International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.760

(03/2009)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

IPTV multimedia services and applications for IPTV – IPTV multimedia application frameworks

Overview of multimedia application frameworks for IPTV services

Recommendation ITU-T H.760

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729
IPTV middleware	H.730–H.739
IPTV application event handling	H.740–H.749
IPTV metadata	H.750–H.759
IPTV multimedia application frameworks	**H.760–H.769**
IPTV service discovery up to consumption	H.770–H.779

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T H.760

Overview of multimedia application frameworks for IPTV services

Summary

Recommendation ITU-T H.760 identifies and describes the relevant standards of multimedia application frameworks for interoperability and harmonization in IPTV services. It gives an overview of standards for declarative application frameworks as well as standards for procedural application frameworks. For declarative application frameworks, standards to be used for IPTV are described (such as HTML, CSS, DOM, ECMAScript, SVG, BML, MHEG-5 and Ginga-NCL), while procedural application frameworks are represented by GEM-based frameworks. It also contains descriptions of ISO/IEC International Standards for application frameworks, namely M3M, BIFS and LASeR. Recommendation ITU-T H.760 also has an annex on common usage of web-related technologies.

Source

Recommendation ITU-T H.760 was approved on 16 March 2009 by ITU-T Study Group 16 (2009-2012) under Recommendation ITU-T A.8 procedures.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Recommendation ITU-T H.760

Overview of multimedia application frameworks for IPTV services

1 Scope

This Recommendation identifies and describes the relevant standards for interoperability and harmonization among IPTV multimedia application frameworks.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T J.201] Recommendation ITU-T J.201 (2008), *Harmonization of declarative content format for interactive television applications*.

[ITU-T T.170] Recommendation ITU-T T.170 (1998), *Framework of the T.170-Series of Recommendations*.

[ITU-T T.172] Recommendation ITU-T T.172 (1998) | ISO/IEC 13522-5:1997, *MHEG-5 – Support for base-level interactive applications*.

[ITU-T T.175] Recommendation ITU-T T.175 (1998), *Application programming interface (API) for MHEG-5*.

[ITU-R BT.1699-1] Recommendation ITU-R BT.1699-1 (2009), *Harmonization of declarative application formats for interactive TV*.

[ITU-R BT.1722-1] Recommendation ITU-R BT.1722-1 (2007), *Harmonization of the instruction set for the execution engine for interactive TV applications*.

[ANSI/SCTE 90-1] ANSI/SCTE 90-1 (2005), *SCTE Application Platform Standard OCAP 1.0 Profile*.

[ARIB STD-B24] ARIB STD-B24 Ver5.3 (2009), *Data coding and Transmission Specification for Digital Broadcasting; ARIB, Association of Radio Industries and Businesses (Japan)*.

[ATSC A/101A] ATSC Document A/101 A (2009), *ATSC Standard: Advanced Common Application Platform (ACAP)*.

[CEA-2014] CEA-2014 (2006), *Web-based Protocol and Framework for Remote User Interface on UPnP Networks and the Internet (Web4CE)*.

[ETSI TS 102 322] ETSI TS 102 322 V1.1.1 (2004), *Specification for a Lightweight Microbrowser for interactive TV applications, based on and compatible with WML*.

[ETSI TS 102 812] ETSI TS 102 812 V1.2.2 (2006), *Digital Video Broadcasting (DVB) – Multimedia Home Platform (MHP) Specification 1.1.1*.

[ETSI TS 102 819] ETSI TS 102 819 V1.4.1 (2008), *Digital Video Broadcasting (DVB) – Globally Executable MHP version 1.0.3 (GEM 1.0.3)*.

[ISO/IEC 14496-11] ISO/IEC 14496-11 (2005), *Information technology – Coding of audio-visual objects – Part 11: Scene description and application engine*.

[ISO/IEC 14496-20] ISO/IEC 14496-20 (2008), *Information technology – Coding of audio-visual objects – Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)*.

[ISO/IEC 16262] ISO/IEC 16262 (2002), *Information technology – ECMAScript language specification*.

[ISO/IEC 23004.x] ISO/IEC 23004.x-series (2007), *Information technology – Multimedia Middleware*.

[W3C CSS1] Cascading Style Sheets, level 1 (2008), *W3C Recommendations: Cascading Style Sheets, level 1*.

[W3C DOM1] Document Object Model (DOM) Level 1 Specification Version 1.0 (1998), *W3C Recommendation: Document Object Model (DOM) Level 1 Specification*.

[W3C DOM2] Document Object Model (DOM) Level 2 Core Specification Version 1.0 (2000), *W3C Recommendation: Document Object Model (DOM) Level 2 Core Specification*.

[W3C SVG 1.1] Scalable Vector Graphics (SVG) 1.1 Specification (2003), *W3C Recommendation: Scalable Vector Graphics (SVC) 1.1 Specification*.

[W3C XHTML] XHTML 1.0 Second Edition (2002), *W3C Recommendation: XHTML 1.0, The Extensible HyperText Markup Language (Second Edition)*.

3 Definitions

3.1 Terms defined elsewhere

This clause is intentionally left blank.

