



INTERNATIONAL TELECOMMUNICATION UNION

**ITU-T**

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

**J.200**

(03/2001)

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**

ITU-T Recommendation J.200

(Formerly CCITT Recommendation)

---

ITU-T J-SERIES RECOMMENDATIONS  
CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER  
MULTIMEDIA SIGNALS

General Recommendations	J.1–J.9
General specifications for analogue sound-programme transmission	J.10–J.19
Performance characteristics of analogue sound-programme circuits	J.20–J.29
Equipment and lines used for analogue sound-programme circuits	J.30–J.39
Digital encoders for analogue sound-programme signals	J.40–J.49
Digital transmission of sound-programme signals	J.50–J.59
Circuits for analogue television transmission	J.60–J.69
Analogue television transmission over metallic lines and interconnection with radio-relay links	J.70–J.79
Digital transmission of television signals	J.80–J.89
Ancillary digital services for television transmission	J.90–J.99
Operational requirements and methods for television transmission	J.100–J.109
Interactive systems for digital television distribution	J.110–J.129
Transport of MPEG-2 signals on packetised networks	J.130–J.139
Measurement of the quality of service	J.140–J.149
Digital television distribution through local subscriber networks	J.150–J.159
IPCablecom	J.160–J.179
Miscellaneous	J.180–J.199
<b>Application for Interactive Digital Television</b>	<b>J.200–J.209</b>

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

## **ITU-T Recommendation J.200**

### **Worldwide common core – Application environment for digital interactive television services**

#### **Summary**

This Recommendation identifies the structure, origins and specification sources for a harmonized environment including a set of Application Programming Interfaces (APIs) for interactive television services.

#### **Source**

ITU-T Recommendation J.200 was prepared by ITU-T Study Group 9 (2001-2004) and approved under the WTSA Resolution 1 procedure on 9 March 2001.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. 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.

## 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, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2001

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from ITU.

## CONTENTS

	<b>Page</b>
1 Scope .....	1
2 References .....	1
2.1 Normative references.....	1
2.2 Informative References .....	1
3 Terms, definitions and abbreviations .....	4
3.1 Terms and definitions .....	4
3.2 Acronyms and Abbreviations .....	13
4 Recommendation .....	17
4.1 Basic architecture .....	17
4.2 Execution Engine.....	18
4.3 Presentation Engine .....	19



## ITU-T Recommendation 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 the result of collaboration in ITU-T and ITU-R of organizations such as DVB, ARIB, ATSC, OpenCable, SMPTE, etc., who have contributed to the process of harmonizing the application environment for a wide range of media throughout the world. It 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, acronyms, and abbreviations. Additional definitions and acronyms relating to the general subject of digital video have also been included.

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

##### 2.1 Normative references

Specification source	Title	Notes
ETSI TS 101 812, version 1.1.1	Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification	1
ARIB	ARIB STD-B24, Data broadcasting coding system and transmission system, Version 1.2	1

##### 2.2 Informative References

Specification source	Title	Notes
OC-OCAP-DS	OpenCable Application Platform (OCAP)	1, 2, 3
ATSC DASE	Draft ATSC Standard, "DTV Application Software Environment, Part 1: Introduction, Architecture, and Common Facilities", T3-528R1, 2001/02/09	1, 2, 3

ATSC DASE-DA	Draft ATSC Standard, DTV Application Software Environment, Part 2: Declarative Applications and Environment, Draft ATSC Standard, T3-529R1, 2001/02/09	1, 2, 3
ATSC DASE-PA	Draft ATSC Standard, "DTV Application Software Environment, Part 3: Procedural Applications and Environment", T3-530R1, 2001/02/09	1, 2, 3
ATSC DASE-API	Draft ATSC Standard, "DTV Application Software Environment, Part 4: Application Programming Interface", T3-531R0, 2001/01/30	1, 2, 3
ATSC DASE-PFR	Draft ATSC Standard, "DTV Application Software Environment, Part 5: Portable Font Resource", T3-532R1, 2001/02/09	1, 2, 3
ECMAScript	ECMA-262, 3rd Edition, "Standardizing Information and Communication Systems Standard ECMAScript Language Specification"	2
DOM1	W3C Recommendation, "Document Object Model (DOM) Level 1 Specification Version 1.0", 1 October 1998, <a href="http://www.w3.org/TR/REC-DOM-Level1">http://www.w3.org/TR/REC-DOM-Level1</a>	2
DOM2 CORE	W3C Recommendation, "Document Object Model (DOM) Level 2 Core", <a href="http://www.w3.org/TR/REC-DOM-Level-2-Core-20001113">http://www.w3.org/TR/REC-DOM-Level-2-Core-20001113</a>	2
DOM2 EVENTS	W3C Recommendation, "Document Object Model (DOM) Level 2 Events", <a href="http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113">http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113</a>	2
DOM2 HTML	W3C Working Draft, "Document Object Model (DOM) Level 2 HTML", <a href="http://www.w3.org/TR/2000/WD-DOM-Level-2-Html-20001113">http://www.w3.org/TR/2000/WD-DOM-Level-2-Html-20001113</a>	2,3
DOM2 STYLE	W3C Recommendation, "Document Object Model (DOM) Level 2 Style", <a href="http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113">http://www.w3.org/TR/2000/REC-DOM-Level-2-Style-20001113</a>	2
DOM2 VIEWS	W3C Recommendation, "Document Object Model (DOM) Level 2 Views", <a href="http://www.w3.org/TR/2000/REC-DOM-Level-2-Views-20001113">http://www.w3.org/TR/2000/REC-DOM-Level-2-Views-20001113</a>	2
CSS1	W3C Recommendation, "Cascading Style Sheets, Level 1" <a href="http://www.w3.org/TR/1999/REC-CSS1-19990111">http://www.w3.org/TR/1999/REC-CSS1-19990111</a>	2
CSS2	W3C Recommendation, "Cascading Style Sheets, Level 2", <a href="http://www.w3.org/TR/1998/REC-CSS2-19980512">http://www.w3.org/TR/1998/REC-CSS2-19980512</a>	2
HAVi	"HAVi (Home Audio/Video Interoperability) User Interface Specification 1.0," HAVi Consortium, <a href="http://www.havi.org/home.html">http://www.havi.org/home.html</a> .	2
HAVi UI	See annex V, ".HAVi Level 2 User Interface", page 591.	2
HTML	W3C Recommendation, HTML 4.01 Specification, <a href="http://www.w3.org/TR/1999/REC-html401-19991224">http://www.w3.org/TR/1999/REC-html401-19991224</a>	2
JMF1.1	"Java Media Player Specification 1.1," Sun Microsystems, <a href="http://java.sun.com/products/java-media/jmf/1.0">http://java.sun.com/products/java-media/jmf/1.0</a>	2



