MHEG-5 and Java
– the basis for a common European API?

A. Mornington-West
ITVA UK

The use of different proprietary APIs in digital television receivers is leading to a fragmented market in which the consumers are losing out, while the broadcasters battle to achieve exclusive ownership of a primary gateway to the viewer.

The Author stresses the need for an open universal API and describes how this could be achieved using the MHEG-5 content decoder in conjunction with a Java-based Virtual Machine layer. He also describes a way forward to enable a practical migration from the use of existing proprietary APIs to the use of a single universal API.

Introduction

The digital television receiver can produce more than just sound and pictures – provided there is some way in which it can be sent additional instructions. These instructions may either be in a form which the receiver can execute, or in the form of content which the receiver can present to the viewer (or listener). Application instructions are written in the language of the specific *application programming interface* (API) which is installed in the receiver.

The use of different proprietary API products has flourished in the short lifetime of digital TV broadcasting (where the API has to satisfy somewhat differing requirements to those found in the PC environment). This has led to a situation where applications[1] that have been devised for one group of receivers cannot be executed by another group. It has also prevented those receivers associated with a particular service provider from being fully useful when using the services from a competing service provider.

While proprietary APIs have undoubtedly helped the launch of digital TV services, they have allowed private broadcasters to determine the features of the receivers which they subsidise, in order to gain a strong if not monopolistic share of the market for digital television broadcasting. Such a strategy will inevitably produce a fragmented market in which the consumer

1. In this article, "application" is used in the same sense as in the computer environment. It is taken to mean the total package of *executable content* (or instructions) and *visual/audible content* which together make up the multimedia programme.

Original language: English
Manuscript received: 17/3/98.

will be the loser as the broadcasters battle to win ownership of a primary gateway to the viewer.

A more beneficial approach would be to use a *universal API*. In this scenario, manufacturers would be able to concentrate on developing the market for receivers which implement a range of enhanced features for which the consumer would be willing to pay, whilst the broadcasters would be able to focus on the delivery of attractively-priced services which appealed to all the consumers.

Why do we need an API?

The composition of software-oriented products can be viewed in terms of *layers*. Each layer provides a suite of functions and provides a service to the layers above and below it. In a well-organized software environment, there should be no reason for processes in any one layer to bypass another layer in order to carry out a function. The most commonly-quoted model is that of the *Open Systems Interconnection* (OSI) *seven layer model*. Although not all software-oriented products can be mapped onto this model, the concept does help us to visualise the structure of many software-oriented products.

Using this model, the API in a digital TV receiver provides a predictable buffer layer between the hardware and the user-oriented executable applications (*Fig. 1*).

The hardware drivers for the decoder chips in receiver are usually provided by the semiconductor manufacturers, as detailed specialist knowledge is frequently required in this domain. In some designs there is a *Virtual Machine* (VM) layer which can provide isolation from the detail of the core processor used in the receiver design.