3.2 Terms defined in this Recommendation

This clause is intentionally left blank.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ACAP	Advanced Common Application Platform
API	Application Programming Interface
BD-J	Blu-ray Disc Java
BIFS	BInary Format for Scene
BIFS-Anim	BIFS Animation
BML	Broadcast Markup Language
CE	Consumer Electronics
CSS	Cascading Style Sheet
DASE	Digital television Application Software Environment
DDE-T	Declarative Data Essence-Transitional

DOM	Document Object Model
EE	Execution Engine
FBA	Face and Body Animation
GEM	Globally Executable MHP
HDTV	High-Definition Digital TeleVision
ISDB-T	Integrated Services Digital Broadcasting-Terrestrial
LASeR	Lightweight Application Scene Representation
M3W	MPEG Multimedia Middleware
MHEG	Multimedia and Hypermedia information coding Experts Group
MHP	Multimedia Home Platform
NCL	Nested Context Language
OCAP	Open Cable Application Platform
PE	Presentation Engine
S&N	Synthetic and Natural
SAF	Simple Aggregation Format
SBTVD-T	Sistema Brasileíro de Televisão Digital Terrestre (Terrestrial Brazilian digital TV system)
SGML	Standard Generalized Markup Language
SMIL	Synchronized Multimedia Integration Language
SVG	Scalable Vector Graphics
UI	User Interface
VCR	VideoCassette Recorder
VM	Virtual Machine
XHTML	Extensible HyperText Markup Language
XML	Extensible Mark-up Language

5 Standards for declarative application frameworks

This clause describes the standards for declarative application frameworks in an alphabetical order, without implying any specific order or preference.

The declarative application platform is a framework on which applications written by a markup language (e.g., HTML) with or without script language (e.g., ECMAScript) can run.

5.1 Binary format for scene (BIFS)

BIFS [ISO/IEC 14496-11] is the scene description language standardized by ISO as a part of the MPEG-4 family of standards. It is designed to allow for the efficient representation of dynamic and interactive presentations, comprising two and three dimensional (2D & 3D) graphics, images, text and audiovisual material. The representation of such a presentation includes the description of the spatial and temporal organization of the different scene components as well as user-interaction and animations.

The main features of MPEG-4 BIFS are the following:

– Seamless embedding of audio/video content: MPEG-4 BIFS allows integration and control of different audio/video objects seamlessly in a scene.

– Rich set of 2D/3D graphical constructs: MPEG-4 BIFS provides a rich set of graphical constructs which enable 2D and 3D graphics. BIFS also provides tools that enable easy authoring of complex face and body animation, tools for 3D mesh encoding, and representation of 2D and 3D natural and synthetic sound models.

– Local and remote interactivity: BIFS defines elements that can interact with the client-side scene as well as with remote servers. Interactive elements allow for text input, mouse events, and other input devices that can trigger a variety of behaviours.

– Local and remote animations: scene properties, such as object positions, colours, and even shapes, etc., can be animated using either predefined scene descriptions or via streams sent from a server.

– Reuse of content: MPEG-4 scenes can contain references to streamed sub-scenes. This means that content can be easily reused, a powerful way to create a very rich user experience from relatively simple building blocks.

– Scripted behaviour: MPEG-4 scenes can have two types of scripted behaviour. A Java API can control and manipulate the scene graph, as well as built-in ECMA script (JavaScript) support that can be used to create complex behaviours, animations, and interactivity.

– Streamable scene-description: the spatial and temporal graphic layout is carried in a BIFS-Command stream. Such a stream operates on the scene-graph through commands which replace, delete and insert elements in the scene-graph.

– Accurate synchronization: audio/visual content can be tightly synchronized with other A/V content, client-side, and server-driven scene animation, due to the underlying MPEG-4 Systems layer.

– Compression: the scene description is converted to binary data and compressed in an efficient way.

Figure 5-1 summarizes the different components of BIFS.

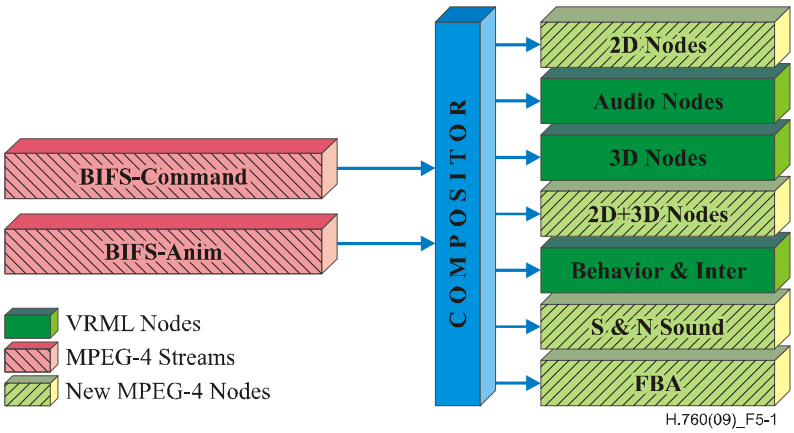

Figure 5-1 – MPEG-4 BIFS scene description

5.2 Broadcasting markup language (BML)

BML (see [ARIB STD-B24] and [ITU-T J.201]) is a declarative application specification for multimedia broadcasting in Japan, standardized by the Association of Radio Industries and Businesses (ARIB). BML consists of XHTML 1.0, CSS 1 and CSS 2, Document Object Model (DOM) 1 and DOM 2, and ECMA-262. BML was revised to include additional functionalities for receivers with digital storage and for terrestrial digital broadcasting including mobile reception.

The specification of the terrestrial integrated services digital broadcasting (ISDB-T) multimedia coding scheme was first based on that of multimedia and hypermedia information coding experts group (MHEG). However, since March 1999, ARIB started developing an XML specification for multimedia coding scheme based on the functionalities of MHEG. This contributed to the development of an XML application language, BML, that enables the presentation of an XML file on BML browsers. The ARIB data broadcasting system standards were enacted as BML in October 1999.

A BML application can be based on synchronization and spatial-temporal relationships of multiple media formats, structural composition, event-action associations, navigation, and user interaction capabilities just as MHEG. BML defines an additional API and object for broadcasting using ECMAScript.

The semantics of BML is based on the following minimum set of MHEG classes:

– Application Class
– Scene Class
– Link Class
– Action Class.

