NIST GCR 02-835

Report on Interactive Television Technology & Standards

Michael A. Dolan
Technical Consultant
Television Data Broadcasting

National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NIST GCR 02-835

Report on Interactive Television Technology & Standards

Prepared for
U.S. Department of Commerce
OU
National Institute of Standards and Technology
Gaithersburg, MD 20899

By
Michael A. Dolan
Technical Consultant
Television Data Broadcasting

September 2002

U.S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary for Technology

National Institute of Standards and Technology
Arden L. Bement, Jr., Director

Report on Interactive Television Technology & Standards

Prepared for the National Institute of Standards (NIST)

By Michael A Dolan

11-August-2002
Final Report

Acknowledgements

The author would like to thank Dr. Glenn Adams and Mr. Sean Hayes for their work in the area of ATSC and DVB markup (HTML) comparison, which formed the basis for parts of Section 4.7.1 of this report; and for Dr. Adams in his overall review and critique of the report.

Table of Contents

1 Introduction

This report provides an overview and comparison of the various Interactive TV data standards specifically including:

ARIB B24 1.1
ATSC DASE-1, Version 1
Cablelabs OCAP 2.0
DVB MHP 1.1
SMPTE DDE-1

All of the above standards make use of some or all of technology that is available to today. They have modified or extended it in many cases, but fundamentally, there are not large amounts of unique development. Some of the technology that is in use can be categorized as:

Application Metadata
Application Lifecycle
Synchronization Mechanisms
Broadcast Data Protocols (MPEG-2, etc.)
Internet Protocols (Internet/Web)
Namespaces

And, the content itself:

Markup Language (W3C HTML)
Layout Control (W3C CSS)
Scripting (ECMA Script)
Script API's (Native Objects, W3C DOM)
Byte Code Virtual Machine (Sun® Java VM)
Byte Code API's (Sun JavaTV®, PJAE, JMF & HAVi UI)
Other Formats (Fonts, PNG, etc)

The above technology will be reviewed in general, and then its application by each standard will be discussed.

Note that references to further reading material are noted in the text by symbols (often related acronyms) enclosed in brackets. For example, [MPEG-SYS], refers to the MPEG-2 Systems Standard as listed in the Bibliography at the end of he report.

2 Overview of the Technology & Architecture

All of the ITV standards under consideration in this report are based on generally available technology that is customized and applied to the ITV application. And, all the standards have a considerable amount of overlap in their use of the technology, especially at the functional level. While there are many details that are different between the standards, it is important to keep in mind that at the very high level, the same core technology is used ubiquitously.

The comparison in this report is based on a couple of high level categorizations listed above. Each of these will be discussed in detail below to provide some basic background for the standards comparison that follows.

In order to discuss some of the ITV-specific characteristics, it will first be helpful to establish some terminology and concepts. First, some basic video and audio concepts are discussed. A *programme*[1] is a finished work that is identified by something such as title and episode and perhaps other characteristics. A single *programme* is often distributed on tape. A *service* (also known as a channel, or virtual channel) is continuous concatenation of *programmes*. An *event* is a binding of a *programme* to a *service* at a particular time. An electronic program guide (*EPG*) is a presentation (and often an interactive user interface for tuning) of *events* on each *service*.

A *data service* is a collection of (data) *applications* that may or may not be associated with a video/audio *service*. An *application* (also known as an *enhancement*) is a collection of *resources*. A *resource* is one of a collection of basic data components. An example of a data component is a *file*, a *stream*, or a *trigger*. A *file* is the same as is used for describing a file in a computer file system – a bounded sequence of bytes. A *stream* is also like that used in a computer system – an unbounded sequence of bytes. A *trigger* has no analogy in normal computer systems. It is a linkage of a *file* to specific point in the *programme* time line.

2.1 Application Metadata

Application metadata is the collection of additional information about the data service application and its components. The metadata describes the organization, the attributes of the components and potentially the emission characteristics of the components. Some of the metadata is consumed by the emission system to configure the transport, and the rest is conveyed in the transport and consumed by the receiver.

An example of where the emission system consumes some metadata in order to configure the transport is: in order to configure a broadcast file system such as a DSM-CC Object Carousel, one must specify a variety of information such as emission rate and aggregate maximum bandwidth. However, that information is currently not standardized by any system, so it would not be meaningful to compare.

[1] The English spelling is used to distinguish it from the MPEG “Program”, which is actually a service and has no relation to this definition.

We will focus here only on the metadata that is transmitted along with the application and its components. For example, for each file in the file system, there is normally a name, as well as other properties.

2.2 Application Lifecycle

The application lifecycle is the definition of the beginning, end of the existence, and other state changes of an application along with its expected behavior in a viewer's television. The lifecycle is usually signaled either explicitly in the transport, or sometimes implicitly in the transport (or both). In addition, there are user events such as disabling the data service and changing channels that affect the application.

Data services and thus applications may be announced in advance for future programming. This is generally coupled with the SI announcement mechanism for the target transport. For example, the Event Information Table (EIT - generally the same concept in all SI) is often used for this purpose.

The presence of a data service and thus applications is usually signaled in the transport through new MPEG-2 tables and descriptors designed specifically for this signaling.

There are several classes of applications. Some applications are "bound" to the service. That is, they may only be active when the service is being actively processed" (typically when viewing the related video and audio). Some applications are "unbounded", and may continue being active after the service is no longer being processed. Note that unbounded applications must at least have been in some active service to become active, or actively retrieved by the receiver over a return channel and started. There are also resident applications which may be built into receiver equipment, but these are not considered here.

Often, the combination of actively receiving the service (tuned to the video and audio of a service, for example) and the signaling of the presence of the application indicates that the application it supposed to start "as soon as possible". But sometimes the signaling is more explicit, depending on the system.

Likewise, when the signaling of an application ceases in the transport, it generally means that the resources for that application are no longer available, and the application may stop being active. The affect of the viewer switching between services and the notion of an unbound application may have varying results.

The above techniques are all implicit life cycle signals, however it is also possible to explicitly signal the state of an application through MPEG-2 structures.

2.3 Synchronization Mechanisms

The idea that video and audio are synchronized is a well-understood (if sometimes poorly implemented) idea. This is commonly known as "lip sync" and even small variations can be quite noticeable to the viewer. MPEG-2 Systems provides the basic mechanism for accomplishing this through the use of the PCR and PTS signaling.

The synchronization of "data" is much more complex. On the surface, one generally wishes to composite a data item with some video (or audio) and have the frames coincide to achieve a composite visual affect. MPEG carefully defined a decoder model to permit small access units of video or audio to be modeled as though they were decoded in zero time[2]. This general model does not extrapolate to "data" which can be arbitrarily large and arbitrarily complex (and thus have an extended decode time that cannot be ignored). Therefore, as a general rule there is no support for synchronized data in current systems except for a single data item, generally known as a trigger.

A trigger is a resource that binds another resource to the receiver's System Time Clock (STC) for this service. Usually, this is meant to synchronize the display of a resource along with a set of video frames. The triggers can be loosely synchronized, or tightly synchronized. The former is when the trigger is sent without any time reference and the receiver is expected to operate on it "now". A synchronized trigger is one that has a time reference and thus can potentially provide sub-frame level accuracy. In general today, the triggers are mostly asynchronous in practice.

2.4 Broadcast Data Protocols

There are some basic transport-layer data protocols that support the delivery of items other than video and audio (i.e. "data"). The basic framework is provided in MPEG-2 Systems [MPEG-SYS]. However, there are some other core elements built on the less well known MPEG standard, Digital Storage Media, Command and Control [DSMCC]. The latter provides the framework for carriage of files and IP packets, the two most common carriage mechanisms used by data applications.

The analog transports are generally constrained to the carriage of IP packets. Thus, higher level functions must build on the IP packet foundation. See section 2.5 for more information on IP.

2.4.1 MPEG-2 Foundation

MPEG-2 Systems [MPEG-SYS] provides the basic framework for the transport of data. It provides the ability to multiplex many information streams into a single transport bitstream. And, it provides the ability to organize these information streams into sets of services (known in

[2] While the model is in zero time, practical implementations are non-zero of course, and account for the difference through additional buffering in the decode pipeline. However, the designs still presume a small bounded time due to access unit size limits, which cannot be done for "data".

MPEG-2 as Programs). A variety of service metadata is also defined, as well as the framework for simple metadata extensions[3] in the form of a common syntax for both tables and descriptors.

MPEG-2 Systems also defines the time reference with which to control the display, buffering and multi-stream synchronization.

Finally, MPEG-2 Systems defines the buffer model framework for both the SI as well as the video, audio and data streams.

Note that MPEG-2 Systems does not define any common means for announcement of future services or programming.

2.4.2 DSM-CC

DSM-CC is the Digital Storage Media Command & Control [DSMCC] defined by MPEG to provide a relatively elaborate framework for video on demand (VOD) services. Overall, it defines a distributed object oriented system for control of the video playback, as well as many infrastructure systems to support this basic function. However, very little of the complex nature of DSM-CC is included in any ITV standard today, and its use is constrained to some supplementary transport mechanisms for files and IP packets.

There are three primary DSM-CC infrastructure components that are used by the ITV systems for the carriage of "data":

Addressable Sections
Data Carousel
Object Carousel

Addressable Sections are used to carry IP packets and are defined in [DSMCC], Section 9 via Amendment #1. This provides physical layer addressing, including support for IP Multicast.

The data carousel is a means for signaling "modules", or files. It was intended originally to provide firmware updates to VOD receivers over the broadcast transport, but its application has been extended by ITV standards to provide relatively robust module delivery. The data carousel is further described in [DSMCC], Section 7.

Finally, a special, enhanced case of the data carousel is the object carousel. This overcomes some limitations inherent in a "firmware download" design of the data carousel and is further described in [DSMCC], Section 11 and Annex F.

[3] The definition of metadata in the form of MPEG-2 descriptors is suitable for "simple" metadata, but not for large amounts or complex metadata, or metadata that requires synchronization. The reader is referred to ISO MPEG work in process, 13818-1:2000 Amendment #1, which addresses these more complex metadata scenarios.

Oddly, these three elements are **not** core to DSM-CC's overall intent and design, but nevertheless have become core to ITV standards for data delivery. The ITV standards have often constrained and/or extended these DSM-CC elements[4].

2.5 Internet Protocols

All ITV receivers have support for some level of Internet Protocols (IP). Some only support forward delivery of IP Multicast, while some of the ITV standards use a bi-directional "return channel". This means that in addition to the broadcast transport, there is another general purpose communications channel available to the receiver. This can take the form of a proprietary modem line connection, but most often it is an Internet link. The basic protocols that are used are:

PPP
IP
UDP
TCP
SAP/SDP
HTTP
SSL (TLS)
NNTP
SMTP
UHTTP

These are discussed further below.

2.5.1 PPP

The point to point protocol [PPP] is used to establish a serial communication link between two devices for exchanging IP packets. It is also used to negotiate Maximum Transmission Unit (MTU), gateway and other IP-related information during the connection establishment. This is the common link layer protocol when using a Plain Old Telephone System (POTS) modem in one's home for example.