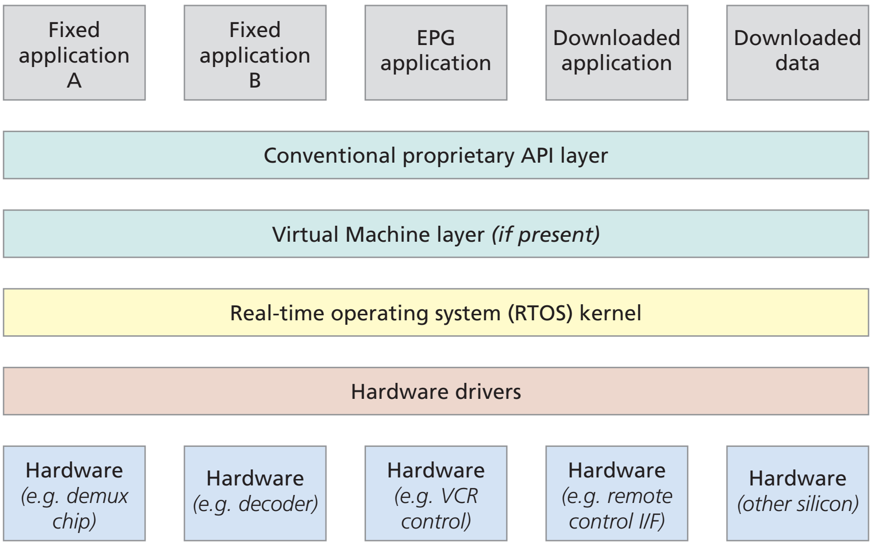


Figure 1
Layered software model of a digital TV receiver employing a proprietary API.

Applications which can be written include the basic man-machine interface (MMI) to the receiver and its remote control. The EPG is the most common example of an application which has been written for receivers and it can, in principle, be implemented as a fixed application, or as an application which is provided by or updated by the broadcast signal. Some receivers have access to a modem and this can be used to provide a return channel for the consumer to interact with remote sources of information. At the present time, the principal use of the modem has been in conjunction with the conditional access (CA) control system which many receivers use.

The use of a VM layer offers a way in which the API can become independent of the host processor. The features offered by a VM differ from one design to the next. Some designs include

Abbreviations

ADSL	Asynchronous digital subscriber line
API	Application programming interface
CA	Conditional access
DAVIC	Digital Audio-Visual Council
DVB	Digital Video Broadcasting
EPG	Electronic programme guide
ESG	Event schedule guide
HDTV	High-definition television
HTML	Hyper-text markup language
I/F	Interface
IEC	International Electrotechnical Commission
IHDN	In-home digital network
IPR	Intellectual property rights
IRD	Integrated receiver/decoder
ISO	International Organization for Standardization
JAVA	Programming language for the WWW (developed by Sun Microsystems)
LMDS	Local multipoint distribution system
MHEG	(ISO/IEC) Multi- and Hyper-media coding Experts Group
MMDS	Multipoint microwave distribution system
MMI	Man-machine interface
OSI	Open systems interconnection
RTOS	Real-time operating system
SI	Service information
STB	Set-top box
VM	Virtual machine

full checking of the address range to which an application should have access; they will report an error if this boundary is exceeded.

Some diagrams of the OSI model omit the real-time operating system (RTOS), although this process is always required.

The API can provide a reliable interface: applications can be written for it without the programmer having to know precise details of the hardware design of a particular receiver. As a consequence, it should be possible to state confidently that an application which executes faultlessly on one model of receiver should execute as reliably on a different model, provided that the API is the same.

In practice there appear to be two reasons why application download is often associated with the CA system. The first is simply that it is not usually in the commercial interests of one broadcaster to allow a competing broadcaster to have unbounded access to receivers which he may have subsidised. The second relates to the fear that a rogue application could modify the behaviour of receivers in an unwanted manner and that this, in turn, could result in dissatisfied viewers or expensive service calls.

Both of these reasons need to be addressed if an open market for unsubsidised receivers is to be established. There should be no requirement which demands that specific proprietary features are implemented. Thus, there should be no need to rely on the use of a CA system. An open API could offer the consumer a wider choice of receivers, provided that all services which could be received were interoperable. To assist in the target of protecting the receiver from the effects of a rogue application, a robust and arguably secure VM needs to be specified.

We can draw a distinction between two classes of API. The first, which we call *declarative APIs*, focus on processing the content of data for the consumer. Obvious examples include HTML browsers, MHEG and even the world standard teletext system. They are characterized by their emphasis on the decoding and presentation of data content in a uniform and consistent manner. One feature of content decoders is their robustness, and this reliability is certainly valuable if viewers are to be saved the problem of learning where the reset button is located!

