OPENCABLE APPLICATION PLATFORM ARCHITECTURE

Allen R. Schmitt-Gordon, Ph.D.
Cable Television Laboratories, Inc.

ABSTRACT

The OpenCable Application Platform (OCAP) is a software middleware layer that resides functionally on top of the Operating System of a OpenCable terminal or receiver. It provides an interface enabling application portability. A fundamental requirement is that applications written for OpenCable be capable of running on any network and on any hardware platform, without recompilation.

The OCAP specification is built on the DVB MHP 1.0 specification with modifications for the North American Cable environment that includes a full time return channel. A major modification to the MHP is the addition of a Presentation Engine (PE), that supports HTML, XML, ECMAScript. A bridge between the PE and the Java Execution Engine (EE), enables PE applications to obtain privileges and directly manipulate privileged operations.

INTRODUCTION

The OpenCable Application Platform (OCAP) is part of a concerted effort, called OpenCable(TM) by the North American cable operators (MSOs) to provide the next generation digital consumer device, encourage supplier competition and create a retail hardware platform. A cable receiver that can be provided at retail must provide interoperability and portability of content and applications across networks and platforms, and be geared towards the full range of interactive services. Current devices are network specific and operate proprietary software that is not portable across platforms or networks. With OCAP, the applications are written primarily to a middleware (software layer between the operating system and the application software) API so that a single application can be deployed across the full range of OpenCable host devices available at retail. Such applications include:

- Electronic Program Guide (EPG)
- Impulse Pay Per View (IPPV)
- Video On Demand (VOD)
- Interactive sports, game shows
- E-mail, Chat, Instant messaging
- Games
- Web Browser: Shopping, Home banking
- Personal Video Recorder (PVR)

Another essential requirement is that the middleware be secure and robust. Stability in the Cable terminal or receiver is imperative and resets are not acceptible. The middleware must also simplify content development, through a publicly available application programming interface. Finally, the OCAP must be operating system and hardware agnostic. That is, OCAP will specify only one profile. Applications and content will be written to only one profile.

Background

This development of the OpenCable middleware was initiated in September, 1999 through the RFP process, 16 vendor companies submitted proposals by October 15, 1999 for a middleware architecture that would enable such portability. Review of the submitted proposals was completed by the end of December of 1999 and several middleware vendors were selected to refine the architecture and develop the OCAP specification. Initial specification development was started in January of 2000 and development teams comprised of the selected vendors, members of the Cablelabs staff and visiting engineers were assembled.

The RFP was worded such that the specific components of the middleware architecture were not specified but left open to the respondents. The architecture that was chosen by the technical team was a middleware that was comprised of two parts: a Presentation Engine (PE) and an Execution Engine (EE). The PE generally was composed of an HTML engine and ECMAScript. The EE included a virtual machine. This architecture is shown in Figure 1. It shows that native applications are supported as well as application written to the middleware via the OCAP interface.

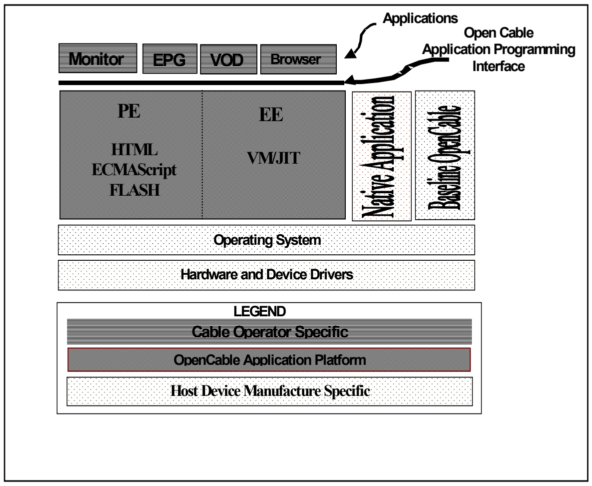

Figure 1

One of the key elements to the development of the OCAP architecture has been the licensing agreement with Sun Microsystems. Sun has provided pertinant portions of the Java API specification and related IP to Cable Labs. Anyone implementing the OCAP specification may implement the Java API without obligation to Sun. Cable Labs will incorporate Sun Technology Compatibility Kit (TCK) as part of Open Cable compliance test suite, and only Sun Java licensees may use Java branding. This agreement enabled the OCAP team to specify the Java Virtual Machine (JVM) and JavaTV as the fundamental components of the EE.

