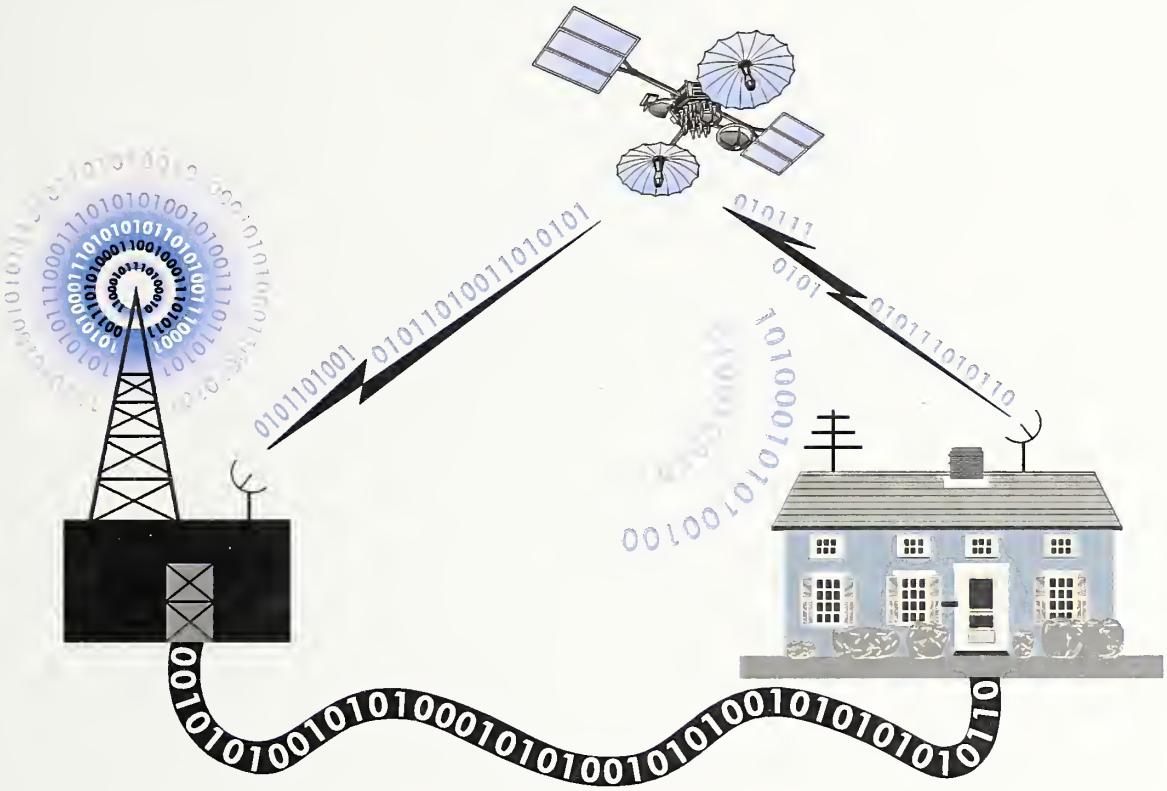
ASE Symposium 2000

NIST IR 6503

Proceedings of the

Symposium on the Foundations of Interactive Digital TV Application Software Environment (DASE)



May 23-24, 2000

National Institute of Standards and Technology
Gaithersburg, Maryland

Sponsored by:

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

NISO
National Information Standards Organization

NISTIR 6503

Proceedings of the

Symposium on the Foundations of Interactive Digital TV Applications Software Environment (DASE)

Edited by:

Alan Mink
Rob Snelick

Information Technology Laboratory

May 2000

U.S. Department of Commerce
William M. Daley, Secretary

Technology Administration
Dr. Cheryl L. Shavers, Under Secretary of Commerce for Technology

National Institute of Standards and Technology
Raymond G. Kammer, Director

Table of Contents

Session 4: DASE Content

Session 5: DASE Conformance Requirements and Testing

Session 6: DASE in Practice

Session 7: DASE and Other Environments

Foreword

As the co-chairs of the DASE Symposium 2000, we would like to welcome you to this inaugural symposium. We have the pleasure of holding the DASE Symposium 2000 at the National Institute of Standards and Technology, just outside our nation's capital, Washington, D.C.

The emergence of interactive digital television (DTV) brings about a host of exciting opportunities for broadcasters, content providers, tool developers, and equipment manufacturers. Interactive DTV combines aspects of traditional television and the internet that inspires applications in e-commerce, targeted advertising, video-on-demand, and enhanced viewing services. An enabling technology for applying interactive DTV is being developed by the Advanced Television Systems Committee (ATSC) Digital TV Application Software Environment (DASE) standards group. The emerging DASE standard, currently a work-in-progress, and how it relates to DTV is the focus of this Symposium. Such a standard environment fosters the interoperability concept of write once, run anywhere applications. The DASE Symposium brings together the DTV industry players to promote commerce and provide an opportunity to learn about DASE technologies in a focused setting.

We hasten to mention that although significant work has been accomplish in the DASE consortium and the structure of the standard is fairly mature, it is important to note that the standard is not finalized and is a work-in-progress.

We would like to thank the speakers for their contributions to both the DASE effort and to this excellent symposium program. We would also like to thank the symposium committee for their support and making this symposium possible. As most of you already know, putting such a symposium together is an arduous task.

Alan Mink
Co-Chair, DASE 2000

Rob Snelick
Co-Chair, DASE 2000

Symposium Committee

Co-Chairs

Alan Mink
Rob Snelick
National Institute of Standards and Technology

Publication Chair

Wayne Salamon
National Institute of Standards and Technology

Exhibit Chair

Pat Harris
National Information Standards Organization

Local Arrangements Chair

Patrice Boulanger
National Institute of Standards and Technology

Registration Chair

Teresa Vicente
National Institute of Standards and Technology

DASE Architecture Overview

Aninda DasGupta

DASE Chairman
Philips Consumer Electronics
<Aninda.DasGupta@philips.com>

DASE -- Impact on Industry and Consumer

Rob Glidden

Market Development Manager
Broadband and Digital Media
Sun Microsystems
<Rob.glidden@sun.com>

The interactive television market today is highly fragmented, with multiple competing proprietary and "walled garden" solutions. Open standards are critical to ultimate market success, and offer to enfranchise stakeholders, increase competition, establish interoperability, and enable content creation. In this context, Java technology offers key benefits for open standards. If the standards challenge is met, then interactive television presents an historic opportunity to empower the media consumer with new levels of information, entertainment, community, and commerce.

ATSC-DASE

Impact of Open Standards on the Interactive Broadcasting Industry

Rob Glidden
Market Development Manager
Broadband & Digital Media
Sun Microsystems, Inc.
rob.glidden@sun.com

Overview

- DTV Market Overview
- Why Open Standards?
- How Java Technology Fits
- The Opportunity

TV Market Overview

- Digital TV Switchover
  - Better video/audio, More channels
  - Interactivity
- Highly Fragmented Marketplace
  - OpenTV, MediaHighway, Wink, WebTV
  - "Walled Gardens"
- Open Standards Becoming Critical
  - ATSC, DVB, W3C, XML, MHEG
    MPEG, CableLabs, ARIB

3

World DTV Standards

4

Why Open Standards?




- DTV Stakeholders Gain Leverage
  - Broadcasters, Network Operators, STB OEMs, Viewers...
- More Competition
  - No One Company Dominates
  - Cost Reduction & Innovation
- Enables Interoperability
  - Create Horizontal Market For STBs
- Encourages Content Creation
  - Removes Fragmentation Barrier

5

Why Java Technology?




- Very Robust Middleware Layer
  - OS & HW Independence
  - Fully-Fledged Programming Language
    - Power and Flexibility
- Portability
  - "Write Once, Run Anywhere™"
- Security
  - Safe Network Delivery Of Content
  - Sandbox Approach

6

Why Java Technology?

- Openness
  - Marketplace Defines Platform, Not One Company
- Extensibility
  - No Static Application Framework
  - Backwards Compatibility For Legacy Content
  - Future Proof
- Code Reusability
  - Reuse Existing Software Components
  - Quick Time To Broadcast-Readiness

7

The Opportunity

- Historic Inflection Point
  - Analog to Digital
  - Narrowband to Broadband
    - DTV, DSL, Cable, DBS
  - Broadcast + Interactive
    - Reach + Personalization
  - The Web + Television
    - Computing + Consumer Electronics
  - The Intelligent Network

8

The Opportunity

- Beyond "The Vast Wasteland"
  - Beyond "500 channels"
  - Beyond "the idiot box"
- The Empowered Media Consumer
  - Information
  - Entertainment
  - Community
  - Commerce

9

Conclusion

- ITV Market Highly Fragmented
- Open Standards Are Critical To Market Success
- Java Technology Offers Key Benefits For Open Standards
- Historic Opportunity to Empower the Media Consumer

10

Contact Information




Sun and Digital Media

http://www.sun.com/media

Rob Glidden
Market Development Manager
Broadband & Digital Media
rob.glidden@sun.com

JAVA

11

Architectural Overview of the DASE Presentation Engine

Glenn Adams

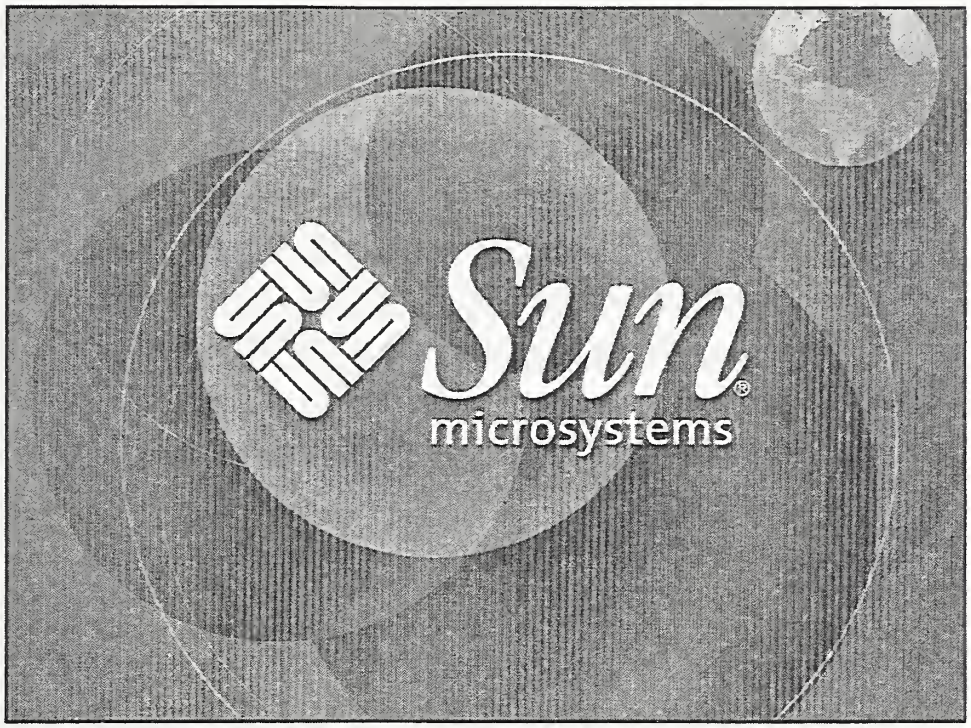
Gemstar International Group
<gadams@genstar.com >

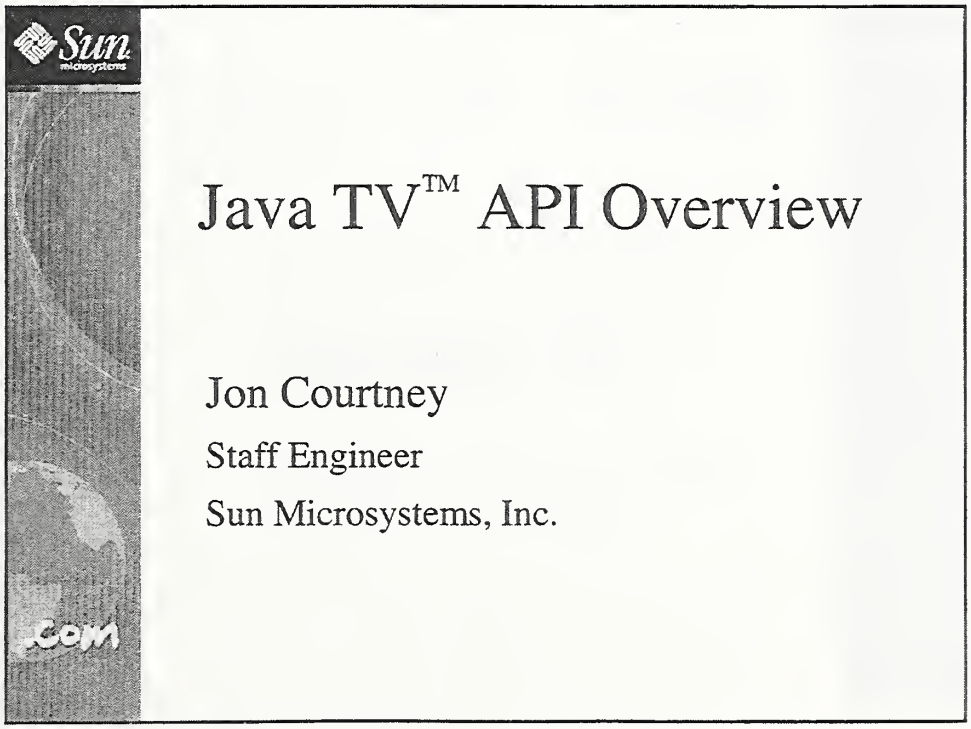
JavaTV API Overview

Jon Courtney

Sun Microsystems, Inc.
<courtney@Eng.Sun.COM >

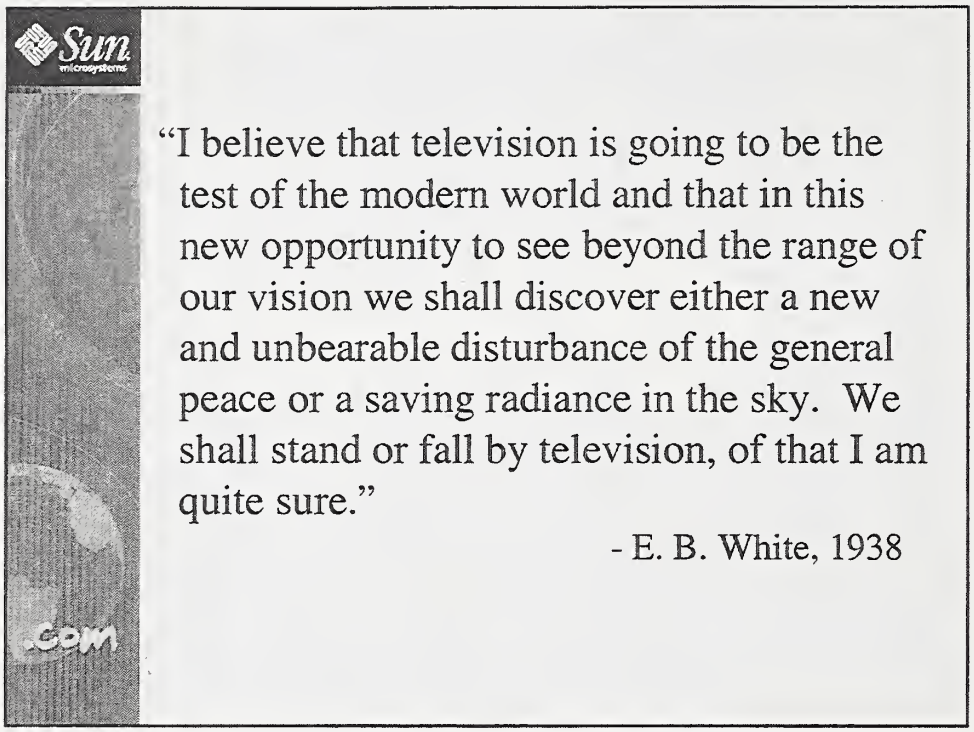
The Java TV API is a standard extension to the Java platform directed at developers who wish to produce interactive content in Java. The Java TV API gives Java programs control of advanced television receivers such as those based on the ATSC DASE standard. This presentation will provide an overview of the architecture of the Java TV APIs and describe their relation to the PersonalJava Application Environment. Five major functional elements of the API will be described: Java TV application life cycle, service information access, service selection, broadcast data access, and broadcast media control. This overview will provide receiver implementers and content creators an introduction to the scope, design and usage of the API.





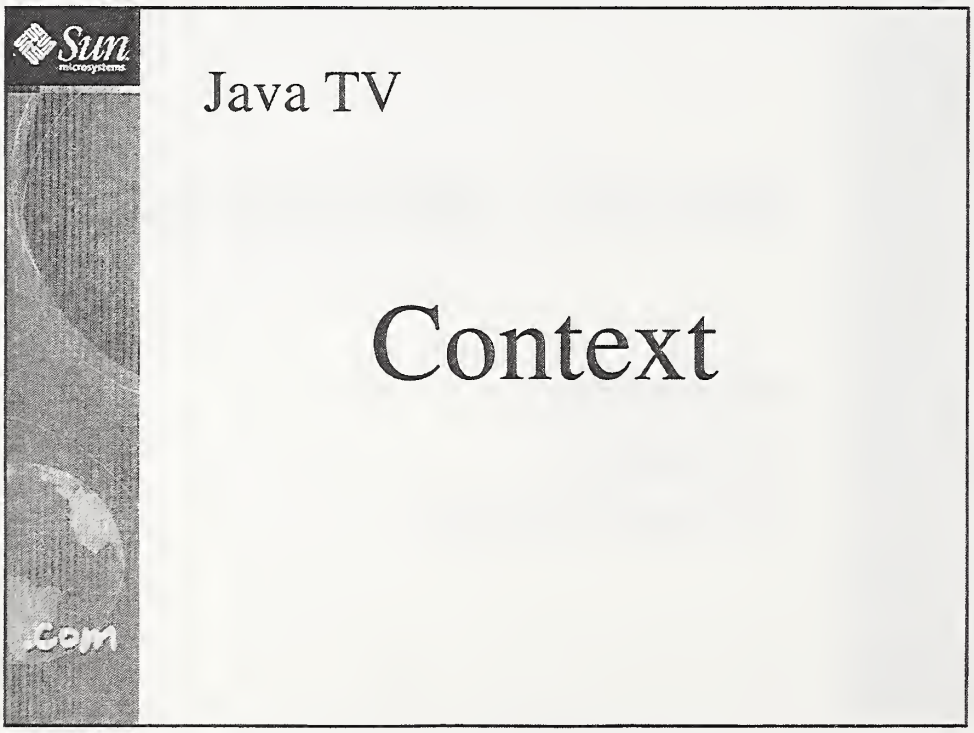
Java TV™ API Overview

Jon Courtney
Staff Engineer
Sun Microsystems, Inc.



"I believe that television is going to be the test of the modern world and that in this new opportunity to see beyond the range of our vision we shall discover either a new and unbearable disturbance of the general peace or a saving radiance in the sky. We shall stand or fall by television, of that I am quite sure."

- E. B. White, 1938



Java TV

Context

Java TV

What it is...

Network Independent
Application Environment
for
Broadcast Networks
and Television Receivers

Java TV API

What it is...

- Java standard extension
- Extends a Java platform
  - J2ME, PersonalJava, JDK
- Provides functionality for advanced television

Experts

Java Community Process

- Sony
- Nokia
- Open TV
- @Home
- OpenCable
- Samsung
- Toshiba
- Matsushita
- LG Electronics
- Philips
- General Instruments
- PowerTV

Standards

Java Community Process

- Advanced Television Systems Committee (ATSC)
- Digital Video Broadcast (DVB) Project

Scope

Receiver Support

- Enhanced Broadcast
  - Broadcast-based, local interaction
- Interactive Television
  - Return-channel, remote-network interaction
- Multi-network Environments
  - Broadband Internet networks
  - Home networks

Scope

Application Support

- Premium Video Service Control
  - PPV, IPPV, VOD, NVOD
- EPGs
  - General purpose, service specific, event specific

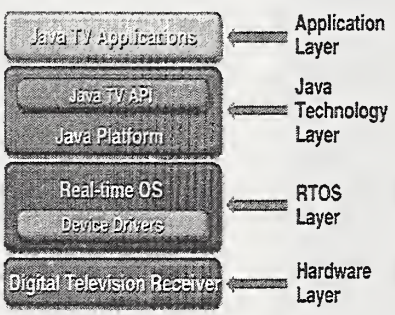
Scope

Application Support

- Television Enhancement & Interaction
  - Video synchronized, data driven, user interactive presentation, animation, simulatio and stream control
- General Applications
  - E-mail, browsing, e-commerce

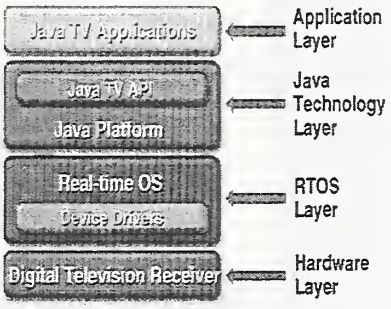
Environment

- Java Platform
  - Virtual Machine
  - Core APIs
  - UI APIs
  - TV extension APIs



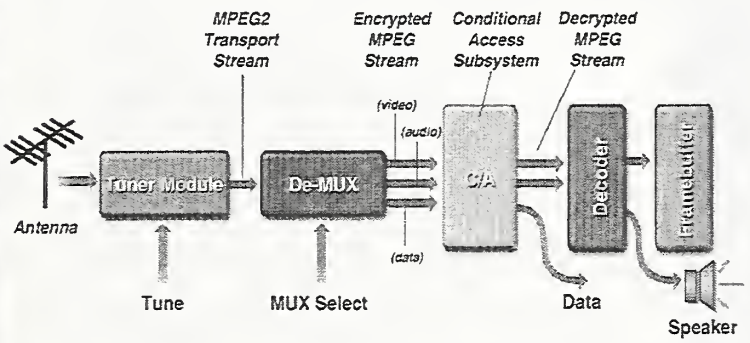
Environment

- Broadcast Platform
  - Operating System
  - Tuner Control
  - Demux Control
  - Conditional Access
    - Media Pipeline
    - Service Information Database

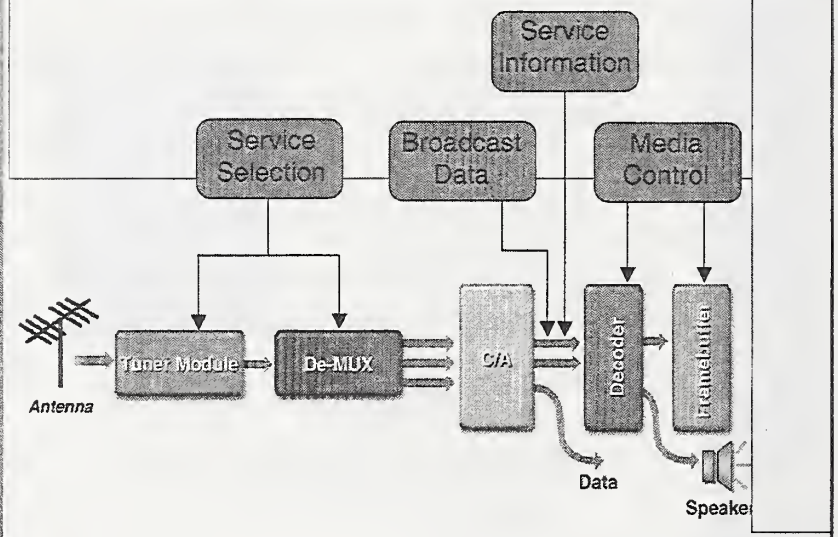


Broadcast Platform

Major Hardware Components



Java TV Components



Java TV

Content

Java Programming Paradigm

Benefits - Applied to Digital Television

- Extensible, reusable, complete
- Security architecture
- Platform independence
- Platform testing & conformance
- Powerful development tools
- Talented developer community

Java Content

Delivery to Digital Television Receivers

- Java application authored alongside A/V content
- Multiplexed with A/V content
- De-multiplexed, loaded and executed along with A/V content at receiver
- Presented to viewer with A/V content
- Influences presentation

