International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.200

(04/2010)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Application for Interactive Digital Television

Worldwide common core – Application environment for digital interactive television services

Recommendation ITU-T J.200

Recommendation ITU-T J.200

Worldwide common core – Application environment for digital interactive television services

Summary

Recommendation ITU-T J.200 identifies the structure, the origins and the specification sources for a harmonized environment, including a set of application programming interfaces (APIs) for interactive television services.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T J.200	2001-03-09	9
1.1	ITU-T J.200 (2001) Cor. 1	2004-05-14	9
2.0	ITU-T J.200	2010-04-29	9

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

Introduction

This Recommendation is the result of collaboration, within ITU-T and ITU-R, of organizations such as DVB, ARIB, ATSC, OpenCable, ABNT, etc., who have contributed to the process of harmonizing the application environment for a wide range of media throughout the world. This Recommendation outlines the structure of application environments specified by organizations such as those listed above and indicates the high level of commonality which has been achieved. It includes the origin and key specification sources used to define the application environment in the normative references, as well as relevant terms, definitions, abbreviations and acronyms. Additional descriptions of terms and acronyms relating to the general subject of digital video have also been included for information.

Recommendation ITU-T J.200

Worldwide common core – Application environment for digital interactive television services

1 Scope

This Recommendation defines the high-level architecture for a harmonized set of interactive content formats and application programming interfaces (APIs) capable of providing the variety of functionalities required by advanced interactive applications to be delivered over television networks to end-users' homes. This application environment can also be used for downloading purposes, display purposes, network control and security.

This Recommendation is intended to provide the "umbrella" for the work in progress preparing the detailed Recommendations defining the specifications for the application environment for various applications and also the mechanisms provided for future extensions. There will of course be some regional or media-specific requirements which demand variation from the common structure recommended. However, this "umbrella" Recommendation is intended to encourage and assist continued harmonization of the work in progress in the ITU on the application environment for digital interactive television services.

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 (2009), *Harmonization of declarative content format for interactive television applications*.

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

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 application: Information that expresses a specific set of observable behaviour.

3.2.2 application entity: A unit of information that expresses some portion of an application.

3.2.3 application environment (environment): The context or software environment in which an application is processed.

3.2.4 application programming interface (API): Software libraries that provide uniform access to system services.

3.2.5 application resource: A bit-stream serialization (a physical embodiment) of an application entity.

3.2.6 application resource collection: The set of application resources that embody an application entity collection.

3.2.7 broadcast markup language (BML) standard: An XML [b-W3C XML] application language [b-ARIB STD B-24] that deals with tags and attributes for multimedia representation exclusively.

3.2.8 cascading style sheets (CSS): Standard for the style sheet for a markup language document.

3.2.9 content: A general term that refers to any of the following: application, application resource collection, or application resource.

3.2.10 declarative application: An application which primarily makes use of declarative information to express its behaviour; an XML document instance is an example of a declarative application.

3.2.11 declarative application environment: An environment that supports the processing of declarative applications; an XML user agent (browser) is an example of a declarative application environment.

3.2.12 digital storage media command and control (DSM-CC): A control method defined in [b-ISO/IEC 13818-6], which provides access to files and streams for digital interactive services.

3.2.13 document object model (DOM): An API that defines the logical structure of XML [b-W3C XML] and HTML [b-W3C HTML] documents and the way a document is accessed and manipulated. It is also called DOM-API. It is an interface independent from platforms and languages.

3.2.14 document object model (DOM) object: An object generated by a HTML [b-W3C HTML] document.

3.2.15 ECMAScript: Programming language defined by [b-ECMAScript].

3.2.16 execution engine: A subsystem in a receiver that evaluates and executes procedural applications consisting of computer language instructions and associated data and media content. An execution engine may be implemented with an operating system, computer language compilers, interpreters and application programming interfaces (APIs), which a procedural application may use to present audiovisual content, interact with a user, or execute other tasks that are not evident to the user. A common example of an execution engine is the JavaTV software environment, using the Java programming language and byte code interpreter, JavaTV APIs, and a Java virtual machine for program execution.

3.2.17 extensible HTML (XHTML): Reformulation of HTML [b-W3C HTML] as an application language of XML [b-W3C XML].

3.2.18 formatter: A subsystem in a receiver that evaluates and presents declarative applications consisting of content in multiple formats. A formatter also responds to formatting information associated with the content, to user inputs, and to script statements that control presentation behaviour and initiate other processes in response to user input and other events. An example of a formatter is a nested context language (NCL) [b-NCL] formatter engine.

3.2.19 function: A process which conveys or transforms data in a predictable way. It may be effected by hardware, software or a combination of both.

3.2.20 GINGA: The middleware specification for the Brazilian digital TV system. It comprises two execution environments, for both declarative and imperative applications.

3.2.21 GINGA-J: Ginga's execution environment for imperative applications written in Java. It also comprises a set of APIs for the development of interactive digital TV applications.

3.2.22 interoperability: The reception and presentation of applications in a vendor-, author- and broadcaster-neutral framework.

