Interactive TV Applications:
Standard APIs for Digital TV Receivers

Petr Peterka
General Instrument Corporation

Abstract

With advanced analog cable settop boxes, early digital satellite boxes and recently with digital set-top boxes, television viewers are getting used to more than just audio-visual (A/V) content. Enhanced broadcast includes graphical and data enhancements to the specific A/V program, such as additional text, graphics, user choices, teleshopping, etc. Standalone applications such as electronic program guides are becoming a norm. All current deployment of such systems is based on proprietary solutions.

The Internet and the Web in particular was enabled by platform independent content formats such as HTML and Java. The same must happen in order to deliver enhanced content and downloadable applications to digital TV receivers of all kinds including terrestrial receivers, cable set-tops, satellite receivers and computers. A platform independent content format is not enough to provide a rich, well-integrated audio/video/data content to all possible receivers. These devices must have a common set of application programming interfaces (API) in order to make downloadable content and applications truly interoperable. The goal of these APIs is to provide access to the receiver functions such as tuning and channel changing, receiver resources such as a return channel and the TV screen, as well as system information necessary for channel navigation and program guides. User-specific data such as user preferences and personal data may also be made available to applications via these APIs.

This paper addresses the current work-in-progress in the ATSC T3/S17 specialist group also known as the DTV Application Software Environment (DASE), specifically the definition of Java APIs. Classification of downloadable applications is presented together with a set of requirements that must be met in order to enable such interoperable applications. Also a detailed description of all DTV receiver system services that are being abstracted by the Java APIs is presented. The focus of this paper is on Java-based downloadable applications that are enabled by the presence of a Java interpreter in the form of a Java Virtual Machine (JVM) and a set of Java APIs providing access to the DTV receiver functionality.

MOTIVATION

As the wide spread use of the WWW was enabled by platform independent content formats such as HTML and Java, we will see more and more enhanced content and downloadable applications available on digital receivers of all kinds including terrestrial receivers, cable set-tops, satellite receivers and computers. A slow processor and a very small amount of memory no

longer limit these devices, which was the case until recently. The point of platform independence is essential to the broadcast environment. As opposed to the computer industry where there is a very small number of dominant operating systems and hardware platforms, the embedded device world uses a wide variety of real-time operating systems as well as hardware platforms and CPUs. Content delivered over the air must be consumable by a large number (majority) of receivers since it is too costly to waste the available bandwidth by sending separate formats of the same content to a wide variety of receivers.

The Internet provides a very interesting model but the DTV network can provide much richer user experience. Delivering the Internet experience to a television audience is a worthwhile goal but the potentials are not limited to those of the Internet. The broadcast world has its own characteristics, advantages and limitations. It is primarily a unidirectional multicast network, although a return channel is becoming a common practice in both digital cable networks with a two-way cable as well as satellite networks with traditional telephone return paths. Another differentiating factor is that the primary content is high-quality and high-resolution audiovisual content with a large amount of bandwidth available on each channel. The capability to synchronize applications with the main video program is often necessary.

DOWNLOADABLE APPLICATIONS

Applications described here are only those that are downloaded from the network and use nonproprietary formats and APIs. Although native (i.e. platform dependent) and resident applications are important, they are outside the scope of this discussion.

The wide variety of downloadable applications suitable for the DTV domain may be classified based on different criteria including (1) level of interactivity, (2) level of synchronization with the audio-visual content, (3) style of authoring or programming, and (4) level of persistency.