BML is applied to reference model for data broadcasting services that are carried out as part of the digital broadcasting. Figure 5-2 shows the structure of BML, which contains the following blocks:

– Data coding:
  - Coding mono-media (1): Coding system for character string and bit map, etc. used in multimedia.
  - Coding of subtitle, superimpose (2): Coding system for subtitle and superimpose.
– XML-based multimedia coding scheme:
  - Multimedia coding (3): Coding system of XML based system adopted as multimedia coding system and its profile.
– Data transmission specification:
  - Content transmission format (4): Content transmission format of data carousel transmission method, etc. to transmit content.
  - Subtitle and superimpose transmission format (5): Independent PES transmission format to transmit subtitle and superimpose.

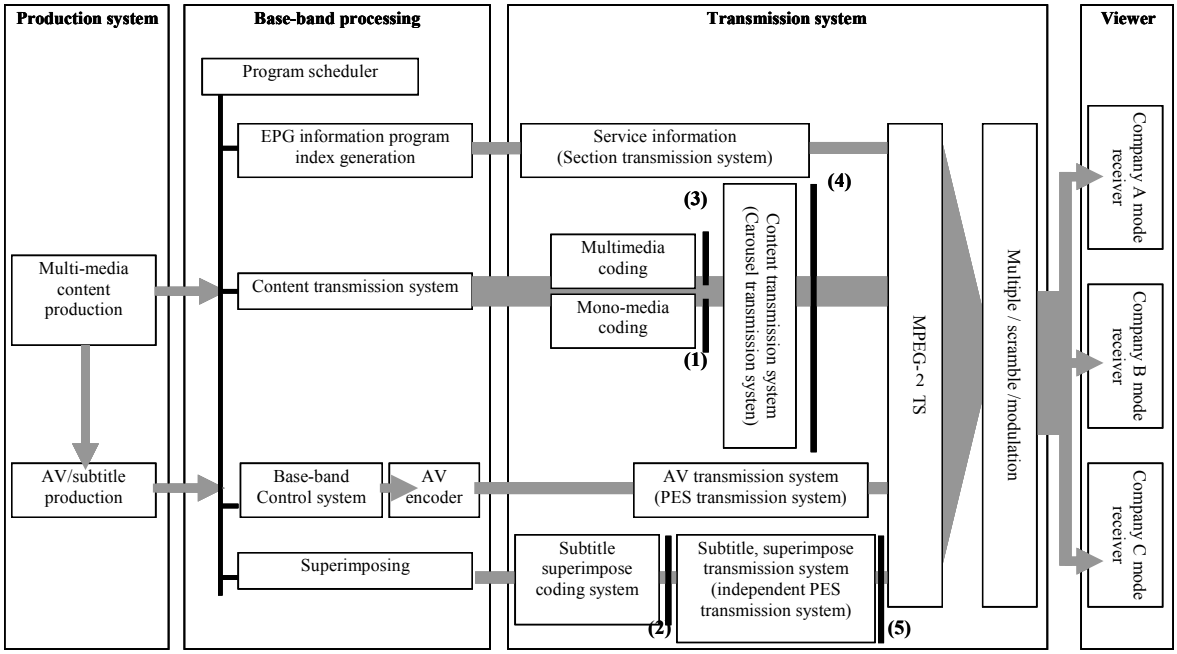

Figure 5-2 – System Structure of BML

5.3 CEA-2014

CEA-2014 is a remote user interface specification that allows a user interface to be remotely displayed on devices. It is based on existing web rendering technologies for consumer electronics (CE) browser with Worldwide Web Consortium (W3C) tags, XHTML 1, ECMA-262, CSS TV profile and DOM 2. CEA-2014 is a web-based protocol and framework for remote user interface (UI) on UPnP home network and over Internet. Two service domains are defined, as shown in Figure 5-3.

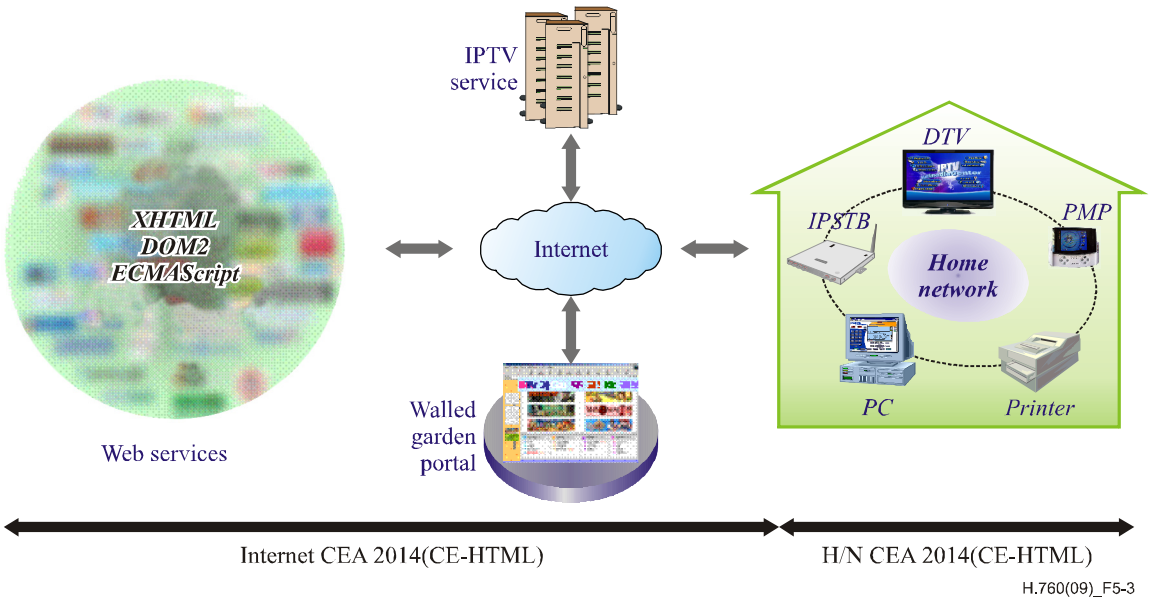

Figure 5-3 – Service model of CEA-2014 (remote user interface)

CEA-2014 defines the mechanisms allowing a user interface to be remotely displayed on and controlled by devices or control points other than the one hosting the logic. The basic device operations are based on the UPnP Device Architecture v1.0 for UPnP networks and UPnP devices in the home. The specification also allows the remote display of user interfaces provided by third-party Internet services on UPnP devices in the home, and covers UI capabilities for TVs, mobile phones and portable devices.