The second group comprises the more conventional *procedural APIs*. These APIs make available to the programmer the full panoply of conventional computer language constructs. One problem area here is to ensure that all models of receivers from a wide range of manufacturers implement the API in exactly the same manner, in order to ensure that any application which is executed will behave in a predictable manner. This can lead to the expensive practice of having to forward the receiver to the owner of the API for conformance testing. It can also slow down the development of applications.

MHEG-5 content decoder

The origins of the MHEG philosophy started within work initiated by DAVIC. The target was to create a content decoder which would intrinsically be aware of the principal characteristics of the digital television system. It would therefore offer:

- awareness of all digital television systems;
- awareness of the pixellations that are available in the DVB bitstream;
- awareness of the video and audio content;
- the ability to link textual and graphical data with specifically timed events;
- a deterministic method of presenting the text blocks and text flow, using a subset of the suite of HTML tags to control the text formatting.
- a suite of MMI tools which could be used to provide the viewer with selection buttons and sliders for actions such as level control, character entry and screen navigation.

MHEG-5 was developed in order to support the distribution of interactive multimedia applications to executing platforms of different types and of different brands. The bulk of the code is declarative but there are provisions for calling procedural code as required. Naturally, the receiver will need to have a run-time interpreter which can process the commands as they are delivered over the air.

DAVIC has produced a suite of specifications which incorporate the MHEG standard. Each of them builds on the previous versions in a compatible manner.

- ***DAVIC 1.0*** was published in January 1996. It provided a set of tools to support basic applications such as TV distribution, near-video-on-demand, video-on-demand and simple forms of tele-shopping.
- ***DAVIC 1.1*** was produced in September 1996. It added tools to support basic "Internet compatibility", the addition of microwave broadcast networks (MMDS and LMDS), set-top units that are network-independent and set-top units that can behave as virtual machines.
- ***DAVIC 1.2*** was released in December 1996. It included some tools to enable TV networks to provide Internet services at high speed to television and PC users, and also defined the HDTV formats and systems for conditional access.
- ***DAVIC 1.3*** is the current version, dating from September 1997. It has added: comprehensive Service and Network Management; multiple broadcast servers; mobile reception; scalable audio; content and meta-data packaging; Java APIs for DVB service information,

and a new concept of *contours* – the first instances are the *Enhanced Digital Broadcast* contour and the *Interactive Digital Broadcast* contour.

Work on ***DAVIC 1.4*** continues and a ***DAVIC 1.5*** version is expected to be released in December 1998. Among the features which are expected to be added in versions 1.4 and 1.5 will be: the addition of the Java API; MHEG5-resident programs to access the DVB Service Information (DVB-SI), and further integration of the DAVIC and Internet content. The size of the DAVIC specifications means that use will be made of the contours concept; features will be carefully selected from the full feature set, in order to suit the required functionality of the receiver and the cost of developing the software.

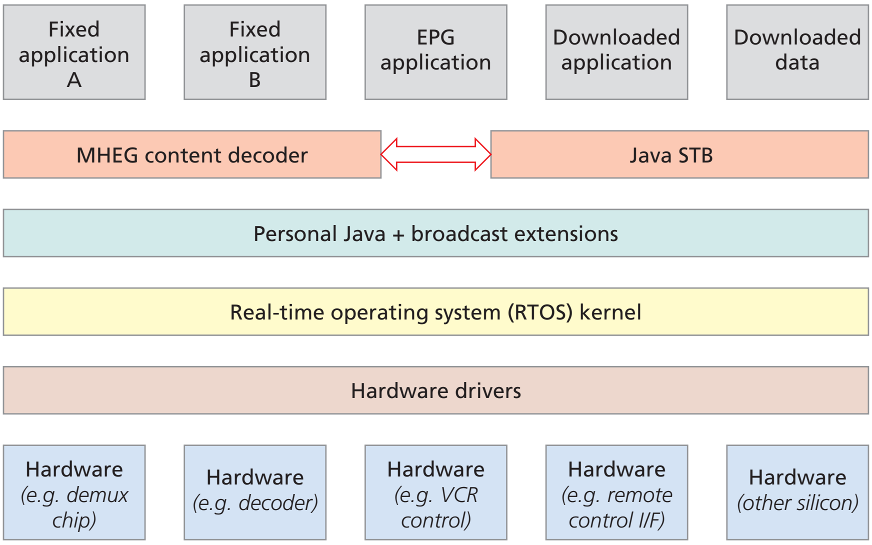


Figure 2
A layered model of a digital television receiver with DAVIC 1.5 components.