Level of Interactivity

The simplest applications provide additional textual or graphical enhancement to the traditional audio-visual programming. Examples of such enhancements may be as simple as subtitles or closed captions, or more sophisticated such as graphical illustrations during weather forecasts, statistics sent along with a sports program, or news headlines provided during a news hour. This class of applications allows very minimal or no user interactivity. The next step in the level of user involvement is local interactivity. This means that all receivers are running the same application but the behavior is different based on individual user actions, choices or settings. Examples of such applications may include additional features of applications mentioned in the previous category, such as user customized news search criteria, team or individual player statistics selected by the user, as well as user-selectable camera views during a football or golf game. This class also includes applications such as electronic program guides, local interactive games and commercials. Finally, the highest class of applications based on interactive capabilities is server-based interactivity or sometimes called transactional applications. A return path and a connection to a network server, such as a cable headend or a service provider, are required. This type of applications includes traditional Internet service, interactive commercials with order

forms, electronic commerce, video on demand (NVOD), etc.

Level of Dependency on the Audio-Visual Program

Many applications may be related to the current audio-video program. These applications fall into the enhanced audio-video program category. Such applications will most likely be delivered in the same MPEG program stream. Within this category, applications may be either loosely associated with the audio-video streams, such as a news ticker during a political news program, or highly synchronized with the video, such as real-time telemetry data during an automobile race or basketball statistics updated in real time.

Other applications will be completely independent of the current A/V program. Examples may include an electronic program guide, stock ticker, e-mail or home banking. These applications may be superimposed on top of the video program if they don't require the entire screen or may replace the video program. Such applications may be delivered as an independent MPEG program.

Authoring Style

This classification for a DTV application describes the authoring style of the application. There are three possible types of authoring: (1) procedural applications, (2) declarative content applications, and (3) mixed-format applications.

Procedural applications are applications that are generally written in a procedural programming language such as Java while declarative content applications are written using a declarative language such as HTML. Mixed-format applications are those that contain both procedural and declarative content components.

Level of Persistency

Broadcast applications are generally played once and not permanently stored on the DTV receiver. Such applications are usually small or are delivered in the data carousel stream in advance so that they can be played when triggered either by synchronization with the audio/video channel, by the signaling protocol or by user interaction.

An application that is used often can be optionally stored locally on the DTV to save time next time it is used. Some applications are too big to be acquiring and downloading every time they are needed. An example of such an application is an electronic program guide or an e-mail client. Another reason for storing an application is that it may be a paid service; once the user pays for the application, it is decrypted and stored for future use.

CURRENT PRACTICE

Current advanced analog cable settop boxes, digital satellite receivers and digital cable set-top boxes provide television viewers with more than just audio-visual content. Enhanced broadcast includes graphical and data enhancements to specific A/V programs, such as additional text, graphics, user choices and customization, teleshopping, etc. Standalone applications such as electronic program guides are becoming a norm. All current deployment of such systems is based on proprietary solutions. Therefore, such services are limited to a specific network provider and CE manufacturer.

Another common practice is the delivery of the Internet to the television. It is enabled

by supporting a two-way communication path using protocols such as HTTP or a mechanism where a large number of web pages, possibly related to the current video programming, are broadcast and the user may browse this limited view of the Internet.

TECHNICAL SOLUTION

Several conditions must be met in order to provide an economically feasible environment for downloadable DTV applications. First, applications must be written in a format understandable by all DTV receivers; therefore, platform independent. Such content format may be HTML [12], Dynamic HTML (HTML, CSS1 [13] and DOM1 [14]) or XML-based XHTML [15], or a procedural programming language format such as Java byte codes [7]. Second, an interpreter capable of decoding the application format is required on the receiver. In the case of HTML or any of its related formats, an appropriate markup language parser is needed. In the other case a Java byte code interpreter such as a Java Virtual Machine (JVM) [8] is required. Third condition is a standardized set of interfaces, which provide access to the DTV receiver functionality that is essential for any downloadable application.

Such an environment enables the deployment of platform independent downloadable applications and still allowing for a relatively high implementation freedom for DTV receivers, including the choice of a platform or middleware implementation language, operating system, CPU and other relevant hardware.