2.5.2 IP/UDP/TCP

The foundation to the Internet protocols are three basic ones, Internet Protocol [IP], User Datagram Protocol [UDP], and Transmission Control Protocol [TCP]. IP provides a basic virtual addressing system using Internet addresses, and provides some route control. It abstracts the

[4] It is important to note that many of the ITV data carriage concepts were only recently added in various amendments to [DSMCC], so the reader is cautioned to get the most current versions of the document(s) with all amendments.

packets of data from the physical addressing needed for physical layer delivery. The IP header itself is simple, however, the routing behavior surrounding the delivery of IP packets is quite complex covered by many, many standards in the Internet Engineering Task Force (IETF).

UDP provides a basic datagram service including a source and destination port numbers. The IP address in the IP packet header plus the port number in the UDP header form a complete service address for delivering packets between two application layers. UDP provides only single modest size blocks of data at a time, with no flow control, and no guaranteed delivery. It is not connection oriented. All connection, flow control and retransmission of dropped data is at the discretion (and responsibility) of the application layer. UDP is most often used for multicast (one to many) delivery. The algorithm for processing UDP packets is relatively simple.

TCP, like UDP, also defines ports, but is connection oriented, provides for flow control, and guarantees delivery, re-transmitting data as necessary. The algorithm for processing a TCP connection is significantly more complex than for UDP.

2.5.3 SAP/SDP

The Session Description Protocol [SDP] is used to announce multimedia (primarily video and audio) sessions, normally on the Internet. However, this protocol has been adapted to broadcast streams where the “media” is data instead of just video and audio.

The Session Announcement Protocol [SAP] is used to provide a very elementary delivery wrapper for the SDP records, which are textual cannot be transmitted alone. This “wrapper” is usually sent in a UDP packet, and often as multicast addressed.

2.5.4 HTTP

HTTP is the HyperText Transfer Protocol [HTTP]. It was designed to be a more efficient (relative to FTP) when there was a single resource being transferred. It forms the basis for most “web” accesses by browsers today.

2.5.5 SSL (TLS)

SSL, or the Secure Socket Layer, is formally known as Transport Layer Security [TLS]. While it is a general purpose protocol for providing encrypted TCP transactions, it most well known for its use when resolving the URL scheme name, “https:” familiar to users of web browsers for “secure” transactions. TLS provides public key exchange and encryption for the life of a TCP connection, and can be used relatively transparently for any application layer that uses TCP.

2.5.6 NNTP

NNTP, or the Network News Transfer Protocol [NNTP], is used to access the Internet News feeds. This is the open Internet protocol that provided an early foundation for the development of “chat rooms”.

2.5.7 SMTP

SMTP, or the Simple Mail Transport Protocol [SMTP], is the basic protocol used to send and route email on the Internet.

2.5.8 UHTTP

The Unidirectional HyperText Transport Protocol [UHTTP] is used to carry broadcast resources. This is functionally analogous to the DSMC-CC data carousel.

2.5.9 DSM-CC U-U

DSM-CC defines objects that use IP as a mechanism for communication between the object skeleton and the object implementation using Remote Procedure Call (RPC) techniques as further defined in the Common Object Reference Broker Architecture (CORBA). CORBA is a system that provides the framework for a distributed object software system.

2.6 URI Schemes

There are several URI schemes in use within the various standards. Some are URL’s which define a specific location; and some are URN’s which require more information to resolve. Some are transport independent, and some are transport dependent. The reader is referred to RFC 2396 [URI] for more details about URI schemes in general. The ones we will discuss are the transport independent ones:

tv: [URI-TV]
lid: [URI-LID]
http: [URI-HTTP]
https: [URI-HTTPS]

URI schemes in general are used by applications to refer to resources related to the broadcast. Some resources are in the broadcast transport itself and the URI can be resolved through inspection of labeling that occurs in the broadcast. Often, this takes the form of the resource being directly labeled with the URI. In other cases, the URI is implied from a directory

hierarchy in the broadcast file system. The latter may also be defined in terms of a transport-dependent scheme. Several of these are in use, but not elaborated on in this report.

2.7 Content Formats

An overview of the basic ITV content formats is provided in this section. They are grouped into:

Markup Languages
Layout Control
Scripting
Script API's
Byte Code Virtual Machine
Byte Code API's
Other Content Formats

2.7.1 Markup Languages

Markup languages have been around for some time – since the invention of standalone word processors. Those early word processors had a (proprietary) way to describe how each part of the document was to appear when printed - for example, what font to use for a word, and then which size and style for that font. This is still true (including the proprietary part) today in the current releases of Microsoft Word and Adobe Acrobat – two well-known proprietary markup languages.

The idea of a standard markup language began in the 1980's with the definition of the Standard Generalized Markup Language [SGML] published as an ISO standard in 1986. SGML allows one to define an application-specific document structure. SGML itself is not the document markup language, but rather it is the **framework** to define specific document structures.

SGML was mostly unheard of until the creation of HTML 1.0 and the idea of hyperlinks. This early SGML language was little more than a way to write a (technical) paper and be able to provide links that allowed the reader to find the references (in the traditional sense) that the paper used. HTML, versions 2 and 3 quickly ran with this basic idea and we have the common baseline for what is published on the web today, namely HTML version 3.2. The first step in a more formal definition of HTML came in version 4, that included a Document Type Definition (DTD). While it is possible to create a generalized SGML parser and then use the HTML4 DTD, this is a complex design and not done in practice. All browser implementations today still have an implied DTD for HTML 4.

The lack of formal DTD usage in practice has led to divergent implementations in products, notably today Microsoft IE and Netscape Navigator. And, further aggravating interoperability is that HTML content is generally poorly formed by the authors and/or authoring tools. To help address these issues, XML was defined.

XML is an ISO-compliant subset of SGML. It is purposely much simpler than the full SGML and permits a generalized parser to be implemented easily and efficiently. So, in a nutshell, XML was designed to motivate manufacturers to build these simpler, but more generalized, environments.

And, W3C is building everything going forward on this XML subset of SGML. The first obvious step was to recast HTML4 into an XML DTD, which they did. This is now called XHTML 1.0, and is functionally equivalent to HTML4. So, the validity of XML seems established through W3C's forward development, as well as implementations that are starting to appear, which confirm its small, compact design goals. So, several formal DTD's exist in practice today.

The DTD's from which the ITV work has been derived are:

HTML 4.0 (all 3)
XHTML 1.0
XHTMLMOD 1.0

2.7.2 Layout Control

Layout is the control of the markup appearance on a particular media. For example, when a markup language defines an element as a "Level One Heading", it is the layout control that determines for the video device (or printer) that this means it should be rendered in Arial, 14pt Bold, for example. This display separation of markup elements and layout is a critical feature of re-purposing content that has so far not be very well exploited in web server designs.

HTML 3 had a mixture of markup language and layout control. The first attempt in separating these was done with HTML, version 4.0 [HTML]. HTML 4 enabled the use of external style sheets and deprecating the built-in style elements of HTML 3.2. The formal definition of Cascading Style Sheets [CSS1] was defined by W3C to once again separate the markup and the rendering style. This has been only mildly successful, with the vast majority of authors still using the simpler inline style elements and attributes of HTML 3.2 (which is still permitted in HTML 4). CSS1 is only now gaining more widespread usage as the repurposing of content becomes more painful to the authors. CSS2 has added media-specific controls, which makes layout even easier to manage.

2.7.3 Scripting

Scripting primarily offers the ability to dynamically alter the markup and layout of an ITV application based on viewer input or receiver capability. The scripting most often used for this purpose is a language defined by [ECMA] called ECMAScript [ECMASCRIPT2]. There are 3 "editions" of ECMAScript. Editions 1 and 2 are substantively the same, the difference being

only editorial changes as part of the ISO 16262 publication process. The 3rd edition [EMCASCRIPT3], is available only from ECMA at this writing, and has primarily added exception processing over the 2nd edition.

[5]*ECMAScript is an object-oriented programming language for performing computations and manipulating computational objects within a host environment. ECMAScript is not intended to be computationally self-sufficient; indeed, there are no provisions in the specification for input of external data or output of computed results. Instead, it is expected that the computational environment of an ECMAScript program will provide not only the objects and other facilities described in the core specification but also certain environment-specific host objects, defined by actual applications and implementations.*

ECMAScript was originally designed to be a Web scripting language, providing a mechanism to enliven Web pages in browsers and to perform server computation as part of a Web-based client-server architecture.

Some of the facilities of ECMAScript are similar to those used in other programming languages; in particular Java [JAVA]. ECMAScript syntax intentionally resembles Java syntax. ECMAScript syntax is relaxed to enable it to serve as an easy-to-use scripting language. For example, a variable is not required to have its type declared nor are types associated with properties, and defined functions are not required to have their declarations appear textually before calls to them.

ECMAScript is object-based. Basic language and host facilities are provided by objects, and an ECMAScript program is a cluster of communicating objects. An ECMAScript object is an unordered collection of properties each with zero or more attributes that determine how each property can be used. A primitive value is a member of one of the following built-in types: Undefined, Null, Boolean, Number, and String; an object is a member of the remaining built-in type Object; and a method is a function associated with an object via a property.

ECMAScript also defines a set of built-in operators that may not be, strictly speaking, functions or methods. ECMAScript operators include various unary operations, multiplicative operators, additive operators, bitwise shift operators, relational operators, equality operators, binary bitwise operators, binary logical operators, assignment operators, and the comma operator.

2.7.4 Script API's

ECMAScript has a variety of native objects that assist the programmer in basic computing functions. In addition, a set of API's that is critical for the primary purpose of scripting is the Document Object Model [DOM].

[5] The remaining text in this section on scripting was taken substantially from [ECMASCRIPT3], Section 4.

2.7.4.1 Native Objects

ECMAScript supports constructors that create objects by executing code that allocates storage for the objects and initializes all or part of them by assigning initial values to their properties. All constructors are objects, but not all objects are constructors. Each constructor has a Prototype property that is used to implement prototype-based inheritance and shared properties. Objects are created by using constructors in new expressions; for example, new String("A String") creates a new String object. Invoking a constructor without using new has consequences that depend on the constructor. For example, String("A String") produces a primitive string, not an object.

The native objects supported are:

Global
Object
Function
Array
String
Boolean
Number
Math
Date
RegExp
Error

For more details, see [ECMASCRIPT3], Section 15.

2.7.4.2 Document Object Model

The Document Object Model (DOM) is a general term used to describe the framework for defining and modifying a “document” within a programming environment. The W3C has defined this in terms of generic HTML and XHTML documents, and defined several sets of DOM API’s supporting this. The W3C DOM specifications in use are DOM-1 [DOM1] and DOM-2 [DOM2]. In addition, SMPTE has defined a DOM level 0 [DOM0]. The general framework includes object definitions (aka API’s) that permit interrogation and modification of the document instance(s). These objects are defined in IDL (see [CORBA]).

DOM-0 describes current practice on the web, including early ITV devices using DDE-1 or something close to DDE-1. DOM-0 also defines a couple of navigation objects, in addition to the DOM objects. This specification, while in IDL, has only ECMAScript bindings.

DOM-1 and DOM-2 have structured the document model into components that are markup-independent, then added the markup language dependent specification (i.e. HTML). The components of the DOM-2 are:

Core
Views
Events
Style
Traversal and Range
HTML

Both DOM-1 and DOM-2 have both ECMAScript and Java language bindings that accompany their IDL definitions.