In order to expedite development of the OCAP specification, it was necessary to utilize existing standards and architectures, as much as possible.

ARCHITECTURAL COMPONENTS

Rationale for Middleware

The current software architecture in general usage, requires that all applications be written to the specific Operating System running on the hardware platform being used. In an environment where cable receivers are leased and are derived from a limited number of suppliers, this model works to some extent. It does not enable platform or network portability. Applications such as EPG, VOD, mail, etc are compiled to the application programming interface determined by the operating system and associated hardware. This is illustrated in figure 2.

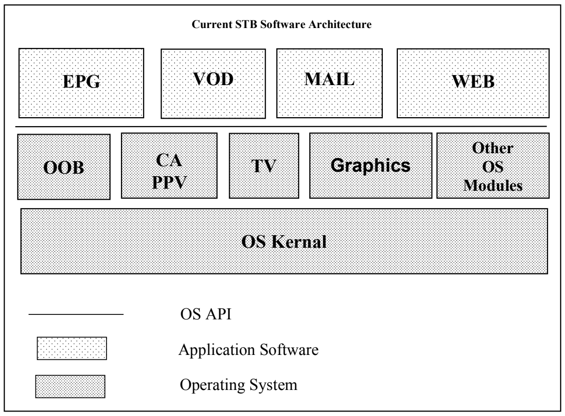

Figure 2

By imposing a middleware layer, that abstracts the functionality of the OS, hardware devices and network interfaces, such as shown in figure 1, a variety of applications can written that will run on any platform. Such applications will run, without recompilation, as would be the case for applications written in C or C++. Careful design will enable network portability as well.

Presentation Engine

The PE enables the use of tools that have been widely used for internet content. The PE renders declarative content such as graphics, text, animations and audio based on formatting rules in the PE itself and formatting instructions such as the markup language.

Its primary components consist of HTML 4.01, XHTML 1.0, CSS 1 and CSS 2, and ECMAScript 3. In addition, advanced TV is supported through Macromedia Flash and Plug-ins to access other Web content formats.

Unless the PE has many extensions added to it, certain functionality will be lacking. Most notable would be tuning. In order to facilitate this functionality and not circumvent the inherent security of Java as well as provide a common mechanism for managing receiver resources, the middleware architecture will include a bridge between the PE and the EE.