STANDARDS WORK

A platform independent content format is not enough to provide a rich, well-integrated audio/video/data content to all possible receivers. These devices must have a common set of application programming interfaces (API) in order to make downloadable content and applications truly interoperable. There are several standards organizations and company consortia which are trying to do exactly that. An open standard definition of such APIs opens up the market, which is currently dominated by proprietary solutions, for competition. DAVIC [18], DVB [19], ATSC [1], OpenCable [20], Sun Microsystems [9] and other organizations are currently in the process of specifying a set of Java APIs that would enable such applications.

The following sections will discuss the current work in progress of the ATSC T3/S17 [2] specialist group also known as Digital TV Application Software Environment (DASE). This group has been working over a year on a selection of an Application Execution Engine (AEE) and a Presentation Engine (PE). The current draft specification includes the JVM as the Application Execution Engine, XHTML-based parser as the Presentation Engine, and a set of Java APIs.

API DESIGN GOALS

The goal of these APIs is to provide access to selected receiver functions such as tuning and channel changing, receiver resources such as a modem or a return channel if available, a conditional access module and the TV screen, system information (e.g. ATSC PSIP, DVB SI, SCTE SI, etc.) necessary for channel navigation and program guides, as well as user preferences and user-related information.

Another important aspect is manageability of these receivers. The

current TV viewer is used to a very reliable appliance which does not put up computer-like error messages, does not require a frequent reboot nor an advanced degree to operate. The TV is becoming a broadcast network computer combining the traditional passive TV experience with an interactive aspect currently expected on computers. It is essential to provide means to remotely diagnose and troubleshoot any problems with no or minimal involvement of the viewer. A number of APIs provides application and resource instrumentation that will enable such remote management.

In order to maximize the implementation freedom of each DTV receiver manufacturer, these APIs must maintain a certain level of abstraction. Consumer electronic manufacturers should be also free to set their own policies for resource and application management.

Security becomes an essential component of a DTV receiver as well. The APIs must be defined so that appropriate security policies may be implemented based on the trust level of each downloaded application, user preferences as well as the network operator's business model.

DASE APIS

This section describes the current work-in-progress in the ATSC T3/S17 specialist group (DASE) [2] and other directly related efforts. First, the DTV reference architecture and a detailed description of all DTV receiver system services that are being abstracted by the DASE APIs is given. Finally, the API definition and a mapping to other related standards such as MPEG-2, DSM-CC, ATSC T3/S13 [4] and T3/S16 [5] conclude the discussion.

Reference Architecture

The purpose of this paper is not to describe the reference architecture of a DTV receiver. The following paragraphs show the basic concepts in order to demonstrate how the DASE APIs relate to other DTV receiver platform components.

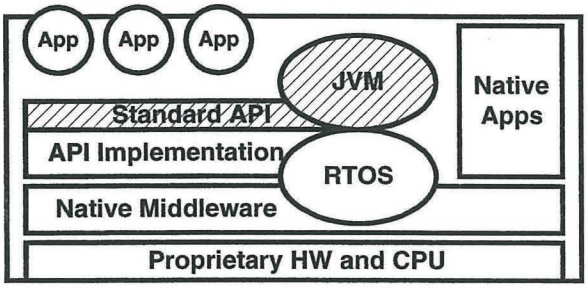

Figure 1 - Reference Architecture

Figure 1 above shows a very simple diagram of a DTV receiver reference architecture just to demonstrate the basic elements of a DTV receiver with respect to downloadable applications and the API they may use. The main point here is that only the shaded components are standardized: the Java Virtual Machine and the Java APIs. Everything else, primarily the API implementation, the real-time operating system, the native middleware, as well as the hardware and CPU may be chosen by the CE manufacturer based on its own architecture.

System Services

The DASE APIs act as an abstraction layer above the operating system's native libraries as well as the receiver middleware. The Java Virtual Machine in the role of an AEE provides a set of APIs which are very generic and typical for operating systems. In order to provide a rich set of DTV receiver specific functions, such as tuning, and a close integration between a DASE

application and the DTV receiver, an additional, well-defined set of APIs must be developed.