2.7.5 Byte Code Virtual Machine

There are some ITV applications that are too complex to develop and maintain easily using a scripting language. The solution to this is a more formal programming language. Since it is desirable to have a machine independent encoding of the program and not require receivers compile source code, an intermediate "byte code" language is used. The byte code language and corresponding interpreter in the standards is Sun Microsystems' Java Virtual Machine [JVM]. In addition, there is a source code language of the same name that encodes (compiles) to the JVM byte codes [JAVA].

The byte codes are generally a compact binary representation of the human readable form of the computer language. Thus, they are efficient to transmit compared to ASCII source code. However, the authoring requires more sophisticated authoring systems and usually a more sophisticated programmer.

2.7.6 Byte Code API's

There are several classes (aka API's) in use for ITV systems. First, there are the basic ones that are not ITV specific:

Personal Java Application Environment [PJAE]
Java Media Framework [JMF]
W3C DOM [DOM2] & CSS [CSS2]

Then, there are classes new to ITV:

Java TV [JAVATV]
HAVi User Interface [HAVI-UI]
DAVIC Media [DAVIC-141]
OpenCard Framework [OCF]

These are discussed in more detail in the following subsections.

2.7.6.1 Sun JavaTV

Java TV [JAVATV] is a transport independent abstraction for access to the MPEG-2 SI (as extended by the transport standards); with additional support for the unique properties of broadcasting traditional data items such as files. It is slightly more favorable to DVB's SI over the other systems, since that was its first supported model, however in general it is 99% abstract.

It contains the following packages:

javax.tv.carousel
javax.tv.graphics
javax.tv.locator
javax.tv.media
javax.tv.media.protocol
javax.tv.net
javax.tv.service
javax.tv.service.guide
javax.tv.service.navigation
javax.tv.service.selection
javax.tv.service.transport
javax.tv.util
javax.tv.xlet

2.7.6.2 Sun PJAE

The Personal Java Application Environment [PJAE] is a profile of the more widely known Java Development Kit [JDK], version 1.1.8 and 1.2.2.

The PJAE contains subsets of the following standard JDK packages:

java.applet
java.awt
java.awt.datatransfer
java.awt.event
java.awt.image
java.awt.peer
java.beans
java.io
java.lang
java.lang.reflect
java.math
java.net

java.rmi
java.rmi.dgc
java.rmi.registry
java.rmi.server
java.security
java.security.acl
java.security.cert
java.security.interfaces
java.security.spec
java.sql
java.text
java.text.resources
java.util
java.util.jar
java.util.zip

In addition, it introduces the following new packages:

com.sun.awt
com.sun.lang
com.sun.util

The derived specifications are allowed to subset these packages (in the context of pJAE) and have often defined "unsupported features" with only the signatures of the class in place. So, there are variations in the standards, as well as variations in the versions of the PJAE specification being referenced, which has led to some divergence.

2.7.6.3 Sun JMF

The Java Media Framework [JMF] is a set of packages that control streaming media in a network setting, and is a generic package used outside the ITV environment. JMF 1.0 was originally designed to control streaming video and audio from a disk file either locally or from a "video" file server over a network. In order to address the unique nature of broadcast streams, JMF was extended by DAVIC with its org.davic.media package, discussed later. The JMF 1.0 used for ITV contains the following packages:

javax.media
javax.media.protocol

The javax.media package has an extensive set of classes and interfaces. Please see [JMF] for more details.

2.7.6.4 HAVi UI

The Home Audio Video Interoperability [HAVI] has defined an extensive set of Java-based packages for consumer electronics devices. One part of HAVi that has been adopted by the standards supporting Java, includes the user interface (UI) packages. This set of packages requires a subset of the standard java.awt package as more fully defined in [HAVI].

The HAVi User Interface contains the following packages:

org.havi.ui
org.havi.ui.event

2.7.6.5 DAVIC Media

The Digital Audio-Visual Council [DAVIC] developed some Java packages for media control as published in their version 1.4.1 Specification [DAVIC-141].

The entire specification is extensive, but the packages used by ITV standards are limited to two packages defined in two of the annexes of Part 9 of [DAVIC-141], and provide some media control for broadcast media (JMF was originally designed for reading media files from disk):

org.davic.media
org.davic.resources

2.7.6.6 W3C DOM

The W3C [W3C] developed Java package bindings of their DOM (including CSS) API's, analogous to the Scripting API's described above. These packages are:

org.w3c.dom
org.w3c.dom.css
org.w3c.dom.events
org.w3c.dom.html
org.w3c.dom.stylesheets
org.w3c.dom.views

2.7.6.7 OpenCard Framework

The OpenCard Framework [OCF] is an open standard Java package that provides access to smart cards and related conditional access services. The Java packages are:

com.gemplus.opencard.terminal
com.ibm.opencard
com.ibm.opencard.access
com.ibm.opencard.buffer
com.ibm.opencard.crypto
com.ibm.opencard.dictionary
com.ibm.opencard.factory
com.ibm.opencard.handler
com.ibm.opencard.isofs
com.ibm.opencard.script
com.ibm.opencard.service
com.ibm.opencard.signature
com.ibm.opencard.terminal
com.ibm.opencard.terminal.pcsc
com.ibm.opencard.terminal.pcsc10
com.ibm.opencard.terminal.pcscmig
com.ibm.opencard.util
com.ibm.tools.rejar
com.ms.security
netscape.security
opencard.core
opencard.core.event
opencard.core.service
opencard.core.terminal
opencard.core.util
opencard.opt.applet
opencard.opt.applet.mgmt
opencard.opt.database
opencard.opt.emv.mgmt
opencard.opt.iso.fs
opencard.opt.javacard
opencard.opt.ms
opencard.opt.netscape
opencard.opt.security
opencard.opt.service
opencard.opt.signature
opencard.opt.terminal
opencard.opt.terminal.protocol
opencard.opt.util
opencard.tests.service
opencard.tests.terminal
opencard.tests.terminal.codes
opencard.tests.terminal.pcsc
opencard.tests.testframe

2.7.7 Other Content Formats

There are a variety of miscellaneous content formats (aka “media types”) that are supported by ITV standards. These include:

Portable Font Resource [PFR]
Portable Network Graphics [PNG]
Multiple Network Graphics [MNG]
Graphics Interchange Format [GIF]
Joint Picture Experts Group [JPEG]
Basic Audio [PCM]
ASCII Text

PFR is a binary font format developed by Bitstream and first published openly by DAVIC. It is an outline format that is in common use on the Internet today and included in many proposed ITV standards that define a broadcast font file format.

GIF is a very popular lossless graphics file format that provides good compression for computer graphics images. It is widely used on the Internet and in many standards. It is generally not good for photographic images.

PNG is a lossless graphics image file format first defined by W3C as a replacement for GIF, explicitly defined to avoid LZW patent assertion claims. It is better designed than GIF and includes alpha channel and gamma correction, but not some animation features of GIF.

MNG, in a nutshell, is a means to animate multiple PNG files. This general feature is available in GIF, but the functions are much more robust in MNG. Typically, only a small subset of MNG is supported in ITV systems.

JPEG is a lossy image compression scheme used for photo images. While the full JPEG specification can be complex, most ITV systems are constrained to the “JFIF” profile made popular on the Internet. It includes the JPEG “baseline” profile, along with a few extensions for thumbnail images.

PCM is a simple mono-channel digital audio format, used for sound affects and voice enhancements.

ASCII Text is the US ASCII charset with common interpretation of the CR and LF characters. This media type is commonly known as “text/plain”.

3 Overview of the Standards and Their Use of Technology

This section discusses the technical elements of the prominent ITV standards, specifically:

ARIB B24 1.1
ATSC DASE-1, Version 1
Cablelabs OCAP 2.0
DVB MHP 1.1
SMPTE DDE-1

Each of the standards organizations addressed in this report has developed (and in most cases is still developing) standards that are based on MPEG-2 Systems. However, each organization has divided and organized the documentation in slightly different ways. Some organizations developed more monolithic standards than others. A comparison of the documentation for each organization is shown in Figure 3-1. Refer to the following sections for more details on the contents of this figure.

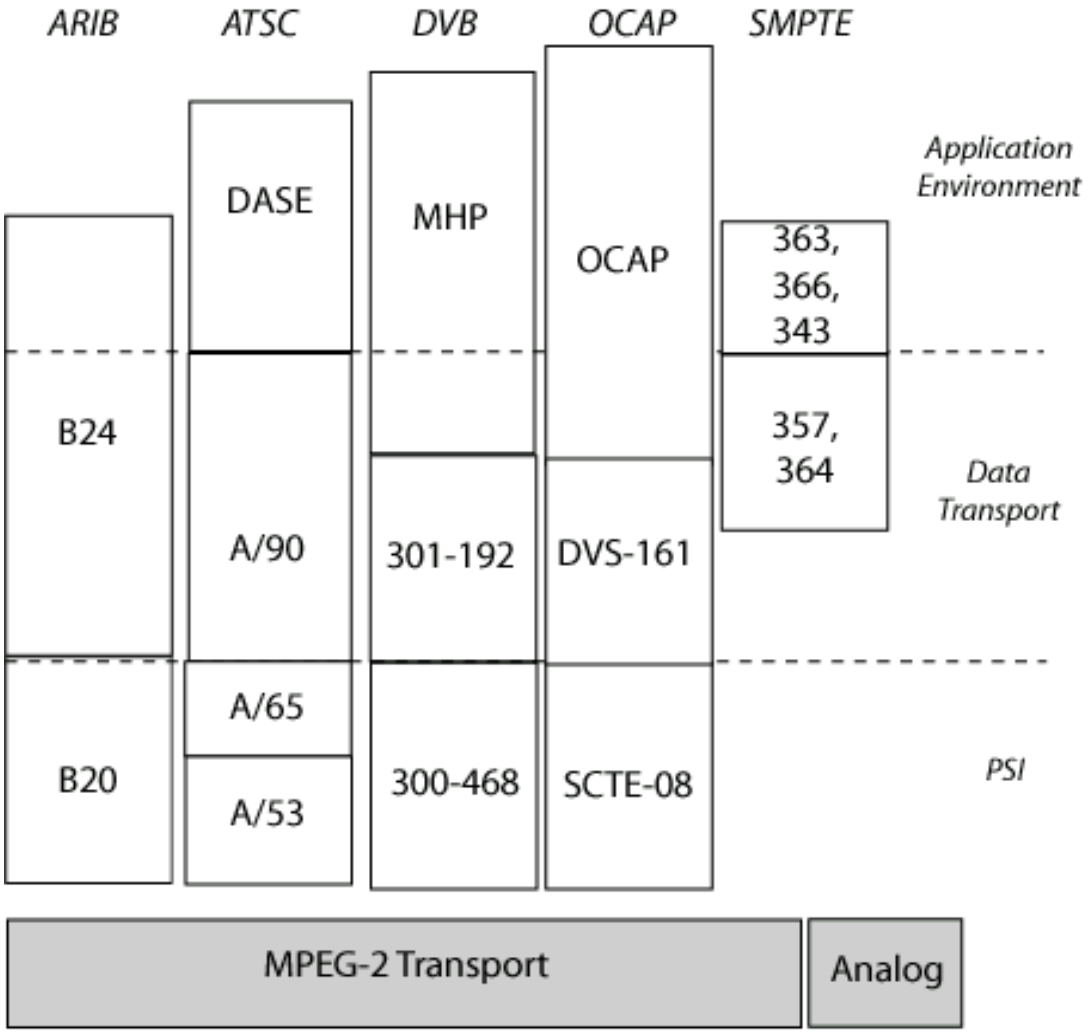

Figure 3-1. Comparison of the Standards Documentation

3.1 ARIB B24 1.1

ARIB is the Association of Radio Industries and Businesses [ARIB], a Japanese government sanctioned industry alliance for radio and television (including digital) broadcasting standards in Japan. ARIB's relationship to Japanese or international standards bodies is not known to the author. But their specifications appear to be mandated by law in Japan, so appears to be analogous to ATSC in its standing.