The combination of the MHEG-5 content decoder and the Java API is represented in *Fig. 2*.

This diagram suggests that the proprietary API layer can be removed and this may be the case for receivers marketed in some regions. There is obviously scope for both MHEG and Java to be either resident in the receiver or transmitted over the air (so-called *airware*). One use of a resident MHEG-5 application would be to provide the essential features of an event schedule guide (ESG). This is a guide which is based on the use of published DVB-SI data tables.

MHEG is an object-oriented environment in which successive scenes can be joined together to construct the overall application. Objects in a scene may include graphics, sound or video, as well as the ability to respond to local actions such as key presses. MHEG does not specify exactly what form a remote control or other input device should take; it simply specifies the functions which shall be provided, irrespective of the approach taken.

Only one application may be active at any one time although the *ingredients* which are referred to in a scene will be available to other active scenes. The ingredients include:

- links;
- procedures;
- palette;
- font;
- variable.

The ingredients are fairly self-explanatory; for example, a *link* provides a facility for handling the choices which a viewer might make, while *variable* indicates some storage space that can be used to pass values to and from procedures and other MHEG objects. These objects are grouped into classes and an important one is the *visible class*. It implements the presentation of displayable objects – such as line art, video, text, sliders and buttons – on a screen.

The route to migration

The inclusion of the Java VM into the specification is significant. It provides digital TV receivers with two key benefits. The first is a definition of a *processing engine* which, as far as any application need be aware, can be independent of the actual hardware processor which has been chosen by the manufacturer, i.e. receivers from different manufacturers should perform in the same way. The second benefit provides *reliability*, because the VM has been defined in order to protect applications from interfering with each other or with the background operating system processes.

The DAVIC standards set out to use existing public standards where possible. A public standard is not necessarily free of the costs of paying for any associated intellectual property rights (IPRs). DAVIC includes such standards as required, provided that any IPR is agreed to be made available on fair and non-discriminatory terms. The Author believes that the current status of MHEG-5 is such that there are no IPR fees which need to be paid and this is part of its attraction to digital television receiver manufacturers (the licence royalties for proprietary APIs can be significant). Of course, the other attraction is that a single API means that the development work can be amortised over a much larger market and this too will help to reduce the costs.

Interoperability of digital STBs has been elusive – sufficient at least to prevent any service operator from having the confidence to place orders for an integrated receiver / decoder (IRD). It seems likely that the next major growth phase in the market for digital television requires the implementation of standards such that the consumer can rely on true interoperability, not only of the primary video and audio signals but also of all the further signal components which may be present. The most heralded of these are the signals for data services, as these provide digital receivers with their full scope for interactivity.

Expansion of this market, and of the whole gamut of products which could be produced for the in-home digital network (IHDN) environment, is unlikely to take place until there is a uniform API. This is perhaps the last step to removing all the individual proprietary technologies that are found within today's receivers. These technologies result in design implementations which do not enable full interoperability, even within the same country!