A major component of CEA-2014 is the CE-HTML profile that uses XHTML content to define user interfaces that can be rendered on screens with different resolutions and sizes ranging from HDTV screens to mobile-phone displays.

5.4 Cascading style sheet (CSS)

CSS is a style sheet language specified by W3C that is used to describe the presentation (e.g., fonts, colours and spacing) of a document written in a markup language. CSS is human readable and writable, and expresses style in common desktop publishing terminology.

CSS has been in use since October 1995 using the media type **text/css** without registration.

A CSS can be either linked to a document through a URI or included in the document.

A superset of [W3C CSS1] and [b-W3C CSS2] includes a number of new capabilities like absolute, relative, and fixed positioning of elements, the concept of media types, support for aural style sheets and bidirectional text, and new font properties such as shadows.

[b-W3C CSS TV Profile] specifies subsets of [b-W3C CSS2], e.g., colour specifications tailored to TV devices.

CSS does not constitute by itself a multimedia framework, but is used with some markup languages such as BML, CEA-2014, and SVG. CSS 1 and CSS 2 are used with BML. CSS TV Profile is used with CEA-2014, and CSS 2 is used with SVG and DVB-HTML.

5.5 Document object model (DOM)

DOM 2.0 [W3C DOM2] defines the Document Object Model Level 2 Core, a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content and structure of documents. The Document Object Model Level 2 Core builds on the Document Object Model Level 1 Core.

The DOM Level 2 Core is made of a set of core interfaces to create and manipulate the structure and contents of a document. The Core also contains specialized interfaces dedicated to XML.

DOM does not constitute by itself multimedia framework but is used with some markup languages such as BML, CEA-2014, and SVG. DOM 1 and DOM 2 are used with BML and SVG 1.1. SVG Tiny is using uDOM. CEA-2014 and DVB-HTML are using DOM 2.

5.6 Digital video broadcasting hypertext markup language (DVB-HTML)

Digital video broadcasting hypertext markup language (DVB-HTML) (see [ITU-T J.201]) is a standard for allowing digital televisions to access Internet content. It is an optional part of the larger MHP 1.1 standard of DVB.

The specification is based around a modularized version of XHTML 1.1, and also includes CSS 2.0, DOM 2.0, and ECMA-262 (also known as ECMAScript).

Among other things, MHP 1.1 specifies the Internet access profile, in which applications can control the basic operations of Open Internet resident clients (web browser, e-mail and news client).

Figure 5-4 outlines the basic architecture of MHP 1.1 and the relationship between the DVB-HTML optional application type and the DVB-J application type.

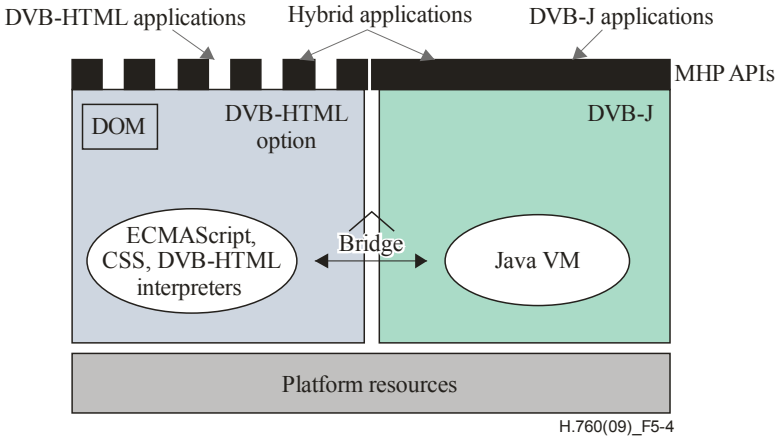

Figure 5-4 – MHP 1.1 and DVB-HTML

5.7 ECMAScript

ECMAScript [ISO/IEC 16262] is a scripting programming language that is used on the web and is often referred to as JavaScript or JScript, after the two earlier implementations of the specification.

ECMAScript is supported in many applications. ECMAScript is also included as a component in many presentation engines (PEs) such as BML and DVB-HTML, which are used for digital data broadcasting. Some implementations have a completely different set of libraries; making applications written in one dialect of ECMAScript will not necessarily work in another.

ECMAScript is an object-oriented programming language for performing computations and manipulating computational objects within a host environment. It was originally designed to be a web scripting language, providing a mechanism to enliven web pages in browsers and to perform server computation as part of a web-based client-server architecture.

A web browser provides an ECMAScript host environment for client-side computation including, for example, objects that represent windows, menus, pop-ups, dialog boxes, text areas, anchors, frames, history, cookies, and input/output. Further, the host environment provides a means to attach scripting code to events such as change of focus, page and image loading, unloading, error and abort, selection, form submission, and mouse actions. Scripting code appears within the HTML and the displayed page is a combination of user interface elements and fixed and computed text and images. The scripting code is reactive to user interaction and there is no need for a main program.

A web server provides a different host environment for server-side computation including objects representing requests, clients, and files; and mechanisms to lock and share data. By using browser-side and server-side scripting together, it is possible to distribute computation between the client and server while providing a customized user interface for web-based applications.

