL*. ITU Telecommunication Standardization Sector, Geneva, Switzerland, November 2014.
- [14] 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.

- [15] 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.