Standard B24 [B24] is the ITV standard for ARIB. This comparison is with the latest translation into English of B24, version 1.1 published in April 2000, and is thus somewhat dated with respect to the other standards[6]. B24 provides a vertical specification for building ITV on top of MPEG-2 Systems [MPEG-SYS] with references to B10, the ARIB standard for basic SI. This specification also includes various profiles. Multiple profiles adds an additional level of complexity to a comparison, so only the most advanced profile is addressed here. Note that some critical elements to fully understanding B24 are located in B10, no version of which is available in English. So, the author cautions the reader about the completeness of this review, especially in regard to application metadata and lifecycle.

Unlike DVB-MHP and ATSC-DASE, this version of B24 defines a declarative environment only – i.e. there is no Java environment.

3.1.1 Application Metadata

The notion of an "application" is a bit loosely defined in B24. There is a root module/file identified with a special moduleId value, and after that everything is linked by name. So there is no application global metadata.

Applications are organized on a single program element, although multiple applications are permitted per program.

For each module there is:

Name
Media type
Estimated Download Time
Expiration Time
Activation Time

[6] The current version is 3.2, but is available in Japanese only at this writing.

3.1.2 Application Lifecycle

Application lifecycle is defined only the presence or absence of the data components, and in particular, a data carousel moduleId=0x0000, along with the existence of events. There is no explicit signaling of the end of an application.

3.1.3 Synchronization Mechanisms

A unique event mechanism is used, which also makes use of the DSM-CC Normal Play Time (NPT) constructs.

3.1.4 Broadcast Data Protocols

3.1.4.1 MPEG-2 Foundation

B24 is bound to the general ARIB MPEG-2 transport defined in B10.

3.1.4.2 DSM-CC

B24 makes use of the DSM-CC data carousel messages to carry the BML files.

3.1.5 Internet Protocols

There are no Internet protocols carried in the broadcast channel, but the return channel is very robust and supports all popular Internet protocols.

3.1.5.1 PPP

PPP is supported.

3.1.5.2 IP/UDP/TCP

IP and UDP are supported for the carriage of IP Multicast packets in the forward broadcast stream; and TCP is supported in the return channel.

3.1.5.3 SAP/SDP

There is no support for SAP or SDP.

3.1.5.4 HTTP

There is support for HTTP 1.0, as well as support for FTP.

3.1.5.5 SSL (TLS)

There is no support for SSL.

3.1.5.6 NNTP

There is support for NNTP.

3.1.5.7 SMTP

There is support for SMTP and POP3.

3.1.5.8 UHTTP

There is no support for UHTTP.

3.1.6 URI Schemes

Several unique schemes are defined in B24, each of which identifies items in various locations in the receiver or broadcast:

arib: (for AV streams)
arib-dc: (for data carousel modules)
arib-ic: (for I Frames)
arib-node: (for events)
arib-series:
nvram: (for items residing in receiver NVRAM)
romsound: (to access a built-in receiver sound)

3.1.7 Content Formats

3.1.7.1 Markup Languages

B24 defines a markup language based on the XHTML 1.0 Strict DTD. There are quite a few extensions, particularly in the area of events.

3.1.7.2 Layout Control

Layout is defined with the use of a subset of CSS2, plus extensions.

3.1.7.3 Scripting

Scripting is supported with ECMAScript.

3.1.7.4 Script API's

Scripting API's include most of DOM Level 1 plus some objects unique to ARIB.

3.1.7.5 Byte Code Virtual Machine

There is no support for a byte code virtual machine.

3.1.7.6 Byte Code API's

There are no byte code API's.

3.1.7.7 Other Content Formats

B24 supports the following additional content formats:

Portable Network Graphics [PNG]
Multiple Network Graphics [MNG]
Joint Picture Experts Group [JPEG]
Basic Audio [PCM]

However, it does not support:

Portable Font Resource [PFR]
Graphics Interchange Format [GIF]

In addition, it does support MPEG-4 video and MPEG-4 audio as separate "monomedia" elements.

3.2 ATSC DASE-1, Version 1

ATSC is the Advanced Television Systems Committee [ATSC], a US government sanctioned industry alliance for terrestrial digital television in the US. It has no relationship to any US or international standards organizations, although compliance (in the US) with their standards is governed by US law.

DASE is the DTV Application Software Environment [DASE] and is ATSC's data application standard. Level 1, version 1.0 was published as a "candidate standard" by ATSC in December, 2001. At this writing, there is no legal mandate to comply with this standard. It builds on some other ATSC standards for data broadcasting (see [DATA-BOOK]), as well as MPEG-2 Systems [MPEG-SYS].

3.2.1 Application Metadata

DASE defines various application metadata in the form of an XML metadata file included as part of the ITV application and included in the transport signaling. This metadata includes for each application:

Identifier
Text Name
Text Description
Initial Resource
Metadata Language
Permissions
Application Runtime Parameters
Misc Required and Optional Receiver Capabilities
Profile/Level

And, for each resource:

Cache Directives

The metadata file containing the above metadata has the content type, "application/dase" and is specified by the transport infrastructure as the root resource of the application.

In addition to the metadata file, [A90] specifies a variety of transport-layer metadata that is not in the scope of this report.

3.2.2 Application Lifecycle

DASE application lifecycle is determined at two levels. The first level is controlled by signaling in the MPEG-2 table sections of the transport as defined in [MPEG-SYS] and [A90]. An application is loaded and started after the receiver first notices it in the transport. It is suspended when the channel is changed or it is no longer signaled in the transport. It is killed when another (possibly new) application that is being signaled needs the resources.

The second level is a function of the type of environment of the application's initial resource – XDML or JavaTV xlet. Each of these environments has its own application lifecycle. The XDML document lifecycle follows the transport layer lifecycle for its display, including any ECMAScript or Java content it references. The JavaTV xlet lifecycle once it is able to load from the lower layer, is documented in [JVM] and further in [DASE].

DASE supports shared and bound lifecycle models, but not unbounded.

3.2.3 Synchronization Mechanisms

Synchronization between the broadcast center and the receiver application is accomplished with the use of ATSC "triggers", used to carry events. While these events may be emitted in a synchronized manner, they are not "tightly" synchronized to the video and audio frames, such as might be done with MPEG-2 transport timing references. The synchronization is done based on the receipt of the event in the environment.

3.2.4 Broadcast Data Protocols

DASE itself is defined independent of the ATSC transport, however it is assumed to make use of the facilities of ATSC A/90 and related ATSC data broadcast standards.

3.2.4.1 MPEG-2 Foundation

DASE is not tied to any specific MPEG-2 construct not already needed to carry video and audio.

3.2.4.2 DSM-CC

DASE relies on the DSM-CC Data Carousel or Object Carousel for delivery of its file resources and DSM-CC Addressable Sections for delivery of IP packets. It relies on ATSC-specific mechanisms defined in A/90 to carry streams. See below for a discussion about IP packets.

3.2.5 Internet Protocols

DASE indirectly supports a variety of Internet protocols through the standard Java language packages. However, without a return channel defined, the IP protocols are confined to those

useful in a broadcast or multicast mode only. They are therefore constrained to those used for what is commonly known as IP Multicast.

3.2.5.1 PPP

There is no support in DASE for PPP, since there is no return channel.

3.2.5.2 IP/UDP/TCP

IP and UDP are supported for the carriage of IP Multicast packets in the forward broadcast stream, but there is no support for TCP.

3.2.5.3 SAP/SDP

There is no support in DASE for SAP or SDP.

3.2.5.4 HTTP

There is no support in DASE for HTTP, and hence generic web access.

3.2.5.5 SSL (TLS)

There is no support in DASE for SSL, and hence 2-way secure transactions.

3.2.5.6 NNTP

There is no support in DASE for NNTP, and hence network news.

3.2.5.7 SMTP

There is no support in DASE for SMTP, and thus Internet email.

3.2.5.8 UHTTP

There is no support in DASE for UHTTP, since the function is provided by the DSM-CC Data Carousel or Object Carousel.

3.2.6 URI Schemes

DASE supports the tv:, lid: and ecmascript: URI schemes. It also supports a unique scheme, arhive:.

3.2.7 Content Formats

DASE, like the other standards, supports two application environments – the declarative and the procedural. In addition, there are some common facilities of both environments. All are discussed here.

3.2.7.1 Markup Languages

DASE defines a W3C XHTML functional derivative, named XDML. It is a constrained subset of full XHTML (and thus HTML4). However, XDML does accommodate the *functionality* of HTML4, provided one can transcode some of the elements properly before emission. Guidelines on how one might do this transuding are contained in [DASE]. DASE also supports the XHTML Basic DTD defined by W3C. The details are summarized in Section 4.

3.2.7.2 Layout Control

DASE defines a subset of W3C CSS2 and minimally supports CSS1 (a proper subset of CSS2). The finer details of the subset are summarized in Section 4.

3.2.7.3 Scripting

DASE supports ECMAScript, 3rd Edition.

3.2.7.4 Script API's

DASE supports all the ECMAScript native objects and a subset of DOM-2, including some support for the extra functions defined in DOM-0. The DOM-2 subset is what is needed to support the functionality of DOM-0, cast into DOM-2; plus Stylesheets, Views, CSS, Mutation Events, and XML Extensions. The details of the subset can be found in Section 4.

3.2.7.5 Byte Code Virtual Machine

DASE supports the Java JVM.

3.2.7.6 Byte Code API's

DASE supports a subset of the following Java packages:

Personal Java Application Environment [PJAE]
Java Media Framework [JMF]
Java TV [JAVATV]
HAVi User Interface [HAVI-UI]
DAVIC Media [DAVIC-141]
W3C DOM [DOM2]

It does not support the OpenCard Framework [OCF].

There are many constraints placed on the above packages by the use in JavaTV (via PJAE) as well as DASE itself. In some cases, classes are defined to exist, but always return "not supported" exceptions.

In addition, DASE defines some packages unique to ATSC:

org.atsc.application
org.atsc.carousel
org.atsc.dom
org.atsc.dom.environment
org.atsc.dom.events
org.atsc.dom.html
org.atsc.dom.views
org.atsc.graphics
org.atsc.management
org.atsc.net
org.atsc.preferences
org.atsc.registry
org.atsc.security
org.atsc.system
org.atsc.trigger
org.atsc.user
org.atsc.xlet

3.2.7.7 Other Content Formats

DASE supports the following additional content formats:

Portable Font Resource [PFR]

Portable Network Graphics [PNG]
Multiple Network Graphics [MNG]
Joint Picture Experts Group [JPEG]
Basic Audio [PCM]

However, it does not support:

Graphics Interchange Format [GIF]

3.3 Cablelabs OCAP 2.0

Cablelabs, or more formally, Cable Research Laboratories, is a commercial research entity sponsored by the US cable system MSO's. It has no relationship to any US or international standards organizations. The cable industry standard organization is the Society of Cable Telecommunications Engineers [SCTE], but there is no connection between Cablelabs and SCTE, and no connection between Cablelabs and ATSC.