3.2.23 Lua: Lightweight embeddable scripting language that combines simple procedural syntax with data description constructs, based on associative arrays and extensible semantics.

3.2.24 markup language: A formalism that describes a document's structure, appearance, or other aspects. An example of a markup language is XHTML [b-W3C HTML1].

3.2.25 multimedia hypermedia experts group-5 (MHEG-5): A specification [b-ISO/IEC 13522-5] for presentation engine applications designed for decoding in interactive television receivers using modest resources. The UK profile [b-MHEG Profile], which is recognized within the ISO standard, extends the specification. It shares common text and graphics formats, and carousel mechanism with multimedia home platform (MHP), thus allowing the use of common data between MHEG-5 and MHP applications, with only a small overhead.

3.2.26 nested context language (NCL): The nested context language [b-NCL] is a declarative language that is used to describe the temporal behaviour of a multimedia presentation to associate hyperlinks (user interaction) with media objects, to define alternatives for presentation (adaptation), and to describe the layout of the presentation on multiple devices.

3.2.27 object: An identifiable entity consisting of data and/or computer code.

3.2.28 packet: A packet is a set of contiguous bytes consisting of a header followed by its payload.

3.2.29 payload: The bytes following the header byte in a packet.

3.2.30 presentation engine: A subsystem in a receiver that evaluates and presents declarative applications consisting of content, such as audio, video, graphics, and text, primarily based on presentation rules defined in the presentation engine. A presentation engine also responds to formatting information, or "markup", associated with the content, to user inputs, and to script statements, which control presentation behaviour and initiate other processes in response to user input and other events. A common example of a presentation engine is an HTML browser, capable of displaying text and graphic content formatted in HTML [b-W3C HTML], with interactive behaviour programmed in ECMAScript [b-ECMAScript].

3.2.31 procedural application: An application which primarily makes use of procedural information to express its behaviour. A Java program is an example of a procedural application.

3.2.32 procedural information: Information expressed in the form of procedures, e.g., do *F* or *F*( ).

3.2.33 receiver platform (platform): The receiver's hardware, operating system and native software libraries of the manufacturer's choice.

3.2.34 resource (system): A well-defined capability or asset of a receiver, which can be used by the application environment. Examples: MPEG decoder, graphics system.

3.2.35 return channel: The communication mechanism which provides connection between the receiver and a remote server.

3.2.36 scripting language: A language to describe the program process, which is embedded in markup documents.

3.2.37 section: A syntactic structure specified in [b-ITU-T H.222.0] for the embedding of data in the transport stream. A data structure comprising a portion of a [b-ITU-T H.222.0] (or a [b-ISO/IEC 13818-6]) defined table, such as the program association table (PAT), conditional access table (CAT), program map table (PMT) or DSM-CC section.

3.2.38 service: Content and applications provided by network operators and broadcasters.

3.2.39 service bound application: An application delivered as part of a broadcast stream.

3.2.40 service information (SI): Data that describes programs and services.

3.2.41 transport stream: MPEG-2 transport stream syntax for the packetization and multiplexing of video, audio, and data signals for digital broadcast systems.

3.2.42 trigger: An event that may cause a change in the behaviour of the application that registers interest in such events. Triggers may come from many sources, e.g., the broadcast stream, or may be generated from other data (such as the system clock). It also can carry some semantically significant payload in order to affect changes in an application based on information not available at the time the application was written.

3.2.43 user agent: An embodiment of a declarative application environment.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

API	Application Programming Interface
BML	Broadcast Markup Language
CAT	Conditional Access Table
CRC	Cyclic Redundancy Check
DAVIC	Digital Audio Visual Council
DNS	Domain Name System
DOM	Document Object Model
DSM-CC	Digital Storage Media – Command and Control
DTD	Document Type Definition
DVB	Digital Video Broadcasting
GEM	Globally Executable MHP
HAVi UI	Home Audio Video Interoperability, User Interface specification
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JMF	Java Media Framework
JPG	JPEG image format
LWUIT	Light Weight User Interface Toolkit
MHEG	Multimedia Hypermedia Experts Group
MHP	Multimedia Home Platform
MIME	Multipurpose Internet Mail Extensions
MPEG	Motion Picture Experts Group
NCL	Nested Context Language
OCAP	Open Cable Application Platform

OS	Operating System
PAT	Program Association Table
PCR	Program Clock Reference
PES	Packetized Elementary Stream
PID	Packet Identifier
PMT	Program Map Table
PNG	Portable Network Graphics
PPP	Point-to-Point Protocol
PSIP	Program and System Information Protocol
PTC	Physical Transmission Channel
PTS	Presentation Time-Stamp
SI	Service Information
UI	User Interface
URI	Uniform Resource Identifier
XHTML	eXtensible HTML
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
XSLT	XSL Transformations

5 Conventions

None.

6 Common application environment

APIs and middleware for digital interactive TV should be based on the structure defined below and the normative references given in clause 2.

6.1 Basic architecture

The high-level architecture of APIs and middleware for digital interactive TV can be regarded as comprised essentially of two components: the execution engine and the presentation engine. However, these two components are not necessarily independent; appropriate bridges may be defined. In addition to the basic components, there will be other native applications, or service-specific software and content, such as MHEG-5, as well as various proprietary formats.

