Open Graphical Framework for Interactive Television

Pablo Cesar, Juha Vierinen, and Petri Vuorimaa
{pcesar, jvierce, pv} @tml.hut.fi
Telecommunications Software and Multimedia Laboratory
Helsinki University of Technology
Finland

Outline

• Introduction
• Background
• Proposed Graphical Software Architecture
• Digital Television
• Prototype (Ubik)
• Screenshots
• Conclusion
• Questions

pcesar@tml.hut.fi
December 11th 2003
Introduction

• “Interacting in Chaos” (1999, by Olsen):
  – “The fundamental assumptions of interactive computing must be reconsidered”
  – Device chaos: ”variability of future interactive devices”
• “Past, present, and future of user interface tools” (1999, Myers et. al):
  – “We are at the dawn of an era where user interfaces are about to break out the “desktop” box”
• In conclusion
  – Not so long ago, desktop computer the only multimedia platform
  – Today, the number of interactive devices is growing (e.g., mobile phones, digital television receiver)
  – So, a new user interface model is needed

Classic Architecture

• ”User Interface Software Tools” (1995, Myers) defines a layered model
• Each layer includes one more level of abstraction
• 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)
System Architecture Requirements

• User Experience
  – Support for multimedia objects (video, image...)
  – User interaction effectively handled

• Developer Experience
  – Need a set of APIs as easy as possible to use to develop the software

• Core Architecture
  – Control different contexts
  – Synchronise audio, video, and applications

• Adaptability
  – Cross platform development

User Interface Software Tools Metrics

• “Past, present, and future of user interface tools” (1999, Myers et. al) proposed several metrics to evaluate user interface software tools
• In this presentation, two of them are widely used:
  – Ceiling: expressional power of a tool
  – Threshold: difficulty of learning the tool
• They are really useful because for example it makes clear that 3D games should not be implemented the same way as Information Systems
• In conclusion:
  – Device chaos + basic requirements
  – Classic architecture + user interface software tools metrics
Proposed Model (1/2)

- Different Layers, which can be use for development (image upside down):
  - HAL: Hardware Abstraction Layer
  - CPLib: Cross-platform graphics library
  - Graphical Environment
  - GUI Toolkit
  - High Level Languages

Proposed Model (2/2)

- Hardware Abstraction Layer
  - Provides a unified way to access the hardware (e.g., access video memory)
- Cross-platform graphics library
  - Provides multimedia primitives for presentation, user interaction, and video and audio
- Graphical Environment
  - Manages the different contexts
- GUI Toolkit
  - A set of “ready-made” user interface widgets
- High Level Languages
  - Fast and easy development of applications
Digital Television (1/5)

**Analog tv:**
1 television program in a 6 or 8 MHz channel  
Little choices  
Receiver: only Television set

**Digital tv:**
Audiovisual signal is compressed and transported by digital means (MPEG-2 stream).  
Spectrum efficiency. Multiple programs in 6 or 8 MHz.  
Interactive Services: VoD, banking, games...  
Receiver: TV set + decoder.

Digital Television (2/5)

- Each Broadcaster decides how to use its own bit-rate:  
  - Only television programs  
  - Only interactive applications  
  - Television program(s) plus interactive application(s)
- Flexible quality of video and audio signals  
  - 3 Mbps is enough for satisfactory video quality.
- Interactive Applications as value added services  
  - Java Xlets and XHTML documents  
  - Internet access (e-mail, web, ...)

pcesar@tml.hut.fi  
December 11th 2003
Digital Television (3/5)

- Standards = Horizontal Market
  - Applications and decoders can be developed by different developers and manufacturers.
- DVB Standards (Digital Video Broadcasting)
  - DVB is the European initiative in Digital Television field
  - Specifies the transmission in different mediums: DVB-S (Satellite), DVB-C (Cable), DVB-T (Terrestrial)
- MHP (Multimedia Home Platform)
  - Set of Java Application Program Interfaces (APIs)
  - Home/Audio Video Interoperability (HAVi) is the GUI library. It is based on a subset of Abstract Windowing Toolkit (AWT)
  - Every DTV receiver include them

Digital Television (4/5)

- Resident applications
  - Navigator
    - Display current program information.
    - Allow control over the decoder settings.
    - Display forthcoming information, Electronic Program Guide.
  - Digital Teletext
    - Mass media information delivery (news, sport results...).
    - Information structured in pages.
    - Contains multimedia objects (audio, images, animations...)
- Broadcast applications
  - All kind of services: VoD, banking, games...
Digital Television (5/5)

- The proposed model can be used for digital television
  - AWT + MHP classes
  - HAVi
  - DVB-HTML
- Hardware Abstraction Layer breaks Java interoperability
- Other High Level Languages not taken into account yet