Java Content

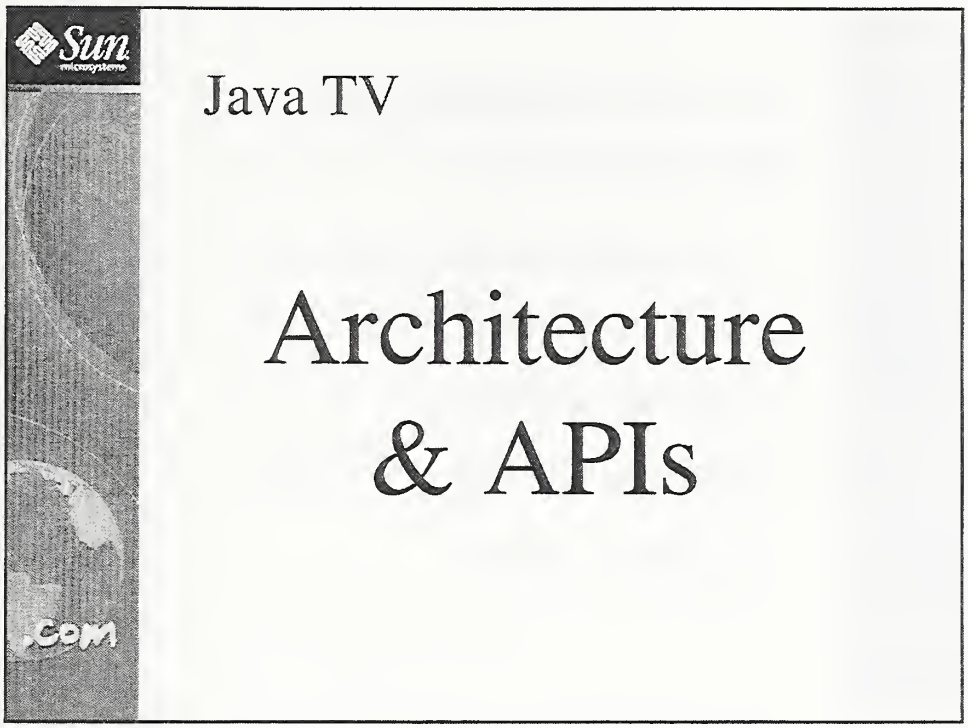
Java Byte Codes: A New Media Asset

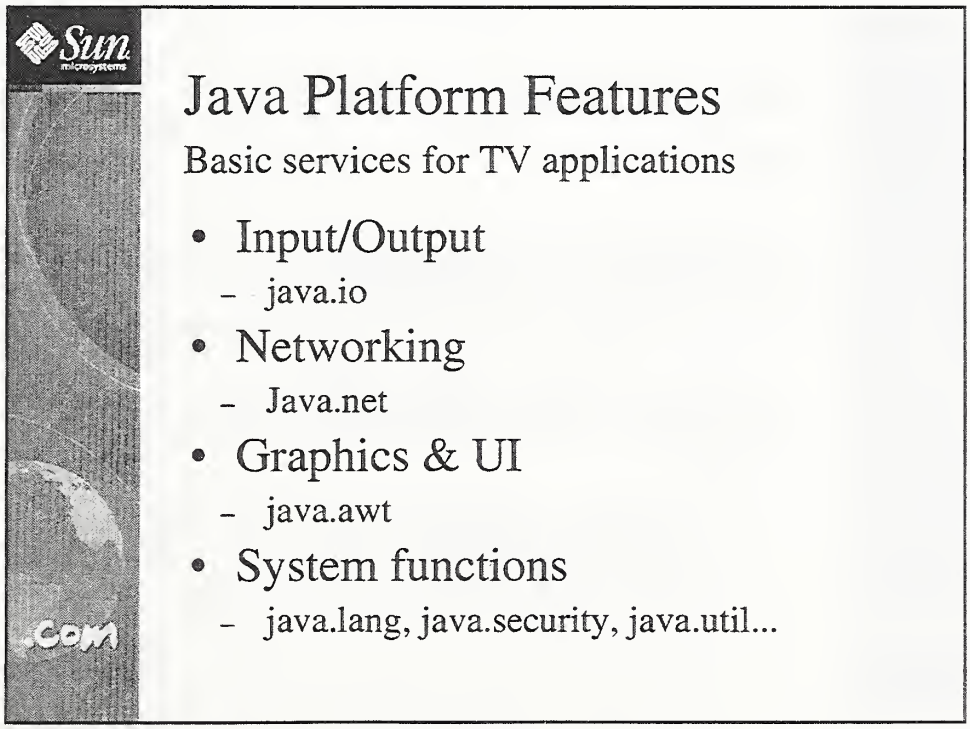
- Important as video, image, text, audio assets
- Business logic
- Simulations & games
- Smart Content

Smart Content

Adaptive Content

- Platform-customized presentation
- User interaction modes
- Viewer specific content
- Preference-based behavior


Java TV
Architecture
& APIs


Java Platform Features
Basic services for TV applications
• Input/Output
– java.io
• Networking
– Java.net
• Graphics & UI
– java.awt
• System functions
– java.lang, java.security, java.util...

Java TV Architecture

Major API Elements

- Application life cycle
- Service Information
- Service Selection
- Broadcast Data
- Media Control

Java TV Architecture

Locators

- Handles to information & resources
- Typically opaque to application
- Created from / externalized to string form
  - LocatorFactory.create(String) -> Locator
  - Locator.toExternalForm() -> String

Java TV Architecture

Security & Resource Management

- Policy is determined by network/platform
- Policy enforced by receiver
- Expressed using exceptions

Java TV Architecture

Application Life Cycle Model

Application Life Cycle

Features

- Ease of use for application implementers
- Separate from:
  - Window system management
  - Resource management
  - Management policy
- Minimal requirements on app managers

Application Life Cycle

Four application states:

- Loaded
  - Code is loaded, uninitialized
- Paused
  - App initialized, quiescent, minimal resource usage
- Active
  - App is executing normally
- Destroyed
  - App has released resources, terminated

Application Life Cycle

Xlet interface

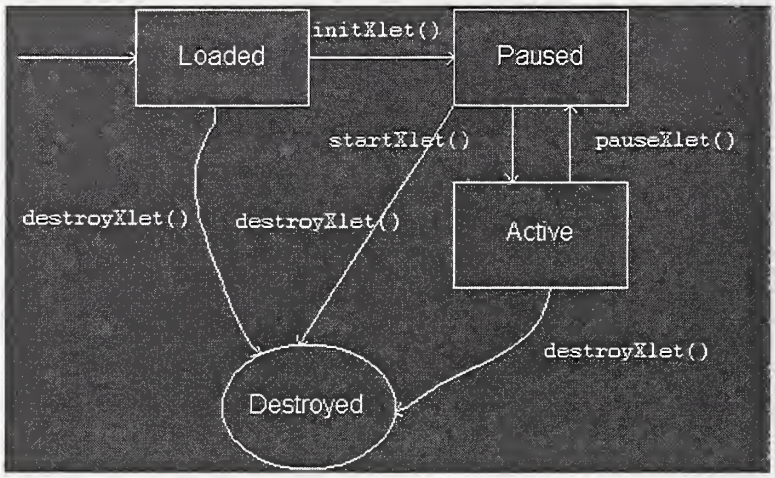
- Implemented by the application
- Methods to signal state transitions
- Xlets managed by Xlet Manager
- Similar to applet model w/o UI

Application Life Cycle

Xlet Interface

```
Package javax.tv.xlet;
public interface Xlet {
  void initXlet(...);
  void pauseXlet();
  void startXlet();
  void destroyXlet(...);
}
```

Application Life Cycle



Application Life Cycle

XletContext

- Provides property interface
- Used by Xlet to signal transitions to application manager
- **Xlet.initXlet(XletContext context);**

Application Life Cycle

XletContext

```
package javax.tv.xlet;

public interface XletContext {
  Object getXletProperty(String);
  void notifyPaused();
  void resumeRequest();
  void notifyDestroyed();
}
```

Java TV Architecture

Service Information API

Service Information

Features

- Protocol independent
- Storage and delivery independent
- Extensible for new SI types
- Cached and non-cached access
- Sync and async access
- Installed services discovery

Service Information

Three "views" of service information...

- Navigation package
  - Traversing through hierarchical SI data
- Guide package
  - EPG support
  - Program schedules, events, rating info
- Transport package
  - Exposes SI delivery mechanisms

Service Information

Asynchronous Retrieval

- Database cannot cache all SI data
- High latency in accessing data not in cache
- Inconvenient for programs to block while waiting for data

Service Information

Asynchronous Retrieval

- Asynchronous retrieval mechanism permits apps to queue requests and continue execution
- Asynchronous data access methods prefixed with 'retrieve'
  - **retrieveProgramEvent(...)**

Service Information

Asynchronous Retrieval

- Interface SIRetrievable extended by retrievable data types
  - **Date getUpdateTime();**
- Interface SIRequestor implemented by applications for retrieving data
  - **void notifySuccess(SIRetrievable[]);**
  - **void notifyFailure(...);**

Service Information

Asynchronous Retrieval

- SIRequest objects returned by asynchronous retrieval calls
  - **boolean cancel();**
- **Example:**
  - **SIRequest retrieveProgramEvent(Locator, SIRequestor);**

Service Information

SI Manager

- Access to SI Database
- Event generator describing SI updates
- Provides lists of available services
- SI filtering operations

Service Information

```
Package javax.tv.service.navigation;

public class SIManager {

  ServiceCollection
     createServiceCollection(ServiceFilter);

  Service getService(Locator);

  Transport[] getTransports();

  SIRequest retrieveSIElement(Locator, SIRequestor);

}
```

Service Information

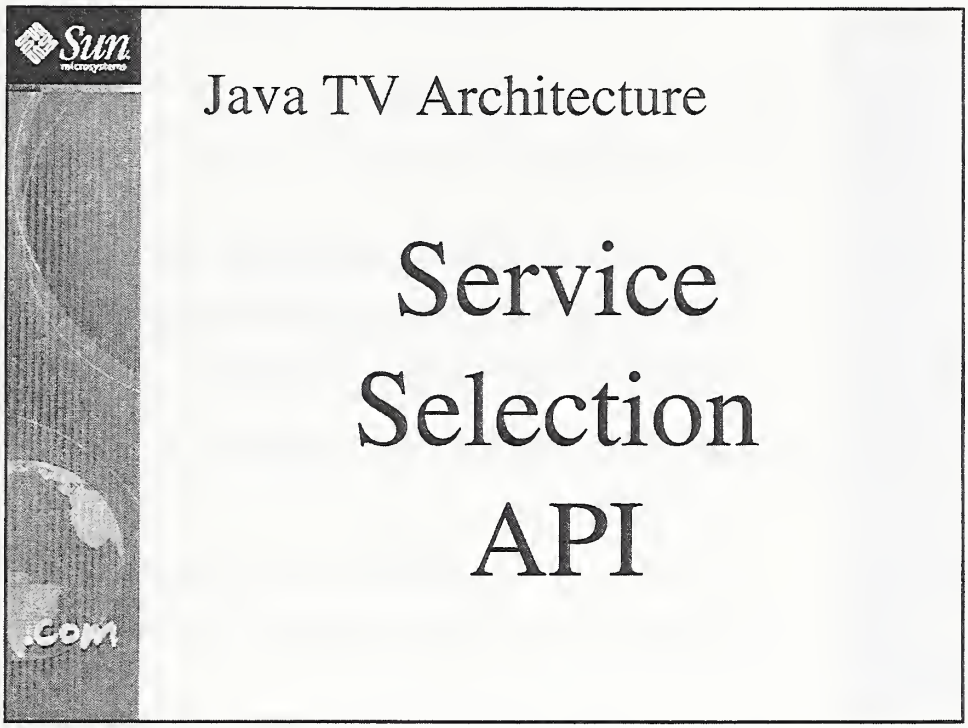
Services

- Represents a source of content, "channel"
- Selectable via service selection API
- Persistent data: name/number, locator
  - Cached, available synchronously
  - "Installed services" for bootstrap
- Asynchronous access to Service "details"

Service Information

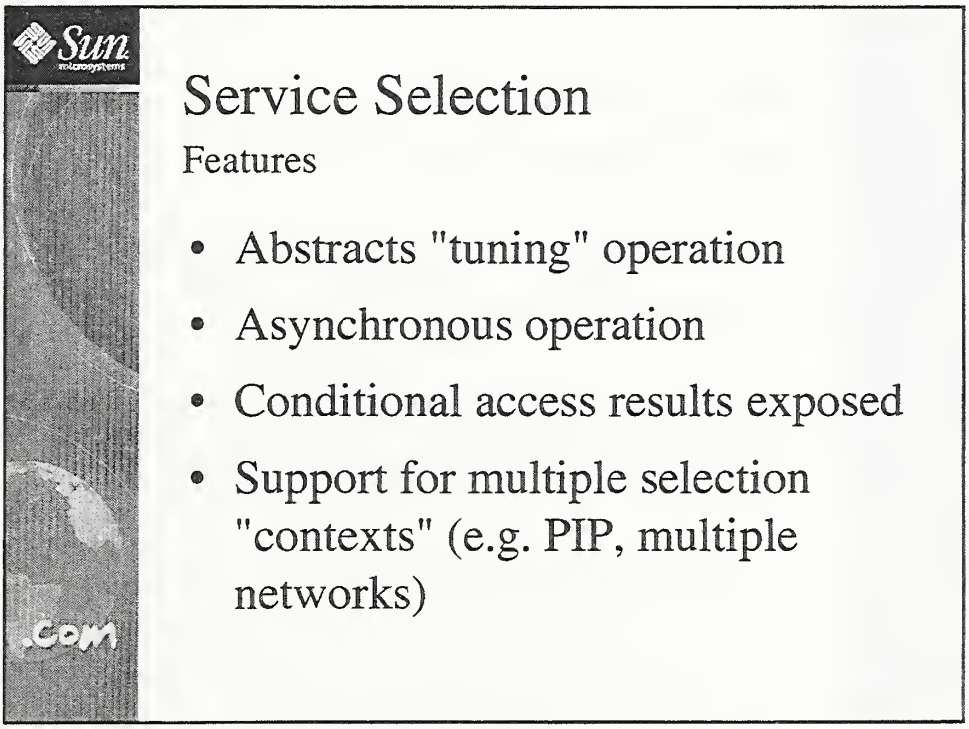
ServiceDetails

- Service meta-data
  - Represents specific instance of a service in the broadcast
  - Reports description, program schedule, etc.
  - Reports service components & types (e.g. audio, video, data)
- Extensible for new meta-data



Java TV Architecture

Service Selection API



Service Selection

Features

- Abstracts "tuning" operation
- Asynchronous operation
- Conditional access results exposed
- Support for multiple selection "contexts" (e.g. PIP, multiple networks)

Service Selection

ServiceContext

- Represents an environment for presenting media and downloaded applications from a Service.
- Provides selection via Service Locators
  - **ServiceContext.select(Service.getLocator());**
- Reports currently selected service

Service Selection

ServiceContext

- Management of multiple contexts
- Access to content "handlers"
- Signals current state via events for completion, redirection, failure

Service Selection

ServiceContext State Model

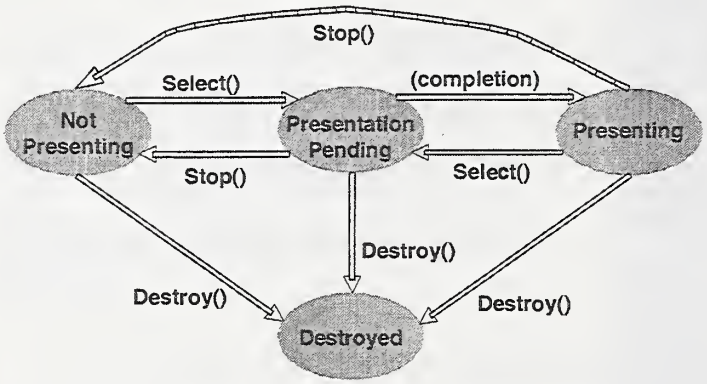
- Not Presenting
  - PresentationTerminatedEvent
- Presentation Pending
  - After select operation, but before completion

Service Selection

ServiceContext State Model

- Presenting
  - NormalContentEvent: Expected content is presented
  - AlternativeContentEvent: C/A redirection
- Destroyed
  - ServiceContextDestroyedEvent

Service Context States



Java TV Architecture

Broadcast Data APIs

Broadcast Data

Features

- File style access to broadcast filesystems
- Push style delivery for streams
- DatagramSocket access to broadcast IP

Broadcast Data

Package javax.tv.carousel

- Provides access to bounded data in hierarchical, cyclically transmitted broadcast filesystem
  - DSMCC object carousel
  - DSMCC data carousel
  - ATVEF UHTTP

Broadcast Data

Package javax.tv.carousel

- CarouselFile extends java.io.File
  - Represents broadcast files
  - Familiar mechanisms from java.io package
    - FileInputStream
    - RandomAccessFile
    - FileReader

Broadcast Data

CarouselFile

- Event notification of content changes
  - Interface CarouselFileListener
- Latency management
  - Instancing a CarouselFile notifies system to asynchronously cache file from broadcast
- Referenced via locators or filenames
  - Broadcast filesystem is mapped into local file name space

Broadcast Data

PushSourceStream

- Represents source of streaming data
- Acquired through JMF Manager
- Delivers data in a non-flow-controlled manner
  - Client is notified when data arrives
- Subinterface throws exceptions for data loss

Broadcast Data

Package javax.tv.net

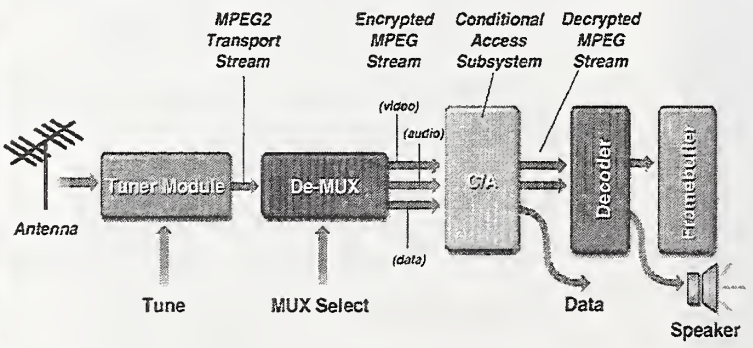
- javax.tv.net.InterfaceMap permits access to broadcast IP through conventional mechanisms
  - Dynamically maps locator to broadcast IP into private local IP address
  - Unicast and multicast supported
  - Access through familiar java.net mechanisms
    - DatagramSocket, MulticastSocket

Java TV Architecture

Media Control APIs

Media Control

- Java Media Framework manages pipeline
- JMF Player wraps decoder, rendering
- JMF DataSource wraps tuner & de-mux



Broadcast Pipeline

JMF Player and Data Source

- Representation of Network interface
- Representation of Rendering pipeline
- Separation allows reuse of pipeline
- Synchronization primitives
  - Media time exposed
- Downloaded s/w decoders enabled

Broadcast Pipeline

JMF Player and Data Source

- A/V control primitives
  - JMF Controls published
  - Runtime extendible
  - Media time control
- Resource management mechanisms
  - Events signal state transitions
- Small framework abstracts hardware

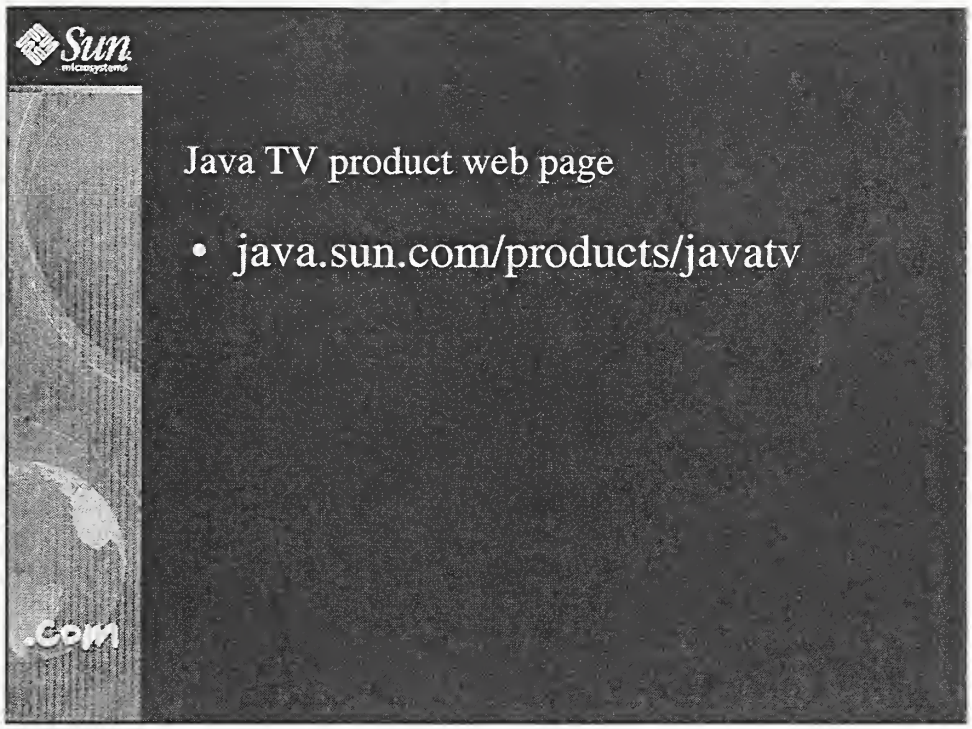
Java TV

Conclusion

Java TV Specification

Status

- Version 1.0 Release Candidate D
  - Released to standards bodies
  - On public web site
- Standards bodies
  - DVB, ATSC: Referenced by draft specifications
- Final release with reference implementation & conformance tests
  - 3Q2000


Sun
microsystems
Java TV product web page
• java.sun.com/products/javatv
.com


Jonathan Courtney
jonathan.courtney@sun.com
We're the
dot in .com™
Sun
microsystems

DASE APIs, Their Use & Relationship to Other Java APIs

Petr Peterka

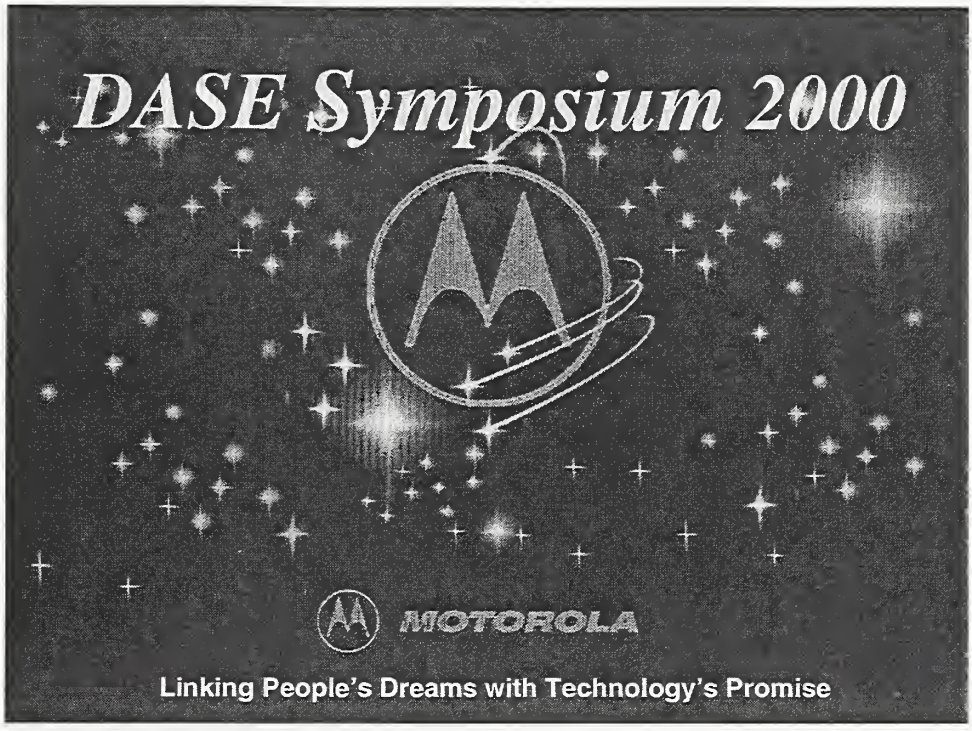
DASE Vice-Chairman and API Architect
Broadband Communication Sector, Motorola
<PPeterka@gi.com>

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 and graphics, user choices, personalization and localization, teleshopping, targeted advertisements, etc. Standalone applications such as electronic program guides are becoming a norm. Most current deployments of such systems are based on proprietary solutions.