Figure 1 shows the structure of the application environment. It identifies the relationship between the presentation engine and the execution engine.

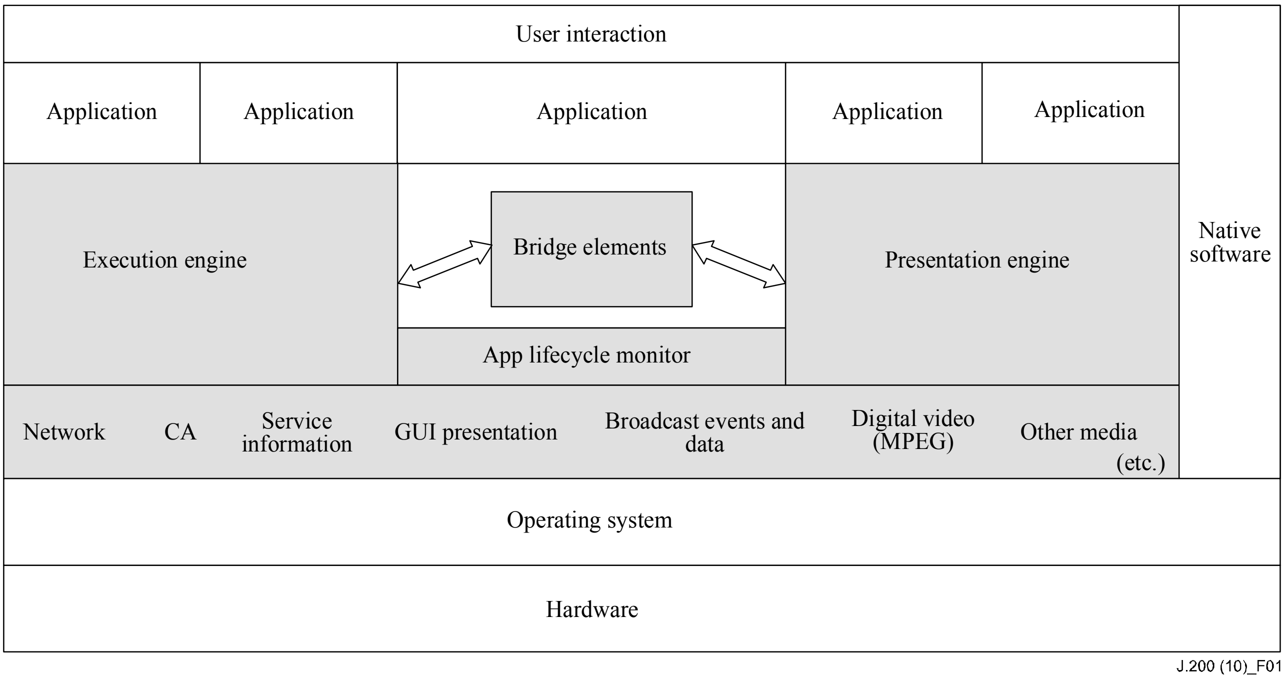


Figure 1 – Application environment system architecture

Where specified, the application environment draws from the following architectural elements:

1) Presentation engine/execution engine – These are described in the normative references and are both required. It is recognized that, in some cases, one or the other has been implemented individually. However, to achieve full harmonization and worldwide interoperability of applications, both, execution engine and presentation engine, are required.

2) Bridge elements – This is a mechanism for applications that allows bidirectional mapping between execution engine and presentation engine.

3) App lifecycle monitor – This controlling software is expressed differently in each specification. It can appear as a full-fledged application or just a simple operating system (OS) monitor to control the state of the software. The general functionality is to manage applications over its entire life cycle, including initiation, termination and control.

4) Applications – In some systems, these applications can be limited to service bound applications or unbounded applications. These applications can be written either to the presentation or to the execution, or to both.

5) User interaction – This is the input layer into remote control, keyboards, and other controllers.

6) Other media – This media includes streaming media, such as audio and data feeds, or monomedia, such as static images and text.

7) Native software – This is a software that is either a legacy software or a software written using additional APIs and functionality outside the specified application environment.

6.2 Execution engine

Figure 2 shows the structure of the execution engine and indicates the origin of the various elements.

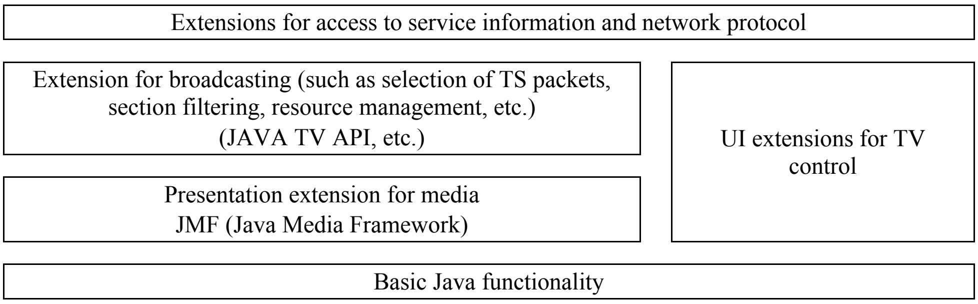

Extensions for access to service information and network protocol
Extension for broadcasting (such as selection of TS packets, section filtering, resource management, etc.) (JAVA TV API, etc.)
Presentation extension for media JMF (Java Media Framework)
UI extensions for TV control
Basic Java functionality

Figure 2 – Structure of the execution engine

The key standards that are supported by the execution engine are:

1) Basic Java functionality – The APIs consisting of the personal basis profile, foundation profile and connected device configuration.