ECMAScript does not constitute by itself a multimedia framework but is used with some markup languages such as BML, CEA-2014, DVB-HTML and SVG.

5.8 Hypertext markup language (HTML)

HTML (hypertext markup language) is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document – by denoting certain text as links, headings, paragraphs, lists, and so on – and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of tags, surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code (such as ECMAScript) that can affect the behaviour of web browsers and other HTML processors.

Elements are the basic structure for HTML markup. Elements have two basic properties: attributes and content. Each attribute and each element's content has certain restrictions that must be followed for an HTML document to be considered valid. An element usually has a start tag (e.g., **<element-name>**) and an end tag (e.g., **</element-name>**). The element's attributes are contained in the start tag and content is located between the tags (e.g., **<element-name attribute="value">Content </element-name>**). Some elements, such as **
**, do not have any content and must not have a closing tag. Listed below are several types of markup elements used in HTML.

5.8.1 HTML and XHTML

HTML is also often used to refer to content in specific languages, such as a MIME type **text/html**, or even more broadly as a generic term for HTML, whether in its XML-descended form (such as XHTML 1.0 [W3C XHTML] and later) or its form descended directly from SGML (such as HTML 4.01 and earlier).

One difference in the latest HTML specifications lies in the distinction between the SGML-based specification and the XML-based specification. The XML-based specification is usually called XHTML to distinguish it clearly from the more traditional definition; however, the root element name continues to be 'html' even in the XHTML-specified HTML. W3C intended XHTML 1.0 to be identical to HTML 4.01 except where limitations of XML over the more complex SGML required workarounds. Because XHTML and HTML are closely related, they are sometimes documented in parallel. In such circumstances, some authors conflate the two names as (X)HTML or X(HTML).

Aside from the different opening declarations for a document, the differences between an HTML 4.01 and XHTML 1.0 document – in each of the corresponding document type definitions – are largely syntactic. The underlying syntax of HTML allows many shortcuts that XHTML does not, such as elements with optional opening or closing tags, and even EMPTY elements which must not have an end tag. By contrast, XHTML requires all elements to have an opening tag or a closing tag. XHTML, however, also introduces a new shortcut: an XHTML tag may be opened and closed within the same tag, by including a slash before the end of the tag like this: **
**. The introduction of this shorthand, which is not used in the SGML declaration for HTML 4.01, may confuse earlier software unfamiliar with this new convention.

5.8.2 Dynamic HTML

Combined with a client-side scripting language (such as ECMAScript), a presentation definition language (such as CSS) and DOM, HTML is often used to create interactive and animated web sites. Such a use is often called dynamic HTML.

Dynamic HTML allows a scripting language to change variables in a page's definition language, which in turn affects the look and function of otherwise "static" HTML page content, after the page has been fully loaded and during the viewing process.

5.9 Lightweight application scene representation (LASeR) and simple aggregation format

MPEG-4 Part 20 [ISO/IEC 14496-20] is a specification designed for representing and delivering rich-media services to resource-constrained devices such as mobile phones. It defines two binary formats: lightweight application scene representation (LASeR), a binary format for encoding 2D scenes, including vector graphics, and timed modifications of the scene; and simple aggregation format (SAF), a binary format for aggregating in a single stream LASeR content with audio/video streams.

The LASeR specification has been designed to allow the representation of 2D scenes describing rich-media services for constrained devices. A rich-media service features a dynamic and interactive presentation comprising 2D vector graphics, images, text and audiovisual material. The

representation of such a presentation includes describing the spatial and temporal organization of its different elements as well as its possible interactions and animations.

The SAF specification defines tools to enable the transport of LASeR content along with its attached audiovisual material according to these requirements. The SAF specification defines a binary format for a SAF stream, made of a LASeR stream with any type of media stream. SAF streams can be delivered using many delivery mechanisms: download-and-play, progressive download, streaming or broadcasting. To achieve reactivity, the SAF specification defines the concept of cache unit which allows sending in advance sub-content which will be used later on in the presentation.

5.10 MHEG-5

MHEG-5 (see [ITU-T T.170], [ITU-T T.172], [ITU-T T.175], [b-ETSI ES 202 184]) represents an application as a set of scenes that contain objects common to all scenes. A multimedia application can be conceived as a set of self-contained objects based on synchronization and spatio-temporal relationships of multiple media formats, structural composition, event-action associations, navigation, and user interaction capabilities. Controlling the playback of time-dependent content (like streams of multiplexed audiovisual data) requires specific support. These streams demand video cassette recorder (VCR) control functions (play, pause, fast forward, etc.), as well as the capability to manage events generated during their presentation. For example, rendering text subtitles can be synchronized with timecode events generated during the playback of a stream.

A scene supports the spatially and temporally coordinated presentation of audiovisual content consisting of graphics, bitmaps, text, and streams (based on the multiplex of audio and video components). Interaction can be performed via graphic elements like buttons, sliders, text entry boxes, and hypertext selections. Every scene, as well as an entire application, is a self-contained entity that can represent its localized behaviour by links that are event-action associations. Events can be generated by users, expiration of timers, playback of streams, and other conditions within the run-time environment.

The global scope of MHEG-5 is to define the syntax and semantics of a set of object classes that can be used for interoperability of multimedia applications across minimal-resources platforms. The developed applications will reside on a server, and as portions of the application are needed, they will be downloaded to the client. In a broadcast environment, this download mechanism could rely, for instance, on cyclic rebroadcast of all portions of the application. It is the responsibility of the client to have a runtime that interprets the application parts, presents the application to the user, and handles the local interaction with the user.