The Internet and the Web in particular was enabled by platform independent content formats such as HTML, JavaScript 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 presentation will address 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. The DTV receiver system services that are being abstracted by the Java APIs include Network Communication, Content Management, Presentation and User Interface, Application and Resource Management, Security Management, Environment Management and Utility Services.

Since there are similar efforts in different realms of the industry, DASE decided to reuse existing APIs where appropriate. As a result, the DASE draft specification includes the following APIs: Sun's JavaTV 1.0 and JMF 1.1 APIs, HAVi 1.0 User Interface API, a subset of DAVIC 1.4 APIs and an ATSC-specific set of APIs. All of these APIs are defined on top of the Java Virtual Machine and a subset of Personal Java 1.2. Personal Java provides the basic Java packages which abstract an operating system; JavaTV provides the core DTV receiver functionality including tuning, access to system and service information, data carousels, extensions to JMF, etc.; HAVi addresses the needs of an embedded device with respect to a user interface. DASE adds APIs for ATSC-specific features including PSIP and ATSC T3/S13 data broadcast protocol. Other extensions include support for application management, user management and user preferences. Downloadable applications are represented by an Xlet, a broadcast version of an Applet, which are delivered as data in the transport stream together with audio, video and supporting data.



DASE APIs, Their Use & Relationship to Other Java APIs

Petr Peterka

DASE Vice-Chairman and DASE API Architect

23 May, 2000

MOTOROLA
Broadband Communications Sector

Purpose of DASE APIs

- Provide access to DTV receiver functions relevant to downloadable applications
- Enable Application portability
  - Applications run on all DASE receivers independent of CE manufacturer
- Provide an abstraction
  - Hide implementation details
  - Hide the choice of Operating System
  - Hide the choice of hardware

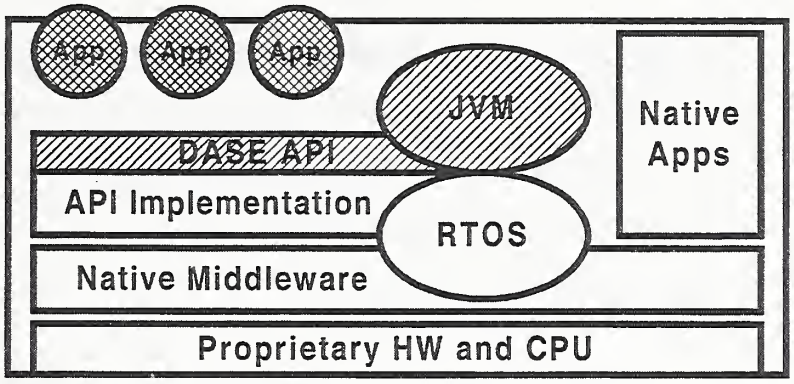
MOTOROLA
Broadband Communications Sector

API Goals

- High-level of abstraction
  - allow implementation freedom
  - allow choices of implementation language
  - allow range of operating systems
  - allow wide variety of HW and CPU
- Consistency
  - event and error handling
- Minimal and complete

MOTOROLA
Broadband Communications Sector

DASE API Reference Architecture



MOTOROLA
Broadband Communications Sector

DASE System Services

1. Network Communication
2. Content Management
3. Presentation and User Interface
4. Application and Resource Management
5. Security Management
6. Environment Management
7. Utility APIs

MOTOROLA
Broadband Communications Sector

Network Communication

- Navigation Service (MPEG PSI, Cable SI, PSIP, etc.)
- Event Information Service (EIT)
- Program Selection Service
- Data Broadcast Service

MOTOROLA
Broadband Communications Sector

Content Management

- Audio Video Decoder Service
- Media Decoder and Playback Control Service
- Audio Control Service
- Video Presentation Service
- Presentation Synchronization Service
- Decoder/Player Synchronization Service

MOTOROLA
Broadband Communications Sector

Presentation and User Interface

- Graphics Presentation Service
- Font Management Service
- Color Management Service
- User Input Service

MOTOROLA
Broadband Communications Sector

Application and Resource Management

- Application Lifecycle Service
- Application Registration Service
- Application Version Management Service
- Application Verification Service
- Application State Service (diagnostics)

MOTOROLA
Broadband Communications Sector

Security Management

- Authentication Service
- Conditional Access Service
- Security Service (policy)
- Cryptography Service

MOTOROLA
Broadband Communications Sector

Environment Management

- Receiver Profile Service
- Version Information Service
- System and User Preferences Service
- User Management Service
- Content Control Service

MOTOROLA
Broadband Communications Sector

Utility Services

- Event Dispatching Service
- Interprocess Communication Service
- Exception Handling Service
- Scheduling Service
- Math Service
- Time Service
- Text Service
- Localization Service

MOTOROLA
Broadband Communications Sector

Current DASE API Components

1. ATSC specific APIs
2. Sun's Java TV API 1.0
3. HAVi User Interface API 1.0
4. Personal Java 1.2 (with exceptions)
5. JMF 1.1
6. DAVIC 1.4 (selected packages)

MOTOROLA
Broadband Communications Sector

Java TV APIs

- Abstract SI - access to MPEG PSI, Cable SI, ATSC PSIP, DVB SI)
- Service Selection - changing channels
- Data Broadcast - access to broadcast data
- Locator - similar to URL
- Xlet - Applet-like downloadable application
- JMF extensions for A/V and CC control
- AWT/Graphics extensions (alpha blending)
- TVTimer - scheduling

MOTOROLA
Broadband Communications Sector

ATSC-specific APIs

- SI extensions for PSIP
- Data Broadcast extensions for S13
- Application package for
  - Xlet extensions
  - Application Manager
  - Application Management
- User Management

MOTOROLA
Broadband Communications Sector

ATSC-specific APIs (cont.)

- Common and User Preferences
- System Properties
- Extensions for Networking
- Extensions for Carousel File Access
- Extensions for Graphics (work in progress)
- Security

MOTOROLA
Broadband Communications Sector

HAVi User Interface APIs

- HAVi UI made independent of other HAVi packages
- Added simple widgets and widget toolkit
- Support for Transparency
- Support for Video
- Display device independence
- Support for remote control events

MOTOROLA
Broadband Communications Sector

JavaTV Package Hierarchy

- javax.tv.service
- javax.tv.service.navigation
- javax.tv.service.guide
- javax.tv.service.transport
- javax.tv.service.selection
- javax.tv.locator
- javax.tv.carousel
- javax.tv.net

MOTOROLA
Broadband Communications Sector

JavaTV Package Hierarchy (cont.)

- javax.tv.xlet
- javax.tv.graphics
- javax.tv.media
- javax.tv.media.protocol
- javax.tv.util

MOTOROLA
Broadband Communications Sector

DASE Package Hierarchy

- org.atsc.si
- org.atsc.si.descriptor
- org.atsc.data
- org.atsc.application
- org.atsc.preferences
- org.atsc.user
- org.atsc.net

MOTOROLA
Broadband Communications Sector

DASE Package Hierarchy (cont.)

- org.atsc.system
- org.atsc.security
- org.atsc.management
- org.atsc.registry

MOTOROLA
Broadband Communications Sector

HAVi UI Package Hierarchy

- havi.ui
- havi.ui.event

MOTOROLA
Broadband Communications Sector

DAVIC API Hierarchy

- org.davic.awt
- org.davic.media
- org.davic.resources

MOTOROLA
Broadband Communications Sector

Personal Java

- Personal Java 1.2
  - based on JDK 1.1.8
  - includes JDK1.2 Security Architecture
- Except
  - see next slide

MOTOROLA
Broadband Communications Sector

Personal Java in DASE

- **java.applet** - Not Required except for profiles supporting Internet
- **java.awt** - Only a subset required as defined in JavaTV and HAVi specifications
- **java.math** - Required
- **java.rmi** - Not required for Profile 1
- **java.sql** - Not required
- **com.sun.util (Ptimer)** - Not required (is in javax.tv.util.TVTimer)

MOTOROLA
Broadband Communications Sector

DASE API Specification: Work in Progress

- Set of requirements
- API description
- API object model
- API behavioral model
- API syntax and semantics
- API JavaDoc online documentation

MOTOROLA
Broadband Communications Sector

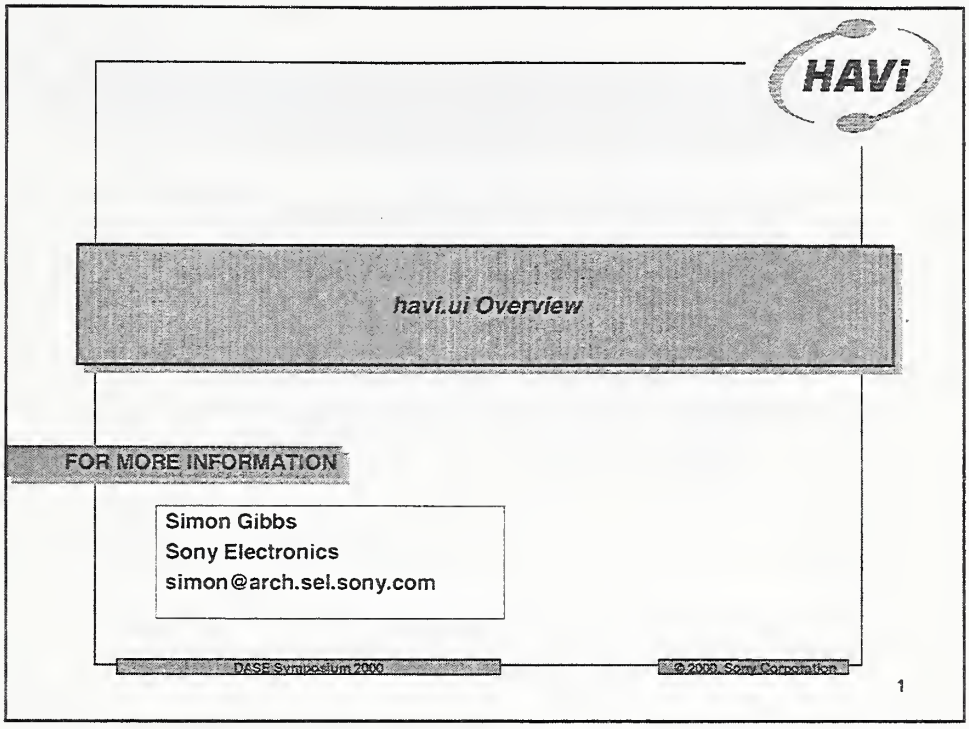
havi.ui Overview

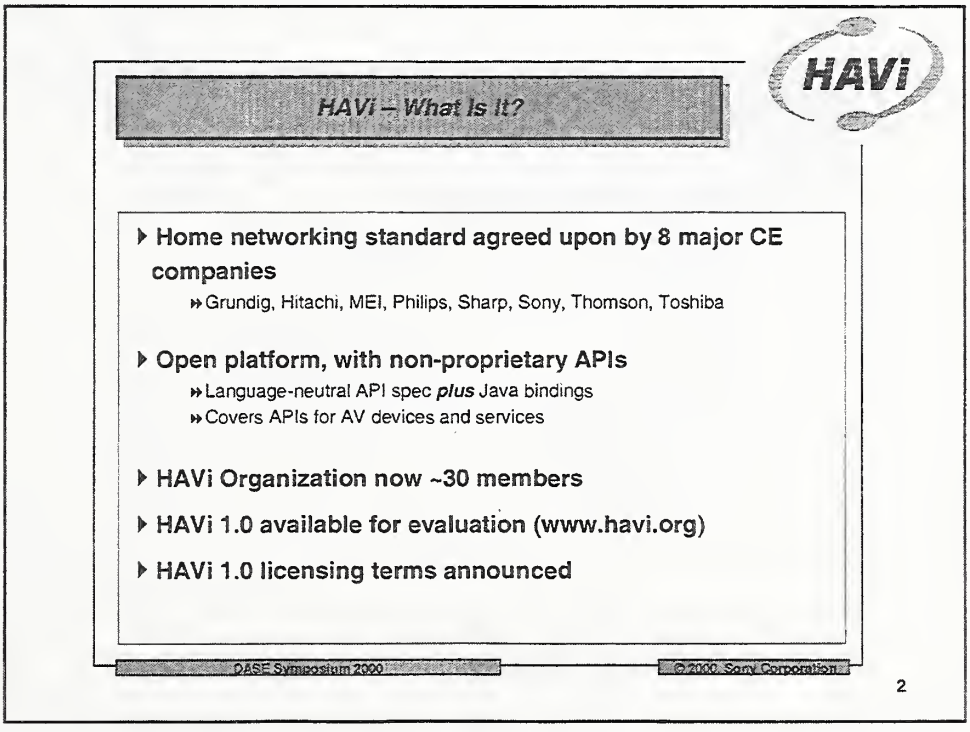
Simon Gibbs

Sony Electronics Inc.
<simon@arch.sel.sony.com>

DASE applications (xlets) typically present some form of user interface and so require Java classes for drawing graphics and reacting to user input. For this purpose, DASE provides parts of java.awt and two additional packages: org.havi.ui and org.havi.ui.event (collectively called "havi.ui"). havi.ui is based on the "lightweight component" subset of java.awt, and adds several extensions explicitly designed to be suitable for use and implementation on television receivers and other consumer electronics devices. These extensions include:

- remote control support and the ability for an application to determine the user-input capabilities of the platform on which it is running.
- ability to determine the resolution and physical characteristics of the current display device and detect modifications to the resolution and physical characteristics of the current display device (e.g., a 4:3 display switching between clipped and letter-boxed renditions of 16:9 content).
- support for graphics/video integration, e.g., "registering" graphics to background video.
- support for visual effects (blending, transitions) using mattes and transparency operations.
- a framework allowing applications to construct their own widget sets and so define their own "look and feel".


HAVi
havi.ui Overview
FOR MORE INFORMATION
Simon Gibbs
Sony Electronics
simon@arch.sel.sony.com
DASE Symposium 2000
© 2000, Sony Corporation
1


HAVi
HAVi – What Is It?
Home networking standard agreed upon by 8 major CE companies
Grundig, Hitachi, MEI, Philips, Sharp, Sony, Thomson, Toshiba
Open platform, with non-proprietary APIs
Language-neutral API spec plus Java bindings
Covers APIs for AV devices and services
HAVi Organization now ~30 members
HAVi 1.0 available for evaluation (www.havi.org)
HAVi 1.0 licensing terms announced
DASE Symposium 2000
© 2000, Sony Corporation
2

Home Networking – What Is It?

- **A general architecture for interconnecting and controlling home devices, *ideally*:**
  - Any device can discover and communicate with any other device
  - Any device can control and 'present' any other device
  - Plug-and-play and future-proof capabilities
  - Open software APIs
  - Control ***and*** content
- **Examples:**
  - HAVi, HomePnP, CEbus,
  - HAPI, UPnP, ...

HAVi and IEEE 1394

HAVi builds on IEEE 1394 technology

- **Designed for AV, plug-and-play**
- **Supported by CE and IT industries**
- **Selected by DVB, DAVIC, VESA, EIA as standard for home networking**
- **Becoming low cost**
- **Extensions being standardized for:**
  - Longer distance
  - Higher speeds
  - Bridges
  - Wireless

IEEE 1394 – Present

HAVi

1394-1995

- 100, 200, 400 Mbps
- 64 "isochronous" channels (guaranteed bandwidth, bounded jitter)
- True plug-and-play (network is self-configuring)
- Protocols for transport of MPEG2, DV etc.
- Available products: DV camcorders, PC cards, printers, disks...
- OSD, high-def pass-through from STB to DTV receiver (EIA-775)

IEEE 1394 – Future

HAVi

P1394a

- Improved reset
- Power management
- Asynchronous streams

P1394b

- 800 Mbps, 1.6, 3.2 Gbps
- Bridging, plastic fiber
- Backwards compatible with 1394a

1394 over IR and RF

Digital copy-protection

HAVi System Components

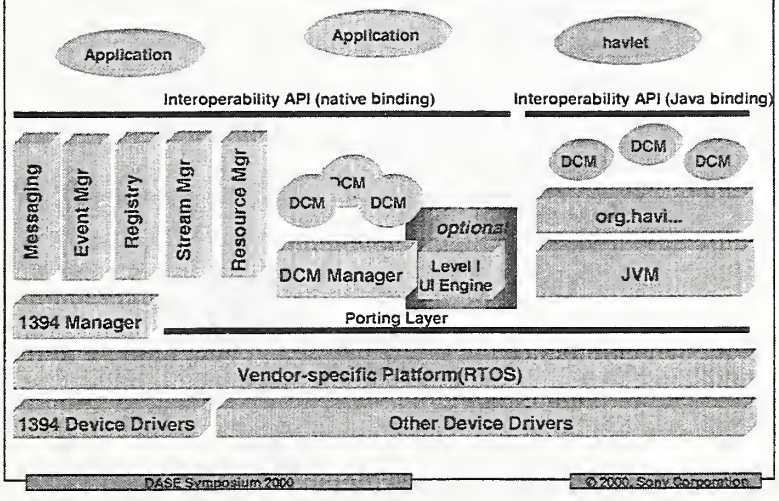
- 1394 Manager
- Messaging System
- Event Manager
- Registry
- DCMs
- DCM Manager
- Stream Manager
- Resource Manager
- Level I UI Engine
- Java runtime – optional

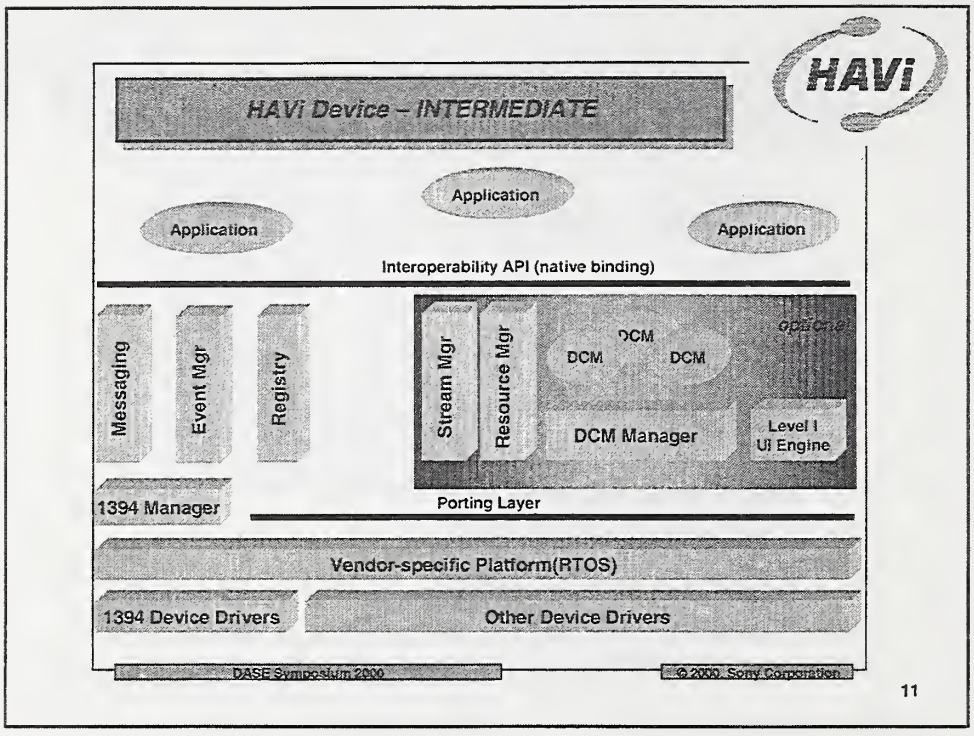
	FULL	INTERMEDIATE	BASE	LEGACY
Java Runtime	✔			
Level I UI Engine	[✔]	[✔]		
Resource Manager	✔	[✔]		
Stream Manager	✔	[✔]		
DCM Manager	✔	[✔]		
Registry	✔	✔		
Event Manager	✔	✔		
Messaging System	✔	✔		
1394 Manager	✔	✔		
HAVi device descriptor	✔	✔	✔	
DCM	✔	[✔]	✔	✔

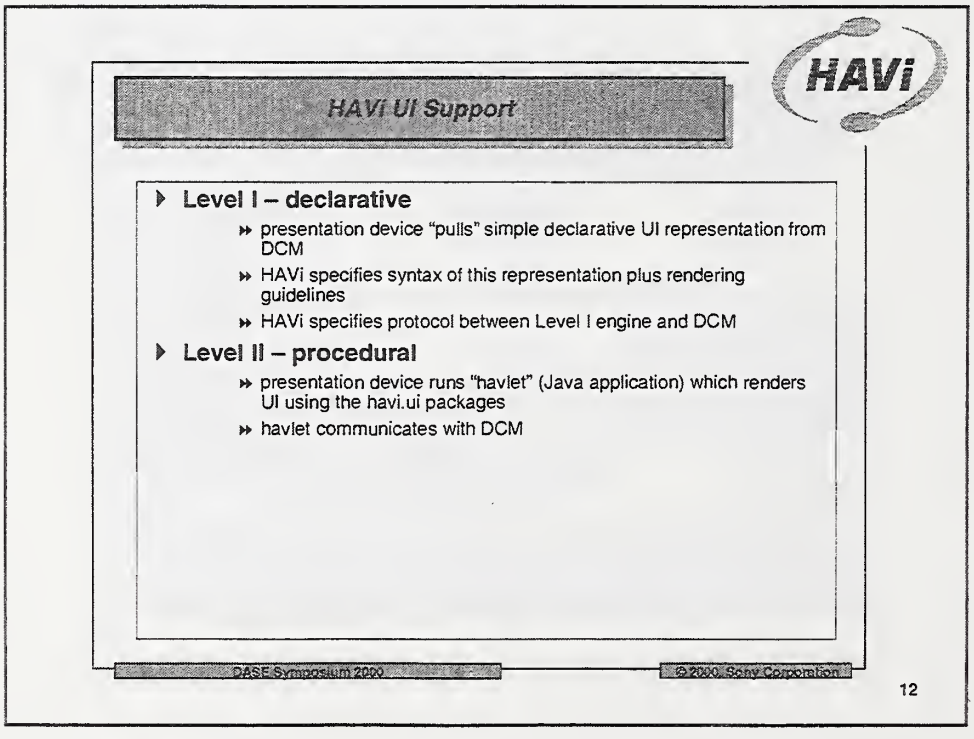
HAVi Device Classification

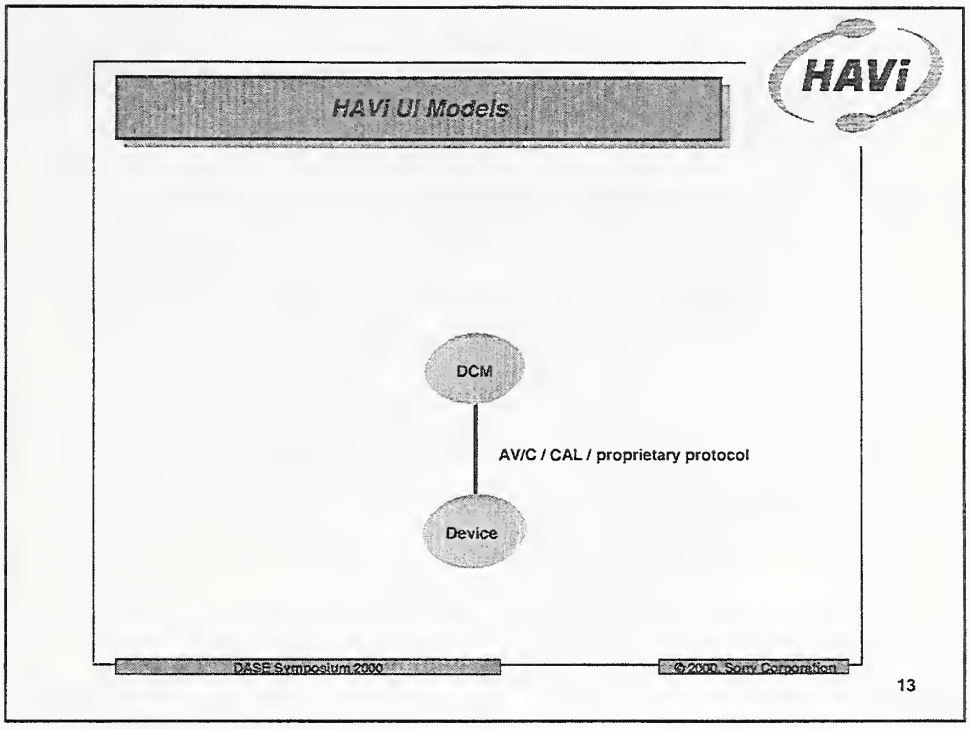
- LEGACY
  - » existing devices, existing control protocols
- BASE
  - » low cost/resource: camcorder, MD, CD
- INTERMEDIATE
  - » intermediate cost/resource: DVTR, DVD
- FULL
  - » higher cost/resource: DTV, STB, PVR, PC