2) Extension for broadcasting – This API consists of the javax.tv.* packages as specified by Sun Microsystems in the JavaTV specification plus additional broadcasting functionalities such as section filtering, transport stream tuning, resource management and return channel connectivity management (e.g., DAVIC or Ginga-J).

3) Presentation extension for media – This API consists of the javax.media.* packages by Sun Microsystems.

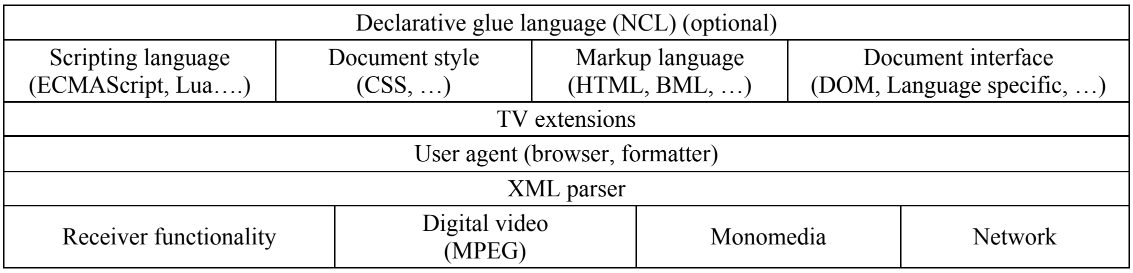
4) UI extensions for TV control – This API provides TV specific functionality for presentation and user interaction, e.g., HAVi UI API [b-HAVi UI] or light weight user interface toolkit (LWUIT).

5) Extensions for access to service information and network protocol (e.g., relevant APIs defined in [b-ETSI ES 201812], [b-ETSI TS 102727], [b-OCAP], [b-ATSC ACAP], [b-ARIB STD B-23], [b-GINGA-J]).

NOTE – Item 1) requires careful consideration of compatibility with systems based on older versions of this Recommendation. See normative references.

6.3 Presentation engine

Figure 3 shows the structure of the presentation engine and indicates the origin of the various elements.



<table>
<tr><td colspan="4">Declarative glue language (NCL) (optional)</td></tr>
<tr><td>Scripting language
(ECMAScript, Lua….)</td><td>Document style
(CSS, …)</td><td>Markup language
(HTML, BML, …)</td><td>Document interface
(DOM, Language specific, …)</td></tr>
<tr><td colspan="4">TV extensions</td></tr>
<tr><td colspan="4">User agent (browser, formatter)</td></tr>
<tr><td colspan="4">XML parser</td></tr>
<tr><td>Receiver functionality</td><td>Digital video
(MPEG)</td><td>Monomedia</td><td>Network</td></tr>
</table>

Figure 3 – Structure of the presentation engine

The key standards that are supported by the presentation engine are:

1) Markup language such as HTML and BML common modules defined in XHTML modularization and/or HTML as defined by W3C.

2) Cascading style sheets (CSS) to describe the presentation style as defined by W3C.

3) DOM APIs to dynamically change the contents of HTML or XHTML documents as defined by W3C.

4) Scripting language such as ECMAScript [b-ECMAScript] and Lua [b-Lua].

5) Declarative glue language such as NCL.

6) TV specific extensions and additional APIs and DOM objects for handling signalling, triggers, protocols and MIME types specific to TV receivers and receiver-specific functions.

7) Standard media types such as image/jpg, image/png, audio/basic and TV specific media types such as video/mpeg as defined by ISO.

8) XML [b-W3C XML] parser as defined by W3C.

Appendix I

Terminology relevant to specific APIs

(This appendix does not form an integral part of this Recommendation)

The following list of terms is provided to facilitate identifying relevant technologies defined in a wide range of standards that specify the details of the APIs identified in this Recommendation.

I.1 **application boundary**: A formal general description of the data elements (HTML [b-W3C HTML] documents, code files, images, etc.) used to form one application and the logical locator of the entry point.

I.2 **application delivery system**: A mechanism by which an application is announced and signalled, and its resources delivered to the application environment.

I.3 **application entity collection**: A collection of application entities that expresses an application as a whole.

I.4 **application manager**: The entity that is responsible for managing the lifecycle of the applications. It manages applications running in both the presentation engine and execution engine if both are present.

I.5 **application root entity**: A specific element of an application entity collection which is processed before all other elements in the collection.

