ETV – CABLE'S INTERACTIVE PLATFORM FOR LEGACY AND ADVANCED RECEIVERS

Frank Sandoval – Director, OCAP Specifications, CableLabs
David Hooley – CableLabs

Abstract

Enhanced Television (ETV) is a cable defined platform for delivering interactive video applications. Interactive games, game shows, reality programming, awards shows, sports, and advanced advertising are some of the anticipated applications. The platform has been defined by CableLabs within the OpenCable project, with the hearty support of the MSO community.

This paper will describe the technical aspects of the platform, including the application format and on-the-wire signaling. CableLabs has also developed User Interface and Operational Guidelines to address deployment issues.

MSOs and programmer partners, along with tools vendors, have collaborated on several interoperability events, and technical field trials are being contemplated.

Finally, a set of anticipated business models will be introduced.

ETV – ENHANCED TELEVISION

Enhanced and Interactive TV

As seems common in the emerging digital TV landscape, new terms almost always mean different things to different people. For our purposes, the term 'Interactive TV' will be used to denote interactive features that augment the viewing experience but are a part of the content of a program; Electronic Program Guides, or stock/news/weather crawls are examples.

'Enhanced TV' will refer to video programming that has an interactive application bound to it; an application that executes concurrently with the program, and terminates when the program is no longer being presented. Examples include an awards show that allows you to vote on upcoming awards in real-time, and an advertisement that lets you request more information about a product.

An enhanced television program is one that has had an interactive component produced to magnify a viewer's experience. The interactive component is called, naturally, an enhancement.

OCAP Bound Applications

OCAP, the OpenCable Applications Platform, is another MSO initiative managed by CableLabs. OCAP is a Java environment, meant to abstract system details away from applications, so that interactive applications may be deployed on a wide range of televisions and set-top boxes. OCAP contains definitions for 'service bound' applications. These applications are embedded into the same MPEG-2 Transport Stream as the associated audio and video. When a receiver tunes to a program, it reads the data PIDs that contain the application, and where an application is present, loads and launches it.

ETV applications are transported and executed in a similar way. They are accessed

when a viewer tunes into an enhanced program, and are terminated when the show ends or the viewer changes channels.

Project Overview

The CableLabs Enhanced Television project was initiated in early 2004 to provide a technical solution for ETV on legacy thin-client cable receivers that would also be appropriate for advanced receivers. MSO and vendor participation has been consistently strong and progress has been rapid.

Mission

Provide a complete technical definition of a national cable ETV platform, to enable deployments near year-end 2006.

Deliverables

ETV applications are compact sets of files that contain the programming logic and resources, like images, that comprise an interactive experience. These files describe application User Interface (UI) and logic and support a set of UI 'widgets', return channel capabilities, and real-time event synchronization mechanisms.

A 'user agent' software module on a receiver reads and renders ETV applications based exclusively on the content of these resource files. The resources are multiplexed with their associated A/V programs in the A/V stream. On-the-wire signaling instructs user agents to acquire and process these resources.

Both legacy (written in native platform code) and advanced (written using OCAP APIs) user agents are supported

The project deliverables are a set of guidelines and specifications that provide an end-to-end platform. The deliverables and timelines are:

- Binary Interchange Format (BIF) Specification. An application format that can be executed on the widest range of receivers, including limited capability devices such as the Motorola DCT 2000, while taking advantage of the advanced capabilities of more advanced devices. This format allows applications to be network and device independent, providing a nationwide cable footprint for ETV. The initial version, Version I01, was issued Q2 '05. Version I02 was issued Q3 '05 and included a number of clarifications and improved features.

  Version I03 will be issued Q1 '06 and will include: support for on-demand sessions, greater graphic display capabilities such as translucency, and an XML format of BIF that enables software tools to easily exchange applications and application elements.

- Application Messaging Specification. Defines network signaling to enable application acquisition by receivers and application synchronization with programming. Based upon OCAP signaling mechanisms, it adds features to support ETV applications and to target limited capability devices.

  Version I01 issued Q2 '05. Version I02 issued Q3 '05.

- Operational Guidelines. This document describes the overall end-to-end architecture and provides information on deployment mechanics, such as downstream and upstream bandwidth management, access to receiver resources, and other relevant topics. A key component of this architecture describes possible national broadcast distributions. Since the initial work was begun, the scope has expanded to include return channel, VOD, testing and other deployment-related issues.

  Version I01 is expected to be issued Q1 '06

- User Interface Guidelines. This document describes how ETV applications are presented to viewers and describes the basic ground rules for providing simple and consistent user experiences. This document defines the cable equivalent of BSkyB's 'red button' commonly used in Britain.

  Version I01 issued Q1 '05

- Metrics Gathering. This document describes an OCAP-oriented approach to gathering metrics data from ETV receivers. Still in its formative stages, this effort will focus on such measurements as click-stream, application lifecycle, and others.

  Version I01 targeted Q4 '06

Features and Functionality

The ETV platform is specifically designed to support relatively simple applications that are delivered with a TV program. Network bandwidth and receiver memory is assumed to be quite limited, constraining the typical size of applications, and they are not generally allowed to perform sensitive operations on a receiver. The platform is not designed to provide a full programming environment for workhorse applications like EPGs.

ETV application make very efficient use of bandwidth and memory resources, through special encoding and data sharing, can be targeted to many receiver classes, executing in a limited fashion on lower end devices while accessing advanced features of others. They can be extended to support innovative receiver and network specific functions.