ATSC T3/S17 (DASE) has identified a number of system services by analyzing the API needs of many example interactive and data broadcast applications submitted by ATSC T3/S17 members. The following system service groups were identified:

1. Network Communication Group
2. Content Management Group
3. PSIP Service Group
4. Presentation and User Interaction Group
5. Application Management Group
6. Environment and User Management Group
7. Resource Management Group
8. Security Service Group
9. Utility Service Group

Network Communication Group

This group represents services related to the network communication between the DTV receiver and external devices. This is primarily the MPEG-2 transport stream in the broadcast environment. It also includes return channel, both batch and real-time. If the DTV receiver communicates with other devices on the home network, this group provides appropriate services to support it. The DTV receiver is expected to be a part of a larger home network. Downloadable applications may use or control other devices on the network. These services should abstract and unify the home network device control functions.

The main services in this group include service selection (i.e. channel changing), access to broadcast data, explicit tuning, access to an optional return channel and home networking.

Content Management Group

This group of services represents content decoding, synchronization of content and media control. It is primarily concerned with audio/video content but includes data of various formats as well as downloadable application code. The content management services may also provide the system with content life cycle, content integrity and stream synchronization functions.

For applications that may need to store content, this service group will also provide content storage and playback functions.

Program and System Information (PSIP) Service Group

The PSIP Group provides the receiver with MPEG-2 Program and System Information (PSI) and the ATSC Program and System Information Protocol (PSIP) [3] data. It is closely related to the Content Management Group because it partially describes the content as received via the network. It supports basic navigation services (i.e. transport streams, virtual channels), program guide services via scheduled program events (i.e. event information tables - EIT) as well as data services via extensions supporting data broadcast as defined by ATSC T3/S13 and object carousels defined by T3/S16 and DSM-CC.

Presentation and User Interaction Group

This group is primarily responsible for presenting content to the end user. Content consists of audio, video, text and graphics. These services provide functions for displaying and controlling the presentation of information to the user.

This group of services also allows the DTV receiver user to interact with the

applications executing on the receiver. These include IR remote control events, front panel, keyboard, pointing device, voice control, etc.

Application Management Group

The application management services provide the support for downloadable applications: code verification and authentication, application registration and life cycle, version management, and application management for the purpose of local or remote monitoring and control.

Environment and User Management Group

Environment services collect and provide information about the overall DTV receiver environment and profile, configuration and versioning, system monitoring and usage statistics.

The DTV receiver may be a multi-user environment. This group also includes system services that support user profiles, permissions, preferences, usage and user identification functions.

Resource Management Group

Resource management services provide a uniform mechanism to reserve, release and request status of scarce resources such as tuner, modem, conditional access module and other devices, as well as local storage resources such as memory, disk space, etc.

Security Group

This group of services supports security-related functions which provide the system with capabilities to control and manage access to applications, content, and services: conditional access for controlling access to audio, video and data services; encryption and decryption of content; authentication of applications and users; digital signatures and certificates for controlling application downloads; privacy for protecting user-specific information; content control permissions based on content rating or end user geographic location,

Utility Service Group

The DTV receiver is expected to be a multitasking/multithreading environment where different components need to communicate with each other. These include event routing and dispatching, exception handling, inter-process communication and other forms of signaling.

This group also provides miscellaneous functions such as time and scheduling, mathematical functions, string manipulation, internationalization and localization functions.

Some applications may need to store their state and other data between executions. These services will provide access to the local storage functions. They will work closely with the resource management services to provide coordinated access to the storage device.