JMF1	Sun Microsystems Java Media Player Specification (javadoc). Edition 1.0, September 2, 1997, <a href="http://jav.sun.com/products/java-media/jmf/forDevelopers/playerapi/packages.html">http://jav.sun.com/products/java-media/jmf/forDevelopers/ playerapi/packages.html</a>	2
Java TV	Java TV API Specification 1.0, Sun Microsystems, <a href="http://java.sun.com/products/javatv/">http://java.sun.com/products/javatv/</a>	2
Personal JAE	Sun Microsystems, PersonalJava Application Environment Specification Version 1.2, <a href="http://java.sun.com/products/personaljava/">http://java.sun.com/products/personaljava/</a>	2
JVM1	The Java Virtual Machine Specification (1st Edition), T. Lindholm and F. Yellin, Addison-Wesley, 1996, ISBN 0-201-63452-X	2
JVM1-ERRATA	Sun Microsystems, Errata for the Java Virtual Machine Specification, T. Lindholm and F. Yellin, <a href="http://java.sun.com/docs/books/vmspec/errata.html">http://java.sun.com/docs/books/vmspec/errata.html</a>	2
JVMX	Sun Microsystems, Inner Classes Specification, February 4, 1997, <a href="http://java.sun.com/products/jdk/1.1/docs/guide/innerclasses/">http://java.sun.com/products/jdk/1.1/docs/guide/innerclasses/</a>	2
JVM2	The Java Virtual Machine Specification (2nd edition), T. Lindholm and F. Yellin, Addison-Wesley, 1999, ISBN 0-201-432943	2
JLS1	The Java Language Specification (1st Edition), James Gosling et al., Addison-Wesley, 1996, ISBN 0-201-63451-1	2
JLS-ERRATA	Sun Microsystems, Clarifications and Amendments to The Java Language Specification, <a href="http://java.sun.com/docs/books/jls/clarify.html">http://java.sun.com/docs/books/jls/clarify.html</a>	2
XML NAMES	W3C Recommendation (14 January 1999) "Namespaces in XML", <a href="http://www.w3.org/TR/1999/REC-xml-names-19990114">http://www.w3.org/TR/1999/REC-xml-names-19990114</a>	2
XHTML1	W3C Recommendation (14 January 1999), "XHTML <sup>TM</sup> 1.0: The Extensible HyperText Markup Language, A Reformulation of HTML 4 in XML 1.0", 26 January 2000, <a href="http://www.w3.org/TR/xhtml1/">http://www.w3.org/TR/xhtml1/</a>	2
XHTMLMOD	W3C Proposed Recommendation (22 February 2001), "Modularization of XHTML <sup>TM</sup> ", <a href="http://www.w3.org/TR/2001/PR-xhtml-modularization-20010222">http://www.w3.org/TR/2001/PR-xhtml-modularization-20010222</a>	2, 3
XML	W3C Recommendation (10 February 1998), "Extensible Markup Language (XML) 1.0", <a href="http://www.w3.org/TR/1998/REC-xml-19980210">http://www.w3.org/TR/1998/REC-xml-19980210</a>	2
XML-stylesheets	W3C Recommendation (29 June 1999), "Associating Style Sheets with XML documents", <a href="http://www.w3.org/TR/xml-stylesheet/">http://www.w3.org/TR/xml-stylesheet/</a>	2
SMPTE DDE-1	Society of Motion Picture and Television Engineers, D27.106-2297B, Specification in ballot	2, 3
UNICODE	Unicode Character Encoding Standard, Version 3.0.1, Unicode Consortium	2