OCAP is the Open Cable Application Platform [OCAP], and is Cablelabs' proposal for ITV on US cable systems. Version 2.0 was published in May 2002. It is built on MPEG-2 Systems [MPEG-SYS] and is a derivation of DVB-MHP [MHP], with some consideration for ATSC data transport layer, and of course, the SCTE SI standards. It is also considered a "profile" and relies heavily on the version 1.0 specification, and is in general a superset of 1.0. Compliance with it has no legal status in the US.

OCAP is unusual amongst these standards in that it is a derivation primarily from another standard (DVB-MHP). For clarity and brevity, this discussion will frame OCAP relative to DVB-MHP rather than repeat the material in the DVB-MHP section.

3.3.1 Application Metadata

The application metadata used in DVB-MHP is adopted with a few minor additions to support the addition of unbounded applications (see the next section).

3.3.2 Application Lifecycle

The application lifecycle is identical to DVB-MHP except for the addition of unbounded and monitor applications. An unbounded application is one where it is not tied to any specific service (channel). Unlike a bound application, channel changes generally have no affect on this type of application.

The monitor application is a special built-in application that performs certain functions under network (MSO) control of the receiver. While also unbounded, it is not delivered via the broadcast and has specific, well-defined functions.

Finally, OCAP has the notion of an abstract service, which is a packaged collection of applications. This is similar to the ATSC Data Service, which may contain multiple applications.

Also, like DVB-MHP, lifecycle has more explicit control from the transport in the form of explicit events such as stop, etc.

3.3.3 Synchronization Mechanisms

Synchronization is provided with the same techniques as in DVB-HTML; namely through the DSM-CC StreamEvent object semantics. In addition, it defines analog stream events, which are mapped from the EIA-608 TEXT2 field using a dsmcc:// URI scheme.

3.3.4 Broadcast Data Protocols

OCAP is bound pretty closely to the cable MPEG-2 transport infrastructure defined in DVS-161 and SCTE-08, as well as components drawn from DVB-MHP. In addition, OCAP defines an Object Carousel based on DSM-CC, like ATSC and DVB, but somewhat divergent from them.

3.3.4.1 MPEG-2 Foundation

OCAP makes use of the same mechanisms of DVB-MHP. In addition, it relies on a special descriptor signaled in the PMT for unbound applications.

3.3.4.2 DSM-CC

As noted above, OCAP relies on a derivation of the DSM-CC Object Carousel for delivery of its file resources, and it relies on DSM-CC Addressable Sections for delivery of IP packets. It relies on ATSC and DVB-specific mechanisms to carry streams.

3.3.5 Internet Protocols

OCAP indirectly supports a variety of Internet protocols through the standard Java language packages, basically the same as DVB-MHP, except that there is support for the OOB channel for full duplex connections over the “broadcast” cable, thereby having a built-in “return channel”.

Also, when carrying IP, it is done according to SCTE-42, which permits both DVB and ATSC encapsulations of IP packets (both of which are derivations of DSM-CC addressable sections).

3.3.5.1 PPP

PPP support is included and is identical to DVB-MHP.

3.3.5.2 IP/UDP/TCP

IP and UDP are supported for the carriage of IP Multicast packets in the forward broadcast stream according to SCTE-42 for IP Multicast. For unicast, there is also support in both the broadcast stream via DSM-CC addressable sections, as well as the OOB channel. Over the OOB channel, there is also support for TCP.

3.3.5.3 SAP/SDP

There is no support for SAP or SDP (as in DVB).

3.3.5.4 HTTP

There is no support for HTTP, and hence generic web access.

3.3.5.5 SSL (TLS)

Support for SSL (TLS) is identical to DVB-MHP.

3.3.5.6 NNTP

There is no support for NNTP (news).

3.3.5.7 SMTP

There is no support for SMTP (mail).

3.3.5.8 UHTTP

There is no support for UHTTP.

3.3.6 URI Schemes

The URI schemes supported are the same as for DVB-MHP except when the scheme refers to the DVB SI. In addition, OCAP defines the ocap:// scheme used primarily for referring to services; and the dsmcc:// scheme for delivering analog service events (from CC TEXT2).

3.3.7 Content Formats

OCAP 2.0, like DVB-MHP, supports two application environments – the declarative and the procedural.

3.3.7.1 Markup Languages

OCAP supports DVB-HTML.

3.3.7.2 Layout Control

OCAP supports DVB-HTML (including the CSS support).

3.3.7.3 Scripting

OCAP supports DVB-HTML (including EMCAScript support).

3.3.7.4 Script API's

OCAP supports DVB-HTML (including EMCAScript DOM an the Java bridge).

3.3.7.5 Byte Code Virtual Machine

OCAP supports the Java JVM (identical to DVB-MHP).

3.3.7.6 Byte Code API's

In general, OCAP supports all the DVB-MHP Java API's, including those in the org.dvb packages. A notable exception is the org.dvb.event, replaced with org.ocap.event. However, there is a rather extensive set of minor deviations in the form of constraints to many of the packages. The reader is referred to the OCAP 1.0 and 2.0 specifications for all the details.

In addition, OCAP has defined a series of extensions in the package namespace, org.ocap.*. In general, the added functions include:

Module Registry Support (org.ocap.system)
Modified Event Handling (org.ocap.event and org.ocap.ui.event)
Receiver Hardware, including POD (org.ocap.hardware)
Extended Lifecycle Management (org.ocap.application)
Error Management (org.ocap.error)
In-band Network Support (org.ocap.net)
Monitor App Support (org.ocap.resource)
Special Class Loader (org.ocap.OcapSystem)
Unbound Application Support (org.ocap.service)
Closed Captioning Support (org.ocap.media)

3.3.7.7 Other Content Formats

Like DVB-MHP, OCAP supports the following additional content formats:

Portable Font Resource [PFR]
Portable Network Graphics [PNG]
Joint Picture Experts Group [JPEG]
Graphics Interchange Format [GIF]
ASCII Text

However, it does not support:

Basic Audio [PCM]
Multiple Network Graphics [MNG]

3.4 DVB MHP 1.1

DVB is the Digital Video Broadcast project [DVB], a European industry alliance for digital television. It has no relationship to individual country's standards bodies, or international standards bodies. However, its standards are regularly adopted by the European Telecommunications Standards Institute [ETSI], an internationally accredited standards body.

MHP is the Multimedia Home Platform [MHP], and is the DVB's ITV standard. Version 1.1.1 was published by ETSI in November, 2001 and is DVB MHP 1.1. Like ARIB, it is a monolithic, vertical specification and it builds primarily on MPEG-2 Systems [MPEG-SYS], although it also has references to related DVB SI specifications. Compliance with it has no legal status in Europe, although discussions are ongoing.

Like ARIB B24, MHP has multiple defined profiles. This comparison is with the complete feature set.

3.4.1 Application Metadata

MHP defines various application metadata in the transport in the Application Information Table (AIT) and related MPEG-2 descriptors.

Application Identifier (ETSI Registry Required)
Profile/Level

For each resource in the application, there is also the possibility of signaling in the Application Description File, the following items:

Filename (just the binding to the object carousel name)
Priority (of acquisition)
File Size

In addition to the items listed above, there are also a variety of transport-layer metadata that is analogous to ATSC and others, but not in the scope of this report.

3.4.2 Application Lifecycle

Application lifecycle in MHP is both implicitly and explicitly signaled in the transport. The implicit signaling occurs when a channel change is initiated by the user. This may cause an application to terminate through the absence of a state signal. The explicit signals defined by MHP are:

Autostart
Present
Prefetch
Remote
Destroy
Kill

Autostart means that the application should load and its entry point (whether Java or HTML) used to begin display or execution. Present indicates that this application is not Autostart. Prefetch is similar to autostart except display of the HTML (not defined for Java) is suspended until receipt of a trigger. Remote indicates that the application is remote and that it is only launchable after tuning to the related service.

Destroy initiates a call to the Java xlet destroy() method with the unconditional flag set to false; Kill is the same except the unconditional flag is set to true. For DVB-HTML applications, the difference is a bit more subtle and effectively the same.

3.4.3 Synchronization Mechanisms

MHP relies on the DSM-CC StreamEvent object for synchronization of data applications to the video and audio, or between applications. In addition MHP supports the concept of DSM-CC Normal Play Time (NPT) which is basically programme elapsed time. The StreamEvent can be of the type, “do it now”, or it can be set to an NPT time.

3.4.4 Broadcast Data Protocols

The MHP specification is much more vertical than ATSC-DASE for example, and includes a variety of transport layer definitions that build on [DVB-SI] and [DVB-DATA] directly in the MHP document.

3.4.4.1 MPEG-2 Foundation

Basic MPEG-2 tables and descriptors are made use of without modification.

3.4.4.2 DSM-CC

MHP makes use of a constrained extension to the DSM-CC Object Carousel, and the DSM-CC Addressable Sections. However, there are some small differences between MHP’s definition and ATSC’s.

3.4.5 Internet Protocols

MHP indirectly supports a variety of Internet protocols through the standard Java language packages. Since a return channel is defined, the IP protocols include both multicast usage as well as full duplex protocols, including those based on TCP.

3.4.5.1 PPP

PPP is supported for dialup connections.

3.4.5.2 IP/UDP/TCP

IP is used for both UDP and TCP. UDP may be used in the forward broadcast channel for the carriage of IP Multicast. Both UDP and TCP may be used in the return channel.

3.4.5.3 SAP/SDP

There is no support for SAP or SDP.

3.4.5.4 HTTP

HTTP is supported on top of IP/TCP in the return channel.

3.4.5.5 SSL (TLS)

SSL is supported on top of IP/TCP in the return channel.

3.4.5.6 NNTP

NNTP is supported on top of IP/TCP in the return channel.

3.4.5.7 SMTP

SMTP is supported on top of IP/TCP in the return channel.

3.4.5.8 UHTTP

UHTTP is not supported.

3.4.6 URI Schemes

MHP uses a few URI schemes. The common ones are http://, https://, file:// and mailto:. However, it also defines the following:

dvb:
exit:
dripfeed:

The dvb:// scheme is used to refer to most resources in the broadcast, specifically including files in the object carousel file system. The exit:// scheme is used in a DVB-HTML application to

provide a means for the viewer to terminate the application. The dripfeed:// scheme is used to refer to MPEG-2 video streams.

3.4.7 Content Formats

MHP supports two main application environments – the declarative and the procedural. There are also a variety of common media types supported.

3.4.7.1 Markup Languages

MHP defines a subset of W3C XHTML 1.0 with some extensions. It is slightly more constrained that ATSC-DASE's XDML, with the specific details summarized in Section 4.7.1 below.

3.4.7.2 Layout Control

MHP defines a subset of W3C CSS2 and it minimally supports CSS1 (a proper subset of CSS2). A couple items are added. The finer details are summarized in Section 4 below.

3.4.7.3 Scripting

In the declarative environment, ECMAScript, Second Edition is support

3.4.7.4 Script API's

MHP supports all the ECMAScript native objects and a subset of DOM-2. The details of the subset can be found in Section 4. In addition, unlike ATSC-DASE, MHP supports all the Java API's through an API bridge mechanism. That is, a DVB-HTML application can access all of the Java API's in ECMAscript.