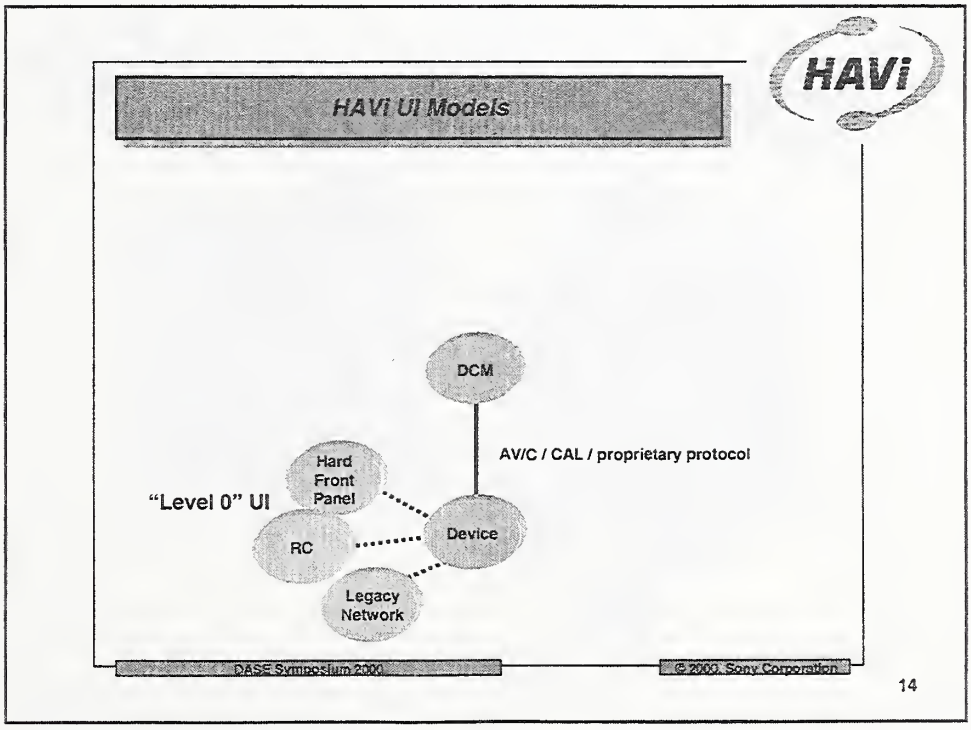
HAVi Device – FULL

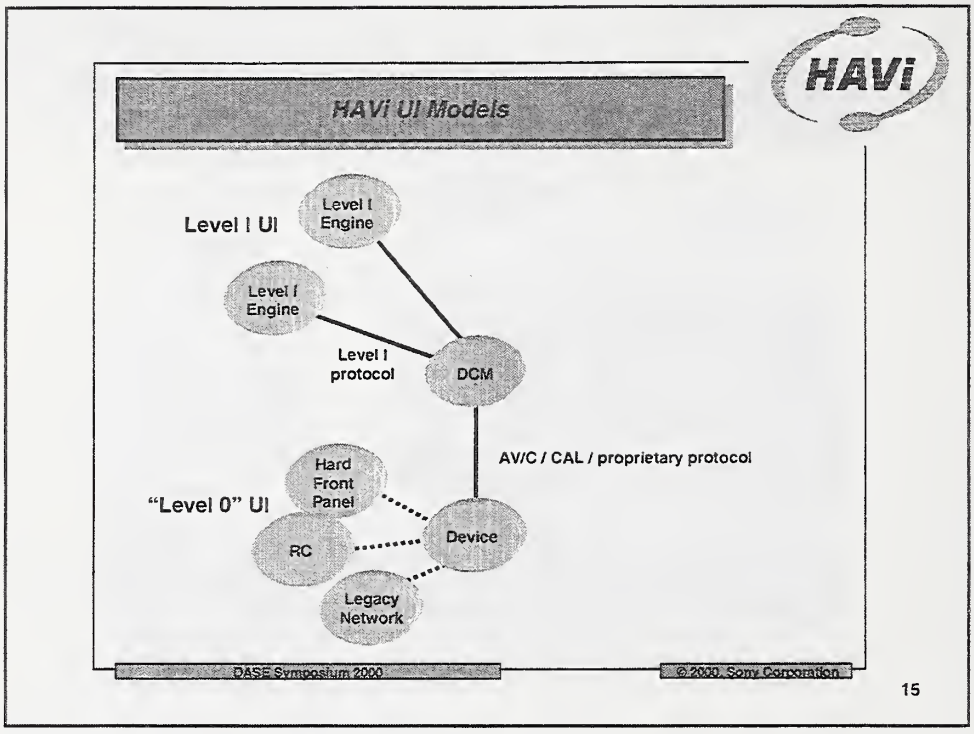


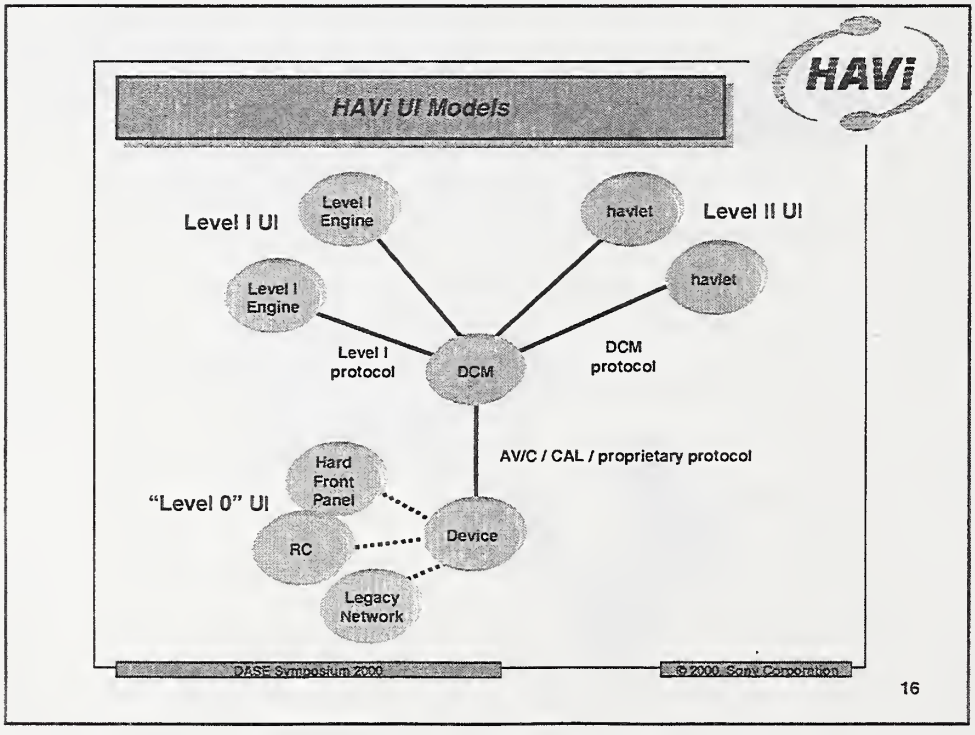

HAVi Device – INTERMEDIATE
Application
Application
Application
Interoperability API (native binding)
Messaging
Event Mgr
Registry
Stream Mgr
Resource Mgr
DCM
DCM
DCM
DCM Manager
Level I
UI Engine
1394 Manager
Porting Layer
Vendor-specific Platform(RTOS)
1394 Device Drivers
Other Device Drivers
DASE Symposium 2000
© 2000, Sony Corporation
11


HAVi UI Support
Level I – declarative
presentation device "pulls" simple declarative UI representation from DCM
HAVi specifies syntax of this representation plus rendering guidelines
HAVi specifies protocol between Level I engine and DCM
Level II – procedural
presentation device runs "havlet" (Java application) which renders UI using the havi.ui packages
havlet communicates with DCM
DASE Symposium 2000
© 2000, Sony Corporation
12


HAVi UI Models
HAVi
DCM
AV/C / CAL / proprietary protocol
Device
DASE Symposium 2000
© 2000, Sony Corporation
13


HAVi UI Models
HAVi
DCM
AV/C / CAL / proprietary protocol
"Level 0" UI
Hard Front Panel
RC
Device
Legacy Network
DASE Symposium 2000
© 2000, Sony Corporation
14

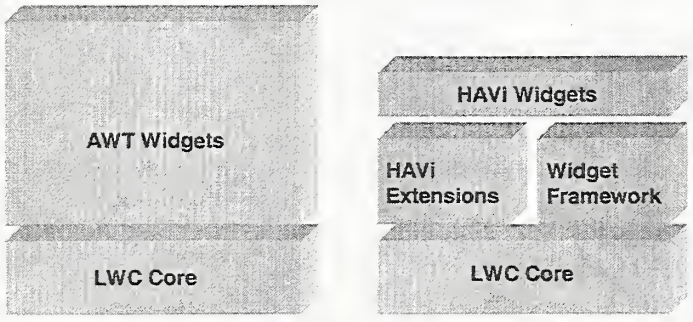

HAVi
HAVi UI Models
Level I UI
Level I Engine
Level I Engine
Level I protocol
DCM
AV/C / CAL / proprietary protocol
"Level 0" UI
Hard Front Panel
RC
Device
Legacy Network
DASE Symposium 2000
© 2000, Sony Corporation
15


HAVi
HAVi UI Models
Level I UI
Level I Engine
Level I Engine
havlet
havlet
Level II UI
Level I protocol
DCM
DCM protocol
AV/C / CAL / proprietary protocol
"Level 0" UI
Hard Front Panel
RC
Device
Legacy Network
DASE Symposium 2000
© 2000, Sony Corporation
16

havi.ui APIs

- uses subset of AWT needed for lightweight components
- no native "peers" necessary
- simple widget set
- "TV friendly" UI extensions:
  - display formats
  - video/graphics registration
  - transparency and effects
  - widget framework
  - up/down/left/right navigation
  - remote control support

havi.ui and AWT



Display Formats

Broadcast

4:3

16:9

DASE Symposium 2000

19

Display Formats

Broadcast

Display Surface

4:3

4:3

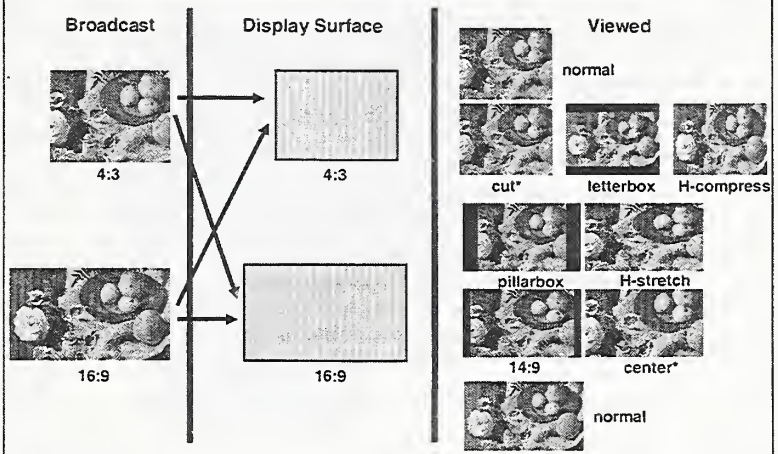
16:9

16:9

DASE Symposium 2000

20

Display Formats



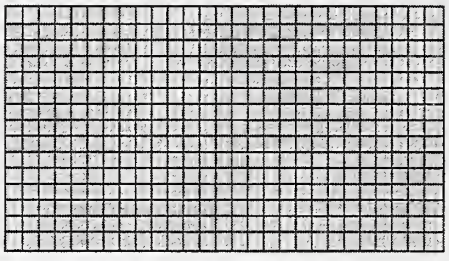
org.havi.ui.HScreenDevice

- configuration and query of display devices
- notification of changes in configuration
- subclasses:
  - HVideoDevice
  - HGraphicsDevice, HEmulatedGraphicsDevice
  - HBackgroundDevice
- associated configuration and template classes, eg
  - HVideoConfiguration
  - HVideoConfigTemplate
- configuration parameters to be determined by havi.ui implementation profile:
  - HAVi: FAV configuration parameters
  - DVB: MHP configuration parameters
  - ATSC: DASE configuration parameters

Video/Graphics Registration

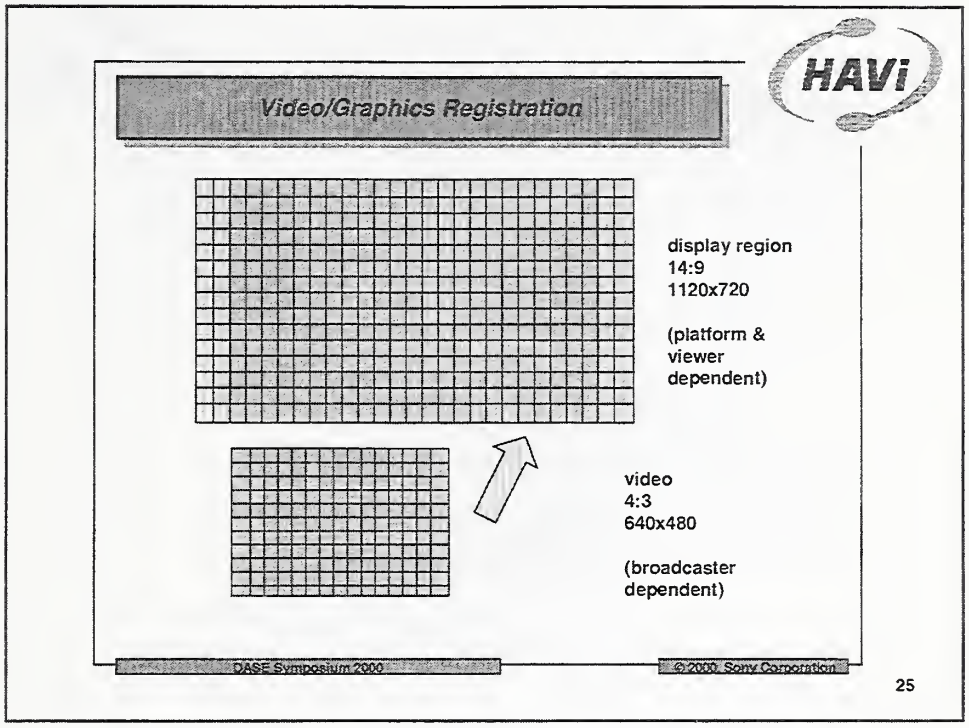
- Application authors want:
  - device independent coordinate system
  - square pixels
  - mapping of authoring coordinates to display coordinates
- Example:
  - original video is 4:3 640 x 480
  - video is displayed in 14:9 region on a 16:9 monitor
  - full screen video display resolution is 1280 x 720
  - full screen graphics (OSD) resolution is 1280 x 720
  - authoring coordinates are 800 x 600, square pixel
  - application author wants this coordinate system to be mapped to wherever the video appears

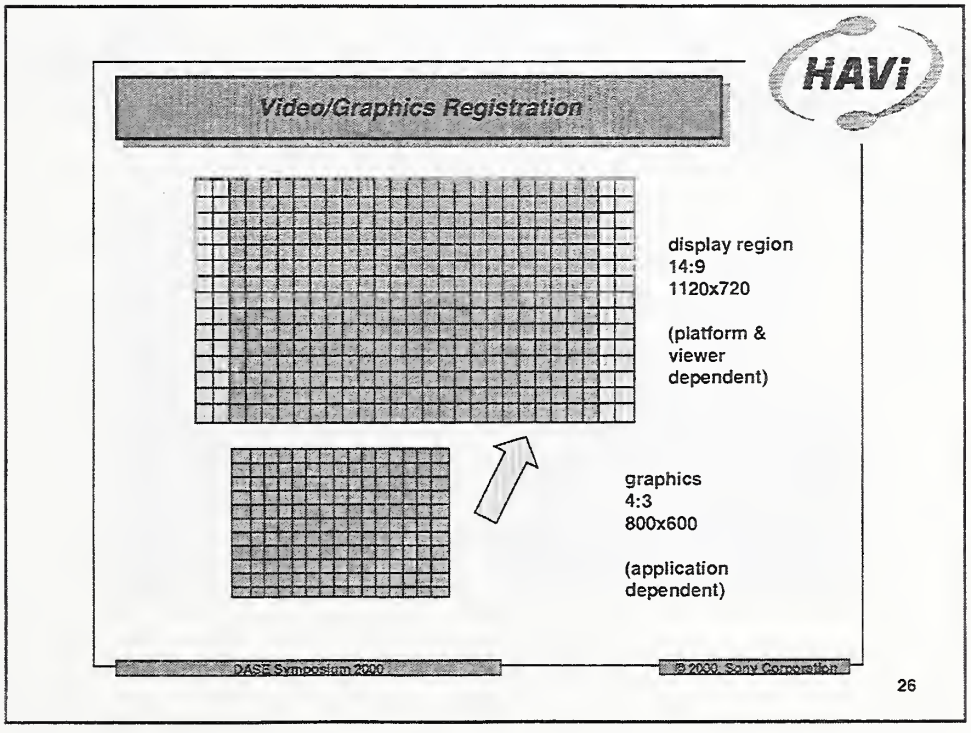
Video/Graphics Registration



screen
16:9
1280x720

(platform
dependent)


HAVi
Video/Graphics Registration
display region
14:9
1120x720
(platform &
viewer
dependent)
video
4:3
640x480
(broadcaster
dependent)
DASE Symposium 2000
© 2000 Sony Corporation
25


HAVi
Video/Graphics Registration
display region
14:9
1120x720
(platform &
viewer
dependent)
graphics
4:3
800x600
(application
dependent)
DASE Symposium 2000
© 2000 Sony Corporation
26

Video/Graphics Registration

- Solution:

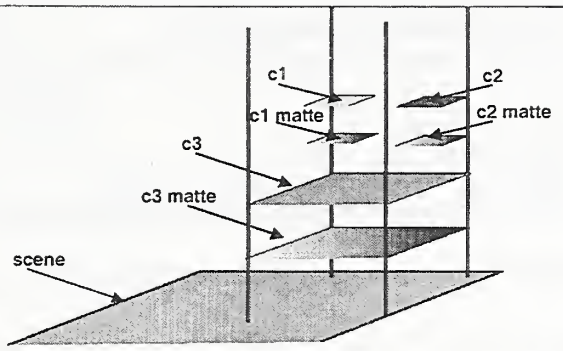
```
configTemplate = new HGraphicsConfigTempate();

configTemplate.setPreference(VIDEO_GRAPHICS_REGISTERED,
       myVideoConfig, REQUIRED);

configTemplate.setPreference(PIXEL_RESOLUION,
       new Dimension(800, 600), REQUIRED);

graphicsConfig = configTemplate.getBestConfiguration(...);

graphicsDevice = graphicsConfig.getDevice();

// create HAVi scene using graphicsDevice
```


Transparency and Effects

- Goals:
  - help UI designers present a "TV look"
  - allow rich image-based UI
  - use image composition model of layering and keys/mattes
  - support fades, transitions and special effects at the component level
  - orthogonal to AWT drawing methods

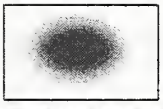
Transparency and Effects

- 2½D layering model
- HMatteLayer interface for HComponent and HContainer



org.havi.ui.HMatteLayer

c1.setMatte(m1)

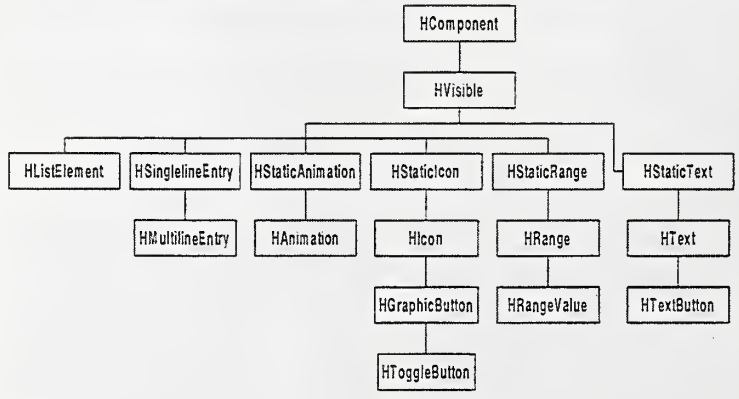


c1.setMatte(m2)

Widget Framework

- Simplifies construction of custom widgets
- Abstraction of Feel:
  - a core of abstract component behaviors
  - HNavigable, HActionable, HSwitchable, HValue
- Separation of Look:
  - supports "pluggable" looks (consistent changes in appearance of widgets)
  - several predefined looks: HGraphicLook, HTextLook, ...
- Application may define new looks, feels and widgets

Resident Widgets



org.havi.ui.event

- Remote control support
  - dedicated keys (CHANNEL_UP, ...)
  - colored keys
- Event Representation
  - string
  - symbol
  - color
- Keyboard/Mouse/RC capability queries

havi.ui Status

- several implementations underway
- havi.ui 1.0 in DVB/MHP specification
- havi.ui 1.0 in ATSC/DASE draft specification

NIST API Reference Implementation

Robert Snelick

National Institute of Standards and Technology
Information Technology Laboratory
<rsnelick@nist.gov>

The National Institute of Standards and Technology (NIST) is developing a reference platform for the Digital TV Application Software Environment (DASE) standard. The NIST/DASE reference platform provides a development environment of the DASE standard for designers, implementers, and content providers. The environment includes a set-top box (STB) simulation, a DASE Application Programming Interface (API) implementation, unit test modules, and sample DASE applications. The goal of the NIST reference platform is to demonstrate proof of concept of the DASE standard, provide the impetus for conformance testing, aid the design and development of other DASE implementations, and provide an environment for developing and testing DASE content/applications. In alignment with these goals, the design of the reference platform emphasizes implementation clarity and portability over performance and system constraints. To achieve these goals, the majority of the system is written in Java.

The NIST API reference implementation is currently built on top of the NIST STB simulation. The simulation is a collection of Java classes that encapsulate the functions of an ATSC STB environment. A central task of the Java simulation classes is to provide the implementation with ATSC data structures and associated data managers. A key aspect of the API reference implementation design is an intermediate software layer, called the Hardware Abstract layer (HAL), that facilitates portability. The HAL provides an interface to the STB environment that hides the details of the underlying architecture from the implementation. The HAL assumes no intelligence at the STB interface and accesses the raw MPEG/ATSC table information. At the API interface the HAL provides a consistent view of the MPEG/ATSC table information in a manner that reflects the API definition. Thus depending on the intelligence of the receiver the HAL allows the API to be glued to the underlying system. Therefore, porting issues are largely contained in the HAL. It is envisioned that this multi-layered design will ease the task of porting the implementation to other receiver platforms.

This talk will give an overview of the design and structure of the NIST Reference Platform. The focus will be given to the API Reference Implementation with a brief introduction of the STB simulation. The major topics include uses and benefits of a 3rd party neutral reference implementation, overview and design of the Reference Implementation, and the status and future plans for the NIST Reference Platform.

Although significant work has been accomplished in the DASE consortium, it is important to note that the standard is not finalized and is a work in progress. As such, the NIST implementation follows a similar path.