ISO/IEC 10646-1	ISO/IEC 10646-1:2000, <i>Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.</i>	
ISO/IEC 13818-1	ITU-T H.222.0 (1995)   ISO/IEC 13818-1:1996, <i>Information technology – Generic coding of moving pictures and associated audio information: Systems.</i>	
ISO/IEC 13818-2	ITU-T H.262 (1995)   ISO/IEC 13818-2:1995, <i>Information technology – Generic coding of moving pictures and associated audio information – Part 2: Video.</i>	
ISO/IEC 13818-3	ISO/IEC 13818-3:1998, <i>Information technology – Generic coding of moving pictures and associated audio – Part 3: Audio.</i>	
ISO/IEC 13818-6	ISO/IEC 13818-6:1998, <i>Information technology – Generic coding of moving pictures and associated audio information – Part 6: Extensions for DSM-CC.</i>	
MIME-MEDIA 2 RFC2046	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, IETF RFC 2046	
ARIB STD B10 V1.3	ARIB STD-B10 V1.3 (June 2000), "Service Information for Digital Broadcasting System", (Japanese)	
ARIB STD B-5	ARIB STD-B5 (August 1996) "Data Multiplex Broadcasting for Standard Television with Transmission Method Using Vertical Blanking Interval", (Japanese)	
MHP045	Digital Video Broadcasting (DVB); commercial requirements Rev.11	2
UK MHEG Profile	Digital Terrestrial Television MHEG-5 Specification 1.05	2

NOTE 1 – Master Specification: This document is considered a "master" Application Environment standard, which contains many normative references that are not listed individually in this Recommendation. However, some of these have been listed because they have special significance in the context of this "umbrella" Recommendation.

NOTE 2 – Specification contributed by an organization not yet recognized by ITU: This document is considered to be normative, but has to be included as informative because the originating organization is not a standards body recognized by the ITU. It will be transferred to normative status when the necessary procedures for ITU recognition have been completed.

NOTE 3 – Specification in process: This document is considered to be potentially normative, but has to be included as informative at the time of publication because it has not yet completed all approval processes required by the relevant standards organization. The document will be transferred to normative status when the required process is completed.

### 3 Terms, definitions and abbreviations

#### 3.1 Terms and definitions

The list of definitions is more extensive than needed to define the text in this Recommendation in order to facilitate the comprehension of the wide range of standards under this "umbrella" specification.

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

- 3.1.2 application boundary:** A formal general description of the data elements (HTML documents, code files, images, etc.) used to form one application and the logical locator of the entry point.
- 3.1.3 application delivery system:** A mechanism by which an application is announced and signalled, and its resources delivered to the application environment.
- 3.1.4 application entity:** A unit of information that expresses some portion of an application.
- 3.1.5 application entity collection:** A collection of application entities that expresses an application as a whole.
- 3.1.6 application environment:** The context or software environment in which an application is processed.
- 3.1.7 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.
- 3.1.8 application programming interface:** Consists of software libraries that provide uniform access to system services.
- 3.1.9 application resource:** A bit-stream serialization (a physical embodiment) of an application entity.
- 3.1.10 application resource collection:** The set of application resources that embody an application entity collection.
- 3.1.11 application root entity:** A specific element of an application entity collection which is processed before all other elements in the collection.
- 3.1.12 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.
- 3.1.13 attribute:** A parameter to represent the character of property.
- 3.1.14 audio-visual event:** An event (see definition below) where elementary streams are all of type video or audio.
- 3.1.15 bit rate:** The rate at which the bit stream is delivered from the channel to the input of a decoder.
- 3.1.16 broadcast markup language (BML) standard:** It specifies an XML application language called BML that deals with tags and attributes for multimedia representation exclusively.
- 3.1.17 broadcast XML (B-XML):** 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.
- 3.1.18 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.
- 3.1.19 built-in object:** An object which is implemented in the execution system of ECMAScript from the start of script execution. There are 9 types of objects (Array, Boolean, Date, Function, Global, Math, Number, Object, String).
- 3.1.20 call:** To transmit the signal for establishing a telecommunication channel from a telephone, etc.
- 3.1.21 cascading style sheets:** Standard for the style sheet for HTML documents.
- 3.1.22 character:** A specific "letter" or other identifiable symbol, e.g. "A".