5.11 Nested context language (NCL)

Nested context language (NCL) is an XML application language that allows authors to write interactive multimedia presentations [b-ITU-T H.761]. Using this language an author can declaratively describe the spatio-temporal behaviour of a multimedia presentation, associate hyperlinks (viewer interaction) with media objects, define alternatives for content and for content presentation (adaptation), and describe the layout of the presentation on multiple exhibition devices. NCL is part of the data coding specifications of the Terrestrial Brazilian Digital TV System (SBTVD-T), and comprises the declarative language used by the presentation engine Ginga-NCL of the SBTVD-T middleware (Ginga).

5.11.1 Ginga and Ginga-NCL

Figure 5-5 depicts an overview of the architecture of the Ginga middleware. It consists of three main modules: the Ginga-cc (common core), the Ginga-NCL declarative environment and the Ginga-J imperative environment.

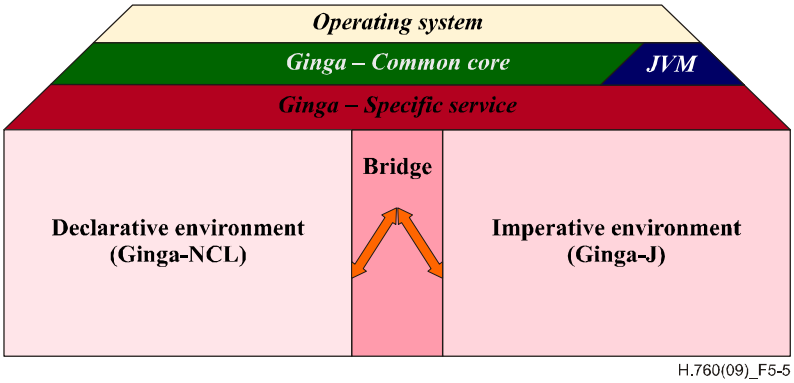

Figure 5-5 – Ginga architecture

Ginga-NCL is the logical subsystem of the Ginga system that processes NCL documents. A key component of Ginga-NCL is the declarative content decoding engine (NCL user agent). The NCL engine is in charge of receiving a NCL document and controlling its presentation, trying to guarantee that the specified relationships among media objects are respected.

5.11.2 NCL and the NCL media players

[b-ABNT NBR 15606-2] and [b-ABNT NBR 15606-5] specify the NCL application language profile for SBTVD fixed and portable receivers, respectively.

NCL defines the glue that holds media objects together in a multimedia presentation. Thus, an NCL document only defines how media objects are structured and related, in time and space. As a glue language, it does not restrict or prescribe the media-object content types. In this sense, we can have image objects (GIF, JPEG, etc.), video objects (MPEG, MOV, etc.), audio objects (MP3, WMA etc.), text objects (TXT, PDF etc.), imperative objects (Xlet, Lua, etc.), etc., as NCL media objects. Which media objects are supported depends on the media players that are embedded in the NCL engine (NCL user agent). One of these players could be a main audio/video decoder/player, usually implemented in a digital TV receiver's hardware. In this way, note that the main video and audio streams are treated like all other media objects that can be related using NCL.

NCL also treats an HTML document as one of its possible media objects. Therefore, Ginga can run any HTML application developed for ISDB, DVB and ATSC, depending only on the XHTML player implementation. Therefore, NCL does not substitute but embed XHTML-based documents.

NCL also includes support for media objects that contain imperative code, extending the language basic model, adding decision-making features that were otherwise not possible. NCL allows imperative objects with java code and with Lua code.

5.11.3 The scripting language Lua

Lua is the scripting language of NCL. Lua combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, runs by interpreting byte-code for a register-based virtual machine, and has automatic memory management with incremental garbage collection.

Among all these Lua's characteristics and advantages as a scripting language, it should be stressed its efficiency. Lua is an efficient dynamic language (interpreted, dynamic typing).

Both NCL and Lua engines are available as free software.

5.12 Scalable vector graphics (SVG)

SVG [W3C SVG 1.1] is a language for describing two-dimensional graphics and graphical applications in XML. SVG allows for three types of graphic objects: vector graphic shapes (e.g.,

paths consisting of straight lines and curves), images and text. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects and template objects.

SVG drawings can be interactive and dynamic. Animations can be defined and triggered either declaratively (i.e., by embedding SVG animation elements in SVG content) or via scripting. Sophisticated applications of SVG are possible by use of a supplemental scripting language which accesses SVG DOM, which provides complete access to all elements, attributes and properties.

SVG Basic and SVG Tiny are targeted to resource-limited devices and are part of the 3GPP platform for third generation mobile phones. SVG Print is a set of guidelines to produce final-form documents in XML suitable for archiving and printing.

5.13 Worldwide TV markup language (WTVML)

Worldwide TV Markup Language (WTVML) [ETSI TS 102 322] is a content format for the delivery of interactive TV applications using Internet servers. A WTVML interactive television technology platform comprises a micro-browser, a markup language, and a significant collection of associated software tools and services. The micro-browser and markup language are both based upon the Open Mobile Alliance WML 1.3 specification. It has all the required elements, attributes and events necessary to describe the user interface and user interaction models for the majority of interactive TV services.

The format supports key features to enable service-oriented applications and also provides a balance between the explicit layout and design requirements necessary for television and flexibility and dynamic capabilities of the Internet. The format fully describes the rendering and interaction of the interactive TV service in such a manner as to allow its implementation in a number of different mechanisms and technologies. The format assumes a user agent behaviour that is more sophisticated and more stateful than a standard Internet based HTML browser.

The format combines explicit pixel-perfect control required for TV user interfaces, and the dynamic layout and Internet compatibility requirements necessary for e-business and dynamic content. It provides a good separation between the layout of high-level objects, and the dynamic layout of elements in the main mark-up language. This separation also allows for the default attribute values to be set for a given layout, effectively creating sets of inherited styles. One useful by-product of this separation is the ability to deploy the service in a functionally identical way by interoperating differently or replacing the high-level layout for different device types.

The format assumes a rich event model and contains explicit state and variable management, allowing the creation of sophisticated user interface effects without the use of scripting.