ETV applications present a user interface through the use of common widgets. Widgets, such as images, text boxes, selectable buttons and so on, can be customized to suite the application's needs. ETV applications can communicate with head-end modules, can save local files if resources permit, and can be tightly synchronized with their associated programs.

The current, baseline ETV platform includes support for some network-specific functions like VOD.

Some Implementation Details

The user agent is responsible for moving an ETV application between these lifecycle states:

- Unloaded
- Loaded
- Suspended
- Running
- Terminated.

A running application is in one of the following navigation states:

- Unrealized
- Disabled
- Enabled
- Focused
- Armed
- Selected.

Multiple applications can run concurrently, and a suspended or not-in-focus application can execute while in the background.

Two-way processing is supported via form submission and form response. Both scheduled and do-it-now stream events (triggers) are delivered to the user agent via in-band data.

An application is characterized by the resources at its disposal, consisting of page, data, platform and other resources such as images and MPEG stills. Each resource structure consists of a header, one or more tables, and a data heap.

All application elements are referenced through tables, and table entries reference data structure offsets in the heap. Some example tables and their contents are:

- *Action Table:* application script references (eg, LoadApp, SelectService, SubmitForm, AddWidget, StoreVariable)
- *Generic Data Table:* self-defining data structures
- *Metadata Table:* application metadata
- *Platform Directory Table:* pointers to platform-specific code and data sections in the resource
- *Reference Table:* acts as a variable store
- *Resource Locator Table:* pointers to external resources and to response data
- *Trigger Table:* trigger schema definitions
- *Widget Table:* provides the hierarchy of pages in the application (eg, Container, Form, Button, Video)

Messages are conveyed to the application using the in-band Enhanced TV Integrated Signaling Stream (EISS). There are three descriptors conveyed in the EISS table:

- *Application Information Descriptor:* used to autostart, present or destroy the application
- *Stream Event Descriptor:* used to send ‘do-it-then’ stream event and generic payload data to the application
- *Media Time Descriptor:* provides the clock for ‘do-it-then’ stream events

ETV and OCAP

ETV complements OCAP by enabling what OCAP calls ‘bound’ applications on non-OCAP receivers. Although the application format is new, the network signaling is common with OCAP. Recall that a receiver ‘user agent’ renders ETV applications, so ETV support can be added to OCAP by defining how a user agent may

operate as an OCAP application, and/or be defining how a user agent can be built into an OCAP implementation.

Interops

The first ETV Interoperability event was held in Aug 2005. There were three User Agents hosted in two major set-top environments and several stream generators and application toolkits represented. This first event provided an excellent demonstration of initial developments, ranging from simple "Hello World" type applications to a sophisticated application designed to accompany a nationally broadcast network program.

The second event was held in February 2006. The target here was for a complete implementation of the platform, and that goal was largely achieved. More extensive use was made of EBIF capabilities, including bitmap images and transparency. Full application signaling was supported, including the sending of real-time stream events. Two-way communication was performed over the return channel, and included retrieval of a VOD asset from a live VOD server. Some basic metrics gathering from receivers was also demonstrated.

Deployment Goals and Field Trials

The CableLabs Executive Committee (CEOs) has directed CableLabs to facilitate and provide project oversight for an ETV field trial in early 2006, with the strategic goal of providing a baseline for national deployment by several MSOs in late 2006.

A key portion of the Operational Guidelines specification is a set of potential national broadcast scenarios, with the issue of affiliate pass-through of ETV programming a particularly thorny one. A field trial is essential in helping to select the viable choices from this list, and ensuring that the affiliate pass-through problem has a solution prior to a deployment decision.

Another gating factor for deployment is a solid framework for gathering and reporting metrics from participating ETV receivers. Again, this is an essential capability that a field trial must demonstrate, and includes considerations of timeliness, chain of flow and security and privacy.

The first phase of the field trial will involve native user agents and alpha versions of applications. Later phases will demand Java user agents and "baked" applications.

Programming Partners

For both the field trials and subsequent deployment, it is vital to have participation from a range of representative programmers and advertising partners. The field trials planned include a broad selection of programmers, broadcast networks, national cable networks, and advertisers.

The revenue-generating business model for ETV deployment must clearly satisfy all of these participants before deployment can become a reality.

Business Models

The notion of enhanced programming has been both widely praised and ridiculed for many years. Time will tell which camp is more closely right, but without a mass medium roll-out one's opinions remain safe from an empirical test. We do have the benefit of some very successful examples on which to pin our hopes. PC/Internet based 'two-screen' applications have been

deployed by many programmers that have shown extremely high take rates. Mobile phone applications have also been deployed to accompany popular awards and games shows. These examples demonstrate a hearty appetite for interactive services among TV audiences.

Another important example is the BSkyB platform in England. Very wide usage of interactive features incorporated into a wide range of programming has been observed. The revenue prospects for BSkyB are improved by the twin facts that return channel communications involve a toll call, which provides a revenue split between the phone and video carrier, and that betting is less regulated that in the US, allowing cash cow applications like real-time horse racing to prosper.

The great hope for enhanced programming on cable is perhaps the opportunity to add interactive features to advertising, thereby increasing the value of ads to both sponsors and networks.

Do not count out the creativity, and pecuniary interest, of the television industry. We may very well see some ‘killer-app’ emerge from an unexpected quarter.