- 3.1.23 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.
- 3.1.24 character set:** See "character encoding".
- 3.1.25 colour lookup table:** Transformation table from index colour value to physical value.
- 3.1.26 communication channel:** A digital medium that transports a digital stream. A communication channel can be unidirectional or bidirectional.
- 3.1.27 communications network:** A system of interconnected entities providing data interchange between points or from a point to multiple points.
- 3.1.28 constant bit rate:** Operation where the bit rate is constant from start to finish of the bit stream.
- 3.1.29 constructor:** A function that generates and initializes objects.
- 3.1.30 CRC:** The cyclic redundancy check used to verify the correctness of the data.
- 3.1.31 content:** A general term that refers to any of the following: application, application resource collection, or application resource.
- 3.1.32 data access unit:** The portion of a synchronized or synchronous Data Elementary Stream that is associated with a particular MPEG-2 Presentation Time Stamp.
- 3.1.33 data carousel:** A transmission scheme defined in 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.
- 3.1.34 data element:** A self-contained subset of a data elementary stream.
- 3.1.35 data elementary stream:** The payloads of a series of consecutive MPEG-2 Transport Stream packets referenced by a unique PID value.
- 3.1.36 data module:** An ordered sequence of bytes of a bounded size.
- 3.1.37 data receiver:** Any device capable of receiving and consuming data carried on an MPEG-2 Transport Stream.
- 3.1.38 data service:** A collection of Applications intended to be provided together as defined by the data service provider.
- 3.1.39 data source:** The provider of data that is being inserted into the MPEG-2 Transport Stream.
- 3.1.40 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.
- 3.1.41 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.
- 3.1.42 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.
- 3.1.43 declarative information:** Information expressed in the form of assertions; e.g. *P* is, *Q* is, *R* is, or, more succinctly,  $\{P, Q, R\}$ .
- 3.1.44 decoded stream:** The decoded reconstruction of a compressed bit stream.
- 3.1.45 decoder:** An embodiment of a decoding process.

- 3.1.46 decoding process:** The process that reads an input coded bit stream and outputs decoded pictures, audio samples, or data objects.
- 3.1.47 digital storage media command and control (DSM-CC):** A control method defined in ISO/IEC 13818-6, which provides access to files and streams for digital interactive services.
- 3.1.48 document object model:** An API that defines the logical structure of XML and 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.
- 3.1.49 document type definition:** Declaration of document type used for XML.
- 3.1.50 DOM object:** An Object generated by an HTML document.
- 3.1.51 domain name system protocol:** Used for a service of mapping host names in a network to IP addresses.
- 3.1.52 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 that the user navigates away from its domain.
- 3.1.53 DTV application software environment (DASE):** Includes software modules that allow decoding and execution of applications that deliver interactive and data broadcast services. The standard environment allows service content and applications to be decoded and executed in a manner independent of the receiver's hardware and operating system.
- 3.1.54 DVB-HTML actor:** The locus of activity or process involved in running the specific set of DVB-HTML documents for some DVB-HTML application, plus any *instantiated* context for that data. The actor runs inside a support application (native, plug-in or downloaded). The nature of the process is not defined explicitly as it depends on the nature of the support application itself. More than one such locus of activity may be present in any given support application.
- 3.1.55 DVB-HTML application:** A set of documents selected from the DVB-HTML family of elements and content formats as defined in the specification. The extent of the set is described by the application boundary.
- 3.1.56 DVB-HTML application states:** Logical states that a DVB-HTML actor can be in (as opposed to states the supporting application may be in); these states may have instance data logically associated with them (e.g. the application id and entry point).
- 3.1.57 DVB-HTML document:** A complete unit of one the HTML family of elements or content formats defined in the DVB specification.
- 3.1.58 DVB-J:** The Java platform defined as part of the MHP and OCAP specifications.
- 3.1.59 DVB-J API:** One of the Java APIs standardized as part of the MHP and OCAP specifications.
- 3.1.60 DVB-J application:** A set of DVB-J classes that operate together and need to be signalled as a single instance to the Application Manager so that it is aware of its existence and can control its lifetime through a lifecycle interface.
- 3.1.61 dynamically redefinable character set:** A scheme to transmit dynamically redefinable external characters using pattern data.
- 3.1.62 ECMAScript:** Programming language defined by Standard ECMA-262.
- 3.1.63 electronic program guide:** A Program table that is presented electronically.
- 3.1.64 element:** A portion of document punctuated by tags.

- 3.1.65 elementary stream:** A generic term for one of the coded video, coded audio or other coded bit streams in PES packets. One elementary stream is carried in a sequence of PES packets with one, and only one, stream id.
- 3.1.66 encoding process:** A process that reads a stream of input pictures or audio samples and produces a valid coded bit stream.
- 3.1.67 entity:** Information which is transmitted as a result of a request or response. It consists of meta information in the format of entity header field and content in the format of the entity body.
- 3.1.68 environment:** See "application environment".
- 3.1.69 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.
- 3.1.70 event group index:** Descriptive information indicating the relationships among the events and/or local events over multiple events. (see above)
- 3.1.71 event handler:** A user defined function which is triggered by key inputs and events invoked by transmitted signals.
- 3.1.72 event index:** A generic term for event group index and local event index.
- 3.1.73 event information table:** A Table containing data concerning events or programmes, such as event name, start time, duration, etc.
- 3.1.74 events:** Asynchronous communication between applications and the application environment on which they are being executed.
- 3.1.75 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 Interfaces (APIs), which a procedural application may use to present audiovisual content, interact with a user, or execute other tasks, which 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.
- 3.1.76 extended object for broadcasting:** The ECMAScript objects, which are specified as an extension in the ARIB B24 Standard. CSVTable object and BinaryTable object are specified and inherit their properties as basic objects.
- 3.1.77 Extended Unix Code (-JP) (EUC-JP):** Japanese character code used in the UNIX environment generally. It is encoded based on ISO/IEC 2022.
- 3.1.78 field:** An element of a two-dimensional binary data table.
- 3.1.79 File Transfer Protocol (FTP) [RFC 959]:** A protocol to share and transfer the files between two hosts through TCP/IP.
- 3.1.80 font:** A mechanism that allows the specific rendering of a particular character to be specified, e.g. Tiresias, 12 points. In practice, a font file format will incorporate some aspects of a character encoding.
- 3.1.81 function:** A process which conveys or transforms data in a predictable way. It may be effected by hardware, software or a combination of the two.
- 3.1.82 Huffman coding:** A type of source coding that uses codes of different lengths to represent symbols, which have unequal likelihood of occurrence.
- 3.1.83 Hypertext Transfer Protocol (HTTP) [RFC 2068]:** An application layer protocol used for data transmission via the World Wide Web.