I.6 **asynchronous data**: Stand-alone or audio/video-related data transmitted with no strong timing requirements in the sense that it is not associated with any transmitted clock references, and that availability of data in a data receiver is not governed by any such clock references.

I.7 **attribute**: A parameter to represent the character of a property.

I.8 **broadcast XML (B-XML)**: XML [b-W3C XML] tags specific to each application are defined in its DTD. XML tags are converted into BML tags by XSLT when they are presented on a terminal. The architecture of XML defined in this way is called B-XML.

I.9 **browser pseudo-object**: The additional objects to realize functions which are unique to broadcasting. They do not inherit properties as basic objects and behave as pseudo-objects.

I.10 **built-in object**: An object which is implemented in the execution system of [b-ECMAScript] from the start of script execution. There are nine types of objects: Array, Boolean, Date, Function, Global, Math, Number, Object and String.

I.11 **character**: A specific "letter" or other identifiable symbol, e.g., "A".

I.12 **character encoding**: A mapping between an integer input value and the textual character that is represented by this mapping, e.g., 'in ASCII, value 65 (decimal) is character A', or shift-JIS for Japanese characters.

I.13 **colour lookup table**: Transformation table from index colour value to physical value.

I.14 **communication channel**: A digital medium that transports a digital stream. A communication channel can be unidirectional or bidirectional.

I.15 **communication network**: A system of interconnected entities providing data interchange between points or from a point to multiple points.

I.16 **constant bit rate**: Operation where the bit rate is constant from start to finish of the bit stream.

I.17 **constructor**: A function that generates and initializes objects.

I.18 cyclic redundancy check (CRC): The cyclic redundancy check used to verify the correctness of the data.

I.19 data access unit: The portion of a synchronized or synchronous data elementary stream that is associated with a particular MPEG-2 presentation time stamp.

I.20 data carousel: A transmission scheme defined in [b-ISO/IEC 13818-6], with which data is transmitted repetitively. It can be used for downloading various data in broadcasting. The scenario of the DSM-CC user-to-network download protocol that embodies the cyclic transmission of data.

I.21 data element: A self-contained subset of a data elementary stream.

I.22 data elementary stream: The payloads of a series of consecutive MPEG-2 transport stream packets referenced by a unique PID value.

I.23 data module: An ordered sequence of bytes of a bounded size.

I.24 data receiver: Any device capable of receiving and consuming data carried on an MPEG-2 transport stream.

I.25 data service: A collection of applications intended to be provided together as defined by the data service provider.

I.26 datagram: The fundamental protocol data unit in a packet-oriented data delivery protocol. Typically, a datagram is divided into header and data areas, where the header contains full addressing information (source and destination addresses) with each data unit. Datagrams are most often associated with connectionless network and transport layer services.

I.27 declarative information: Information expressed in the form of assertions, e.g., *P* is, *Q* is, *R* is, or, more succinctly, {*P*, *Q*, *R*}.

I.28 decoder: An embodiment of a decoding process.

I.29 decoding process: The process that reads an input coded bit stream and outputs decoded pictures, audio samples, or data objects.

I.30 domain of an application: An application cannot run outside its domain. The maximum lifetime of an application extends from the moment the user navigates to its domain until the moment the user navigates away from its domain.

I.31 DVB-HTML: HTML-based markup language (declarative language) specified as an option within MHP specification.

I.32 DVB-J: The Java platform defined as part of the MHP and the OCAP [b-OCAP] specifications.

I.33 environment resource: A physical or logical component of an application environment, e.g., a region of the graphics frame buffer, an input device, a shared semaphore, a memory pool, etc.

I.34 event group index: Descriptive information indicating the relationships among the events and/or local events over multiple events (see above).

I.35 event handler: A user-defined function which is triggered by key inputs and events invoked by transmitted signals.

I.36 event index: A generic term for event group index and local event index.

I.37 event information table: A table containing data concerning events or programmes, such as event name, start time, duration, etc.

I.38 events: Asynchronous communication between applications and the application environment on which they are being executed.

I.39 extensible stylesheet language (XSL): Stylesheet recommendation for XML [b-W3C XML].

I.40 field: An element of a two-dimensional binary data table.

I.41 GINGA-NCL: Ginga's presentation environment for declarative applications [b-GINGA-NCL] written in NCL [b-NCL] and its scripting language Lua [b-Lua].

I.42 inheritance: When a new interface with method and property is generated, the interface has the method and property of the parent interface.

I.43 instance: An occurrence of a process or application.

I.44 interface definition language: A language to define the interface for access to, and operation on, objects.

I.45 Java API: A standard interface for use by platform-independent application software. It is expressed in the Java language.

I.46 language binding: A specification for binding of DOM API and a programming language. For instance, DOM API and ECMAScript [b-ECMAScript] are bound in HTML [b- HTML].

I.47 lifetime of an application: The time between a moment at which the application is loaded and a moment at which the application is destroyed.

I.48 locator: A link, expressed in the syntax in [b-IETF RFC 2396], which provides a reference to an application or resource.

I.49 method: A property of an object and particularly a function that is associated with an object and is allowed to manipulate the object's data.