NIST Reference Platform and API Implementation

Robert Snelick
National Institute of Standards & Technology (NIST),
Information Technology Laboratory (ITL)
rsnelick@nist.gov
www.dase.nist.gov
Support: ITL & NIST ATP (Advanced Technology Program)

NIST Reference Platform

- Open API Reference Implementation
- Sample DASE Applications
- API Unit Tests
- STB Simulation Platform
- Other Receiver Platforms

Benefits of a NIST RI (I)

- Neutral, 3rd party
  - no bias
  - no preconceived notions
- Proof of Concept
  - does it work?
  - detect inconsistencies and incompleteness in API
  - "benchmark" implementation

Benefits of a NIST RI (II)

- Conformance Testing
  - test against an implementation
- Prototype Source Code
  - prototype (starting point, placeholder, etc.)
- Application Development & Testing
  - accelerate application development process
  - application testing

Reference Platform Scope (I)

- API Implementation
  - What we are Doing:
    - Java APIs (javax.tv, org.atsc, org.davic, org.havi)
    - Application Management
    - JMF Support
  - What we are NOT Doing (potential follow on):
    - Presentation Engine

Reference Platform Scope (II)

- Set-top Box environment
  - Simulation
    - Java simulation
  - Real-time Emulation
    - Targeted to begin in the Fall/2000
  - Commercial Receivers
    - Will work with manufacturers in a collaborative effort

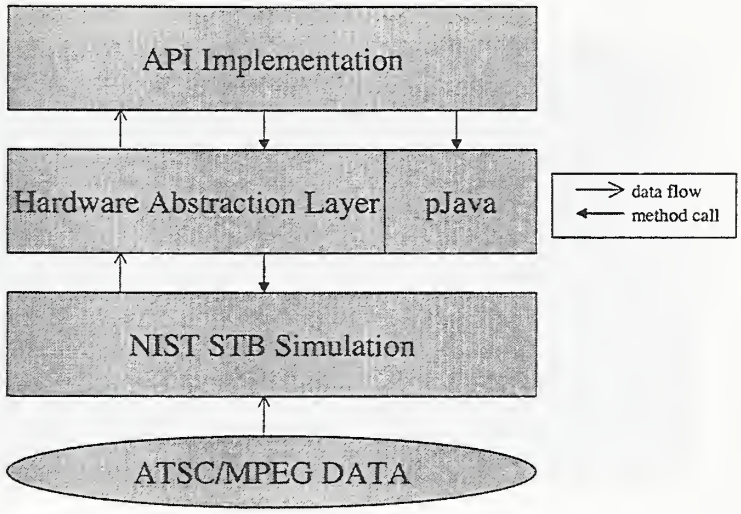
Reference Platform Overview

- API Implementation
  - API definitions as specified
  - gov.nist implementation classes
- Hardware Abstraction Layer (HAL)
  - hides the details of the underlying STB
  - information management
- STB Simulation Environment
  - data management
  - Solaris, Linux, Windows NT

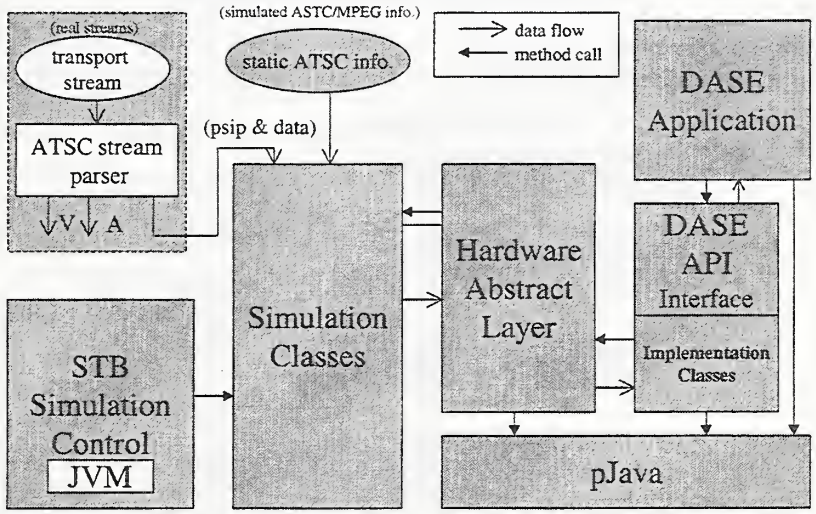
API Design Goals

- Portable
  - Java implementation
  - Intermediate Software Layer between API implementation and STB environment
- Semantic Clarity > Performance
  - implement APIs with semantic correctness in a straight-forward manner
  - initially low priority given to performance and system constraint issues

Implementation Design Stack



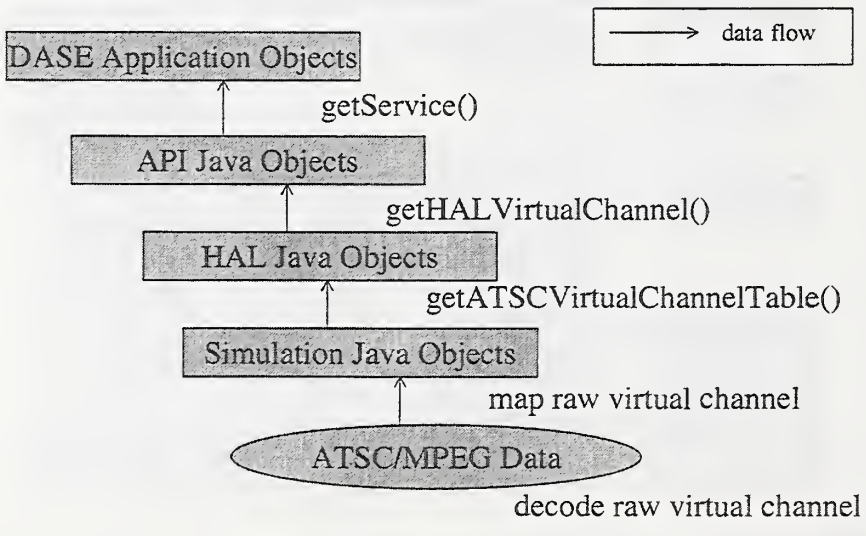
Reference Platform Architecture



Hardware Abstract Classes

Management DataManager XletManager CarouselManager	SI Database VirtualChannelTable EventInformationTable RegionRatingTable
X-let Control XletThread XletState XletClassLoader	Users/Preferences User Preferences FavoriteChannels

Data Flow Example



Sample Applications, Unit Tests

- Native EPG
  - SI Database access
  - User's Preferences
- Downloadable X-lets
  - Application Manager
- JMF Player
- Unit tests

Going Forward

- Complete prototype implementation
  - adjust to changes in specification
  - provide feedback to DASE(S17)
  - encourage review and feedback to ensure correct interpretation
- Port to other STB environments
  - real-time emulation
  - commercial receivers

NIST Implementation Products

- NIST Reference Platform
  - Reference Implementation Source Code
  - STB Simulation Platform Source Code
  - Sample Applications
  - Unit Tests
  - JavaDoc
  - Documentation (SOW, User's Guide, etc.)
- Free and Available to anyone
- www.dase.nist.gov

NIST Settop Box Simulation Environment

Wayne Salamon

National Institute of Standards and Technology
Information Technology Laboratory
<wsalamon@nist.gov>

The NIST Settop Box (STB) simulation environment provides the underlying platform for the execution of the NIST DASE API Reference Implementation. The simulation is coded entirely in Java with a small C language program used to control the operation of the simulation.

The STB simulation contains three major components. The first component within the simulation processes the ATSC and MPEG data tables and the Data Carousel after they have been extracted from the MPEG transport stream. The second component consists of a set of Java classes that maintains the data from in a consistent manner such that the tables are complete and will not be presented in the middle of an update. The third component is a set of Java classes that provides support for STB environment settings, such as user registration, common settings, and hardware state simulation, and control of external services such as an ATSC transport stream parser.

This talk will provide an overview of the STB simulation, including a discussion of the Java classes making up the simulation. The presentation will cover the flow of data from the MPEG streams through the simulation into the Hardware Abstraction layer of the NIST API implementation. Part of the presentation will discuss the reusability of the Java classes outside of the simulation environment. The final part of the presentation will show how the simulation can be used to execute native applications as well as Xlets.

NIST Settop Box Simulation

Wayne Salamon
<wsalamon@nist.gov>
May 23, 2000

Overview

- The need for the STB simulation
- Simulation design
- Simulation components
- Data flow from simulation to API
- Java class code reuse
- Executing native applications and Xlets

2

The Need for an STB Simulation

- DASE API retrieves info from underlying system
- Simulate successful as well as error scenarios
- Maintain state of users and preferences across API test runs
- Used to test Xlets

3

Simulation Design

- Independent of the API and HWAbstract classes
- Implementation based on API requirements
- Performs data management, not information management
  - For example, applies no semantic meaning to the contents of the PSIP tables

4

Simulation Design (cont.)

- Maintains the table consistency and will not return a partially completed table
- Extracts modules from the Data Carousel but doesn't interpret the data
  - For example, Xlet classes and data are maintained as arrays of bytes

5

Simulation Components

- Small C program for initialization, remainder in Java
- ATSC and MPEG table processing and management
- Data carousel module processing
- Other data managers: user, preferences, etc.
- STB state: Power status, resource availability, etc.

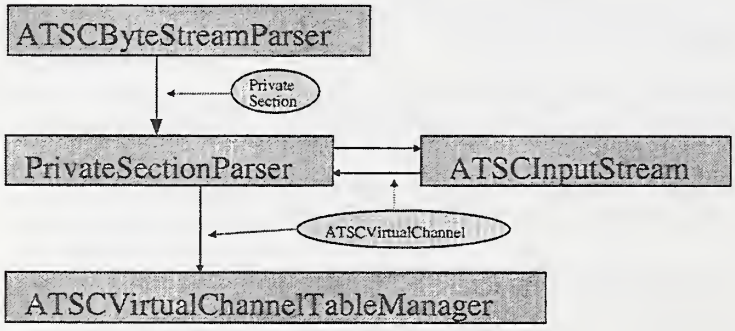
6

Data Flow Example

- Example of how the Virtual Channel Table is extracted by the simulation and presented to the HWAbstract layer
- Trace the class interactions needed to provide this data
- All other tables are handled in a similar manner

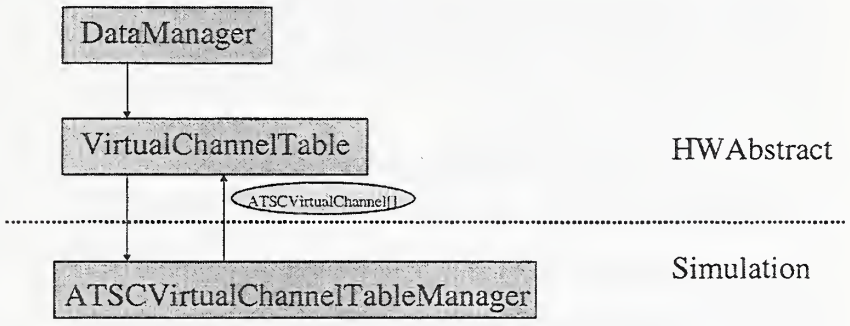
7

Extracting Virtual Channels



8

HWAbstract Channel Retrieval



9

Reuse of the Java classes

- Parsing is separated from table management
- The Huffman decoding is done by a separate utility class which is used by the parser classes
- Carousel module processing is an important function and the classes in the simulation can be reused for S13 protocol handling

10

Executing a Native Application

- Native applications have access to the DASE API and public methods of the HWAbstract classes
- Can be executed from the simulation command line or another native application
- Classes must be found in the CLASSPATH or the STBSIMCLASSPATH

11

Running an Xlet

- Xlets can be executed from the simulation command line by using a wrapper class or injected through the data stream
- Injected Xlets are controlled by the Application Manager and either auto-started or user-started

12

Conclusion

- STB Simulation is not dependent on the API implementation
- Forms the basis of a Xlet development environment
- Portable: Runs on multiple platforms
- Many of the STB classes can be reused
- Will be included in the NIST Reference Implementation distributed via the Web site *www.dase.nist.gov*

13

Developing Programs for Digital Television

Ed Blackmond
Eureka! Computing Solutions
<eb@teameureka.com>

Michael O'Rourke
Dimension 7
<mor@dimension7.com>

This presentation presents digital TV features we believe will motivate consumers to buy digital TV. These features go far beyond better pictures and sound, more channels, and electronic program guides. We leverage the viewer model of television as opposed to the user model of computers.

Computer oriented activities such as browsing the web, processing e-mail, and electronic transactions are not going to be the reason viewers switch to digital TV. We believe television is used as a portal for viewing entertainment. Television certainly presents quite a bit of information but it is not simply a tool for accessing and processing information. In addition, while the technology makes it possible to create sophisticated advertising with buttons to press allowing the viewer to make impulsive decisions to buy a product, this will not be the reason viewers embrace digital TV either. People will buy digital television only when there is compelling content which can not be viewed through the current television paradigm.

Until there are viewers watching digital television programming, advertisers will be reluctant to make a significant commitment to the new technology. However, once a viewer community is established, advertisers will begin to invest heavily into even more sophisticated methods to reach the new audiences.

We present two demonstration applications as examples of our digital TV paradigm. "Multiple Dimensions" presents a model for viewing live entertainment expanding the concept of music videos. Our edu-tainment (educational entertainment) program, "À la Carte," applies our techniques to a "how-to" show. Other shows, including sporting events and drama series, can also be enhanced with these digital television programming techniques.

With these two programs, we hope to stimulate creativity among producers of current television programs. Once they see the capabilities they will begin to visualize new ideas leading to a dynamic new television viewing experience. The longer it takes to expose television producers to the capabilities of the new technologies, the longer it will be before the digital television revolution begins.

Developing Programs for Digital Television

Ed Blackmond
Eureka! Computing Solutions

Michael O'Rourke
Dimension 7

Introduction

- Digital Television Programs
  - Features
  - Viewer vs. User Paradigm
  - Compelling Content Will Drive Market
- Two Digital Television Programs
  - À la Carte
  - Multiple Dimensions

DTV - Features

- Clearer Picture -- Better Sound
- EPG
- Clickable Graphics
- Internet (e-mail, web browsing)
- Conditional Access
- Parental Control

DTV - Paradigm

- Traditional TV Viewer Paradigm
- DTV (minor improvements)
  - Clearer picture
  - Better sound
- DTV (paradigm shift)
  - E-mail
  - Web Browsing
- DTV (new viewing experience)

DTV Programs

- Traditional TV - single A/V stream
- DTV - multiple streams
  - Multiple A/V streams (multicast)
  - Data streams (datacast)

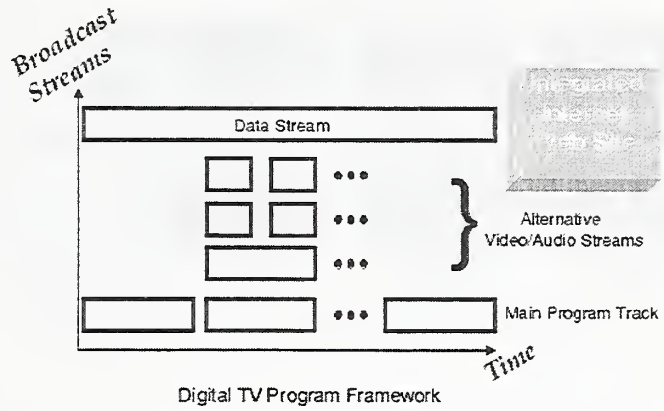
Parallel Viewing

- PGA tournament
  - Spectators follow particular players
  - Spectators sit in grandstand at 18
- Traditional TV
  - Who ever is swinging their club
- DTV
  - Viewers follow particular players
  - Viewers watch all action at 18
  - Viewers browse integrated data

DTV Framework

- 1 or more video streams
- 1 or more audio streams
- Optional datastream
- Optional integrated Web site

DTV Framework

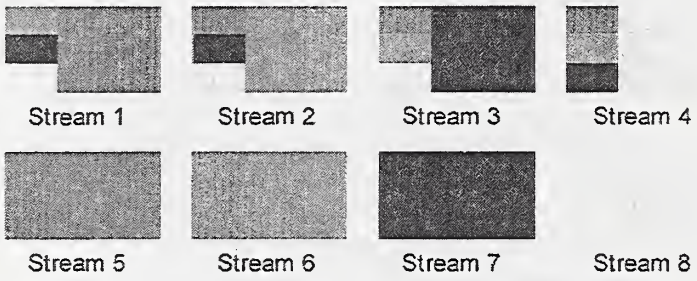


Digital TV Program Framework

Multiple Dimensions

- Draws from MTV, Soul Train, American Bandstand
- Allows viewers to navigate through the virtual dance environment
- Four Cameras
  - Wide angle shot of dance floor
  - Roving camera
  - Performance camera
  - Video mixer output

Multiple Dimensions

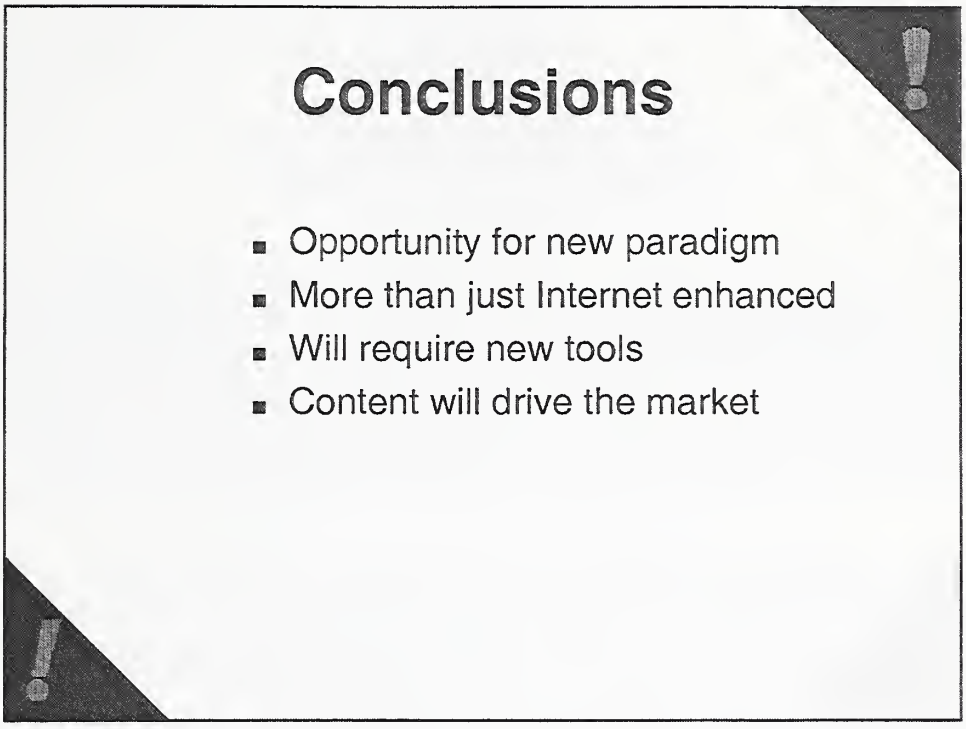


of

Figure 2. Stream configuration for "Multiple Dimensions"

À la Carte

- "How To" Cooking show
- Pasta with seared scallops and shrimp in a ginger-garlic beurre blanc
- 4 Streams
  - Gourmet
  - Quick and easy
  - Health conscious gourmet
  - Quick and easy and low fat


Conclusions
Opportunity for new paradigm
More than just Internet enhanced
Will require new tools
Content will drive the market

Audience Measurement for DASE Compliant Receivers

Scott Brown

Nielsen Media Research
<William_Feininger@tvratings.com>

Nielsen Media Research is the leading provider of television audience measurement and related services in the United States and Canada. Its National People Meter Service provides audience estimates for all national program sources, including broadcast networks, cable networks, Spanish language television, and national syndicators. Local rating services estimate audiences for each of 210 television markets in the U.S., including electronic metered service in 47 markets. These services establish the currency by which broadcasters and advertisers buy and sell advertisements on television.

Nielsen Media Research has a long history of developing technology to meet the measurement needs of the changing television environment. As we enter a new era in the distribution of entertainment programming via digital television, many new products and services including enhanced/interactive broadcasts and e-commerce will be offered to consumers. The Digital Application Software Environment offers a robust platform for consumers to receive these new services, and Nielsen Media Research is developing applications to meet the industry's goals in this environment.

Migrating "Two Screen" content to "One Screen"

Scott Watson

VP Online and Advanced Media
Walt Disney Imagineering
<Scott@disney.com>

Over the last 24 months, Disney/ABC has been producing Enhanced Television content on a regular basis. However, in order to reach the largest population, we have not targeted settop boxes, but have instead used the same Enhanced Television production infrastructure to produce for what we call the 'Two Screen' platform. I will show examples of our content, namely 'Who Wants To Be a Millionaire' and 'Monday Night Football', describe their respective features and talk about how we plan to port them to the DASE AEE/PE.

DASE Conformance and Conformance Testing

Alan Goldfine

National Institute of Standards and Technology
<goldfine@nist.gov>

The success of the emerging DASE standard will depend in large part on the quality of DASE receivers and applications. We can help guarantee this quality by ensuring that these receivers and applications do in fact conform to DASE, i.e., that they faithfully meet the requirements of the standard specification. To this end, the DASE community is:

- incorporating conformance statements into the DASE standard
- developing conformance test assertions and conformance test suites for the standard
- publishing guidance on procedures and policies for conformance testing.

This talk will provide a brief overview and status report on these topics.

DASE Conformance and Conformance Testing

Alan Goldfine
Computer Scientist
National Institute of Standards and Technology

May 24, 2000

Ensuring Conformance

- The success of DASE depends on the quality of DASE receivers and applications
- The quality of the receivers and applications depends on their conformance to the DASE standard

DASE Conformance Activities

- Three planned DASE activities will assist the development of receivers and applications that conform to the standard:
  - Incorporating an effective conformance statement into the DASE standard
  - Developing comprehensive and useable conformance test assertions and profiles
  - Publishing guidance on appropriate conformance testing software, procedures and policies

DASE Conformance Statement

- Is the text that defines what it means to be in conformance to the standard
- Deals with both receiver conformance and application conformance
- Includes both API conformance and PE conformance
- Will be a normative part of the DASE standard

DASE Conformance Test Specifications

- Includes conformance test assertions, which are being developed by UniSoft, Inc.
- Includes profile definitions, and any other optionality specifications
- Will be a normative part of the DASE standard

DASE Conformance Test Guidelines

- Discusses conformance testing issues, including such topics as:
  - Conformance test suites
  - Testing policy
  - Testing procedures
  - Certification
- Will be an informative part of the DASE standard

Conformance Test Development

Andrew Twigger

UniSoft Corporation
<att@unisoft.com>

Overview

This paper is presented in two parts, the first looking at the development of test plans for the ATSC DASE specification and the second looking at the development of a test environment for testing ATSC DASE receiver implementations

Test Plan Development

The presentation reviews the benefits of the assertion driven approach that is being used in the development of a test plan for ATSC DASE. This process is providing feedback to the specification authors to assist in the maturing of the specification.

The presentation provides examples of the problems that can occur during standards definition and updates on the current progress of test planning for ATSC DASE.

Test Framework

This part of the presentation describes some of the problems associated with developing an automated test structure for a digital television environment and outlines some of the requirements that need to be met by receiver implementers to enable automated testing.

The presentation looks at the benefits that can accrue from the use of a common test harness to address the different television standards and the need for abstractions to be implemented to allow for the differences in standards and receiver test environments.