- 3.1.84 inheritance:** When a new interface with method and property is generated, the interface has the method and property of the parent interface.
- 3.1.85 instance:** An occurrence of a process or application.
- 3.1.86 interface definition language:** A language to define the interface for access to, and operation on, objects.
- 3.1.87 Java API:** A standard interface for use by platform-independent application software. It is expressed in the Java language.
- 3.1.88 language binding:** A specification for binding of DOM API and a programming language. For instance, DOM API and ECMAScript are bound in HTML.
- 3.1.89 lifetime of an application:** Characterizes the time from which the application is Loaded to the time the application is Destroyed.
- 3.1.90 interoperability:** The reception and presentation of applications in a vendor-, author- and broadcaster-neutral framework.
- 3.1.91 locator:** A link, expressed in the syntax in RFC 2396, which provides a reference to an application or resource.
- 3.1.92 mail address:** A mail address sent with a protocol such as SMTP.
- 3.1.93 markup text:** Text that is added to the primary information content of a document in order to convey information about that content.
- 3.1.94 markup language:** A formalism that describes a document's structure, appearance, or other aspects. An example of a markup language is XHTML.
- 3.1.95 maximum transmission unit:** The largest amount of data that can be transferred in a single unit across a specific physical connection. When using the Internet Protocol, this translates to the largest IP datagram size allowed.
- 3.1.96 method:** A method is a property of an object and particularly a function that is associated with an object and is allowed to manipulate the object's data.
- 3.1.97 MHEG-5:** ISO/IEC 13522-5 a specification for presentation engine applications designed for decoding in interactive television receivers using modest resources. The UK profile, which is recognized within the ISO standard, extends the specification. It shares common text and graphics formats, and carousel mechanism with MHP, thus allowing the use of common data between MHEG-5 and MHP applications, with only a small overhead.
- 3.1.98 MHP:** The Multimedia Home Platform (MHP) consists of an MHP viewer terminal, including all possible low to high functionality implementations, its associated peripherals and the in-home digital network.
- 3.1.99 MHP solution:** The MHP solution encompasses the whole set of technologies necessary to implement the MHP including protocols and APIs.
- 3.1.100 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.
- 3.1.101 module (DSM-CC):** In the data carousel scheme, data is divided into blocks and transmitted. A module is one of the blocks.
- 3.1.102 MPEG:** Refers to standards developed by the ISO/IEC JTC 1/SC 29 WG 11, *Moving Picture Experts Group*. MPEG may also refer to the Group.
- 3.1.103 MPEG-2:** Refers to the collection of ISO/IEC standards 13818-1 through 13818-6.
- 3.1.104 multipart format:** An entity, which has a single entity body consisting of more than one encapsulated entities.

**3.1.105 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.

**3.1.106 multiprotocol encapsulation:** The encapsulation of datagrams in addressable sections.

**3.1.107 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.

**3.1.108 name server:** Administration name server based on DNS. In general, it means a server machine, which resolves symbolic names to numeric IP addresses.

**3.1.109 native object:** An object included in the Document Object Model.

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

**3.1.111 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.

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

**3.1.113 object carousel:** A repetitively broadcast file system.

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

**3.1.115 opportunistic data:** Data inserted into the remaining available bandwidth in a given transport stream after all necessary bits have been allocated for video, audio and other services.

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

**3.1.117 packet identifier (PID):** A unique integer value used to associate elementary streams of a program in a single or multi-program transport stream.

**3.1.118 payload:** Payload refers to the bytes following the header byte in a packet.

**3.1.119 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.

**3.1.120 PES packet:** The data structure used to carry elementary stream data. It consists of a packet header followed by PES packet payload.