API Definition

Access to the above described system services is provided via the standardized set of DASE APIs. There are several high-level requirements related to the API definition: (1) The API must be at such an abstraction level so that it does not dictate any particular implementation or DTV receiver architecture; it is important to allow DTV receiver manufacturers to differentiate their products. (2) The API definition must be consistent across all groups of system service APIs. Event mechanisms, error and exception handling are examples of such

areas. (3) The API definition should be minimal but complete. This means that any overlap between APIs should be avoided while all application related aspects of the DTV functionality should be accessible through the API.

The current definition of the complete set of DASE APIs consists of several groups, also called Java packages. Some were designed by other industry consortia and some were developed directly for DASE. Specifically, DASE adopted the Sun Microsystem JavaTV APIs [9] and Personal Java APIs [10], Java Media Framework APIs [11] and HAVi [16] User Interface API.

JavaTV APIs

A substantial portion of the DASE API set comes from the Sun Microsystems JavaTV APIs [9]. It provides a high-level, protocol independent set of APIs that are not adopted only by ATSC but by other standards organizations such as DVB. This fact is very important because JavaTV provides a common subset of APIs for applications intended to run on different systems (e.g. ATSC or DVB). JavaTV includes the following APIs:

1. Abstract SI API provides protocol independent access to the basic information delivered to the receiver via a system information (SI) protocol. This API supports DVB SI, ATSC PSIP (A65) as well as SCTE [6] SI protocol.

This set of APIs is divided into several subpackages which represent different views into the SI database depending on the application's needs. Specifically, the navigation package representing a customizable collection of channels, the guide package representing current and future program events, the pipeline package providing information about groups of related channels and the descriptor package which allows an application to retrieve raw MPEG descriptor data. This is necessary to support extensions to SI protocols such as PSIP.

2. Service Selection API allows an application to select services or change channels. When a channel is selected, this API provides the application with a set of presentation controls depending on the nature of the service provided by the particular channel, e.g. video, audio, subtitles, graphics, applications, etc. An event mechanism is included to provide the calling application with information about changes in the selected service.

3. Data Broadcast API provides an abstraction over different types of data delivered to the receiver. This includes streaming as well as file-based data (carousels), asynchronous, synchronous and synchronized data, as well as different data encapsulation mechanisms including IP.

4. Applications Life Cycle API is represented by an Xlet which is the broadcast equivalent of a Web-based Applet. It can be used to do basic application control functions, e.g. initialization, start, pause, stop. This API also includes a mechanism to report application's immediate state back to the controlling entity, e.g. an application manager.

5. Media API: JavaTV also includes portions of the Java Media Framework [11] API and several extensions defined by DAVIC [18]. Through these APIs applications can control the media presentation, audio and video attributes, and get access to media stream events for synchronization.

DASE APIs

APIs designed specifically for DASE consist of extensions that provide access to ATSC specific features such as PSIP, Data Broadcast protocol, etc. or functionality that was not included in other adopted APIs. DASE-specific APIs include:

1. SI API provides extensions to the JavaTV SI API in order to include ATSC specific features as defined in PSIP (T3/S8).

2. Data Broadcast API extends the JavaTV data broadcast API in order to provide access to ATSC specific data signaling features as represented by the data broadcast specification defined by T3/S13 [4] including an abstraction of the Data Service Table (DST).

3. Application Management API extends the JavaTV Xlet API in several ways. The Xlet is initialized with some ATSC specific objects including the data service information the application is part of. This API also adds a concept of an Application Manager where applications can be registered for a persistent download and which provides access to other running applications, in the form of an application proxy, subject to security policy limitations. A significant functional extension is provided by the manageability instrumentation API, which is based on the ITU-T X.731 state management standard [17]. This API enables content providers or network operators to monitor the activities on each DTV receiver, get notified of problems, collect usage statistics and control the behavior of downloadable applications. This capability will be essential in order to provide a very reliable and fail-safe operation of each receiver.