It is common to implement WTVML functionality by way of a micro-browser and gateway architecture. In this architecture, the gateway processes the raw WTVML and generates compiled byte-code to be passed to the user agent to execute. It is expected that different networks may implement their own output formats from the gateway, consisting of WTVML byte-codes extended from the standard WAP Forum byte codes, HTML and ECMAScript, Java execution classes or even code that can be run directly on legacy set-top boxes' middleware or APIs.

6 Standards for procedural application frameworks

This clause describes the standards for procedural application frameworks in an alphabetical order, without implying any specific order.

The procedural application platform is a framework on which applications written by APIs based on procedural language (e.g., Java) can run.

6.1 GEM based application frameworks

Globally Executable MHP (GEM) [ETSI TS 102 819] specifies the core of Blu-ray's BD-J, OCAP and MHP, see Figure 6-1. For broadcast applications, it can best be thought of as the overlap between the MHP and OCAP standards for interactive television. It is a formally standardized Java-based platform for interactive content and applications. GEM has been standardized by DVB, and adopted by ETSI, ITU, CableLabs, ARIB, ACAP and the Blu-ray Disc Association.

6.1.1 Advanced common application platform (ACAP)

ACAP, (see [ATSC A/101] and [ITU-T J.201]) is applicable for specifications and standards based on the ACAP APIs, content formats, and semantic guarantees.

ACAP applications are classified into two categories depending upon whether the initial application content processed is of a procedural or a declarative nature. These categories of applications are referred to as a procedural (ACAP-J) and declarative (ACAP-X) applications, respectively. An example of an ACAP-J application is a Java TV Xlet composed of compiled Java byte code in conjunction with other multimedia content such as graphic, video, and audio. An example of an ACAP-X application is a multimedia document composed of extensible hypertext markup language (XHTML) markup, style rules, scripts, and embedded graphics, video, and audio.

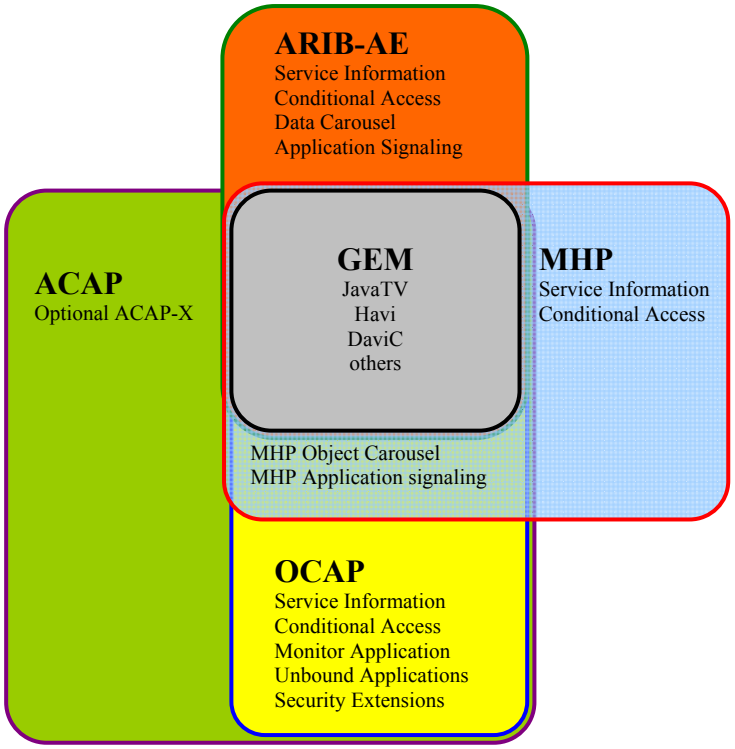

Figure 6-1 – GEM based Standards for multimedia application frameworks

Application environments are similarly classified into two categories depending upon whether they process procedural or declarative applications. These categories are referred to as ACAP-J and ACAP-X environments, respectively. An example of an ACAP-J environment is a Java Virtual Machine and its associated API implementation. An example of an ACAP-X environment is an XHTML multimedia document browser, also known as a user agent.

The architecture and facilities of ACAP are intended to apply to broadcast systems and receivers for terrestrial (over-the-air) broadcast and cable TV systems.

ACAP is primarily based on GEM and digital television application software environment (DASE) and includes additional functionality from OCAP. The following are the relevant specifications which ACAP includes:

– Globally executable MHP (GEM): ACAP includes GEM 1.0.2 in its entirety, except as explicitly modified by the ACAP.
– Open cable application platform (OCAP): ACAP includes OCAP 1.0 in its entirety, except as explicitly modified by the ACAP for ACAP terminal operating in a terrestrial broadcast environment.
– Addition on non-ACAP Interfaces: terminal specifications based on ACAP may add public interfaces, provided that they are added in a namespace that does not conflict with ACAP. ACAP terminal specifications and ACAP terminals shall not require that such extension interfaces be called by ACAP applications in order to enable behaviour that is normatively required by ACAP.

6.1.2 Recommendation ITU-R BT.1699

[ITU-R BT.1699-1] identifies functional commonality among the declarative application environments for interactive TV application specifications ACAP-X, BML and DVB-HTML. Elements which are common to these three standards are identified as a "common core". The value of the common core is to assist program authors to exchange declarative content internationally using these standards. [ITU-R BT.1699-1] also notes features outside of the common core of the covered standards. The goal of [ITU-R BT.1699-1] is to note these differences to encourage efforts toward increasing commonality between the standards to further improve functionality and enhance economies of scale.

[ITU-R BT.1699-1] is intended to harmonize the application environment for declarative content for interactive TV. It specifies common elements, media types and APIs at the syntactic level of the declarative application environment to satisfy regional application requirements for the three standards ACAP-X, BML and DVB-HTML as specified in the normative references. It is noted that there are other declarative formats such as MHEG-5 and Society of Motion Picture Television Engineers (SMPTE) DDE-T that are not covered in [ITU-R BT.1699-1].