**3.1.121 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.

**3.1.122 PES stream:** A continuous sequence of PES packets of one elementary stream with one stream is stream id.

**3.1.123 physical 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.

**3.1.124 physical transmission channel:** See "physical channel".

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



- 3.1.126 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.
- 3.1.127 Point-to-Point Protocol (PPP) [RFC 1661]:** A protocol which enables transfer of multiple protocols over a point-to-point link. It is used for dial-up connections.
- 3.1.128 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, with interactive behaviour programmed in ECMA script.
- 3.1.129 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.
- 3.1.130 presentation unit (PU):** A decoded audio access unit or a decoded picture.
- 3.1.131 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.
- 3.1.132 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.
- 3.1.133 procedural information:** Information expressed in the form of procedures, e.g.  $do F$  or  $F()$ .
- 3.1.134 profile:** A specification for a class of capabilities providing different levels of functionality in a receiver.
- 3.1.135 program stream:** A sequence of audio and/or video packets encoded in the MPEG format.
- 3.1.136 program clock reference (PCR):** A time stamp in the transport stream from which decoder timing is derived.
- 3.1.137 program specific information:** Defined in ISO/IEC 13818-1 and consists of normative data, which is necessary for the demultiplexing of Transport Streams and the successful regeneration of programs.
- 3.1.138 property:** An attribute of an object. For instance, the properties of ECMAScript DOM objects include 5 types of data values (number, string, Boolean, null and undefined), objects and methods.
- 3.1.139 prototype:** A property of an object used for sharing and inheritance of other objects.
- 3.1.140 Program and System Information Protocol (PSIP):** A collection of tables describing virtual channel attributes, event features, and other information.
- 3.1.141 receiver platform (platform):** The receiver's hardware, operating system and native software libraries of the manufacturer's choice.
- 3.1.142 record:** A set of data fields in a database.
- 3.1.143 regular expression:** A method of capturing a large, possibly infinite, set of strings in a compact representation.
- 3.1.144 reserved:** The term 'reserved', when used in the clauses defining the coded bit stream, indicates that the value may be used in the future.
- 3.1.145 resident application:** An application available from non-volatile storage in receiver.

- 3.1.146 resource (system):** A well-defined capability or asset of a receiver, which can be used by the application environment. Examples: MPEG decoder, Graphics system.
- 3.1.147 resource:** A network data object or a service, which is uniquely identified in the network. An application resource or an environment resource.
- 3.1.148 resource identifier:** An identifier that labels a resource, e.g. a URI.
- 3.1.149 resource reference:** The use of a resource identifier to refer to a resource.
- 3.1.150 return channel:** The communication mechanism which provides connection between the receiver and a remote server.
- 3.1.151 sandbox:** The context to which an application is restricted.
- 3.1.152 scripting language:** A language to describe the program process, which is embedded in HTML documents.
- 3.1.153 secret key cryptosystem:** The same key is used for encryption and decryption in the cryptosystem.
- 3.1.154 section:** A syntactic structure specified in ISO/IEC13818-1 for the embedding of data in the transport stream. A data structure comprising a portion of an ISO/IEC 13818-1 – or an 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.
- 3.1.155 service:** Content and applications provided by network operators and broadcasters.
- 3.1.156 service application (bound):** An application delivered as part of a broadcast stream.
- 3.1.157 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.
- 3.1.158 service information (SI):** Data which describes programs and services.
- 3.1.159 simple mail transfer protocol (SMTP) [RFC 821]:** A protocol to relay and distribute an e-mail.
- 3.1.160 stream:** A unidirectional continuous flow of content. Example: MPEG2 video.  
An ordered series of bytes. The usual context for the term *stream* is the series of bytes extracted from Transport Stream packet payloads that have a common unique PID value (e.g., video PES packets or Program Map Table sections).
- 3.1.161 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.
- 3.1.162 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).
- 3.1.163 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.
- 3.1.164 system services:** The receiver platform provides application programs with various common functions that the applications may use to implement data services. For example: the tuning services, the user interface services, the remote control input services, the communication protocol and channel navigation services.
- 3.1.165 system software:** Software implementation below the API for a specific platform entirely under control of the manufacturer.

- 3.1.166 table:** The collection of reassembled sections bearing a common version number.
- 3.1.167 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.
- 3.1.168 tap:** A reference to a data resource, including but not limited to: a data elementary stream, a data carousel module, or a network resource.
- 3.1.169 time-stamp:** A term that indicates the time of a specific action such as the arrival of a byte or the presentation of a presentation unit.
- 3.1.170 transport stream:** Refers to the MPEG-2 Transport Stream syntax for the packetization and multiplexing of video, audio, and data signals for digital broadcast systems.
- 3.1.171 trigger:** An event that may cause a change in the behaviour of 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 an application was written.
- 3.1.172 UCS transformation format:** A transformation method for UCS.
- 3.1.173 Uniform Resource Identifier (URI):** An addressing method to access a resource in local storage or on the Internet.
- 3.1.174 Universal (coded) Character Set (UCS):** An international character code created by ISO (ISO/IEC 10646).
- 3.1.175 user agent:** An embodiment of a declarative application environment.
- 3.1.176 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.
- 3.1.177 X.28:** An ITU-T Recommendation which defines the access function from a public subscriber telephone network to a public data network with packet switching.
- 3.1.178 eXtensible HTML (XHTML):** Reformulation of HTML as an XML application.
- 3.1.179 Xlet:** Interface used for Java application life cycle control.
- 3.1.180 extensible stylesheet language (XSL):** Style sheet recommendation for XML.