3.4.7.5 Byte Code Virtual Machine

MHP supports the Java JVM.

3.4.7.6 Byte Code API's

MHP supports the following Java packages:

Personal Java Application Environment [PJAE]

Java Media Framework [JMF]
Java TV [JAVATV]
HAVi User Interface [HAVI-UI]
DAVIC Media [DAVIC-141]
W3C DOM [DOM2]
OpenCard Framework [OCF].

There are many constraints placed on the above packages by the use in JavaTV (via PJAE) as well as MHP itself. In some casesclasses are defined to exist, but always return "not supported" exceptions.

In addition, MHP defines the following additional packages specific to DVB:

org.dvb.ixc
org.dvb.net
org.dvb.application
org.dvb.user
org.dvb.media
org.dvb.security
org.dvb.smartcard
org.dvb.lang
org.dvb.event
org.dvb.dom
org.dvb.dom.dvbhtml
org.dvb.dom.css
org.dvb.dom.environment
org.dvb.dom.event

3.4.7.7 Other Content Formats

MHP supports the following additional content formats:

Portable Font Resource [PFR]
Portable Network Graphics [PNG]
Joint Picture Experts Group [JPEG]
Graphics Interchange Format [GIF]
ASCII Text

However, it does not support:

Basic Audio [PCM]
Multiple Network Graphics [MNG]

3.5 SMPTE DDE-1

SMPTE is the Society of Motion Picture and Television Engineers [SMPTE], and is a US and ISO accredited body for international television standards. Its focus has traditionally been on studio and facility equipment standards and not digital emission standards (although the world analog standards were published by SMPTE).

DDE-1 is Declarative Data Essence, Content Level 1 [DDE], and is a refinement to the Advanced Television Enhancement Forum [ATVEF] Specification, version 1.1r26. DDE-1 provided a compliant, but more interoperable standard for this specification. In general, it is transport independent and more of a content specification (in contrast to a receiver specification). Some related SMPTE specifications cover its encapsulation on IP Multicast, and over analog transports. Compliance with DDE-1 has no legal status in the US or elsewhere.

DDE-1 operates in 2 "modes" – Transport A and Transport B, each of which have some different metadata and lifecycle properties.

3.5.1 Application Metadata

DDE-1 Transport A has no support for application metadata – there are only triggers and the resource to which they refer. Transport B has more information in the form of special fields in the SDP records. The use of SDP is meant more for advance announcement than for immediate signaling. Nevertheless it is used for both and thus, the application is loosely defined by the SDP record, which can signal the following metadata:

Session ID
Session Name
Session Description
Language
Bandwidth
Total Size
Profile/Level

For each resource, any HTTP response field may be used. But the only one required to be supported is the Expires, which effectively controls caching for each resource.

3.5.2 Application Lifecycle

The application lifecycle is generally defined by the trigger arrivals. There are some clues in the SDP records about when it starts and ends, but these are not strictly used to control the lifecycle.

An application begins when a trigger is received that contains a URL of a top level HTML page. It ends when a new trigger is received and the next application is started. A good discussion of this is in the SMPTE Engineering Guideline [DDE-EG].

There is no requirement for a receiver to terminate an application on channel (even though this may be a good idea in practice). And, there is no explicit signal of the end of an application. Thus, DDE-1 supports shared, bound and unbounded lifecycle models.

3.5.3 Synchronization Mechanisms

Applications are synchronized to the video and audio (or standalone) through the use of triggers. Triggers are basically URL's to top level content. They are not tightly synchronized, but are acted upon receipt on a best effort basis by the receiver.

3.5.4 Broadcast Data Protocols

3.5.4.1 MPEG-2 Foundation

There is nothing specific about the basic MPEG-2 transports used by DDE-1.

3.5.4.2 DSM-CC

There is no implicit support for DSM-CC in DDE-1 – the carriage usually relies on IP/UDP with UHTTP for file transfer. However, it **could** be carried using the DSM-CC object carousel, and when encapsulated with IP, can be carried using DSM-CC Addressable Sections.

3.5.5 Internet Protocols

3.5.5.1 PPP

There is no support for PPP in DDE-1.

3.5.5.2 IP/UDP/TCP

IP/UDP is often used as the encapsulation to carry the basic DDE-1 content, but there is no support for generic access to IP networking from a DDE-1 application. Similarly, a Transport A **environment** supports HTTP, and thus TCP, but there is no requirement for this in Transport B.

3.5.5.3 SAP/SDP

SAP/SDP are supported in Transport B and are used to carry the application metadata as defined in the section above. Various extensions to SDP were defined in order to permit this.

3.5.5.4 HTTP

Transport A requires HTTP, but Transport B does not.

3.5.5.5 SSL (TLS)

There is no support for SSL in DDE-1.

3.5.5.6 NNTP

There is no support for NNTP in DDE-1.

3.5.5.7 SMTP

There is no support for SMTP in DDE-1.

3.5.5.8 UHTTP

There is no explicit support for UHTTP from the application, but the environment may use IP/UDP with UHTTP for delivery of the application resources.

3.5.6 URI Schemes

DDE-1 supports the javascript:, tv: and lid: URI schemes, and when operating in Transport A, the http: scheme. However, the tv: scheme is constrained to the literal use, “tv:”, without the broadcaster field.

3.5.7 Content Formats

DDE-1, by definition, is only a declarative environment, so there is no procedural environment except for support for ECMAScript.

3.5.7.1 Markup Languages

DDE-1 fully supports all three HTML 4.0 DTD's – strict, transitional and frameset. There are no extensions or subsets defined.

3.5.7.2 Layout Control

DDE-1 supports full CSS1. There are no extensions or subsets defined.

3.5.7.3 Scripting

DDE-1 supports ECMAScript, 2nd Edition with a few minor constraints.

3.5.7.4 Script API's

DDE-1 defines and thus supports DOM-0 in addition to the ECMAScript native objects.

3.5.7.5 Byte Code Virtual Machine

DDE-1 does not support a byte code virtual machine.

3.5.7.6 Byte Code API's

DDE-1 does not support a byte code API (since it does not support the VM).

3.5.7.7 Other Content Formats

DDE-1 supports the following additional content formats:

Portable Network Graphics [PNG]
Joint Picture Experts Group [JPEG]
Basic Audio [PCM]
ASCII Text

However, it does not support:

Portable Font Resource [PFR]
Multiple Network Graphics [MNG]
Graphics Interchange Format [GIF]

4 Summary of the Standards and Technology

This is a summary of the discussion in the previous chapters, provided in table form. In all the tables below:

"ARIB" is the ARIB B24 1.1 [B24]
"ATSC" is the ATSC DASE-1, Version 1 [DASE]
"DVB" is the DVB MHP 1.1 [MHP]
"OCAP" is the OpenCable OCAP 2.0 [OCAP]
"SMPTE" is the SMPTE 363M [DDE]

4.1 Application Metadata

Element\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
ID		X	X	X	X
Name		X			X
Description		X			X
Root Resource	X	X	X	X	
Permissions		X	X	X	
Runtime Params		X	X	X	
Receiver Capabil.		X			
Resource Caching	X	X	X	X	X
Bandwidth					X
Total Size					X
Profile/Level		X	X	X	X
Security/Authorization			X	X	

4.2 Application Lifecycle

Element\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
Bound		X	X	X	X
Shared		X	X	X	X
Unbounded				X	X

4.3 Synchronization Mechanisms

Element\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
Triggers		X			X
StreamEvents			X	X	
Analog StreamEvents				X	

4.4 Broadcast Data Protocols

Element\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
MPEG-2					
Data Piping		X			
PES	X	X	X	X	
DSM-CC					
Download		X			
Data Carousel	X	X			
Object Carousel		X	X	X	
Address. Sections		X	X	X	

4.5 Internet Protocols

Element\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
Internet Protocols					
PPP	X		X	X	
IP	X	X	X	X	X
UDP	X	X	X	X	X
TCP	X		X	X	X
SAP/SDP					X
HTTP	X		X		X
SSL			X	X	
NNTP	X		X		
SMTP	X		X		
POP	X				
UHTTP					X

4.6 Namespaces

Element\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
URI Schemes					
tv:		X			X
lid:		X			X
javascript:		X[7]			X
archive:		X			
file:			X		
mailto:			X		
http:			X		X
https:			X		
dvb:			X	X	
exit:			X	X	
dripfeed:			X	X	
ocap:				X	
dsmcc:				X	
arib:	X				
arib-dc:	X				
arib-ic:	X				
arib-series:	X				
nvram:	X				
romsound:	X				

4.7 Content Formats

4.7.1 Declarative Content

Function\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
Markup DTD	BML	XDML & XHTMLBasic	DVB-HTML & XHTML1.0	DVB-HTML	HTML4
Layout	CSS2-	CSS2-	CSS2-	CSS2-	CCS1
Scripting	ECMAScript1	ECMAScript3	ECMAScript2	ECMAScript3	ECMAScript2
DOM Level	1-	2-	2-	2-	0

Note the post-pended minus sign ("-") indicates a subset of the indicated specification.

[7] This is implemented with the more politically correct, but unused in practice, "ecmascript:" scheme.

4.7.1.1 Markup

The ATSC and DVB columns are heavily derived from summary work done by Dr. Glenn Adams and Mr. Sean Hayes and provided to various ITU and SMPTE working groups.

Element\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
Encoding – 8859		X			X
Encoding – UTF-8		X	X	X	
Encoding – UTF-16	X				
Encoding – EUC-JP	X				
applet	X				X
base	X	X	X	X	X
basic forms	X				
basic tables	X		X	X	
bidi	X	X	X	X	X
client-side image map	X	X	X	X	X
edit	X				X
forms	X	X	X	X	X
frames	X	X	X	X	X
hypertext	X	X	X	X	X
iframe	X		X	X	X
image	X	X	X	X	X
intrinsic events	X	X			X
legacy	X				X
link	X	X	X	X	X
list	X	X	X	X	X
metainformation	X	X	X	X	X
name	X	X			X
object	X	X	X	X	X
presentation	X	X	X	X	X
scripting	X	X	X	X	X
server-side image map	X				X
structure	X	X	X	X	X
style attribute	X	X	X	X	X
style sheet	X	X	X	X	X
tables	X	X			X
target	X	X	X	X	X
text	X	X	X	X	X

4.7.1.2 Layout

The ATSC and DVB columns are heavily derived from summary work done by Dr. Glenn Adams and Mr. Sean Hayes.

