Graphics Software Architecture for High-End Interactive Television Receivers

Pablo Cesar
Helsinki University of Technology
Finland

pcesar@tml.hut.fi
http://www.tml.hut.fi/~pcesar
Outline

• Introduction
  – Device Chaos

• Background
  – Graphics Architectures
  – Content Authoring Formats
  – Digital Television
  – Related Work

• Research Questions

• Thesis

• Results / Contributions

• Conclusions
Introduction: Device Chaos

- Variety of multimedia devices
  - Desktop (PCs)
  - Non-desktop (STB)
- Number of networks
  - Broadcast
  - Mobile
- Diversity of Content
  - MPEG-2
  - Java
  - XML based
### Background: Graphics Architectures (1/3)

<table>
<thead>
<tr>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Higher-level Tools</td>
</tr>
<tr>
<td>Toolkit</td>
</tr>
<tr>
<td>Windowing System</td>
</tr>
<tr>
<td>Operating System</td>
</tr>
</tbody>
</table>

- Myers definition of a layered architecture
- Applications implemented using higher-level tools
- Toolkit: a library of widgets used by applications
- Windowing System: helps user to monitor and control different contexts (input and output functionality)

- Digital Television as Non-WIMP User Interfaces (based on Green and Chorianpoulos findings):
  - Temporal based nature (Multimedia)
  - Embedded System
  - Hypermedia capabilities
Scene Based Architectures

- Can be considered as a subset of windows architectures
- Implies the performance of one task at the time
Background: Graphics Architectures (3/3)

- **Digital Television Architecture:**
  - Three overlapping layers
  - Using transparencies (alpha channel)
  - Background + Video + Graphics

- **Graphics Layer:**
  - Use of video objects
  - Use of 3D graphics objects
  - Use of 2D graphics objects
Background: Content Formats (1/2)

Multimedia (Heller/Purchase)
- Modality: aural and visual
- Nature of the sign: concrete iconic (photorealistic image), abstract iconic (map), symbolic (written word)
- Syntax / Arrangement: spatial and temporal

Interaction (Aleem)
- Passive: only visualization
- Reactive: limited interaction (e.g., Scroll Pane functionality).
- Proactive: choose a path or make selections (e.g., Button).
- Reciprocal: corresponds to user authoring of information

<table>
<thead>
<tr>
<th>DVD</th>
<th>Broadcast TV</th>
<th>Web Surfing</th>
<th>Interactive TV</th>
<th>Video Games</th>
<th>Multimedia Authoring</th>
<th>Face-to-Face Communications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level of Interaction</td>
<td>One</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Zero One

Broadcast TV

Face-to-Face Communications
Background: Content Formats (2/2)

Requirements

- Supported Media Types: audio, video, text, graphics, and animation
- Arrangement of the signs: spatial and temporal
- Interaction: passive, reactive, proactive, and reciprocal
- Difficulty to use (threshold)
- Expressional power (i.e., ceiling)
- Safety of Distribution
- Interoperability

<table>
<thead>
<tr>
<th></th>
<th>Threshold</th>
<th>Ceiling</th>
<th>Interoperability</th>
<th>Safety of Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiled Languages</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>VM Languages</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Markup Languages</td>
<td>+</td>
<td>+</td>
<td>+++</td>
<td>+++</td>
</tr>
</tbody>
</table>
## Background: Digital Television (1/3)

### Worldwide Application Environment

- **Procedural Env.:** GEM (DVB-J)
  - Basic Java (e.g., java.lang)
  - Java TV: applications lifecycle
  - JMF: audio and video
  - HAVi: 2D User Interface
- **Declarative Env.: Undecided**
  - XHTML: structure of document
  - CSS: look
  - Media types: images, video
  - TV Specific extensions: for example remote control interaction
  - Needs scripting for handle events (broadcast, user)

<table>
<thead>
<tr>
<th>Region</th>
<th>Execution Env.</th>
<th>Presentation Env.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe (DVB)</td>
<td>DVB–J</td>
<td>bridge DVB–HTML</td>
</tr>
<tr>
<td>USA (ATSC)</td>
<td>ACAP–J</td>
<td>bridge ACAP–X</td>
</tr>
<tr>
<td>Japan (ARIB)</td>
<td>ARIB–AE (STD–B23)</td>
<td>BML (STD–B24)</td>
</tr>
<tr>
<td>ITU</td>
<td>GEM</td>
<td>ITU J.202</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITU J.200</td>
</tr>
</tbody>
</table>
Background: Related Work (1/2)