## 3.2 Acronyms and Abbreviations

This Recommendation uses the following abbreviations:

API	Application Programming Interface
ATSC	Advanced Television Systems Committee
AV	Audio/Video
AWT	Abstract Windowing Toolkit (from Java™)
bit/s	Bits per second
BML	Broadcast Markup Language
B-XML	Broadcast XML
CA	Conditional Access

CAS	Conditional Access System
CAT	Conditional Access Table
CI	Common Interface
CLUT	Colour Look-up Table
CSS	Cascading Style Sheets
CVCT	Cable Virtual Channel Table
DASE	DTV Application Software Environment
DAVIC	Digital Audio Visual Council
DCT	Discrete Cosine Transform
DECT	Digital Enhanced Cordless Telecommunications
DES	Data Elementary Stream
DET	Data Event Table
DNS	Domain Name System
DOM	Document Object Model
DRCS	Dynamically Redefinable Character Set
DSM-CC	Digital Storage Media – Command and Control
DSM-CC-OC	Digital Storage Media – Command and Control Object Carousel
DSM-CC-UU	Digital Storage Media – Command and Control User to User
DST	Data Service Table
DTD	Document Type Definition
DTV	Digital Television
DVB	Digital Video Broadcasting
ECMA	European Computer Manufacturers Association
EIT	Event Information Table
EPG	Electronic Programme Guide
ES	Elementary Stream
ETM	Extended Text Message
ETSI	European Telecommunications Standards Institute
ETT	Extended Text Table
EUC-JP	Extended Unix Code (-JP)
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GPS	Global Positioning System
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol

I/O	Input/Output
IDL	Interface Definition Language
IEC	International Electrotechnical Commission
IHDN	In-Home Digital Network
IP	Internet Protocol
IPR	Intellectual Property Rights
IRD	Integrated Receiver Decoder
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JDK	Java Development Kit
JFIF	JPEG File Interchange Format
JMF	Java Media Framework
JPEG	Joint Photographic Experts Group
JVM	Java Virtual Machine
LMDS	Local Multipoint Distribution System
Mbit/s	1 000 000 bits per second
MGT	Master Guide Table
MHEG	Multimedia Hypermedia Experts Group
MHP	Multimedia Home Platform
MIME	Multipurpose Internet Mail Extensions
MMDS	Multipoint Microwave Distribution System
MPAA	Motion Picture Association of America
MPEG	Moving Picture Experts Group
NaN	Not a Number
NPT	Normal Play Time
NRT	Network Resources Table
NVOD	Near Video on Demand
OC	Object Carousel
OCAP	OpenCable™ Application Platform
OID	X.509 Object Identifier
OS	Operating System
OSD	On-Screen Display
PAT	Program Association Table
PCR	Program Clock Reference
PES	Packetized Elementary Stream
PFR	Portable Font Resource

PID	Packet Identifier
PJava	Personal Java Profile of Java technologies
PMT	Program Map Table
PNG	Portable Network Graphics
PPP	Point-to-Point Protocol
PSI	Program Specific Information
PSIP	Program and System Information Protocol
PSTN	Public Switched Telephone Network
PTC	Physical Transmission Channel
PTS	Presentation Time-Stamp
RAM	Random Access Memory
ROM	Read Only Memory
RRT	Rating Region Table
SCTE	Society of Cable Telecommunications Engineers
SDT	Service Description Table
SI	Service Information
SMATV	Satellite Master Antenna Television
SMTP	Simple Mail Transfer Protocol
STD	System Target Decoder
STT	System Time Table
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TS	Transport Stream
TTC	Telecommunication Technology Committee (Japan)
TVCT	Terrestrial Virtual Channel Table
UCS	Universal Multiple-Octet Coded Character Set
UCS	Universal (Coded) Character Set
UDP	User Datagram Protocol
UI	User Interface
unicode	Unicode™
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
UTF	UCS Transformation Coding
UTF	UCS Transformation Format
UU	User to User

VCT	(Cable) Virtual Channel Table
VM	Virtual Machine
W3C	World Wide Web Consortium
WAN	Wide Area Network
XHTML	eXtensible HTML
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language