I.50 MHP: The multimedia home platform (MHP) consists of a MHP viewer terminal, including all possible low to high functionality implementations, its associated peripherals and the in-home digital network.

I.51 MHP solution: The MHP solution encompasses the whole set of technologies necessary to implement the MHP including protocols and APIs.

I.52 MHP terminal: A single piece of physical equipment conforming to the MHP specification, in particular in that it contains a virtual machine and an instance of the MHP API.

I.53 MPEG: Standards developed by ISO/IEC JTC 1/SC 29/WG 11, *Moving Picture Experts Group.* MPEG may also refer to the group.

I.54 MPEG-2: The collection of ISO/IEC 13818 standards.

I.55 multipart format: An entity that has a single entity body consisting of more than one encapsulated entity.

I.56 multiplexer/demultiplexer: A physical device that is capable of inserting MPEG-2 transport stream packets into and extracting MPEG-2 transport stream packets from an MPEG-2 transport stream.

I.57 multiprotocol encapsulation: The encapsulation of datagrams in addressable sections.

I.58 multipurpose Internet mail extensions (MIME): An application layer protocol. It features a content architecture to facilitate multimedia data such as text other than US-ASCII code, sound, image, etc. to be handled in Internet mails.

I.59 name server: A server machine or an entity to resolve symbolic names to numeric IP addresses based on DNS.

I.60 native object: An object included in the document object model (DOM).

I.61 navigator: A resident application which the end-user can use to select services and applications.

I.62 node; DOM node: A branch point of a tree configured with generated DOM objects. It is a node of the graph defined for describing the relationship among events, local events, etc. A unique node which is not a child of any other nodes of the generated tree is called the root node. A node which is a parent of a node is called the parent node. Nodes that have same parentage are called sibling nodes.

I.63 normal play time: The absolute temporal coordinates that represents the position in a stream at which an event occurs.

I.64 object carousel: A repetitively broadcast file system.

I.65 packet identifier (PID): As defined in [b-ITU-T H.222.0].

I.66 persistent storage: Memory available that can be read/written to by an application and may outlive the application's own life. Persistent storage may be volatile or non-volatile.

I.67 packetized elementary stream (PES) packet header: The leading fields in a PES packet up to, but not including, the PES packet data byte fields where the stream is not a padding stream. In the case of a padding stream, the PES packet header is defined as the leading fields in a PES packet up to, but not including, the padding byte fields.

I.68 PES stream: A continuous sequence of PES packets of one elementary stream with one stream id.

I.69 physical channel; physical transmission channel: A generic term to refer to each of the 6- or 8-MHz frequency bands where television signals are embedded for transmission. Also known as the physical transmission channel (PTC). One analogue virtual channel fits in one PTC, but multiple digital virtual channels typically coexist in one PTC.

I.70 plug-in: A set of functionalities which can be added to a generic platform in order to provide additional functionality.

I.71 plug-in-type architecture: A software architecture that allows companion modules to be introduced in the receiver to aid in execution of applications and provision of data services.

I.72 point-to-point protocol (PPP): A protocol which enables transfer of multiple protocols over a point-to-point link [b-IETF RFC 1661]. It is used for dial-up connections.

I.73 presentation time-stamp (PTS): A field that may be present in a PES packet header that indicates the time that a presentation unit is presented in the system target decoder.

I.74 presentation unit (PU): A decoded audio access unit or a decoded picture.

I.75 procedural application environment: An environment that supports the processing of procedural applications. For example, a Java virtual machine and its APIs constitute an example of a procedural application environment.

I.76 profile: A specification for a class of capabilities providing different levels of functionality in a receiver.

I.77 program stream: A sequence of audio and/or video packets encoded in the MPEG format.

I.78 program clock reference (PCR): A time stamp in the transport stream from which decoder timing is derived.

I.79 program specific information: Normative data (defined in [b-ITU-T H.222.0]), which are necessary for the demultiplexing of transport streams and the successful regeneration of programs.

I.80 **property**: An attribute of an object. For instance, the properties of ECMAScript [b-ECMAScript] DOM objects include five types of data values (Number, String, Boolean, Null and Undefined), objects and methods.

I.81 **prototype**: A property of an object used for sharing and inheritance of other objects.

I.82 **program and system information protocol (PSIP)**: A collection of tables describing virtual channel attributes, event features, and other information.

I.83 **record**: A set of data fields in a database.

I.84 **resource**: A network data object or a service, which is uniquely identified in the network, e.g., an application resource or an environment resource.

I.85 **resident application**: An application available from non-volatile storage in receiver.

I.86 **resource identifier**: An identifier that labels a resource, e.g., a URI.

I.87 **resource reference**: The use of a resource identifier to refer to a resource.

I.88 **service description framework**: The information conveyed in the program element and providing the data service table and, optionally, the network resource table of a single data service.

I.89 **stream**: An ordered series of bytes extracted from the transport stream packet payloads that have a common unique PID value (e.g., video PES packets or program map table (PMT) sections).

I.90 **stream data**: A stream is a data object, which has no specific start or end. The decoding system may need only a small fraction of the total data to activate a given application.