- Most relevant research groups:
  - Blendo ([http://www.blendomedia.com](http://www.blendomedia.com)): Markup Language based on VRML
  - 3DTV ([http://3dtv-research.org](http://3dtv-research.org)): capture, transmission, and display of 3D TV
  - PISTE ([http://piste.intranet.gr](http://piste.intranet.gr)): animated 3D content and television broadcast
  - SAMBITS ([http://www.irt.de/SAMBITS](http://www.irt.de/SAMBITS)): MPEG-4 player integration in receiver
- Based on MPEG-4, but little interoperability with MHP/GEM
Background: Related Work (2/2)

• Commercial systems
  – Europe (ADB, Philips, Thomson...)
    • Procedural Environment, DVB-J: 2D graphics and video
    • Declarative Environment: commercial browsers that claim DVB-HTML compatibility (e.g., Espial, Ortikon)
  – Tivo (http://www.tivo.com): Personal Video Recorder
  – MSNTV (http://www.webtv.com): Internet browsing in the TV
  – Game Consoles:
    • XBox can access the TV stream

• Open Source Projects
  – LinuxTV (http://www.linuxtv.org): provides applications and drivers for creating your own set-top box
  – MythTV (http://www.mythtv.org/): Personal Video Recorder
  – OpenMHP (http://www.openmhp.org): MHP stack
Research Questions

1. What are the limitations of current digital television receivers implementations?
2. How the GEM standard can be extended, so it defines a Declarative Environment?
3. How the GEM Standard can be extended, so it includes support for all kinds of multimedia objects (2D, 3D, and video) and WWW access?
4. How manufacturers, following GEM, can differentiate their products?
5. What graphics architecture is needed to implement GEM and the extensions proposed in the Thesis? Is this architecture valid for other media stations?
Pablo Cesar
Graphics Software Architecture for High-End Interactive Television Terminals

Thesis: Publications

P1 (2001): general study on the system software for DTV
P2 (2001): description of a system model to run different DTV applications
P3 (2002): Graphical User Interface Toolkit for DTV
P4 (2002): portable SMIL player running on different devices
P5 (2003): possible use case scenarios for SMIL in DTV
P6 (2003): description of two services transmitted over a DTV network
P7 (2004): definition of an XML based profile for developing interactive multimedia services (SMIL+XForms)
P8 (2004): layered graphics architecture for non-desktop devices
P9 (2005): extension to P8 with detailed study on graphics architectures alternatives. Complete reference implementation
Chapter 1: Introduction
(Big Picture of multimedia systems)
- Multimedia Devices
- Convergence of Networks
- Content Authoring Formats
- Graphics Architecture

Chapter 2: Digital Television
(Overview of DTV Environment)
- Architecture
- Standards (DVB, MHP, OCAP, ARIB...)
- State of the Art
- A Look Towards the Future

Chapter 3: Content Authoring Formats
(Definition of a worldwide App. Env.)
- Compiled Languages (C)
- Virtual Machine Languages (Java)
- Markup Languages (XML)
- Application Environment for DTV

Chapter 4: Extensions to GEM
(Proposals to Enhance GEM)
- Related Work
- 3D Graphics Support
- Audio and Video Support
- WWW Convergence

Chapter 5: Graphics Architecture
(Device Side Implementation)
- Window Based
- Toolkit Based
- Scene Based
- Proposed Model

Chapter 6: Reference Implementations
(Concept Implementation)
- Otadigi: DTV Broadcast System
- Ubik: Configurable DTV Receiver
Results: Graphics Architecture

- Lack of pointing device
- Usage: One task at the time
- Screen composed of multimedia objects
- Seamless integration of video, 2D and 3D objects
- A layered architecture, so developers can implement at any level

- HAL: renders the final graphics output
- Graphical Context: cross-platform abstraction of the rendering region
- Graphical Environment: means to control different contexts
- GUI Toolkit: "ready-made" user interface widgets
- HLL: to develop simple services
Results: Graphics Architecture Validation

- Hardware Abstraction Layer
- Graphical Context
- Game Console
- Graphic Environment
- GUI Toolkit
- Proprietary API
- High Level Languages
- DVB−MHP
- AWT and DVB Classes
- HAVi
- DVB−HTML
Results: Digital Television Profiles (1/2)

**Broadcast: Basic**
- Requirements:
  - Television usage: sit on the sofa and watch the favourite show at a given time
  - Catch majority of the population
- Proposed profile:
  - Audio Visual Content Digitised
  - Examples: DVB-T/C/S (Europe), ISDB (Japan), ATSC (USA)