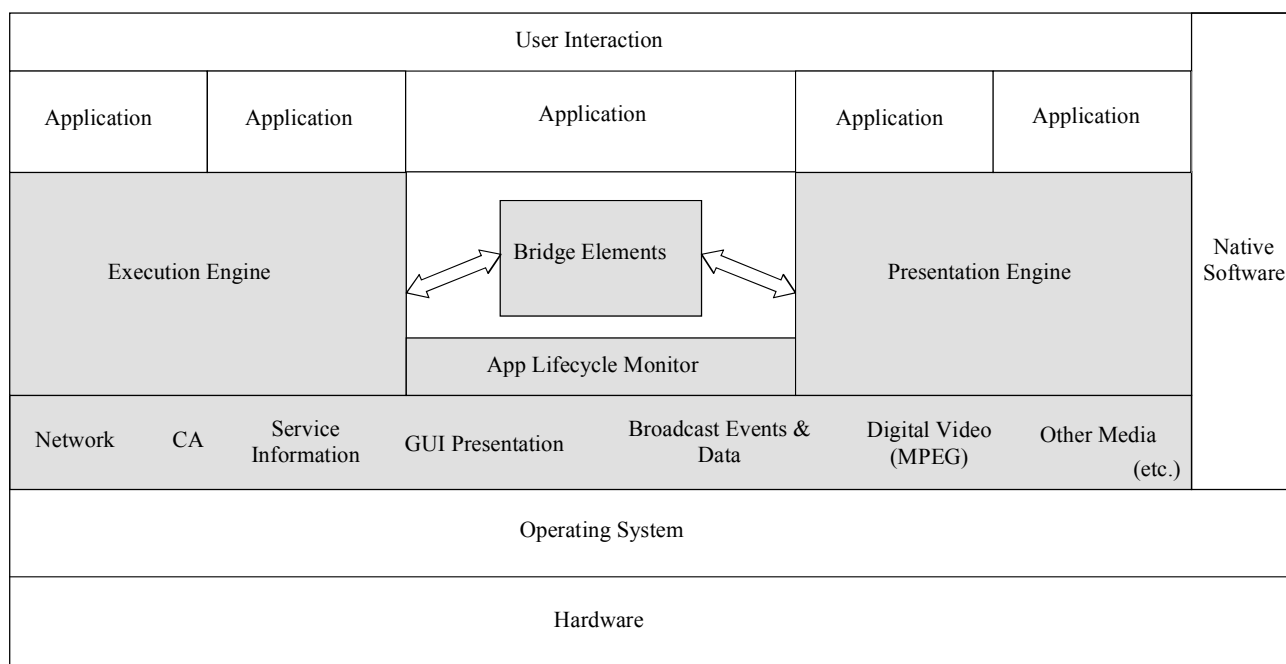
#### 4 Recommendation

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

##### 4.1 Basic architecture

The high-level architecture of APIs and middleware for digital interactive TV can be regarded as comprising 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 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.



T0910150-00

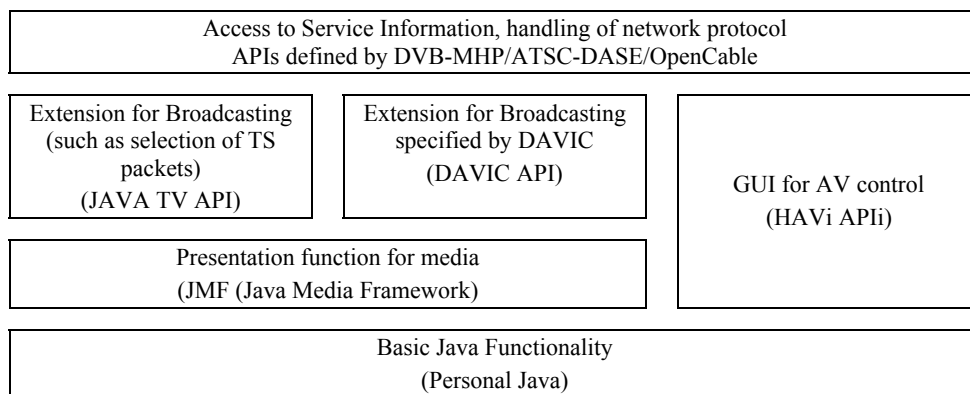
**Figure 1/J.200 – Application environment system architecture**

Where specified, the application environment should draw from the following architectural elements:

- 1) Presentation Engine/Execution Engine – These vary between the different specifications. In some cases one or the other is optional.
- 2) Bridge Elements – This is a mechanism for applications that allows bidirectional mapping between Java APIs and DOM, ECMAScript objects and methods.
- 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 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 software that is either legacy software or software written using additional APIs and functionality outside the specified Application Environment.

#### 4.2 Execution Engine

Figure 2 shows the structure of the Execution Engine and indicates the origin of the various elements.



**Figure 2/J.200 – Structure of the Execution Engine**

Key standards that should be supported by the Execution Engine are:

- 1) Personal Java Application Environment – This API consists of the java.\* package as specified by Sun Microsystems.
- 2) Java TV – This API consists of the javax.tv.\* packages as specified by Sun Microsystems.
- 3) Java Media Framework – This API consists of the javax.media.\* packages by Sun Microsystems.
- 4) DAVIC1.41 Specification part 9 – This API consists of the org.davic.\* package as specified by DAVIC.
- 5) HAVi Level 2 User Interface – This API consists of the org.havi.ui as specified by HAVI and subsets of java.awt.



### 4.3 Presentation Engine

Figure 3 shows the structure of the Presentation Engine and indicates the origin of the various elements.

CSS	ECMAScript Engine & DOM	HTML/XHTML DTD	
TV Extensions			XML Parser (optional)
User Agent (Browser)			
Receiver Functionality	Digital Video (MPEG)	Monomedia	Network

**Figure 3/J.200 – Structure of the Presentation Engine**

The key standards that should be supported by the Presentation Engine are:

- 1) Common modules defined in X-HTML Modularization and/or HTML as defined by W3C.
- 2) CSS style sheet to describe presentation style as defined by W3C.
- 3) DOM APIs to dynamically change the contents of HTML or X-HTML documents as defined by W3C.
- 4) ECMAScript.
- 5) 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.
- 6) Standard media types such as image/jpg, image/png, audio/basic and TV specific media types such as video/mpeg as defined by ISO.
- 7) XML parser as defined by W3C.





## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
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
<b>Series J</b>	<b>Cable networks and transmission of television, sound programme and other multimedia signals</b>
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
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 and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