I.91 **synchronized data**: Data that uses MPEG-2 PCRs and MPEG-2 PTSs with the objective of matching presentation and/or display of data units with access units of other streams (typically audio and video).

I.92 **synchronous data**: Data that uses MPEG-2 PCRs and MPEG-2 PTSs with the objective of delivering data units with timing constraints, these data units being processed for presentation and/or display as a stand-alone stream.

I.93 **system services**: Various common functions in the receiver platform that the applications may use to implement data services such as the tuning services, the user interface services, the remote control input services, the communication protocol and the channel navigation services.

I.94 **system software**: Software implementation below the API for a specific platform entirely under control of the manufacturer.

I.95 **table**: The collection of reassembled sections bearing a common version number.

I.96 **table instance**: Tables are identified by the table id field. However, in cases such as the data event table or the long term service table, several instances of a table are defined simultaneously. All instances are conveyed in transport stream packets of the same PID value and have the same table id field value but different table id extension field values.

I.97 **tap**: A reference to a data resource, including, but not limited to, a data elementary stream, a data carousel module, or a network resource.

I.98 **time-stamp**: The time of a specific action such as the arrival of a byte or the presentation of a presentation unit.

I.99 **uniform resource identifier (URI)**: An addressing method to access a resource in local storage or on the Internet.

I.100 virtual channel: The designation, usually a number that is recognized by the user as the single identification that will provide access to an analogue TV programme or a set of one or more digital elementary streams. It is called "virtual" because its identification (name and number) may be defined independently from its physical location.

I.101 Xlet: Interface used for Java application life cycle control.

Bibliography

[b-ITU-T H.222.0] Recommendation ITU-T H.222.0 (2006) | ISO/IEC 13818-1:2007, *Information technology – Generic coding of moving pictures and associated audio information: Systems.*

[b-ITU-T H.262] Recommendation ITU-T H.262 (2000) | ISO/IEC 13818-2:2000, *Information technology – Generic coding of moving pictures and associated audio information: Video.*

[b-ATSC ACAP] ATSC standard A/101 (2009), *Advanced Common Application Platform (ACAP).*

[b-ARIB STD-B5] ARIB standard STD-B5 V1.0 (1996), *Data Multiplex Broadcasting System for the Conventional Television Using the Vertical Blanking Interval* (Japanese).

[b-ARIB STD-B10 V4.6] ARIB standard STD-B10 V4.6-E2 (2008), *Service Information for Digital Broadcasting System.*
<http://www.arib.or.jp/english/html/overview/doc/6-STD-B10v4_6-E2.pdf>

[b-ARIB STD-B23] ARIB standard STD-B23 V1.2 (2009), *Application Execution Engine Platform for Digital Broadcasting.*

[b-ARIB STD-B24] ARIB standard STD-B24 (1999), *Data Coding and Transmission Specification for Digital Broadcasting.*

[b-ECMAScript] Standard ECMA-262 (2000), 3rd Edition, *ECMAScript Language Specification.*

[b-HAVi UI] *HAVi Level 2 User Interface, HAVi Consortium.*

[b-HAVi] *HAVi (Home Audio/Video interoperability) User Interface Specification 1.0, HAVi Consortium.*

[b-OCAP] OpenCable Application Platform Specifications (2009), *OCAP 1.1. Profile.*
<http://www.cablelabs.com/specifications/OC-SP-OCAP1.1.2-090930.pdf>

[b-ETSI TS 102 727] ETSI TS 102 727 V1.1.1 (2010), *Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2.*

[b-ETSI TS 102 728] ETSI TS 102 728 V1.1.1 (2010), *Digital Video Broadcasting (DVB); Globally Executable MHP (GEM) Specification 1.2.2 (including IPTV).*

[b-ETSI ES 201 812] ETSI ES 201 812 V1.1.2 (2006), *Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.0.3.*

[b-IETF RFC 1661] IETF RFC 1661 (1994), *The Point-to-Point Protocol (PPP).*

[b-IETF RFC 2046] IETF RFC 2046 (1996), *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types.*

[b-IETF RFC 2396] IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax.*

[b-ISO/IEC 10646] ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS).*

[b-ISO/IEC 13522-5] ISO/IEC 13522-5:1997, *Information technology – Coding of multimedia and hypermedia information – Part 5: Support for base-level interactive applications.*

[b-ISO/IEC 13818-3] ISO/IEC 13818-3:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio*.

[b-ISO/IEC 13818-6] ISO/IEC 13818-6:1998, *Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC*.