Element\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
@Rules	X	X	X	X	
Selectors					
:active	X	X	X	X	
:after		X	X	X	
:before		X	X	X	
:first-child	X	X	X	X	
:first-letter		X	X	X	
:first-line		X	X	X	
:focus	X	X	X	X	
:hover	X		X	X	
:lang	X	X	X	X	
:link	X	X	X	X	
:visited	X	X	X	X	
adjacent		X	X	X	
attribute		X	X	X	
attribute value		X	X	X	
class		X	X	X	X
descendent		X	X	X	
grouping		X	X	X	
id		X	X	X	X
type		X	X	X	
universal		X	X	X	
Properties					
azimuth	X				
background	X	X	X	X	X
background-attachment	X	X	X	X	X
background-color	X	X	X	X	X
background-image	X	X	X	X	X
background-position	X	X	X	X	X
background-repeat	X	X	X	X	X
border	X	X	X	X	X
border-bottom	X	X	X	X	X
border-bottom-color	X	X	X	X	
border-bottom-style	X	X	X	X	
border-bottom-width	X	X	X	X	X
border-collapse	X		X	X	
border-color	X	X	X	X	X
border-left	X	X	X	X	X
border-left-color	X	X	X	X	
border-left-style	X	X	X	X	
border-left-width	X	X	X	X	X

border-right	X	X	X	X	X
border-right-color	X	X	X	X	
border-right-style	X	X	X	X	
border-right-width	X	X	X	X	X
border-spacing	X		X	X	
border-style	X	X	X	X	X
border-top	X	X	X	X	X
border-top-color	X	X	X	X	
border-top-style	X	X	X	X	
border-top-width	X	X	X	X	X
border-width	X	X	X	X	X
bottom		X	X	X	
caption-side	X	X	X	X	
clear		X	X	X	X
clip		X	X	X	
color	X	X	X	X	X
content		X	X	X	
counter-increment		X	X	X	
counter-reset		X	X	X	
cue	X				
cue-after	X				
cue-before	X				
cursor	X		X	X	
direction			X	X	
display		X	X	X	X
elevation	X				
empty-cells	X		X	X	
float	X	X	X	X	X
font	X	X	X	X	X
font-family	X	X	X	X	X
font-size	X	X	X	X	X
font-size-adjust	X		X	X	
font-stretch	X		X	X	
font-style	X	X	X	X	X
font-variant	X	X	X	X	X
font-weight	X	X	X	X	X
height		X	X	X	X
left		X	X	X	
letter-spacing	X	X	X	X	X
line-height		X	X	X	X
list-style		X	X	X	X
list-style-image		X	X	X	X
list-style-position		X	X	X	X
list-style-type		X	X	X	X
margin	X	X	X	X	X

margin-bottom	X	X	X	X	X
margin-left	X	X	X	X	X
margin-right	X	X	X	X	X
margin-top	X	X	X	X	X
marker-offset			X	X	
marks					
max-height			X	X	
max-width			X	X	
min-height			X	X	
min-width			X	X	
orphans					
outline	X	X	X	X	
outline-color	X	X	X	X	
outline-style	X	X	X	X	
outline-width	X	X	X	X	
overflow		X	X	X	
padding	X	X	X	X	X
padding-bottom	X	X	X	X	X
padding-left	X	X	X	X	X
padding-right	X	X	X	X	X
padding-top	X	X	X	X	X
page					
page-break-after					
page-break-before					
page-break-inside					
pause	X				
pause-after	X				
pause-before	X				
pitch	X				
pitch-range	X				
play-during	X				
position		X	X	X	
quotes			X	X	
richness	X				
right		X	X	X	
speak	X				
speak-header	X				
speak-numeral	X				
speak-punctuation	X				
speech-rate	X				
stress	X				
table-layout	X		X	X	
text-align	X	X	X	X	X
text-decoration	X	X	X	X	X
text-indent	X	X	X	X	X

text-shadow	X		X	X	
text-transform	X	X	X	X	X
top		X	X	X	
unicode-bidi			X	X	
vertical-align		X	X	X	X
visibility	X	X	X	X	
voice-family	X				
volume	X				
white-space	X	X	X	X	X
windows					
width		X	X	X	X
word-spacing	X	X	X	X	X
z-index	X	X	X	X	
Descriptors					
ascent	X				
baseline	X				
bbox	X				
cap-height	X				
centerline	X				
definition-src	X				
descent	X				
font-family	X	X	X	X	
font-size	X	X	X	X	
font-stretch	X	X	X	X	
font-style	X	X	X	X	
font-variant	X	X	X	X	
font-weight	X	X	X	X	
mathline	X				
panose-1	X				
slope	X				
src	X	X	X	X	
stemh	X				
stemv	X				
topline	X				
unicode-range	X	X	X	X	
units-per-em	X		X	X	
widths	X				
x-height	X				

4.7.1.3 DOM

The ATSC and DVB columns are heavily derived from summary work done by Dr. Glenn Adams and Mr. Sean Hayes. Note that the first set of objects, Anchor through Window, are from “DOM-0”. ARIB, ATSC, DVB and OCAP all support many of these objects indirectly

through their equivalent DOM-2 object names. However, since the syntax is different, they are not listed in the table below.

Object\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
Anchor					X
Button					X
Checkbox					X
Document					X
Form					X
Hidden					X
History		X			X
Image					X
Link					X
Location		X	X	X	X
Navigator		X	X	X	X
Option					X
Password					X
Radio					X
Reset					X
Select					X
String (extended function)					X
Submit					X
Text					X
TextArea					X
Window		X	X	X	X
*Ext		X			
AbstractView		X	X	X	
Attr	X	X	X	X	
BML*	X				
CDATASection	X	X			
CharacterData		X	X	X	
Comment		X	X	X	
Counter		X			
CSS*		X			
CSS2Properties			X	X	
Document	X	X	X	X	
DocumentCSS		X			
DocumentEvent		X	X	X	
DocumentFragment	X	X	X	X	
DocumentStyle		X			
DocumentType		X			
DocumentView		X	X	X	
DOMException	X	X	X	X	
DOMImplementation	X	X	X	X	
DOMImplementationCSS		X			

DVB*			X	X	
Element	X	X	X	X	
ElementCSSInlineStyle		X			
Entity	X	X			
EntityReference	X	X			
Event		X	X	X	
EventException		X	X	X	
EventListener		X	X	X	
EventTarget		X	X	X	
HTMLAnchorElement	X	X			
HTMLAppletElement	X				
HTMLAreaElement	X				
HTMLBRElement	X				
HTMLBaseFontElement	X				
HTMLBlockQuoteElement	X				
HTMLBodyElement	X	X			
HTMLButtonELement	X				
HTMLCollection	X	X			
HTMLDListElement	X				
HTMLDivElement	X				
HTMLDocument	X	X			
HTMLDOMImplementation		X			
HTMLElement	X	X			
HTMLFieldSetElement	X				
HTMLFormElement	X	X			
HTMLFrameElement	X				
HTMLFrameSetElement	X				
HTMLHRElement	X				
HTMLHeadElement	X				
HTMLHeadingElement	X				
HTMLHtmlElement	X				
HTMLIFrameElement	X				
HTMLImageElement	X	X			
HTMLInputElement	X	X			
HTMLLIElement	X				
HTMLLabelElement	X				
HTMLLegendElement	X				
HTMLLinkElement	X				
HTMLMapElement	X				
HTMLMetaElement	X				
HTMLModElement	X				
HTMlOListElement	X				
HTMLObjectElement	X	X			
HTMLOptGroupElement	X				
HTMLOptionElement	X	X			

HTMLParagraphElement	X				
HTMLParamElement	X				
HTMLPreElement	X				
HTMLQuoteElement	X				
HTMLScriptElement	X				
HTMLSelectElement	X	X			
HTMLStyleElement	X				
HTMLTableCaptionElement	X				
HTMLTableCellElement	X				
HTMLTableColElement	X				
HTMLTableElement	X				
HTMLTableRowElement	X				
HTMLTableSectionElement	X				
HTMLTextAreaElement	X	X			
HTMLTitleElement	X				
HTMLUListELement	X				
LinkStyle		X			
MediaList		X			
MouseEvent		X	X	X	
MutationEvent		X	X	X	
NamedNodeMap		X	X	X	
Node	X	X	X	X	
NodeList	X	X	X	X	
Notation	X	X			
ProcessingInstruction	X	X			
Rect		X			
RGBColor		X			
StyleSheet		X			
StyleSheetList		X			
Text	X	X	X	X	
UIEvent		X	X	X	
ViewCSS		X			

4.7.2 Procedural Content

VM\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
Byte Code VM		Java	Java	Java	

The table below summarizes the packages, and in some cases, groups of packages for each system. Where the group of packages is obviously only supported by one system (e.g. org.atsc.*), the packages are collapsed into a single row using the asterisk syntax.

This summary does not make the distinction between a package being fully supported or not. That is, in many cases, a system has defined there to be only signature support for certain classes or interfaces. This means that the class, while present in form, is not functional. The only way to capture this would be to enumerate every class, or perhaps every attribute, too. If the package is included (at least in part), it is checked in the table, regardless of whether or not it has to be entirely implemented. This leaves the actual support between systems a little more vague than for the other comparison sections, but otherwise would require too much detail here.

Package\Standard	ARIB	ATSC	DVB	OCAP	SMPTE
com.gemplus.opencard.terminal			X	X	
com.ibm.opencard*			X	X	
com.ibm.tools.rejar			X	X	
com.ms.security			X	X	
com.sun.awt					
com.sun.lang			X	X	
com.sun.util					
java.applet			X	X	
java.awt		X	X	X	
java.awt.datatransfer			X	X	
java.awt.event		X	X	X	
java.awt.image		X	X	X	
java.awt.peer					
java.beans		X	X	X	
java.io		X	X	X	
java.lang		X	X	X	
java.lang.reflect		X	X	X	
java.math			X	X	
java.net		X	X	X	
java.rmi			X	X	
java.rmi.dgc					
java.rmi.registry					
java.rmi.server					
java.security		X	X	X	
java.security.acl					
java.security.cert		X	X	X	
java.security.interfaces					
java.security.spec			X	X	
java.sql					
java.text		X	X	X	
java.text.resources					
java.util		X	X	X	
java.util.jar					
java.util.locale			X	X	

java.util.zip		X	X	X	
javax.media		X	X	X	
javax.media.protocol		X	X	X	
javax.net			X	X	
javax.net.ssl			X	X	
javax.security.cert			X	X	
javax.tv.carousel		X			
javax.tv.graphics		X	X	X	
javax.tv.locator		X	X	X	
javax.tv.media		X	X	X	
javax.tv.media.protocol		X			
javax.tv.net		X	X	X	
javax.tv.service		X	X	X	
javax.tv.service.guide		X	X	X	
javax.tv.service.navigation		X	X	X	
javax.tv.service.selection		X	X	X	
javax.tv.service.transport		X	X	X	
javax.tv.util		X	X	X	
javax.tv.xlet		X	X	X	
netscape.security					
opencard.core.*			X		
opencard.opt.*			X		
opencard.tests.*			X		
org.atsc.*		X		X[8]	
org.davic.media		X	X	X	
org.davic.resources		X	X	X	
org.dvb.*			X	X	
org.havi.ui		X	X	X	
org.havi.ui.event		X	X	X	
org.ocap.*				X	
org.w3c.dom		X	X	X	
org.w3c.dom.css		X			
org.w3c.dom.events		X			
org.w3c.dom.html		X			
org.w3c.dom.stylesheets		X			
org.w3c.dom.views		X			

4.7.3 Other Formats

Content\Standard	ARIB	ATSC	DVB	OCAP	SMPTE

[8] Relatively small subset only.

PFR		X	X	X	
PNG	X	X	X	X	X
MNG	X	X			
GIF			X	X	
JPEG	X	X	X	X	X
PCM	X	X			X
ASCII Text			X	X	X

5 Glossary of Terms and Acronyms

Announcement – the description of future services and programmes expected to be in the broadcast.

Application - a collection of resources.