Specification Review and Test Plan Development

Presented by

Andrew Twigger

Managing Director

UniSoft Limited

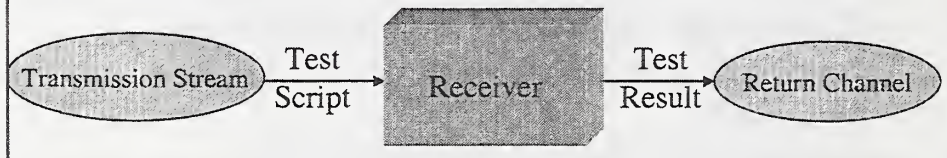
Developing a Test Plan for DTV Testing

- **Determining what needs to be tested**
- **Designing a structure for the test suite**
- **Identifying extensions that will enhance testability**
- **Designing a structure for test case development**
- **Dealing with profiles and configurable options**
- **Measuring results from a test campaign**

Test Suite Goals that affect the Test Plan

- Needs to be platform independent
- Needs to thoroughly test the API Specification
- Needs to identify and report deviations on the system under test
- Needs to provide repeatable and reproducible results

Platform Independence - Black-box Testing



The Test Suite assumes that the Receiver is capable of receiving a transmission stream and providing the specified API services

Preparing the API Specifications for Testing

- **The API specification needs to be written with testability in mind**
  - **Ambiguities need to be identified and resolved**
  - **Both normal and abnormal behaviour needs to be specified**
  - **Dependencies on external interactions need to be identified**
  - **Implementation requirements must be clearly stated and feature profiles defined**

Extensions to the API Specifications to improve testability

- **API extensions can increase testability, for example:**
  - **Simulation of human interaction**
  - **Capture of Presentation Graphics, Video and Audio to allow automated evaluation**
  - **Simulation of Conditional Access Module removal**

Assertion Based Testing - Methodology (1)

- Each "assertion" describes a unit of behaviour specified in the API and leads to a test for that "assertion" in isolation in order to determine its truth value for the Receiver Under Test

Assertion Based Testing - Methodology (2)

- Developing "assertions" from the specification determines the full set of tests needed to cover the API
- Each assertion is written as a plain language description that can be cross referenced to statements in the specification

Assertion Based Testing - Methodology (3)

- Each assertion is assessed for testability with the outline of a software test method being developed as the assertions are generated
- During this process ambiguities in the specification are identified and improvements to assist testability are suggested

Assertion Based Testing - Benefits

- Each "assertion" has a clear purpose
- The "assertion" text assists the user in identifying the specific cause of a deviation identified by the Test Suite
- The methodology enforces good software development practices during test suite development

Example from DAVIC Tuning API

NetworkInterfaceController Constructor

Reserve Event

"If this NetworkInterfaceController has already reserved another NetworkInterface, then it will either release that NetworkInterface and reserve the specified one, or throw an exception."

Problems in the Example

- **Omission** - does not state that a NetworkInterfaceReleaseEvent is sent to the listeners
- **Ambiguities** - does not state how the either/or clause is determined (or whether it is consistent). Does not state whether the Release/Reserve pair is atomic.
- **Application Requirement** - implies that the application must handle either success or error (and recover gracefully on error)

Assertion Development Process

- Iterative process as API Specifications are developed to produce "assertions", feed back questions and perceived problems.
- Start with adopted APIs from other sources and move on to DASE specific APIs

Performance Testing of the API Implementation

- Where specific metrics are included in the API Specification, these will be measured by the Test Suite
- General performance metrics will be available but are not usually used in conformance assessment

Important Features of the Assertion Development Process

- Helps to identify problems in the specification
- Helps to identify constraints on testability
- Relates test purposes to profiles
- Provides a matrix of acceptable test results for each profile

Current Status

- Assertion Development for Java APIs in DASE Draft 1.08.01
- Assertion Development for Application Execution Engine to be commenced against next draft document

A Framework for Digital TV API Testing

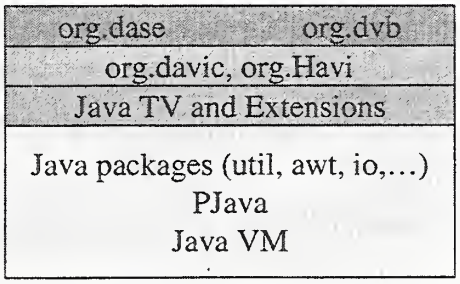
Presented by

Andrew Twigger

Managing Director

UniSoft Limited

What Needs To Be Tested?



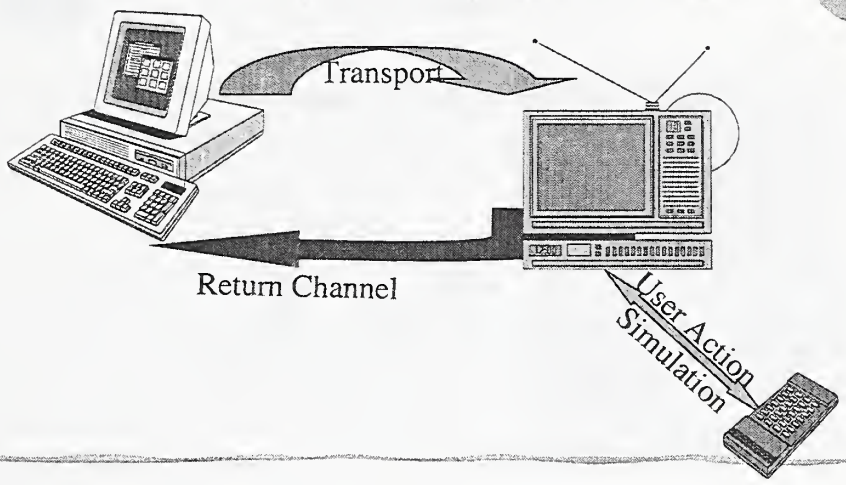
How are these layers tested today?

- No industry tests currently available for the higher layers
- Java layers are tested in a "computer environment" where tests are referenced from known locations

Differences for the DTV environment

- Test framework does not reside on the System Under Test (STB)
- Test cases need to be delivered to the STB using the appropriate delivery stream
- STBs in development may use a different 'delivery stream'
- Test results need to be returned by the STB

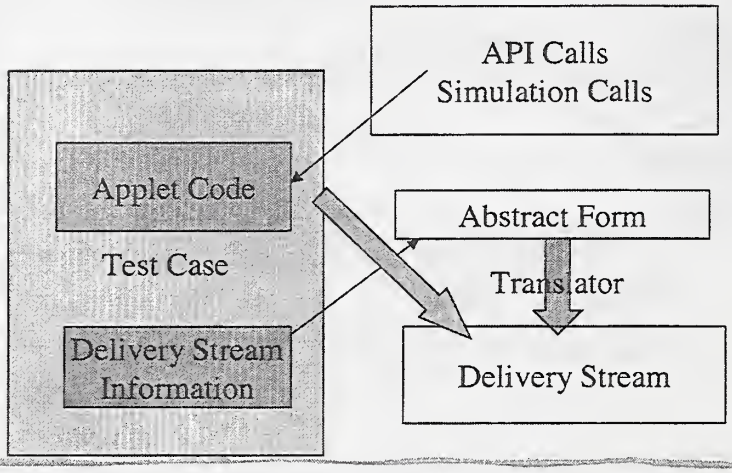
Simple Representation of a DTV Test Environment



Test Re-usability Considerations

- **Test code should be independent of transport and response channel**
- **Transport and response channel code needs to be implementable in a range of different scenarios**
- **User action simulation will probably be an additional API**

Test Case Content



Proposed Translators

- **Simple translator to allow test cases to run in a simulated development environment**
- **Translators to generate streams suitable for use in DASE and in DVB**

Using DASE to Enhance TV: A PBS Perspective

Dave Johnston

Senior Director of Technology
PBS Online
<djohnston@PBS.ORG>

Key to the Public Broadcasting Service's educational mission is the need to engage, inform, enlighten and entertain, and in doing so exploit the educational power of the television medium. We treat the casual viewing experience as a sort of "passive learning opportunity".

PBS has long been engaged in the business of "enhancing the television broadcast", though initially through fairly "low tech" means. We have distributed hundreds of thousands (if not millions) of printed viewer guides and teacher guides to enable learners and viewers to get a fuller understanding of the corresponding television program subject material. We've created television series that have corresponding textbooks and study guides as enhancements to be used as complete college credit courses. More recently we've been creating digital enhancements in the form of websites that enhance the viewing experience by creating a place where "to learn more" is just a few mouse clicks away.

So how does the Digital television Application Software Environment (DASE) platform support the furthering of the PBS mission? This session looks at some models of data enhancement, both nationally and locally focused, that expose additional opportunities for the viewer through synchronous and asynchronous enhancements. The technical distribution path and strategic roll-out of content by PBS will be discussed. Concepts such as viewer acceptance of synchronous enhancements, and application stability requirements will also come to light.

The presenter of this session does not claim to have all the answers with regards to enhanced television, nor will he tell you what is the "killer application" in advanced DTV. He will, however, share some of the progress made by PBS in developing concepts for enhancing the digital television broadcast, and expose some of the many questions still remaining.

Using DASE to Enhance TV: A PBS Perspective

DASE Symposium
May 23-24, 2000
National Institute of Standards & Technology
Gaithersburg, MD

Overview

- PBS's Mission
- Evolution of Enhancement
- National (network-emitted) and Local (station-inserted) content
- Templated and "Always On" Model

Overview (cont'd)

- Post-viewing and Data Broadcasting
- Consumer Acceptance
- Leveraging DASE
- Distribution Model
- Conclusions

PBS's Mission

- Engage, inform, enlighten, entertain!
- Leverage the **educational power** of television
- Have you ever wanted to learn more?

In the Beginning...

We've always extended the broadcast:

- Community outreach
- Viewers' Guides
- K12 Teacher resources
- Formal Higher Ed resources

In the Beginning...

Enter the Internet:

- Late 80's - Usenet News program information, virtual communities emerge
- Early 90's - PBS publishes NPS, K12 & ALS via Gopher

In the Beginning...

- March 1, 1995 - PBS announces "the PBS Website"
- Program extending and **enhancing** in-depth websites
- September 1995 - closing the loop with "on-air" web "tags"

Evolution of Enhancement

What does all of this internet stuff have to do with enhanced TV?

All of our cyberspace activities have been aimed at asynchronously *enhancing* the viewing experience!

Exploring Models of Enhanced TV

- Killer App: (If I knew, would I tell you?)
- Models to be tested...
  refined...
  kept or...
  discarded!

National & Local Content Model

- National content is distributed to stations for broadcast from the PBS Satellite Operations Center
- Local content inserted either into national enhancement or as stand-alone enhancement

National & Local Content Model

- Typical commercial "buy now" application
- PBS Home Video
- Pay per ***not*** view - "pledge-break free" programming
- Backchannel and conditional access

Templated & "Always on Demand" Synchronous

- Collaborative work with program producers
- Need for common navigation
- Make it easy to do, and always available
- Customized on a per program basis

Data Broadcasting and Post-viewing Asynchronous

- Post viewing enhancements currently targetted to PCs
- Explorations, 3-D walkthroughs, simulations
- Teachers' or viewers' guides, databases

Data Broadcasting and Post-viewing Asynchronous

- Data Broadcast model of the late 1980's?
- Internet connectivity
- Not re-creating web over broadcast

Consumer Acceptance of Enhanced Content

- How much is enough?
- How much is too much?
- Program attributes: factually intensive?
- Target audience

Leveraging the DASE Receiver

- Start with the default PE
- Some post-viewing enhancements
- Closely monitor receiver roll-out
- Augment PE-centric enhancements w/Java xlets

Distribution Model

Sample SDTV "bit-budget" model:

4-way multicast @ 4mbps	16
4 enhancements @ .5	2
1 commercial payload @ .5	.5
1 reserved for station use .5	.5

Total: 19 mbps

Distribution Model

Sample HDTV "bit-budget" model:

1 HDTV program @ 17.5mbps	17.5
2 enhancements @ .5	.5
1 commercial payload @ .5	.5
1 reserved for station use .5	.5

Total: 19 mbps

Distribution Model

- Phased roll-out of services allows pass-through w/o costly decode/re-encode adding data enhancement through "re-mux"
- Stations can add decode/encode when prices come down, features and quality come up

Conclusions

- Start by exploiting expertise with Presentation Engine
- Test, modify, test more
- Slowly incorporate more sophisticated applications

Digital Television and Home Networking Paradigm

Alexander D. Gelman, Rajesh B. Khandelwal

Panasonic Information and Networking Technologies Laboratory
2 Research Way, Princeton, NJ 08540
<adg@research.panasonic.com>

Rapid growth of the Internet, expansion of the World Wide Web, and proliferation of Personal Computers created an environment where a person in Central Africa is capable of accessing the same information, use the same e-commerce vendors, and trade on-line as a person in the United States. This enormous progress still leaves most people even in developed countries deprived of access to information that could greatly improve their lives. This situation may persist for a very long time, if we don't address the needs and specificity of consumer mass market information networking. Even those who spend most of their working day at the screen of a computer could still benefit other times from applications that rely on information access.

While there is grows in the number of households with multiple PCs, most people on this Planet still don't use computers for various reasons. Meantime consumer electronics industry managed to reach unprecedented affordability and user-friendliness levels for its high-tech appliances. The challenge at hand is to make use of these appliances in the Global Information Infrastructure by making them network-connected, i.e. enabling them to support consumer communications applications.

Most consider consumer applications to be entertainment-related, which may be so, but we must also include voice, data, education, secure transactions for support of various forms of e-commerce, etc. In order to implement this mass market Information Networking, carefully crafted applications need to be created as well as services support mechanisms that will hide the complexity of the technology from the consumer. Luckily, there exist already various non-PC intelligent devices that could be networked and thus support Information Networking Applications.

It must be noted, that as software layers get "thicker", consumer devices become more complex and their life span becomes shorter. This situation requires introduction of a Consumer Electronics support infrastructure on the part of the CE companies and developing long-term relationships between CE suppliers and consumers. Networked appliances, Residential Networks, and Internet allow to create such infrastructure.

Digital Television, though for the most part one-way, but still constitutes a broadband communication service. And the DTV set, with or without the return channel, is a network-connected consumer device. DASE platform presents a powerful mechanism for support of consumer applications and consumer communications services management.

We present examples of Mass Market Information Networking applications and Home Information Infrastructure management schemes. We offer a view on the Consumer Communications Value Chain composition that insures user-friendly mechanism for provisioning, service creation, and support of the Home Networking environments as well as their integration into the Global Information Infrastructure.

Panasonic Technologies, Inc.

DTV and Residential Networking Paradigm

Alexander D. Gelman, Rajesh B. Khandelwal
Panasonic Information and Networking Technologies Laboratory
May 24, 2000

Outline

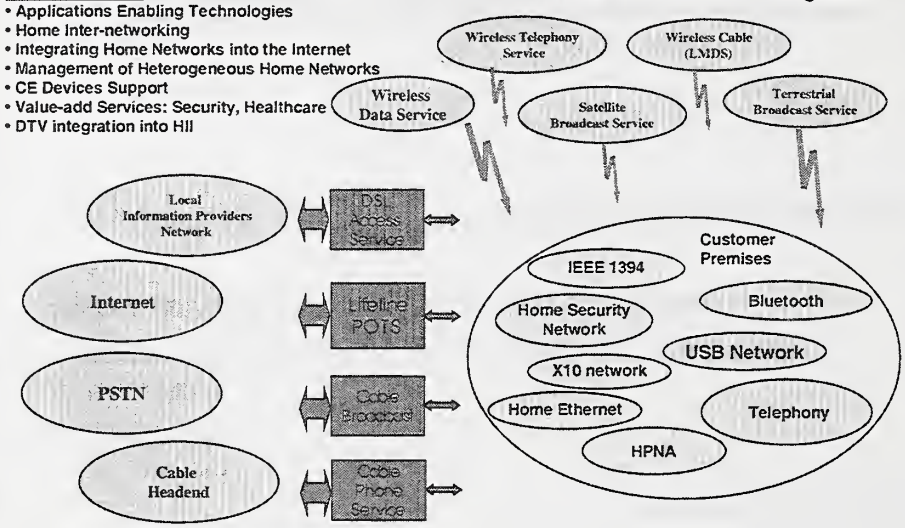
Panasonic Technologies, Inc.

- Residential Networking Paradigm
- DTV Communications Scenarios
- HII Management and Support
- Mass Market Information Networking Applications
- Consumer Communications Value Chain
- Conclusions

Consumer Communications

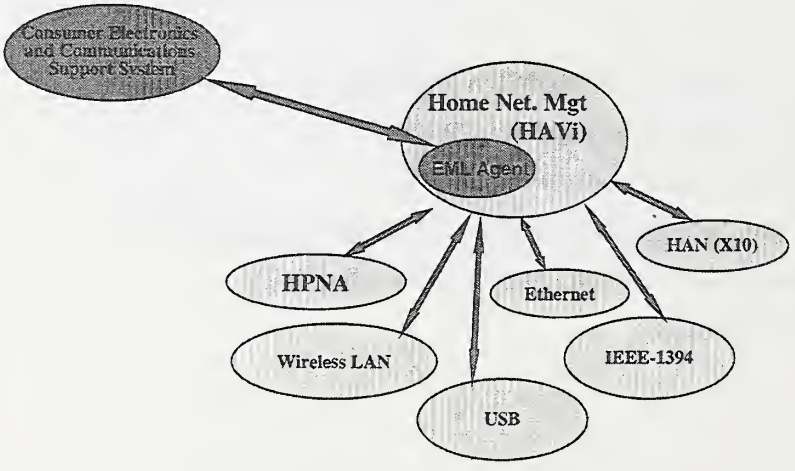
Panasonic Technologies, Inc.

Problem Areas:
- Applications Enabling Technologies
- Home Inter-networking
- Integrating Home Networks into the Internet
- Management of Heterogeneous Home Networks
- CE Devices Support
- Value-add Services: Security, Healthcare
- DTV integration into HII



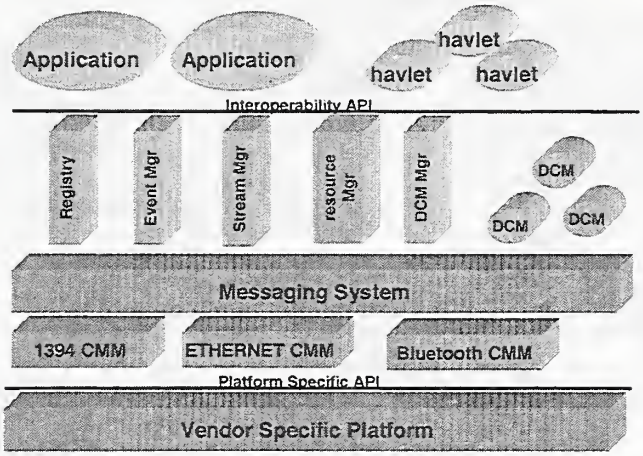
Residential Networks/Appliances Management

Panasonic Technologies, Inc.



HAVi Architecture

Panasonic Technologies, Inc.

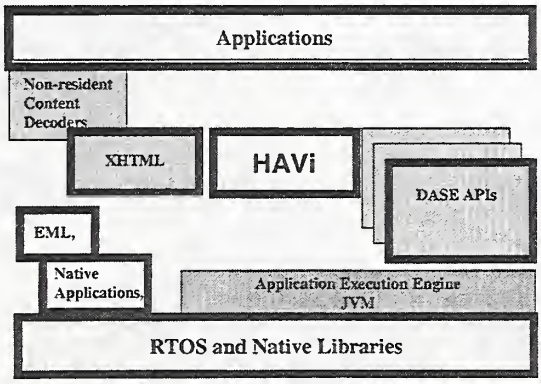


CMM - Communication Media Manager

DMM - Device Control Manager

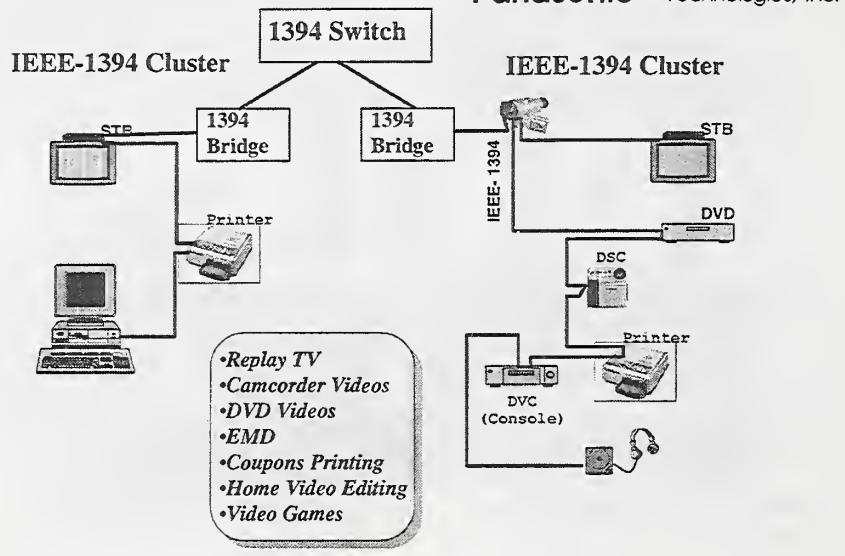
ATSC/DASE Platform

Panasonic Technologies, Inc.



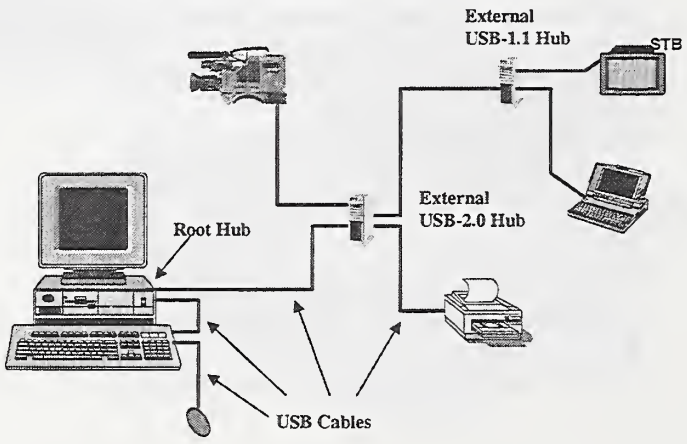
IEEE-1394 Architecture and Applications

Panasonic Technologies, Inc.



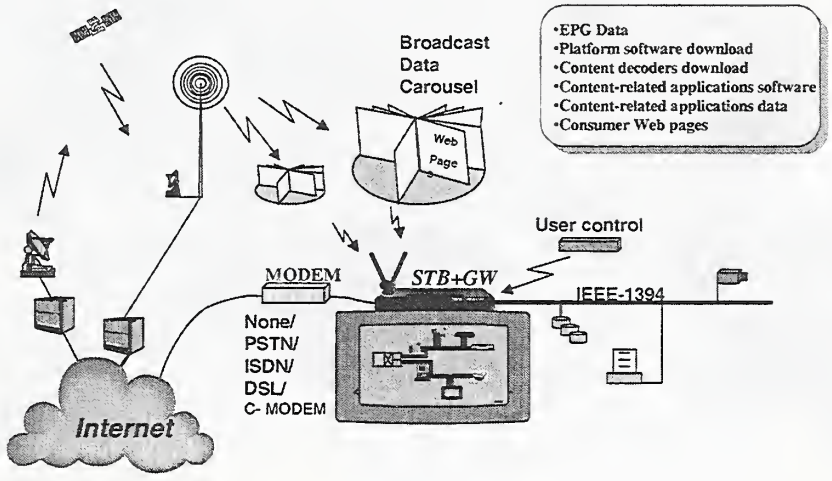
USB Architecture and Applications

Panasonic Technologies, Inc.



Terrestrial TV Communication Scenario

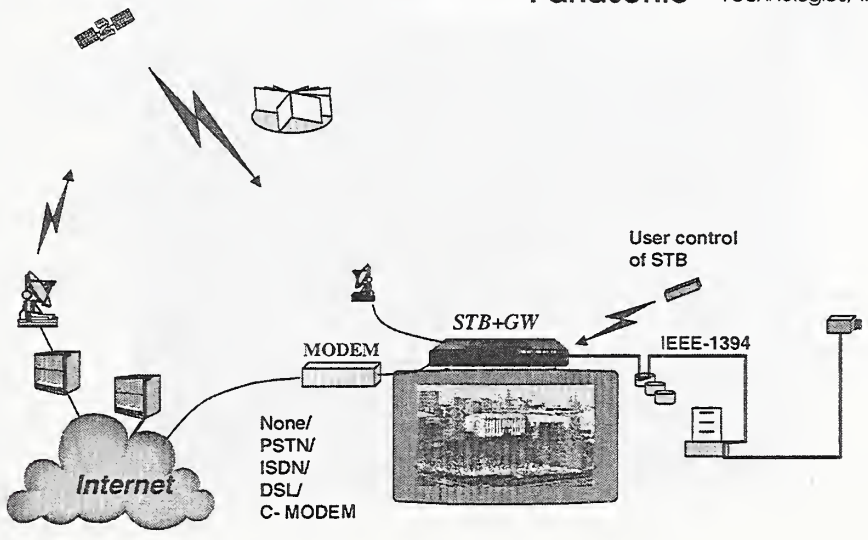
Panasonic Technologies, Inc.