[b-W3C CSS1] W3C Recommendation (1999), *Cascading Style Sheets, level 1*.
<http://www.w3.org/TR/1999/REC-CSS1-19990111>

[b-W3C CSS2] W3C Recommendation (1998), *Cascading Style Sheets, level 2*.
<http://www.w3.org/TR/1998/REC-CSS2-19980512>

[b-W3C DOM1] W3C Recommendation (1998), *Document Object Model (DOM) Level 1 Specification, Version 1.0*.
<http://www.w3.org/TR/REC-DOM-Level-1>

[b-W3C DOM2 CORE] W3C Recommendation (2000), *Document Object Model (DOM) Level 2 Core Specification, Version 1.0*.
<http://www.w3.org/TR/REC-DOM-Level-2-Core-20001113>

[b-W3C DOM2 EVENTS] W3C Recommendation (2000), *Document Object Model (DOM) Level 2 Events Specification, Version 1.0*.
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113>

[b-W3C DOM2 HTML] W3C Working Draft (2000), *Document Object Model (DOM) Level 2 HTML Specification*.
<http://www.w3.org/TR/2000/WD-DOM-Level-2-Html-20001113>

[b-W3C DOM2 STYLE] W3C Recommendation (2000), *Document Object Model (DOM) Level 2 Style Specification*.
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113>

[b-W3C DOM2 VIEWS] W3C Recommendation (2000), *Document Object Model (DOM) Level 2 Views Specification*.
<http://www.w3.org/TR/2000/REC-DOM-Level-2-Views-20001113>

[b-W3C HTML] W3C Recommendation, *HTML 4.01 Specification* (1999).
<http://www.w3.org/TR/1999/REC-html401-19991224>

[b-W3C XHTML1] W3C Recommendation (2000), *XHTML^{TM} 1.0: The Extensible HyperText Markup Language, A Reformulation of HTTML 4 in XML 1.0*.
<http://www.w3.org/TR/xhtml1>

[b-W3C XHTMLMOD] W3C Proposed Recommendation (2001), *Modularization of XHTML^{TM}*.
<http://www.w3.org/TR/2001/PR-xhtml-modularization-20010222>

[b-W3C XML] W3C Recommendation (1998), *Extensible Markup Language (XML), Version 1.0*.
<http://www.w3.org/TR/1998/REC-xml-19980210>

[b-W3C XML NAMES] W3C Recommendation (1999), *Namespaces in XML*.
<http://www.w3.org/TR/1999/REC-xml-names-19990114>

[b-W3C XML-style] W3C Recommendation (1999), *Associating Style Sheets with XML documents, Version 1.0*.
<http://www.w3.org/TR/xml-stylesheet/>

[b-CDC] *JSR 218: Connected Device Configuration (CDC) 1.1* (2006).
<http://jcp.org/en/jsr/detail?id=218>

[b-FP] JSR219: Foundation Profile 1.1 (2006). <http://jcp.org/en/jsr/detail?id=219>

[b-GINGA-J] ABNT NBR 15606-4 (2010), *Digital terrestrial television – Data coding and transmission specification for digital broadcasting – Part 4: Ginga-J – The environment for the execution of procedural applications*.

[b-GINGA-NCL] 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*.

[b-Gosling] Gosling, J., Joy, B., Steele, G. (1996), *The Java Language Specification (1st Edition),* Addison-Wesley, ISBN 0-201-63451-1.

[b-Gosling-ERR] Sun Microsystems, *The Java Language Specification, Clarifications and Amendments to the JLS.* <http://java.sun.com/docs/books/jls/clarify.html>

[b-Java TV 1.1] Sun Microsystems (2008), *Java TV API Specification 1.1.* <http://jcp.org/aboutJava/communityprocess/mrel/jsr927/index2.html>

[b-JavaDTV] *JavaDTV 1.3*, *Java DTV Applications Programming Interface 1.3 (2009).* <http://www.forumsbtvd.org.br/materias.asp?id=200>

[b-JMF2.1] Sun Microsystems, *Java Media Player Specification 2.1.1e.* <http://java.sun.com/products/java-media/jmf/1.0>

[b-Lindholm] Lindholm, T., Yellin, F., (1996), *The Java™ Virtual Machine Specification (1st edition),* Addison-Wesley, ISBN 0-201-63452-X.

[b-Lindholm-ERR] *Errata for The Java™ Java Virtual Machine Specification*, T. Lindholm and F. Yellin. <http://java.sun.com/docs/books/vmspec/errata.html>

[b-Lindholm2] Lindholm, T., Yellin, F. (1999), *The Java™ Virtual Machine Specification (2nd edition),* Addison-Wesley, ISBN 0-201-432943.

[b-Lua] Ierusalimschy, R., Figueiredo, L.H., Celes, W. (2006) *Lua 5.1 Reference Manual,* Lua.org, ISBN 85-903798-3-3.

[b-MHEG Profile] *Digital Terrestrial Television MHEG-5 Specification* (1998). <http://www.mheg.org/users/mheg/archives/doc/MHEG-5_Profile_Issue_1.pdf>

[b-NCL] *Nested Context Language 3.0, Part 8 – NCL Digital TV Profiles*, L.F. Gomes Soares and R. Ferreira Rodrigues (2006), ISSN 0103-9741. <http://www.ncl.org.br/documentos/NCL3.0-DTV.pdf>

[b-PBP] *JSR217 Personal Basis Profile 1.1 (2005).* <http://jcp.org/en/jsr/detail?id=217>

[b-UNICODE] *The Unicode Character Encoding Standard (2009)*, Version 5.2, The Unicode Consortium.

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