6.1.3 Recommendation ITU-R BT.1722-1

[ITU-R BT.1722-1] defines APIs, semantic guarantees and system aspects of platform behaviour for the harmonized instruction set for the execution engines for interactive TV applications.

[ITU-R BT.1722-1] includes the following updated and newly developed instruction set for the execution engines: GEM, MHP 1.0 and 1.1, OCAP 1.0, ARIB-AE and ACAP. Harmonization is based on GEM 1.0.1 which has been developed with contributions from DVB, SCTE, ARIB and ATSC. GEM 1.0.1 is the normative reference in [ITU-R BT.1722-1].

6.1.4 Multimedia home platform (MHP)

MHP [ETSI TS 102 812] is the set of specifications for multimedia broadcasting that has been developed by the DVB Project. MHP has two functional categories. Version 1.0 series covers the execution engine (EE) environment and version 1.1 series covers the presentation engine (PE) environment in addition to version 1.0. The PE portion of the version 1.1 series is not standalone; it mandates the existence of the EE environment. The first release of the DVB MHP version 1.0 series was in May 2000. The MHP 1.0 specification employs Java technology for an EE environment.

6.1.5 Open cable application platform (OCAP)

OCAP [ANSI/SCTE 90-1] is the set of specifications for the interactive multimedia services of digital CATV that has been developed by the OpenCable project. OCAP 1.0 is based on MHP 1.0.2, and includes the extensions for the cable system in the United States.

7 Other related standards

7.1 MPEG multimedia middleware (M3W)

M3W [ISO/IEC 23004.x] provides, next to the architecture and component model descriptions and the reference software, two sets of APIs: multimedia platform APIs, the functional APIs, and Support platform APIs, the non-functional APIs (see Figure 7-1):

– The architecture and component model:
  - Part-1: Architecture;
  - Part-3: Component Model.

– The functional APIs:
  - Part-2: Multimedia API.

The non-functional APIs;
  - Part-4: Resource and quality management;
  - Part-5: Component download;
  - Part-6: Fault management;
  - Part-7: System integrity management.

– The reference software:
  - Part-8: Reference Software.

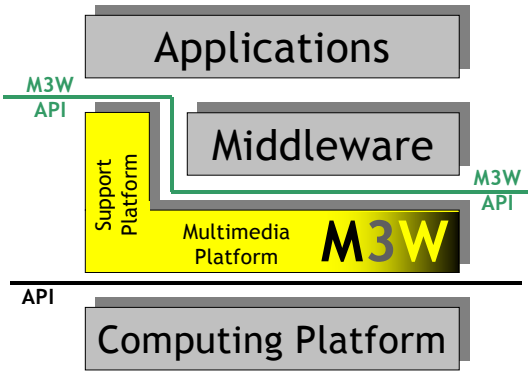

Figure 7-1 – The structure of M3W

The M3W multimedia platform APIs define mainly audio and video processing APIs handling front-end, decoders and post-processing of A/V and trust management APIs handling key management, signature management, license management, and certificate management. The M3W multimedia APIs offer a high-level API that is independent of the target hardware. By standardizing things that are common, the multimedia APIs simplify the work of middleware and application developers.

Annex A

Common usage

(This annex forms an integral part of this Recommendation)

For a typical multimedia framework for IPTV based on web-related technologies, the following combinations are considered foundational:

– one or more markup languages, such as HTML, XHTML, BML, and SVG;

– some profiles of style sheets, that is, CSS;

– some profiles of standard scripting, e.g., ECMAScript;

– DOM.

The particular combination of these technologies will define a "profile" of a standard.

The following clauses give profiles used in the standards mentioned in this Recommendation.

A.1 BML profile

– XHTML 1.0, CSS 1, CSS 2, DOM 1, DOM 2, ISO/IEC-16262 (ECMAScript).

A.2 CEA-2014

– XHTML 1, CSS TV profile, DOM 2, ISO/IEC-16262 (ECMAScript).

A.3 DVB-HTML profile

– XHTML 1.0, CSS 2, DOM 2, ISO/IEC-16262 (ECMAScript).

A.4 SVG profiles

– SVG 1.1: CSS 2, DOM 1, DOM 2, ISO/IEC-16262 (ECMAScript);

– SVG Tiny: CSS 2, uDOM, ECMA ISO/IEC-16262 (ECMAScript).

Bibliography

[b-ITU-T H.761] Recommendation ITU-T H.761 (2009), *Nested context language (NCL) and Ginga-NCL for IPTV services*.

[b-ITU-T J.202] Recommendation ITU-T J.202 (2008), *Harmonization of procedural content formats for interactive TV applications*.

[b-ABNT NBR 15606-2] ABNT NBR 15606-2 (2007), *Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 2: Ginga-NCL for fixed and mobile receivers – XML application language for application coding; Brazilian Association for Technical Standards*.

[b-ABNT NBR 15606-5] ABNT NBR 15606-5 (2007), *Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 5: Ginga-NCL for portable receivers – XML application language for application coding; Brazilian Association for Technical Standards*.

[b-ETSI ES 102 543] ETSI ES 102 543 (2007), *Digital Video Broadcasting (DVB) Globally Executable MHP (GEM)*.

[b-ETSI TS 202 184] ETSI TS 202 184 (2004), *MHEG-5 Broadcast Profile*.

[b-W3C CSS TV Profile] W3C Candidate Recommendation. *CSS TV Profile 1.0 (2003)*.

[b-W3C CSS2] W3C Candidate Recommendation. *Cascading Style Sheets, Level 2 Revision 1 (CSS2.1) Specification (2009).*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2010