**Broadcast: Enhanced**
- Requirements:
  - Interactive applications with internal logic
  - Procedural environment: Java
- Proposed profile:
  - DVB-J (as part of GEM, worldwide accepted version of MHP 1.0)
## Results: Digital Television Profiles (2/2)

### Interactive: Basic
- **Requirements:**
  - Browse simple XML documents
  - Limited navigation
  - Images, text
- **Proposed profile:**
  - XHTML Basic + CSS

### Interactive: Internet Access
- **Requirements**
  - Browse XML Compound Documents (complex)
  - Interactive
  - Temporal Synchronisation
  - Video and audio
- **Proposed profile:**
  - XML Compound Documents (SMIL + XForms)

### Interactive: High End
- **Requirements**
  - Integration of all kinds of multimedia objects
  - Temporal Dimension
- **Proposed profile:**
  - DVB-J + OpenGL Java Wrappers
  - JMF media player (all kind of formats)
Results: Reference Implementation (Ubik)

- DVB-T reception and visualisation of the A/V stream
- HAL: DirectFB and OpenGL
- Resource Manager: to control the different processes
- Graphical Context: SDL
- Graphical Environment: SDLAWT (java.awt)
- GUI Toolkit
  - FTV: 2D widgets
  - JMF: other video than A/V
  - Java OpenGL: 3D Graphics
- HLL: X-smiles (XHTML, SMIL, XForms)
Pablo Cesar
Graphics Software Architecture for High-End Interactive Television Terminals

Applications
- XML Documents
- DVB–J
- Java–OpenGL
- SDL Games
- Channel Tuning
- Stop/Play Stream
- Native UI

System Software Architecture
- X–Smiles XML User Agent
  - Extended GEM Environment
    - FTV
    - JMF
    - TVGL
  - SDLAWT
- SDL
- Device Dependent Libraries
  - DirectFB
  - OpenGL Drivers
  - FusionSound
  - video
  - audio
  - libavcodec
  - libsoftmpeg
  - LinuxTV
  - ALSA / OSS
- Matrox 550
- Keyboard
- Nova–t PCI
- Sound Card

Legend
- Home–built
- 3rd Party (modified)
- 3rd Party
Results: Ubik HAL

- Example composition of scenes:
  - 3D graphics object
  - A/V Stream
- Performance
  - Around 60 FPS
Results: Ubik Graphical Context

• Example native 3D Graphics
  – Some games downloaded from a portal
Results: Ubik Graphical Environment + GUI Toolkit

- Java Applications
- Profile:
  - Broadcast Enhanced
- Target:
  - Procedural Environment Applications (internal logic)
- Languages:
  - DVB-J
- Interaction:
  - Colour Buttons
  - HAVi Widgets
- Multimedia Objects:
  - Images/Text
  - Animations
  - Video/audio
Results: Ubik High Level Languages (1/2)

- XML Based Document
- Profile:
  - Interactive Television Basic
- Target:
  - Information Services (e.g., Teletext)
- Languages:
  - XHTML Basic
- Interaction:
  - Navigation
- Multimedia Objects:
  - Images
  - Text
Results: Ubik High Level Languages (2/2)

- XML Based Document
- Profile:
  - Interactive Internet Access
- Target:
  - Complex Applications (e.g., E-learning Portal)
- Languages:
  - SMIL (or XHTML+Timesheets)
  - XForms
- Interaction:
  - Buttons/Selections...
  - Navigation
- Multimedia Objects:
  - Images/Text
  - Video/Audio
## Conclusions (1/2)

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the limitations of current digital television receivers implementations?</td>
<td>WWW Convergence, Streamed media and 3D graphics</td>
</tr>
<tr>
<td>How the GEM Standard can be extended, so it defines a Declarative Environment?</td>
<td>Procedural (GEM), declarative (XHTML, CSS, continuous media objects, TV extensions such as remote control)</td>
</tr>
<tr>
<td>How the GEM Standard can be extended, so it includes support for all kinds of multimedia objects (2D, 3D, and video) and WWW access?</td>
<td>GEM procedural extensions (Java bindings of OpenGL), Declarative (XHTML + XForms + SMIL syntax)</td>
</tr>
</tbody>
</table>
Conclusions (2/2)

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>How manufacturers, following GEM, can differentiate their products?</td>
<td>For example, game console convergence</td>
</tr>
<tr>
<td>What graphics architecture is needed to implement High-End Interactive Television receivers? Is this architecture valid for other media stations?</td>
<td>Five layers proposed model: HAL, Graphical context, Graphical Environment, GUI Toolkit, and HLL</td>
</tr>
</tbody>
</table>