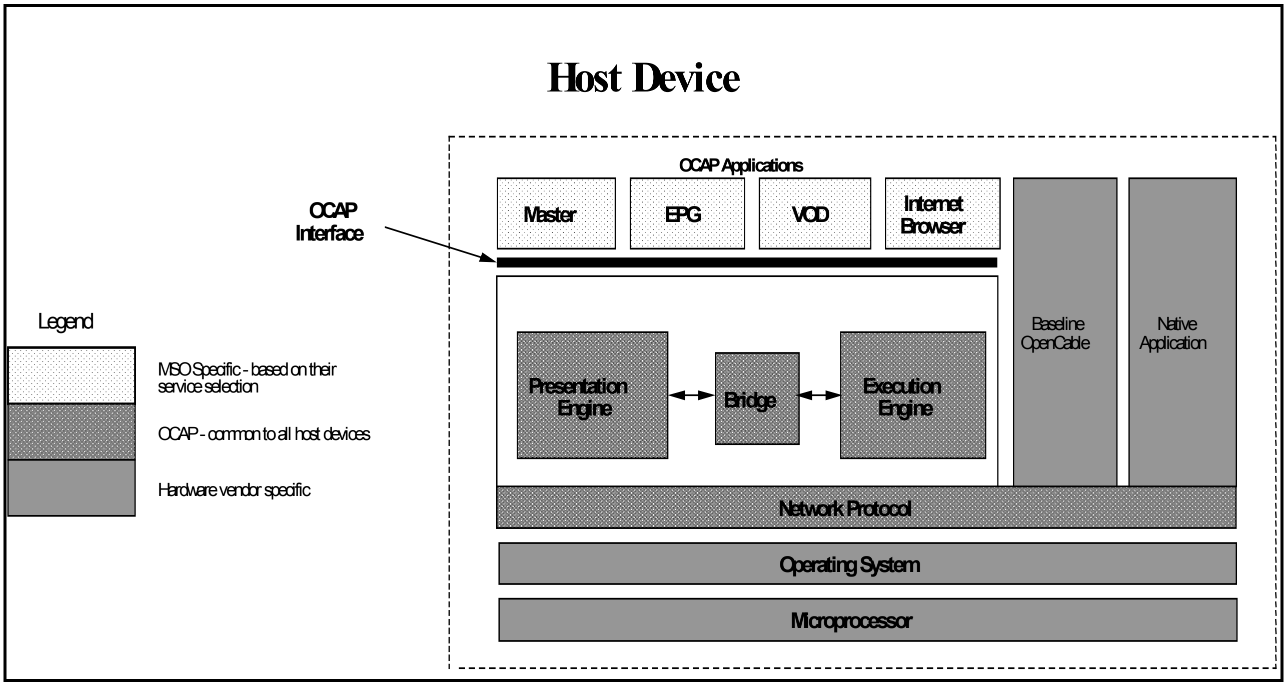

Figure 3

For example, as part of the EE that is accessible via the bridge, JavaTV offers a common point of control and management of various system resources that includes tuning. Thus, a PE application will access device resources through the bridge. This ensures that device resource contention is managed through a common control point for a PE and EE application. that are vying for the same resources.

Execution Engine

The EE provides a general application programming evironment for networking, file I/O, graphics, etc. The OCAP EE (with MHP) provides a full TV application environment.

Java is more portable than C or C++ and provides a platform independent byte code. The EE provides a platform independent set of API's.

As a starting point in the development of the EE, the DVB-MHP 1.0 specification was chosen to build upon in order to expedite the specification development process. OCAP specific extensions have been added to cover those elements such as a full time return channel, application management, and resource, service information.

The major elements of the EE are control of application management through the pJava APIs, control of application management through the pJava APIs, service information and selection through the JavaTV API's, media control through the JMF, broadcast data through the MHP DSMCC APIs. In addition, the EE provides network management and IP data access and extensions from HAVI and DAVIC and DASE.

A fundamental feature of the EE utilizing Java is that security is built into the architecture from the ground up.

Bridge

In order to enable the browser to take full advantage of the resources in the receiver, the design of figure 1 was expanded to include a bridge between the EE and the PE. This is also shown in figure 2. The bridge permits access by ECMAScript applications to the Java Class Libraries and Java programs access to the DOM files. Thus a full programming environment is available for application written for the EE or the PE.

Through the use of the bridge to extend PE functionality beyond what would normally be possible for a browser, the use of plug-ins is minimized. Plug-ins would normally be written in native code, so that many ports of each plug-in would have to be maintained.

Conclusions

The OCAP architecture offers the widest range of support for high quality, attractive, TV centric applications and display of content currently being proposed or offered. It offers a very high degree of portability and uniformity for content display as well as offering a platform for the broadest possible range of application support. The OCAP architecture and certification process ensures security and robustness.

References

Kar, M.L., Vang, S., and Brown, R., Architecture of Retail Set-Top Box Application Platform for Digital Cable Netork, Proc. ICCE, June, 2001.

Zundel, J-P, Emergence of Middleware in Home Telecommunication Equipment, IEEE Communications, June, 2001.

DVB MHP 1.0 -- TM220 3 Rev.4 CM195 Rev.1 SB28(00)07, DVB Multimedia Home Platform Revision 16

ATSC DASE AEE, Doc. T3-530 09, Feb 2001, Rev 1.

Havi Level 2 User Interface, section 2.5.2, http://www.havi.org

Documents relating to the PE including DOM, CSS, HTML can be found at http://www.w3c.org

Documents relating to the Davic specification can be found at http://www.davic.org