4. User Management and Preferences API. The Preferences API allows applications to get access to the user-defined settings and preferences, and potentially modify such preferences. These may include preferred language, user favorite channels, rating ceiling, etc. If the DTV receiver supports a multi-user environment, the User Management API provides a basic mechanism to create, delete and activate users, associate a list of preferences and permissions with each user and invoke user authentication functions.

5. System API is a simple set of properties used to communicate the specific receiver API specification and implementation version numbers, system profile and similar attributes.

6. Security API enables the internal security policy to be enforced through any DASE API. This group of APIs defines a set of security-related exceptions and security-related permissions, which may be used in defining the security policy as well as aid in implementing such a policy.

HAVi APIs

Home Audio/Visual Interoperability (HAVi) is a home audio-video networking specification [16]. It includes a set of Java APIs whose subset is currently included in the DASE specification. HAVi User Interface APIs provide additional features not found in Java Abstract Windowing Toolkit (AWT) such as transparency, support for video, remote control events, etc. There are two packages included from the HAVi specification: the User Interface package and the Remote Control Event package.

Personal Java

The entire DASE API set is expected to be implemented on top of the Personal Java platform [10]. Such a platform provides the

basic APIs necessary for any device to run downloadable applications and provide basic OS-like functions. Note that some packages or classes may be included only optionally, such as AWT or the Applet package.

CONCLUSION

The purpose of this article was to provide a background on the current standardization work in the area of enabling enhanced interactive content for digital TV receivers, specifically the ATSC effort to define a set of Java APIs providing access to the receiver functionality and user data. A standardized set of APIs is one of several requirements for a platform supporting downloadable interoperable applications.

Details about other parts of the DASE architecture, the Presentation Engine, as well as the draft specification including a detailed JavaDoc documentation may be obtained from the DASE Web site [2]

REFERENCES

[1] Advanced Television Systems Committee (ATSC): http://www.atsc.org

[2] ATSC T3/S17 DTV Application Software Environment (DASE): http://toocan.philabs.research.philips.com/misc/atsc/dase/

[3] Program and System Information (PSIP): http://www.atsc.org/Standards/A65/

[4] T3/S13 Data Broadcasting: http://toocan.philabs.research.philips.com/misc/atsc/t3s13/

[5] T3/S16 Interactive Services: http://toocan.philabs.research.philips.com/misc/atsc/t3s16/

[6] Society for Cable Telecommunications Engineers (SCTE): http://www.scte.org/

[7] Java language specification: http://java.sun.com/docs/books/jls/index.html

[8] Java Virtual Machine (JVM): http://java.sun.com/docs/books/vmspec/index.html

[9] Sun Microsystems JavaTV http://java.sun.com/products/javatv/

[10] Personal Java: http://java.sun.com/products/personaljava/spec-1-1/pJavaSpec.html

[11] Java Media Framework (JMF): http://java.sun.com/products/java-media/jmf/forDevelopers/playerguide/index.html

[12] HyperText Markup Language (HTML): http://www.w3.org/TR/REC-html40/

[13] Cascading Style Sheets (CSS): http://www.w3.org/Style/CSS/

[14] Document Object Model (DOM): http://www.w3.org/DOM/

[15] Extensible HyperText Markup Language (XHTML): http://www.w3.org/TR/WD-html-in-xml/

[16] Home Audio/Visual Interoperability (HAVi): http://www.havi.org

[17] ITU-T X.731 State Management: http://www.itu.int/itudoc/itu-t/rec/x/x500up/x731.html

[18] Digital Audio Visual Council (DAVIC): http://www.davic.org/

[19] Digital Video Broadcasting (DVB): http://www.dvb.org/

[20] OpenCable: http://www.opencable.com/

About the Author

Petr Peterka is a Senior Staff Engineer in the Advanced Development department of General Instrument Corporation. He serves as a vice-chairman of the ATSC T3/S17 specialist group and as a Java API software architect for this group. He has also actively participated in the development of a number of APIs discussed in this paper. Petr can be reached via e-mail at ppeterka@gi.com.