4/28/00 9

Satellite TV Communications Scenario

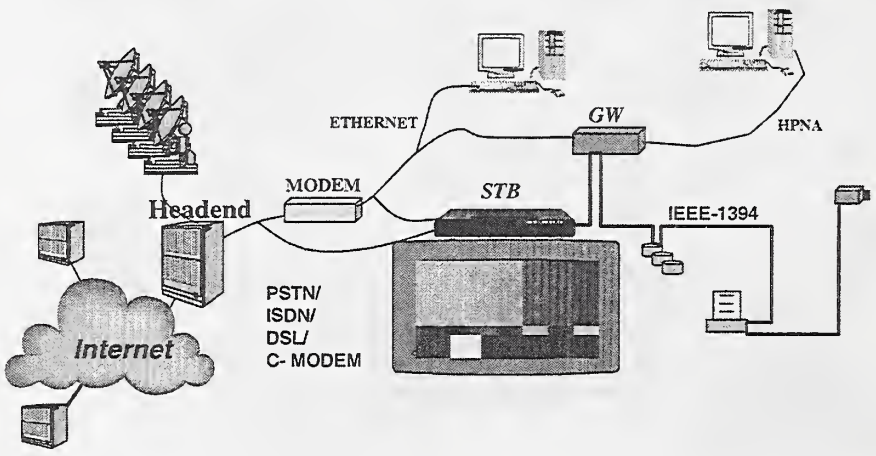
Panasonic Technologies, Inc.



4/28/00 10

Cable TV Communications Scenario

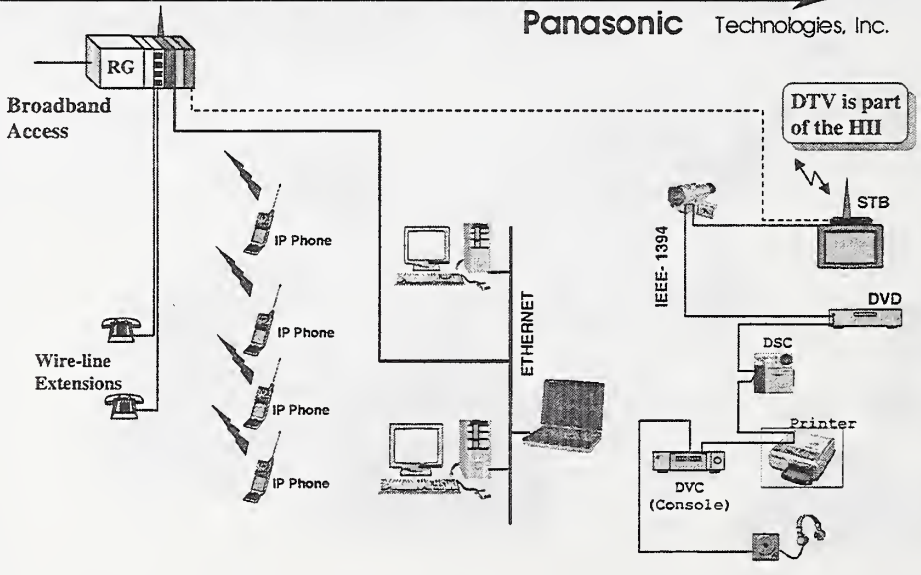
Panasonic Technologies, Inc.



4/28/00 11

Residential Gateway Concept

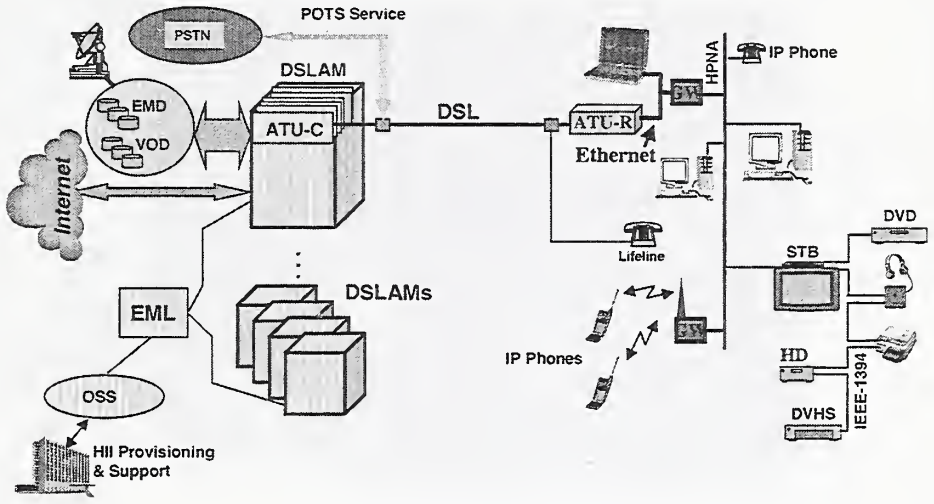
Panasonic Technologies, Inc.



4/28/00 12

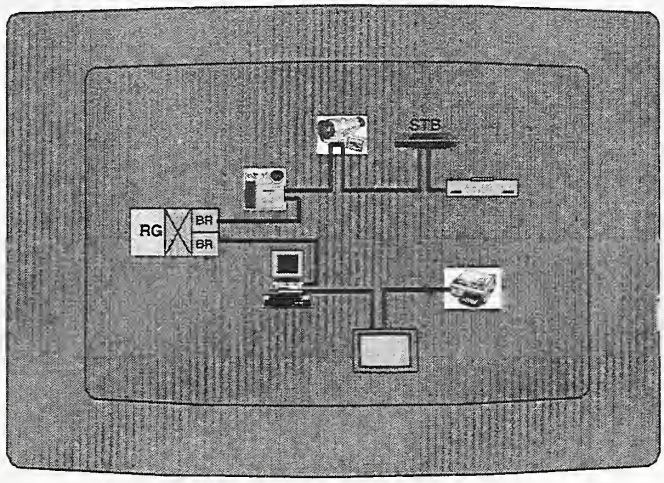
DTV and Broadband Internet Access Services

Panasonic Technologies, Inc.



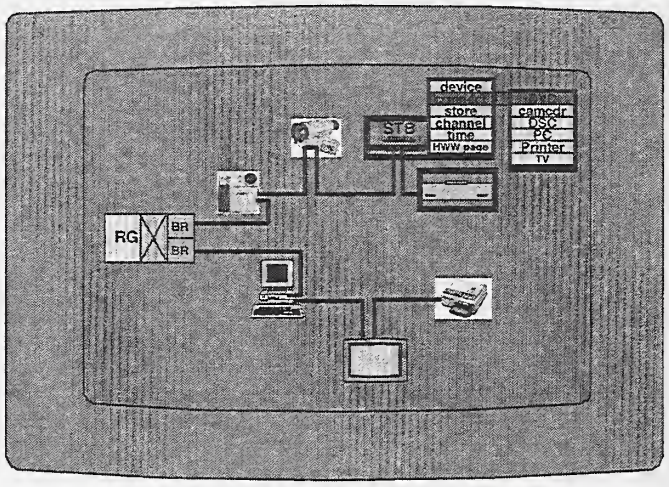
Residential Network Management

Panasonic Technologies, Inc.



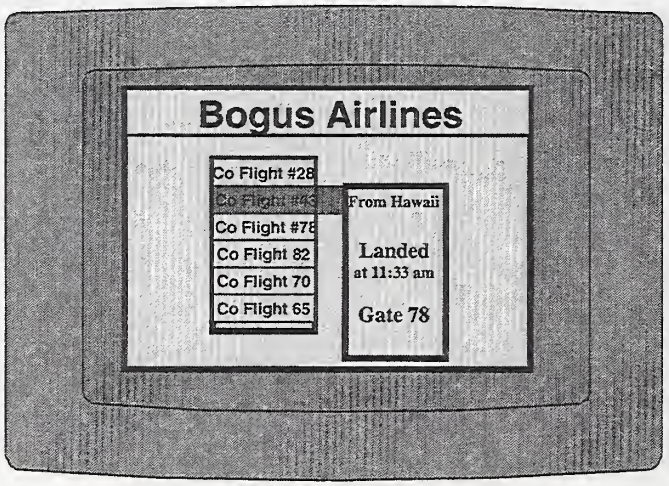
AV Applications

Panasonic Technologies, Inc.



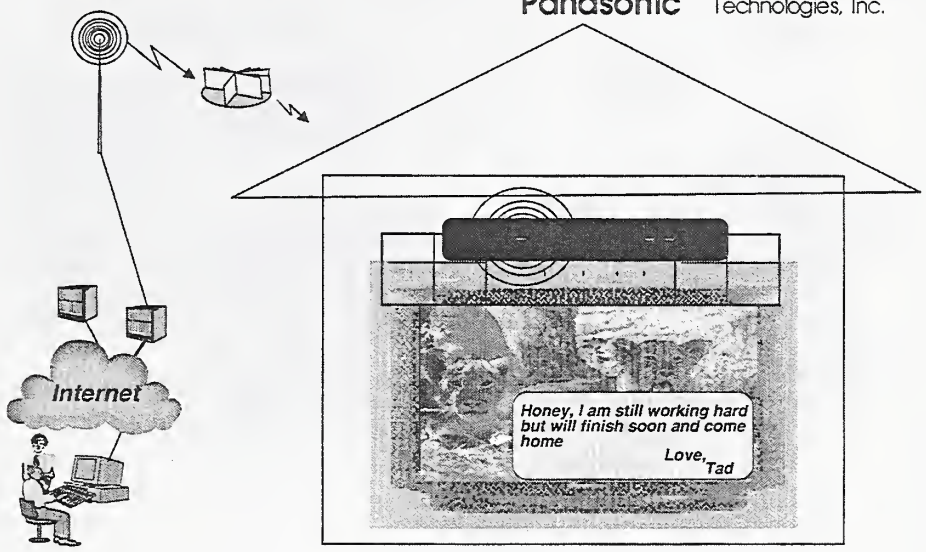
Datacasting Application: Flight Information

Panasonic Technologies, Inc.



Datacasting Application: Messaging

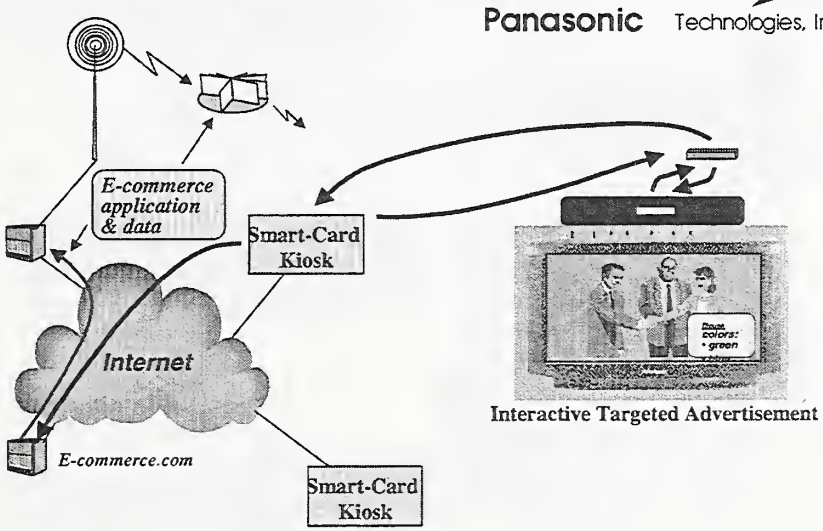
Panasonic Technologies, Inc.



4/28/00 17

Datacasting: E-commerce Application

Panasonic Technologies, Inc.

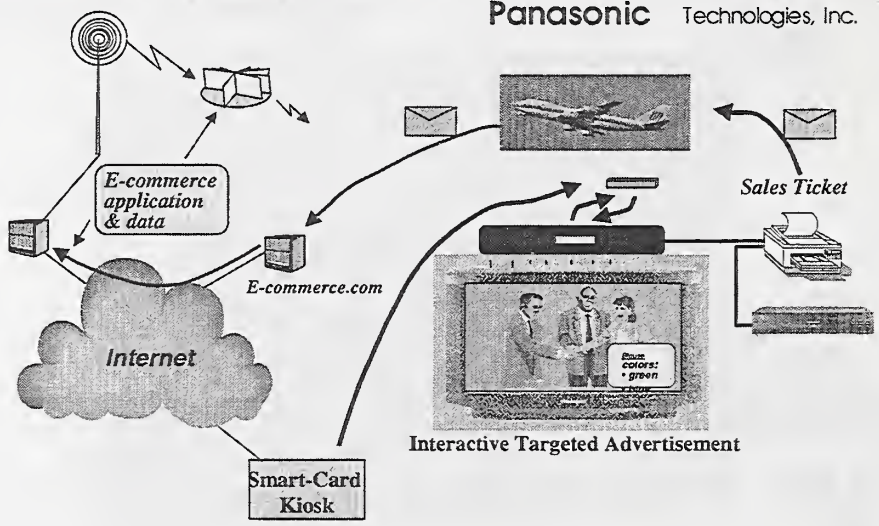


Interactive Targeted Advertisement

4/28/00 18

Datacasting: E-commerce Application

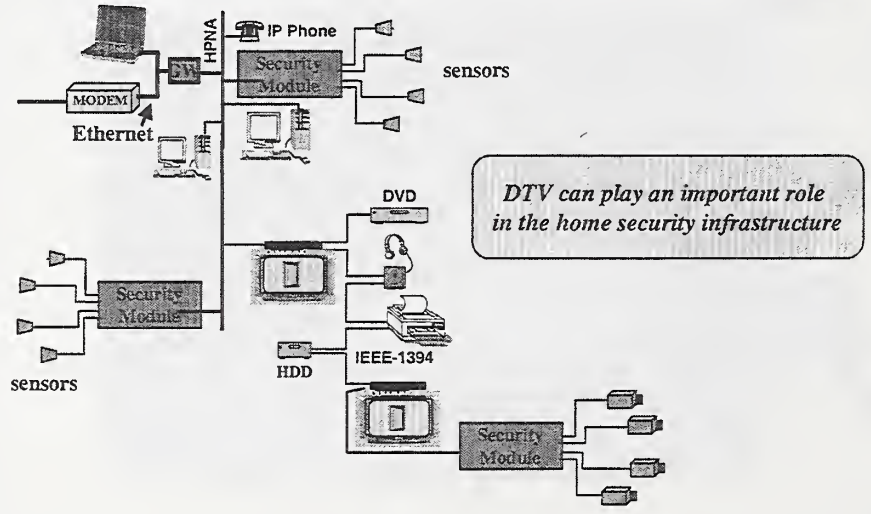
Panasonic Technologies, Inc.



4/28/00 19

Home Security Application

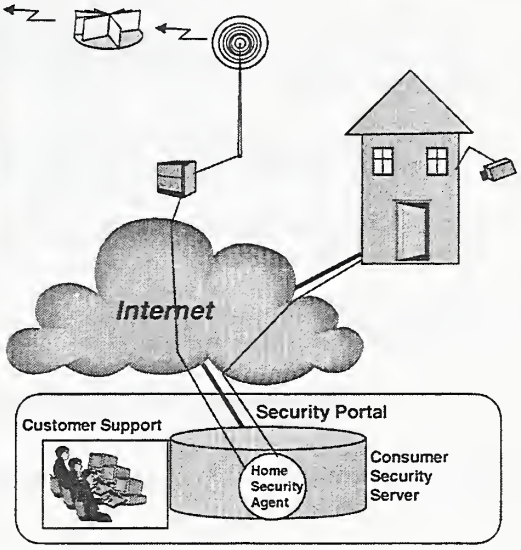
Panasonic Technologies, Inc.



4/28/00 20

Home Security Application

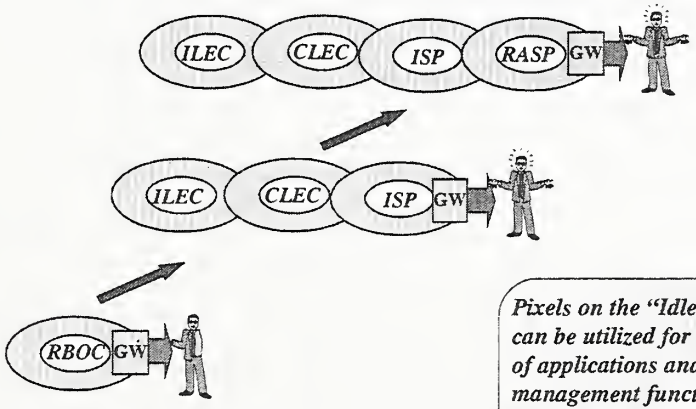
Panasonic Technologies, Inc.



4/28/00 21

Communications Value-Chain Evolution

Panasonic Technologies, Inc.



ILEC=Incumbent Local Exchange Carrier
CLEC=Competitive Local Exchange Carrier
RASP=Residential Applications Service Provider

Pixels on the "Idle Screen" can be utilized for a rich set of applications and HII management functions. In fact these pixels may be Key to the position in the value chain

4/28/00 22

Home Networking Service/Applications Provider Concept

Panasonic Technologies, Inc.

- *Home Networking Service will get inserted into the Communication Services Value Chain. They must become a part of end-to-end provisioning and operations support.*
- *Elements of home networks will be MODEMs, RGWs, (distributed, dedicated and/or embedded within CE devices), and Networked Appliances*
- *Residential Applications Provisioning opens new opportunities for long-term relationships with consumers*
- *HII technologies suppliers' challenge is to emerge as Residential Networking Service and Applications Providers*
- *Display-equipped Intelligent Devices present enormous value for HII*
- *DASE environment is a great platform for support of HII and Mass Market Information Networking*

A DTV Solution That Includes HD, Multicasting, PVR, pJava and Web Browsing

Mark O'Brien

TeraLogic, Inc.
<Mark@teralogic-inc.com>

Open Interface Solutions for DTV Datacasting Systems: Requirements, Products, Directions and Standards

David Catapano

Triveni Digital, Inc.
<dcatapano@lgerca.com>

ATSC DASE datacasting provides many opportunities for DTV broadcasters. This presentation discusses key issues in this exciting area. Topics include:

- A taxonomy of data broadcasting in terms of target audience and application characteristics
- An overview of the emerging ATSC Data Broadcast Standard
- Challenges in implementing end-to-end data broadcast solutions for enterprise-to-enterprise and enterprise-to-consumer applications
- System components required for end-to-end flow of DASE applications and corresponding product requirements

One key challenge is managing the end-to-end flow of data, with suitable architectural support for content providers, broadcasters, and users. Other challenges include bandwidth allocations, error correction, compression, and security.

MEDIAHIGHWAY & DASE

Philippe PIOVESAN

Canal+ Technologies
<ppiovesa@canal-plus.fr>

With 15 years experience in PayTV, CANAL+ is one of the heavyweights of the television industry. Awareness of the huge potential offered by digital technologies led to the creation of CANAL+ TECHNOLOGIES a fully owned subsidiary of CANAL+. With headquarters in Paris and offices in Cupertino (California) and New York, CANAL+ TECHNOLOGIES employs over 450 highly trained engineers whose expertise supports a solid industry-wide reputation. Over 5 million digital set-top boxes based on the systems developed by CANAL+ TECHNOLOGIES are currently deployed in the world, a number growing rapidly.

MEDIAHIGHWAY is a complete end-to-end solution for Digital TV Interactive applications. It provides authoring tools to develop and design interactive applications, head-end server software, a catalogue of existing application and a TV terminal middleware. MEDIAHIGHWAY is made of a set of interpreters that can easily be added to meet our customers' requirements.

Our research teams are constantly anticipating the standards to come through the development of a multi-standard platform for instance (MHEG5, Java, and HTML engines). We have taken an active role in the European DVB standardization process, and have begun to work on similar fields in other continents. Migration towards open standards is indeed a key element of Canal+ Technologies strategy.

The first step of this open process was ONdigital, the world's first digital terrestrial operator that launched in the UK in November 1998. In just a few months, CANAL+ TECHNOLOGIES developed a version of the ISO-standardised graphics MHEG-5 interpreter (MHEG-5 was selected by the British Digital Terrestrial Group (DTG) as the standard for all digital terrestrial broadcasters in the UK).

In parallel, CANAL+ TECHNOLOGIES has developed its own Java^{tm} Virtual Machine. TV specific APIs have been designed and developed around it to build MEDIAHIGHWAY+, the first Java-based interactive TV middleware. This system is currently deployed and operational in the US through MediaOne.

Deeply involved in the DVB-MHP standard process, CANAL+ TECHNOLOGIES has now nearly finished an implementation of this standard and first DVB-MHP platforms with MEDIAHIGHWAY middleware will be deployed at the end of this year.

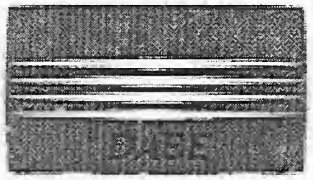
Similarities between DVB-MHP and ATSC-DASE are numerous: core APIs (lang, io, utils, ...), common parts of JavaTV and Davic APIs. Deltas lie in the specification of APIs that deal with the system specificities of each specification: application signaling, application management, security model, transport protocols, user preferences, event model, application model, ... Presentation engine specifications are at the time of the writing not stabilised enough in both standards to be able to draw any comparisons. At the API and system level however, those relatively low functional deltas make it easy to quickly deploy a MEDIAHIGHWAY middleware, compliant with ATSC-DASE AEE when this standard is published.

CANAL+ TECHNOLOGIES

Summary
DVB-MHP
TV & @
Products
Roadmap
Conclusion

MEDIAHIGHWAY

&



ADVANCED TELEVISION SYSTEMS COMMITTEE

23th May 2000

CANAL+ TECHNOLOGIES

Summary
DVB-MHP
TV & @
Products
Roadmap
Conclusion

SUMMARY

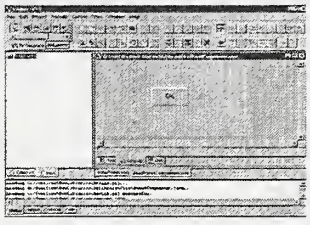
- *MEDIAHIGHWAY OVERVIEW*
- *MIDDLEWARE ARCHITECTURE*
- *MEDIA*HIGHWAY *& STANDARDS*
- *DVB-MHP & ATSC-DASE COMMON BASE*
- *CONCLUSION*

CANAL+ TECHNOLOGIES

MEDIAHIGHWAY OVERVIEW

CANAL+ TECHNOLOGIES

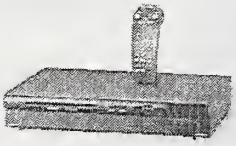
MEDIAHIGHWAY OVERVIEW



An authoring tool

A catalogue of applications

Broadcast servers



A middleware in the terminal

© 2000 CANAL+ Technologies. MEDIAHIGHWAY is a registered trademark of CANAL+ Technologies.

CANAL+ TECHNOLOGIES

HIGHWAY

Studio+

Summary

DVB-MHP

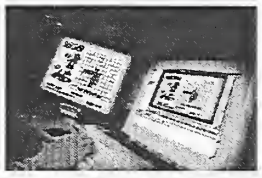
TV & @

Products

Roadmap

Conclusion

- An Integrated Development Environment (IDE)
- Easy to use GUI
- WYSIWYG tool
- Automatic generation of Java™ code and documentation
- Includes a simulation tool
- Connected to a Set-top-box for debugging purposes



Studio+ makes the development of digital TV applications as easy and fast as for the Internet

CANAL+ TECHNOLOGIES

HIGHWAY

CATALOGUE OF APPLICATIONS

Summary

DVB-MHP

TV & @

Products

Roadmap

Conclusion

- ENHANCED TV
  Electronic Program Guide, TV Pilot, Program Mosaic, Games, Weather Forecast, Magazines, ...
- INTERACTIVE TV
  Pay-Per-View, Teleshopping, Telebetting, Telebanking, Quizz, Interactive advertising, ...
- INTERNET ON TV
  Web browsing, e-mail, chat, ...