Ubik

- A prototype system for digital television receivers based on the proposed model
- Open and extensible system
- MHP compliant, overcoming today’s restrictions (interaction channel capabilities, video player implementation…)
- Allows the study of new ideas for MHP
- It is based on Linux Operating System
- Combine different relevant open source projects
Ubik Layered Model

- **Hardware Abstraction Layer**: DirectFB
- **Cross Platform Libraries**: SDL
- **DVB-MHP**: SDLAWT
- **Graphical Environment**: SDLAWT
- **GUI Toolkit**: HAVi (FTV)
- **High Level Languages**: SMIL

---

**Hardware Abstraction Layer**

- Uses DirectFB: [http://www.directfb.org](http://www.directfb.org)
- Renders directly in the framebuffer console, so no extra windowing system is needed (e.g., X Windows)
- Includes hardware acceleration
- Supports controlling different applications in the same screen (e.g., audio-visual content + MHP service)
- Easy to develop a video player for the audio-visual content
Cross Platform Multimedia Libraries

- Uses Simple DirectMedia Layer (SDL): http://www.libsdl.org
- SDL can be compiled in a variety of Operating Systems:
  - Linux, Palm OS
- SDL provides native multimedia primitives:
  - Presentation (visualisation)
  - User Interaction
- Other libraries used in Ubik:
  - SDL_image: support for images
  - SDL_ttf: support for true type fonts

Graphical Environment (1/2)

- Ubik includes Kaffe 1.1 Virtual Machine:
  - http://www.kaffe.org
  - Open source Virtual Machine
- AWT was modified (Java Native Interface bindings), so it renders using SDL library
- Final result is an AWT subset (only essential classes to manage graphics) specified by HAVi:
  - Toolkit, Graphics, Font, FontMetrics, Image, Color…
  - Shape, Polygon, Rectangle, Point..
  - Control of Windows (Component, Container)
  - Event handlers and Listeners (java.awt.event)
Graphical Environment (2/2)

- The actual screen is defined in *Toolkit* as a *SDLSurface*
- *Scene* is a context (like *awt.Window*)
- Scene includes *Components*
- Each *Component* is drawn by its *Graphics* class

---

**Toolkit**

- All widgets are visible, in addition they could be:
  - Navigable: navigated using the remote control
  - Actionable: launch functionality (buttons)
  - Selectable: selection of an item or set of items (Lists)
- Each widget has associated a *Look* class (i.e., view)
High Level Languages

- eXtensible Markup Language (XML) is a meta-language to define other languages
- World Wide Web Consortium (W3C) has defined different languages, two of them:
  - Synchronized Integration Markup Language (SMIL), multimedia presentations on the Web (presentation, time based)
  - XForms, structured web forms (interaction)
- A Language profile SMIL+XForms hybrid documents are integrated in Ubik
- Advantages:
  - Easy and fast to implement (no compilation needed)
  - Small applications (only a document + multimedia objects)
  - Structured and interchangeable content
Screenshots Navigator

Channel Guide

<table>
<thead>
<tr>
<th>Time</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>07:00</td>
<td>Vaara Vuoteessa</td>
</tr>
<tr>
<td>08:30</td>
<td>Kulissien takana</td>
</tr>
<tr>
<td>09:50</td>
<td>Nu talar pappa igen</td>
</tr>
<tr>
<td>11:00</td>
<td>Esko Oinas lavoilla</td>
</tr>
<tr>
<td>12:30</td>
<td>Entertainers '03</td>
</tr>
<tr>
<td>13:15</td>
<td>Pelastusvene</td>
</tr>
</tbody>
</table>


Screenshots Teletext

Super Teletext

MAIN INDEX

- News
- Weather
- TV Guide
- Finance
- Shopping
- Documentaries
- Travel
- Film

GOTO  BACK  HOME  HELP

YLE: 18:00 - 21:30
European Cities, Helsinki
Otadigi

- New multiplex (2002) for academic and research purposes in the Helsinki University of Technology campus area ([http://www.otadigi.tv](http://www.otadigi.tv))
- DVB-T network
- Stable environment
- Reception by commercial set-top boxes
- In Otadigi, different MHP applications are running using our HAVi widgets:
  - Navigator
  - SMIL player
- Real network can study today’s digital television

pcesar@tml.hut.fi
December 11th 2003
Conclusions

- Today situation: Device Chaos = growing number of multimedia terminals
- A Graphical Software Architecture for multimedia platforms has been presented
- Multiple configurations can be obtained (e.g., MHP, game console)
- A prototype for digital television receivers based on the model has been implemented (Ubik)
- Ubik proposes two extension profiles for MHP (low level and high level)
- Otadigi is a real DVB-T network, where Ubik can be tested
Ubik Future

- DVB-T reception (DirectFB video player)
- Java Video Player (use of JMF)
- Communication mechanism such as Bluetooth
- Deeper study of the interaction channel: heterogeneous network support
- SIP client (e.g., Video conference)
- Inclusion of minimal XML parsers

Questions

Thank you very much!

Questions ?
Comments ?