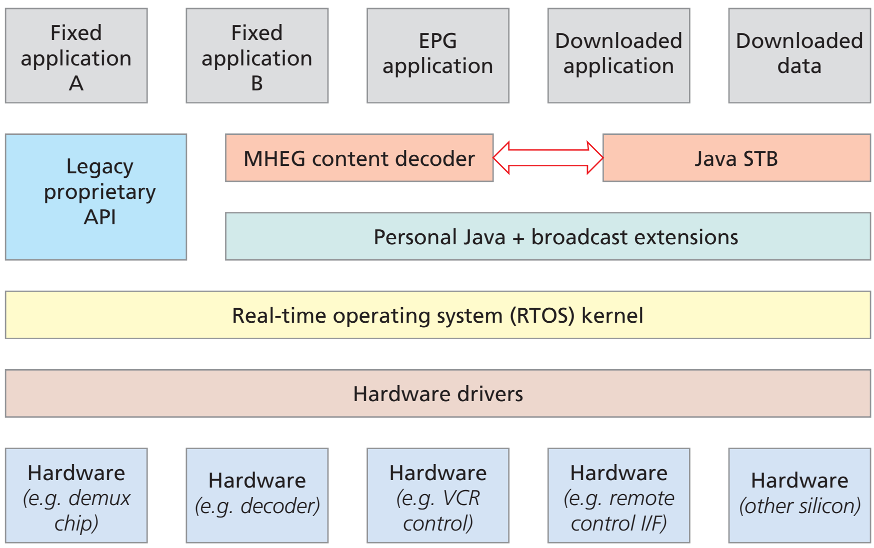
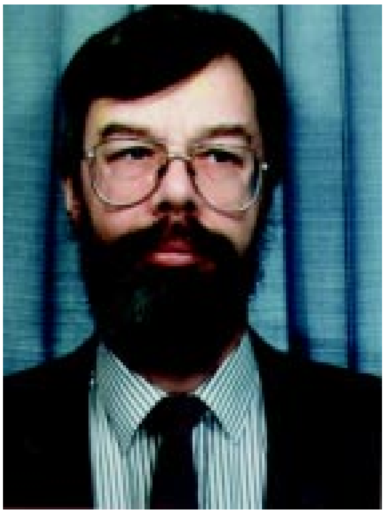


Figure 3
An intermediate state in which a legacy proprietary API may co-exist with MHEG-5 and Java.

Fig. 3 shows one of the many scenarios in which current proprietary APIs may still be used within receivers in the future. One use for the legacy API would be to provide an interpreter for the MHEG content. This may provide a slower performance but it may still provide a reliable product which can be delivered relatively quickly to the marketplace.

Some legacy applications could be supported for the full lifetime of the STB and this may involve the service providers in sending application content for both the legacy receivers and



Allen Mornington-West *graduated from Durham University in north-east England many years ago and spent his formative years designing audio mixing desks, audio effects units and audio power amplifiers for a number of manufacturers. A period of some 8 years was spent at the Independent Broadcasting Association, working in the areas of quality control and measurement practice within the independent television companies.*

More recently Mr Mornington-West held the post of Engineering Director, Quad Electroacoustics, until the challenge arose to assist the Independent Television Association to enter the digital television era.

Allen Mornington-West is a Chartered Engineer, an MIEE and a Fellow of the Institute of Acoustics. He has been active in standards work, including assisting the professional audio and video product-family EMC standards, and the AES 24 communication standard. He is currently working in various ITU-R, EBU, DVB, DigiTAG and UK DTG working groups, some of which he chairs. He is a regular presenter of papers at UK and European conferences and he frequently lectures and writes articles.

the new receivers for a number of years. This would only be feasible where the cost of such "double illumination" with *airware* was cost-effective.

These concerns have formed the background for the work of the DVB multimedia home platform (DVB-MHP) group [1]. The key requirement is to chart a migration path so that receivers which are due to be introduced to the market – and perhaps those which may already be on the market – can be adapted to conform to an agreed uniform API.

At the present time this work is still in progress although it should be noted that the emphasis now is on a solution based on MHEG with a Java VM. This approach is strongly supported by a number of receiver manufacturers and by many broadcasters – particularly those planning to introduce terrestrial digital television services.

Bibliography

[1] J.-P. Evain: **Multimedia Home Platform – an overview**
EBU Technical Review, No. 275, Spring 1998.