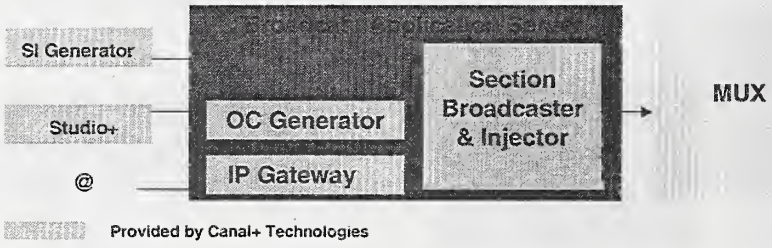
- The GUI is customisable by the customer
- The applications include the back-office part

CANAL+ TECHNOLOGIES

MEDIAHIGHWAY

BROADCAST SERVER

- Signalling server
- Application and data broadcast server (DSMCC)
- Service Information server (DVB-SI)
- Interactive server IP, HTTP, TLS



CANAL+ TECHNOLOGIES

MEDIAHIGHWAY

MIDDLEWARE ARCHITECTURE

CANAL+ TECHNOLOGIES

HIGHWAY

MIDDLEWARE

- Summary
- DVB-MHP
- TV & @
- Products
- Roadmap
- Conclusion

MEDIASTART Resident Applications

Broadcast Applications

[redacted]

Device Layer Interface

Manufacturer 's Firmware

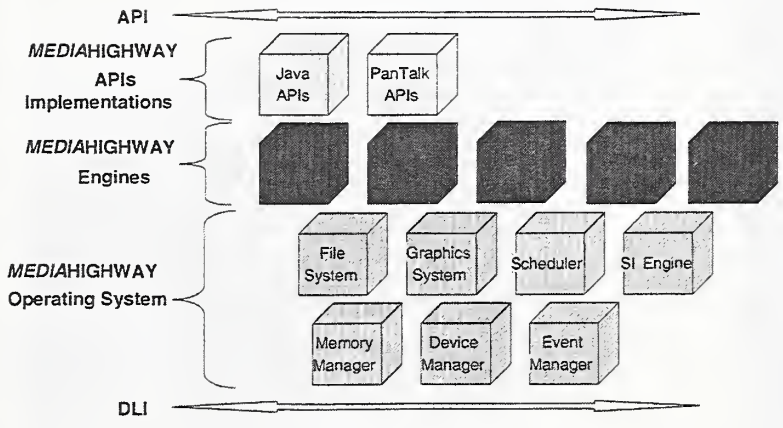
Hardware

CANAL+ TECHNOLOGIES

HIGHWAY

MIDDLEWARE ARCHITECTURE

- Summary
- DVB-MHP
- TV & @
- Products
- Roadmap
- Conclusion



CANAL+ TECHNOLOGIES

HIGHWAY

MEDIASTART

Summary

DVB-MHP

TV & @

Products

Roadmap

Conclusion

- Set of resident applications
- Customizable Look & Feel
- Application and Resource Management
- User's preferences, Parental code, DVB-CI, ...

CANAL+ TECHNOLOGIES

HIGHWAY

Device Layer Interface

Summary

DVB-MHP

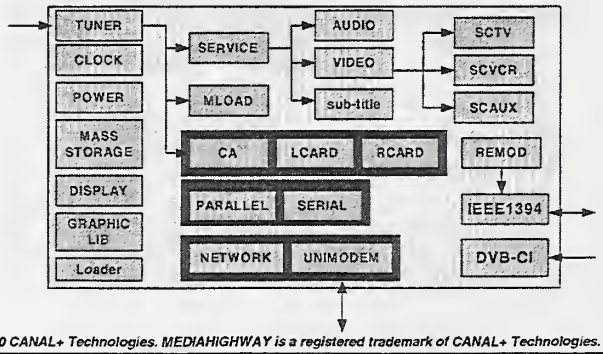
TV & @

Products

Roadmap

Conclusion

- DLI 3.X code name "black box"
- DLI 4.X
  - 2 boxes : -"MediaWeb"
    -"basic box"



CANAL+ TECHNOLOGIES

MIDDLEWARE KEY FEATURES

- ***Interoperability***

Edit Once, Play Everywhere

- ***Portability***

Hardware, firmware and RTOS independent

- ***Flexibility***

The DLI allows for the introduction of specific devices

- ***Multi standard***

Support for multiple languages and formats

CANAL+ TECHNOLOGIES

MIDDLEWARE KEY FEATURES

- ***Specifically designed for digital TV terminals***
  - ***No port of PC or Workstation software***
  - ***Efficient and compact***
- ***Already operational***

CANAL+ TECHNOLOGIES

MEDIAHIGHWAY & STANDARDS

CANAL+ TECHNOLOGIES

HISTORY of Middleware Standards 1/2

- First MEDIAHIGHWAY deployment in Europe in 1996
  - MPEG2 A/V with PanTalk (Procedural language)
- MEDIAHIGHWAY for Terrestrial UK in 1998
  - PanTalk + MHEG-5 (Profile defined by the DTG)
- MEDIAHIGHWAY for US Cable in 1999
  - Java + DAVIC 1.4 APIs + MEDIAHIGHWAY APIs (MH+ APIs)

CANAL+ TECHNOLOGIES

HISTORY of Middleware Standards 2/2

- **MEDIAHIGHWAY in 2000**
  - DVB-MHP (Java + ML)
  - ATSC-DASE (Java + ML)
  - ARIB (BML)

CANAL+ TECHNOLOGIES

DVB-MHP & ATSC-DASE COMMON BASE

CANAL+ TECHNOLOGIES

CONTENT OF SPECIFICATION

Summary

DVB-MHP

TV & @

Products

Roadmap

Conclusion

DVB-MHP is made of :

- **DVB-J :**
  a Java™ based software platform
  comprising a Java™ Virtual Machine and a set of TV-specific APIs

- **DVB-HTML :**
  an HTML based software platform
  adapted to the TV environment and a script language (ECMAScript)

CANAL+ TECHNOLOGIES

Summary

DVB-MHP

TV & @

Products

Roadmap

Conclusion

...

- **An application model** *(for Java & HTML)*
  specifying the lifecycle of the application, an application manager, a resource manager

- **Broadcast protocols**
  (DVB object carousel and Multi Protocol Encapsulation)
  and interactive protocols
  (IP, HTTP)

CANAL+ TECHNOLOGIES

...

- **A selection of contents formats**
  (for text, graphics, sounds, still pictures, etc.)
- **An application signalling mechanism**
  (extension to the PMT, specification of the AIT table)
- **A security scheme**

CANAL+ TECHNOLOGIES

...

DVB-MHP defines 3 profiles :

- **Enhanced broadcast** *(local interactivity)*

Electronic Program Guide, TV Pilot, Program Mosaic, Games, Weather Forecast, Magazines, ...

- **Interactive broadcast** *(interaction channel)*

Pay-Per-View, Teleshopping, Telebetting, Telebanking, Quizz, Interactive advertising, ...

- **Internet access** *(internet services)*

Web browsing, e-mail, chat, ...

CANAL+ TECHNOLOGIES

PLANNING

Summary

DVB-MHP

TV & @

Products

Roadmap

Conclusion

- Specification has been accepted by the Technical Module ... (27 Jan 00)
- ... checked by the Commercial Module ... (16 Feb 00)
- ... and finally agreed by the Steering Board (22 Feb 00)
- Will be published by ETSI
- IPR to be solved in parallel

CANAL+ TECHNOLOGIES

Conformance & Interoperability

Summary

DVB-MHP

TV & @

Products

Roadmap

Conclusion

- An Experts Group and an User group are being organised for interoperability and conformance considerations
- Conformance process based on self-certification
- Test suite will be selected and maintained by Experts Group for conformance assessment by implementers
- Interoperability problems will be reported by the User Group to DVB

CANAL+ TECHNOLOGIES

DVB-MHP KEY DRIVERS

- *Cable TV goes digital*
- *Terrestrial TV goes digital*
- *Satellite TV continues expanding*
- *Free-To-Air operators go interactive*
- *Content providers want to deliver their enhanced-TV channels on several networks*
- *Interactivity to become a source of revenue*

(t-commerce to be part of e-commerce)

- *From a Pay-TV sub or FTA viewer to an interactive consumer*

CANAL+ TECHNOLOGIES

DVB-MHP / ATSC DASE

- *Java*
  - *Rich commonalities between ATSC-DASE and DVB-MHP Java APIs, due to common parts :*
    - common subset of JavaTV®,
    - JMF APIs,
    - HAVi® ui APIs,
    - some of the DAVIC APIs (awt, media and resource notification)
  - *functional deltas lie in APIs in the org.dvb and in the org.atsc namespaces*
    - application management, transport protocols, applications launching, user preferences, SI regional specificities
  - *additional DVB APIs*
    - event model (applications can receive events even when out-of-focus),
    - Object Carousel API
    - Davic APIs : MPEG, MPEG section, tuning, Conditional Access, DVB Locator

CANAL+ TECHNOLOGIES

DVB-MHP / ATSC DASE

- ***HTML***
  - ***both solutions based on XML which provides better grounds for interoperability***

    (not stabilised enough at the time of the writing to draw a delta line)
- ***System***
  - ***deltas in :***
    - comprehensive security model in DVB (based on certificates)
    - transport (in DVB, DSMCC Object Carousel mandatory in DVB, MPE optional)
    - signalisation (SDF versus AIT)
    - application lifecycle (defined for HTML in DVB)

CANAL+ TECHNOLOGIES

CONCLUSION

CANAL+ TECHNOLOGIES

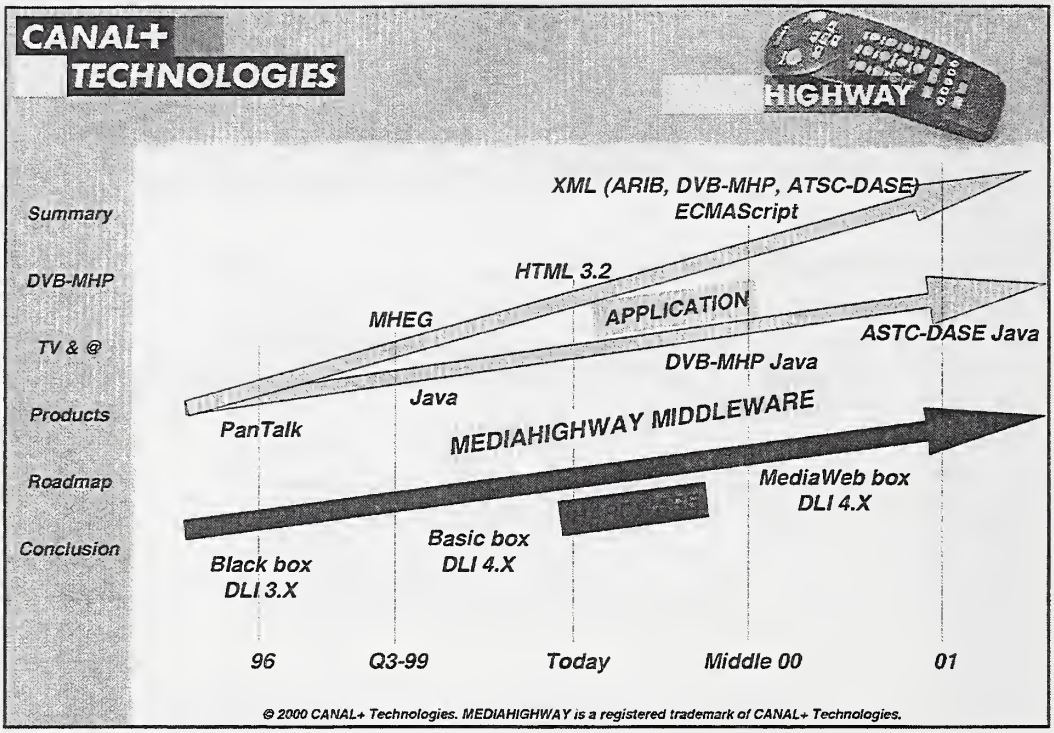
PARTNERSHIPS AND AVAILABILITY

- Adopted by CE manufacturers
- Sony and Philips have announced that they will deliver DVB-MHP compliant digital TV terminals running CANAL+ TECHNOLOGIES' middleware
- Availability : DVB-MHP, fall 2000

CANAL+ TECHNOLOGIES

CONCLUSION

- Keep tracking on Standards
- Suitable solution for DigitalTV and Internet Merging
- Will include Enhanced features : time shifting, VOD, home network, ...


CANAL+ TECHNOLOGIES
HIGHWAY
Summary
DVB-MHP
TV & @
Products
Roadmap
Conclusion
XML (ARIB, DVB-MHP, ATSC-DASE)
ECMAScript
HTML 3.2
MHEG
APPLICATION
ASTC-DASE Java
DVB-MHP Java
Java
PanTalk
MEDIAHIGHWAY MIDDLEWARE
MediaWeb box
DLI 4.X
Basic box
DLI 4.X
Black box
DLI 3.X
96
Q3-99
Today
Middle 00
01
© 2000 CANAL+ Technologies. MEDIAHIGHWAY is a registered trademark of CANAL+ Technologies.

The Use of Lid: and Tv: URIs

Craig A. Finseth

Firwood Consulting
<craig@firwood.net>

This presentation describes how DASE applications (both PE and AEE) use URIs to identify their resources and how these resources are accessed.

It begins by listing the common types of resources accessed and the typical lid: and tv: URI forms used to identify those resources. The presentation then shows how these accesses are performed from both PE and AEE environments. Finally, it summarizes the status of the URI forms.

The material in this presentation is likely to change between this submission and the conference. Updated versions of these slides will be available for download after the conference.

The Use of Lid: and TV: URIs

Craig A. Finseth
Firwood Consulting

DASE Conference
23-24 May 2000, Gaithersburg, MD, US

Contents

What These URIs Do

- URIs in general are used to identify/name resources
- lid: and tv: URIs are used to do this within the context of a television broadcast

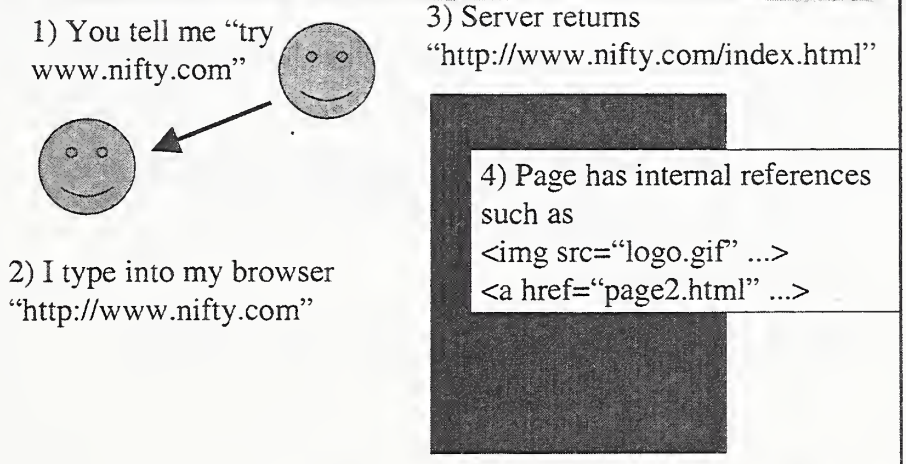
Which Resources?

- Quick list
  - Virtual Channels (tv:abc.com)
  - Events (Programs) (lid://startrek.com, lid://startrek.com/the-cage, lid://upid.smpte.com/1234566/789/123/456)
  - Applications (lid://applics-r-us.com/applA)
  - "Data Files" (lid://applics-r-us.com/dataA)

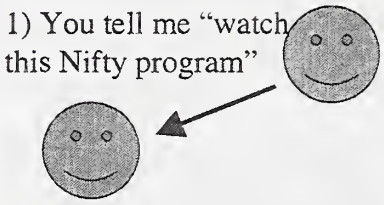
Which Resources (cont'd)?

- Full list:
  - transmission multiplex
  - virtual channels
  - events (anything in the EIT)
  - applications
  - data modules (files)
  - anything in the DIT
  - ...anything in PSIP

URI Usage on the Web



URI Usage for PE



2) Program's SDT has this application:
"lid://www.nifty.com/index.html"

3) PE loads the page

4) Page has internal references such as
<img src="logo.gif" ...>
<a href="page2.html" ...>

Differences Between Web and PE

- On web, I have to type the URI. On TV, it runs automatically.
- On web, uses http: (usually). On TV, it uses lid: (usually).
- DTDs will differ
- But overall, these two are very similar.

URI Usage for AEE

- Just like any other URI
- *This and next subject to change!*
- Opening a data module:
  - Locator = Locator.createLocator("lid:...")
  - if (Locator.getType() == data_module) {
    - cf = Locator.getCarouselFile();
    - stream = CarouselFile.open();
    - ...
  - ...

URI Usage for AEE (cont'd)

- Tuning a channel:
  - Locator = Locator.createLocator("tv:...")
  - if (Locator.getType() == channel) {
    - ch = Locator.getATSCChannel();
    - ch.tune();
    - ...
  - ...

Transmission

- Scheme is transport-independent
- Each transport must have a mechanism defined to carry the URIs and the information required to map them to the correct resource.

Status

- Overall approach is stable and hasn't changed significantly for over six months
  - Will start push for Internet standards track shortly
  - No significant opposition (we hope!)
- Efforts to define carriage
  - T3/S8
  - SMPTE
- Not yet begun
  - MPEG (may build off SMPTE)
  - DVB
  - ...

Reference

- "Uniform Resource Identifiers for Television Broadcasts", Mark Vickers, Dan Zigmond, 02/21/2000. draft-zigmond-tv-url-04.txt
- "The Local Identifier (lid:) URI Scheme", C. Finseth, Dean Blackketter, Dan Zigmond, Gomer Thomas, Michael Dolan,02/24/2000. draft-blackketter-lid-00.txt
- "Guide to TV Broadcast URLs", C. Finseth, Gomer Thomas, 3/28/2000. draft-finseth-guide-01.txt
- http://www.finseth.com/~fin/uri
  - has background, will have updated version of these slides after the conference
- Questions? Contact me at craig@firwood.net

ATVEF and DASE: Opportunity for Harmony

Patrick Griffis

Director, Worldwide TV Standards and Strategy
Microsoft Corporation
<pgriffis@microsoft.com>

The Advanced TV Enhancement Forum (ATVEF) was an industry-driven initiative to jumpstart the development of standards for creation of interactive television content. It emerged in response to the "Tower of Babel" effect occurring at the time wherein a variety of incompatible platform approaches were emerging each requiring different type of interactive content. Many in the content community saw this pattern as a non-starter. The concept of ATVEF was to use the interactive tools/technology from the Internet and combine them with traditional analog as well emerging digital television to create a common denominator catalyst for the expansion of interactive television content. The initial group completed a standard over a year ago and has sunset with over 80 plus adopters around the world. Many of these adopters have already begun deployment of content and products. This presentation will review the history of ATVEF, its marketplace situation and relevant importance to the ATSC DASE activity. ATVEF has a very complementary role to play with the ATSC DASE activity and in fact, the DASE Presentation Engine team has been charged with developing a harmonization strategy approach for ATVEF.

NIST DASE Symposium 2000

DASE and ATVEF: Opportunity for Harmony

Patrick D. Griffis
Director
Worldwide TV Standards & Strategy
Microsoft Corporation
May 25, 2000

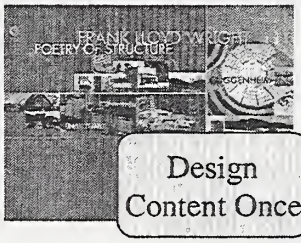
Goals of this Presentation

- Background on ATVEF and Marketplace reality
- Why is ATVEF important to ATSC?
- Related Efforts
- Opportunities for Harmony with DASE

Advanced TV Enhancement Forum (ATVEF) Objectives

- Develop specification that defines layered protocols used for enhancing television programming
  - For analog and coming digital spectrums
- Encourage broad industry adoption of spec with necessary licensing
- Build critical mass in industry leading geographies, accelerate market development
- Enable international coalescence of standards

Solving the Standards "Problem"



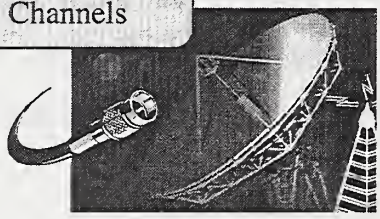
Broadcast content from widespread Web content, tools, & expertise

Design Content Once

Display Universally

Set top boxes, digital/analog TV's, PC's

Multi-distribution Channels



Analog & digital - cable, satellite, terrestrial

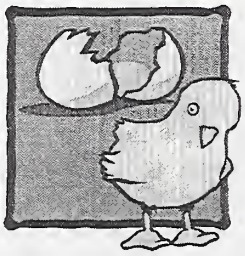
Jumpstarted by Alliance of Leading Companies

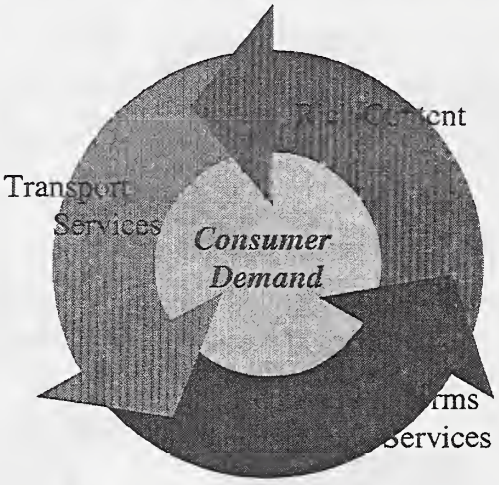
Based on Existing Standards

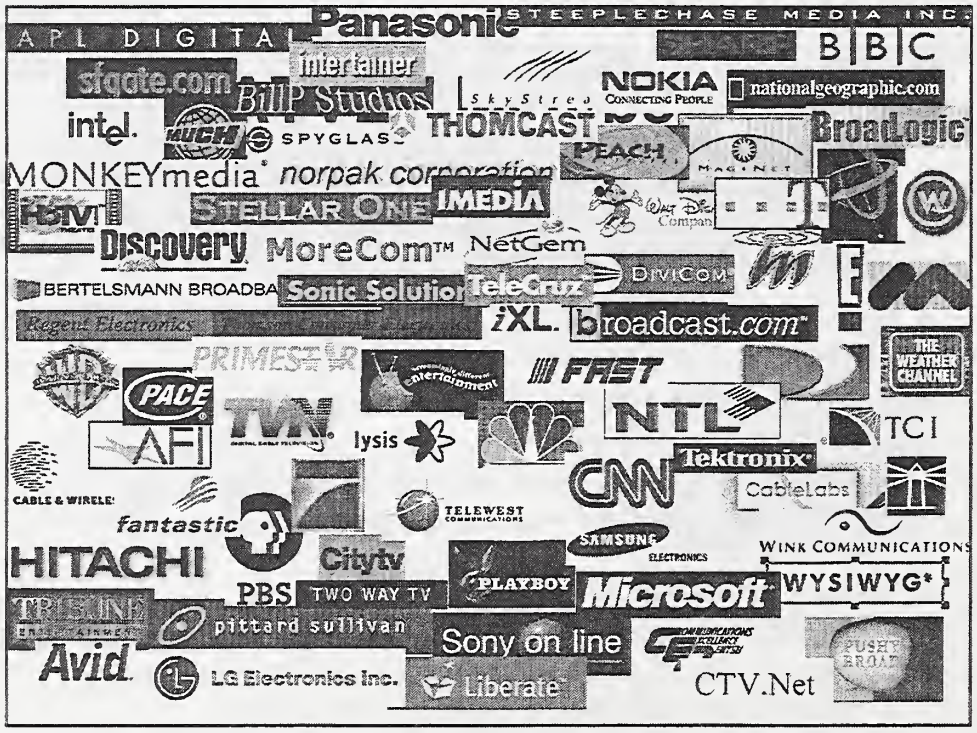
Fully Web-Compatible, with Broadcast Extensions

- Built on current standards
  - **HTML 4.0, EcmaScript 1.1, Multicast IP**
- Low Cost: Existing Authoring Tools support this platform
  - **Same tools for authoring Web content**
  - **Programming development can begin immediately**
  - **Tools evolve with W3C foundational standards, assuring rapid availability**
- Standards bodies alignment
  - **OS and platform independent**
  - **Supports spectrum of existing & developing infrastructures**
  - **Migrates with next generation W3C HTML version**

Accelerating the market







Data-Enhanced TV is here today-in NTSC

- 350+ hours/week
- Top three syndicated shows: Judge Judy (on going), Wheel of Fortune, Jeopardy, started in USA last year
- NBC Interactive football with interactive player statistics
- MSNBC and Weather Channel on 24/7