ATSC Advanced Television Systems Committee (US-centric digital television standards body – see [ATSC])

ATVEF - Advanced Television Enhancement Forum (see [ATVEF]).

ARIB - Association of Radio Industries and Businesses (Japan-centric digital television standards body – see [ARIB])

Author – the generator of essence, which may be from a studio, programmer, service aggregator or even the viewer.

Broadcaster – the programmer that generates a service in a local market.

Content – see Essence.

CSS – Cascading Style Sheets (see [CSS1] and [CSS2]).

DASE - DTV Application Software Environment (see [DASE]).

Data Service - collection of (data) applications (that may or may not be associated with a video/audio service).

DAVIC – Digital Audio-Visual Council (see [DAVIC]).

DBS - Direct Broadcast Satellite.

DDE - Declarative Data Essence (see [DDE]).

DOM – Document Object Model.

DSM-CC – Digital Storage Media Command & Control (see [DSMCC]).

DTD – Document Type Definition (see [W3C]).

DTV - Digital TeleVision

DVB - Digital Video Broadcast (European-centric digital television standards body – see [DVB])

ECMA – European Computer Manufacturers Association (see [ECMA]).

Enhancement – see Application.

EPG - Electronic Program Guide.

Essence – the video, audio or data elements of a programme or service that result in some experience for the viewer (in contrast to metadata).

ETSI - European Telecommunications Standards Institute (see [ETSI]).

Event – binding of a programme to a service at a specific time.

FTP – File Transfer Protocol (see [FTP]).

GIF – Graphics Interchange Format (see [GIF]).

HAVi – Home Audio Video Interoperability (see [HAVI]).

HTML – HyperText Markup Language (see [HTML]).

HTTP – HyperText Transfer Protocol (see [HTTP]).

IDL – Interface Definition Language (see [CORBA]).

IP – Internet Protocol (see [IP]).

ITV – Interactive TeleVision.

JMF – Java Media Framework (see [JMF]).

JPEG – Joint Picture Experts Group.

JVM – Java Virtual Machine (see [JVM]).

Metadata – the information about essence that the television distribution system generates or consumes, but generally does not result in a direct experience to the viewer.

MHP - Multimedia Home Platform (see [MHP])).

MNG – Multiple Network Graphics (see [MNG]).

MPEG – Moving Picture Experts Group (see [MPEG]).

MSO –Multiple System Operator.

MTU – Maximum Transmission Unit.

NNTP – Network News Transport Protocol (see [NNTP]).

NPT – Normal Play Time (see [DSMCC]).

OCF – Open Card Framework (see [OCF]).

OMG – Object Management Group (see [OMG]).

PCM – Pulse Code Modulation (see [PCM]).

PFR – Portable Font Resource (see [PFR]).

PNG – Portable Network Graphics (see [PNG]).

POTS – Plain Old Telephone System.

PPP – Point to Point Protocol (see [PPP]).

PJAE – Personal Java Application Environment (see [PJAE]).

Programme – the video, audio, data and metadata generated by a studio, usually a finished work.

Programmer – the entity that generates a service.

Resource – data component of an application.

SAP – Session Announcement Protocol (see [SAP]).

SCTE – Society of Cable Telecommunications Engineers (see [SCTE]).

SDP – Session Description Protocol (see [SDP]).

Service – the serialized broadcast of programmes (also known as a channel in the US).

Service Aggregator – a cable or DBS system that aggregates services and adds value to the viewer.

SI – System Information (see [MPEG-SYS])

Signaling – the indication of a current service or programme in the broadcast.

SMPTE - Society of Motion Picture and Television Engineers (see [SMPTE]).

SMTP – Simple Mail Transfer Protocol (see [SMTP]).

Socket – an Internet abstraction for an IP address and port number pair.

SSL – Secure Socket Layer (see TLS).

STC – System Time Clock (see [MPEG-SYS]).

Studio – the entity that creates essence.

TCP – Transmission Control Protocol (see [TCP]).

TLS – Transport Layer Security (also known as SSL, see [TLS]).

Trigger - linkage of a resource to specific point in the programme time line.

UDP – User Datagram Protocol (see [UDP]).

UHTTP – Unidirectional HyperText Transfer Protocol (see [UHTTP]).

URI – Uniform Resource Identifier (see [URI]).

URL – Uniform Resource Locator (see [URI]).

VOD – Video On Demand.

Viewer – the target person for the consumption of essence.

W3C – World Wide Web Consortium (see [W3C]).

XHTML – Extensible HyperText Markup Language (see [XHTML]).

XSL – Extensible Style Language (see [W3C]).

6 References

[A90] ATSC Standard A/90, “ATSC Data Broadcast Standard”.

[ARIB] Association of Radio Industries and Businesses, http://www.arib.co.jp

[ATSC] Advanced Television Systems Committee, http://www.atsc.org

[ATVEF] Advanced TeleVision Enhancement Forum, http://www.atvef.com.

[B24] ARIB Standard B24, “Data Coding and Transmission Specification for Digital Broadcasting”.

[CABLELABS] Cable Television Laboratories Inc., http://www.cablelabs.com.

[CORBA] OMG Specification, “The Common Object Request Broker: Architecture and Specification”.

[CSS1] W3C Recommendation, “Cascading Style Sheets, level 1”.

[CSS2] W3C Recommendation, “Cascading Style Sheets, level 2”.

[DASE] ATSC Candidate Standards[9] CS/100-1…8, "DTV Application Software Environment”.

[DATA-BOOK] McGraw-Hill, “Data Broadcasting: Understanding the ATSC Data Broadcast Standard”, Chernock, et al.

[DAVIC] Digital Audio-Visual Council, http://www.davic.org.

[DAVIC-141] DAVIC, “DAVIC 1.4.1 Specification”, Part 9, Annex L.

[DDE] SMPTE Standard 363M, "Declarative Data Essence, Content Level 1" [10].

[DDE-EG] SMPTE Engineering Guide EG39, “Declarative Data Essence”.

[DOM0] SMPTE Standard 366M, “Document Object Model Level 0 (DOM-0) and Related Object Environment”.

[DOM1] W3C Recommendation, “Document Object Model (DOM) Level 1 Specification”.

[9] This is a collection of 8 documents, CS/100-1 through CS/100-8.
[10] Formerly ATVEF 1.1r26

[DOM2] W3C Recommendations[11], “Document Object Model (DOM) Level 2”.

[DSMCC] ISO/IEC 13818-6, “Information technology -- Generic coding of moving pictures and associated audio information -- Part 6: Extensions for DSM-CC”.

[DVB] Digital Video Broadcast, http://www.dvb.org

[ECMA] European Computer Manufacturers Association, http://www.ecma.ch.

[ECMASCRIPT2] ISO/IEC 16262, “Information technology - ECMAScript language specification”.

[ECMASCRIPT3] ECMA-262:1999, “ECMAScript Language Specification, 3rd Edition”.

[ETSI] European Telecommunications Standards Institute, http://www.etsi.org.

[FTP] IETF RFC 959, “File Transfer Protocol”.

[GIF] Compuserve Product Specification GIF89A, “GRAPHICS INTERCHANGE FORMAT Version 89a”.

[HAVI] Home Audio Video Interoperability, http://www.havi.org.

[HAVIUI] HAVi Specification, “The HAVi Specification”, Version 1.1, Section 8.

[HTML] W3C Recommendation, “HTML 4.0 Specification”.

[HTTP] IETF RFC 2616, “Hypertext Transfer Protocol -- HTTP/1.1”.

[HTTPS] IETF RFC 2818, “HTTP Over TLS”.

[IP] IETF RFC 791, “Internet Protocol”.

[JAVA] Addison Wesley, 1996, Gosling, Joy and Steele, “The Java Language Specification”, ISBN 0-201-63451-1.

[JAVATV] Sun Microsystems Product Specification, “Java TV API 1.0“.

[JDK] Addison Wesley, Gosling & Yellin, “The Java Application Programming Interface, Volume 1: Core Packages (ISBN 0-201-63453-8) and Volume 2: Window Toolkit and Applets (ISBN 0-201-63459-7)”.

[11] This is a collection of 6 documents. One, the HTML Specification, is still in Working Draft status at this writing. See [W3C] for more details and current status of the documents.

[JMF] Sun Microsystems Product Specification, “Java Media Framework Specification, Version 1.0”.

[JPEG] ISO 10918-1, “Digital Compression and Coding of Continuous-Tone Still Images”

[JVM] Addison Wesley, Lindholm & Yellin, “The Java Virtual Machine Specification”, ISBN 0-201-63452-X

[MHP] ETSI TS 102 812 v1.1.1:2001, “Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1”

[MNG] PNG Development Group, “Multiple Network Graphics, Version 1.0”

[MPEG] Moving Picture Experts Group, ISO JTC1/SC29/WG11, http://mpeg.telecomitalialab.com/

[MPEG-SYS] ISO/IEC 13818-1, “Information technology -- Generic coding of moving pictures and associated audio information: Systems”.

[NNTP] IETF RFC 977, “Network News Transport Protocol”.

[OCAP] CableLabs OC-SP-OCAP2.0-I01-020419, “OpenCable Application Platform Specification OCAP 2.0 Profile”.

[OCF] OpenCard Framework Specification, “OpenCard Framework version 1.2”.

[OMG] Object Management Group, http://www.omg.org.

[PCM] ITU-T G.711, “Pulse Code Modulation of Voice Frequencies”.

[PFR] ISO/IEC 16500-6:1999, “Information technology – Generic digital audio-visual systems – Part 6: Information representation, Annex A, Coding of Outline Fonts”.

[PJAE] Sun Microsystems Product Specification, “Personal Java Application Environment Specification“.

[PPP] IETF RFC 1661, “The Point-to-Point Protocol (PPP)”.

[SAP] IETF RFC 2974, “Session Announcement Protocol”.

[SCTE] Society of Cable Telecommunications Engineers, http://www.scte.org.

[SDP] IETF RFC 2327, “SDP: Session Description Protocol”.

[SGML] ISO/IEC 8879:1986, “Standard Generalized Markup Language”.

[SMTP] IETF RFC 821, “Simple Mail Transport Protocol”.

[TCP] IETF RFC 793, “Transmission Control Protocol”.

[TLS] IETF RFC 2246, “The TLS Protocol Version 1.0".

[UDP] IETF RFC 768, “User Datagram Protocol”.

[UHTTP] SMPTE Standard 364M, “Declarative Data Essence – Unidirectional Hypertext Transport Protocol”.

[URI] IETF RFC 2396, “Uniform Resource Identifiers (URI): Generic Syntax”.

[URI-HTTP] IETF RFC 2616, “Hypertext Transfer Protocol -- HTTP/1.1”, Section 3.2.

[URI-HTTPS] IETF RFC 2818, “HTTP Over TLS”, Section 2.4.

[URI-LID] SMPTE Standard 343M, “The Local Identifier (lid:) URI Scheme”.

[URI-TV] IETF RFC 2838, “Uniform Resource Identifiers for Television Broadcasts”.

[UUID] ISO/IEC 11578, “Information Technology – Open Systems Interconnection – Remote Procedure Call (RPC)”, Annex A.

[W3C] Worldwide Web Consortium, http://www.w3.org.

[XHTML] W3C Recommendation, “XHTML 1.0: The Extensible HyperText Markup Language”.

[ZIP] ATSC CS/100-5, “DASE Part 5: ZIP Architecture Resource Format”.