A Device Independent XML User Agent for Multimedia Terminals

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

Outline

• Introduction/ Problem
• Background
  – SMIL / XForms
  – Digital Television
• Overview of the Paper
• Requirements
  – Language Profile
  – UI Compatibility
• XML User Agent Implementation
• Case Study
  – Screenshots
  – Evaluation
• Conclusion & Future Work
Introduction

• “Interacting in Chaos” (1999, by Olsen):
  – 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, Interactive Multimedia Software running on different devices
    will be a consumer expectation in the near future

Problem

• Development of Cross-Platform Interactive Multimedia Applications
  – Multimedia Objects: audio, video, images, fonts
  – User Interaction
  – Temporal Dimension (synchronisation of objects)
• Distribution of multimedia applications using higher abstraction languages: XML based
• XML user Agent: Java most interoperable option available
• Still, Java has different APIs depending on the platform: Component Factory is needed
A Device Independent XML User Agent
for Multimedia Terminals

Background: SMIL and XForms

- XML based languages: easy to write, like HTML
- Recommendations by World Wide Web Consortium

SMIL
- Language intended for multimedia presentations
- Defines spatial dimension of the document (layout)
- Defines temporal dimensions of the document (synchronisation)
- User input included as links
- Does not define media formats, only integrates them

XForms
- Next generation of web forms
- Not intended as a self-standing document type
- Needs of a host language to provide the document layout (e.g., XHTML, SMIL)
- User input includes: text input, select one, select many, submit
- User input can be validated in the client-side

Background: Platforms

- In this paper, we consider three different platforms:
  - PC: Java 2 Standard Edition (J2SE)
  - Handheld device: Java 2 Micro Edition (J2ME)
    - High End: Connected Device Configuration (CDC) (personal Java) e.g., Nokia Communicator
    - Low End: Connected, Limited Device Configuration (CLDC), mobile phones
  - Digital Television Receivers: (J2ME, CDC)

<table>
<thead>
<tr>
<th></th>
<th>PC</th>
<th>Handheld</th>
<th>Handheld</th>
<th>DTV Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>High End</td>
<td>Low End</td>
<td></td>
</tr>
<tr>
<td>UI Library</td>
<td>Swing</td>
<td>AWT</td>
<td>MIDP</td>
<td>HAVi</td>
</tr>
<tr>
<td>Input</td>
<td>Keyboard, Mouse</td>
<td>Keypad, stylus</td>
<td>Keypad</td>
<td>Remote control</td>
</tr>
<tr>
<td>Output</td>
<td>1800x1440</td>
<td>240x320</td>
<td>176x192</td>
<td>640x496</td>
</tr>
</tbody>
</table>

pcesar@tml.hut.fi
December 13th 2004
Background: Platforms (DTV)

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.

• 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 13th 2004
Background: Platforms (DTV)

- Standards = Horizontal Market
  - Applications and decoders can be developed by different developers and manufacturers.
- DVB Standards (Digital Video Broadcasting)
  - Specifies the transmission in different mediums: DVB-S (Satellite), DVB-C (Cable), DVB-T (Terrestrial)
- MHP (Multimedia Home Platform)
  - Platform independent middleware between applications and receivers
  - Every DTV receiver include them
  - Defines to languages: DVB-J and DVB-HTML
  - DVB-J includes different packages: Home/Audio Video Interoperability (HAVi) is the GUI library

Overview of The Paper

- Application Layer:
  - XML languages used for implementing interactive multimedia applications
- Platform Layer:
  - User Agent supports the language profile
  - UI Compatibility Layer (Component Factory)
  - Specific Device configuration
Requirements: Language Profile

- Temporal Dimension
- Spatial Layout
- Multimedia Objects Support
  - Continuous
  - Discrete
- User Interaction
  - Links
  - Validated Entry
  - Submission (server-side)

Requirements: UI Compatibility

- Platform uses Java
- Most platform independent option
- The actual APIs for user interface development differ between multimedia terminals:
  - Digital TV: HAVi
  - PC: Swing
  - Handheld: MIDP
## Requirements: Summary (XML Profile)

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal Dimension</td>
<td>SMIL</td>
</tr>
<tr>
<td>Spatial Layout</td>
<td>SMIL (no flow layout)</td>
</tr>
<tr>
<td>Multimedia Object Support</td>
<td></td>
</tr>
<tr>
<td>Continuous Media</td>
<td>SMIL</td>
</tr>
<tr>
<td>Discrete Media</td>
<td>SMIL</td>
</tr>
<tr>
<td>User Interaction</td>
<td></td>
</tr>
<tr>
<td>Links</td>
<td>SMIL</td>
</tr>
<tr>
<td>Validated Entry</td>
<td>XForms</td>
</tr>
<tr>
<td>Submission</td>
<td>XForms</td>
</tr>
</tbody>
</table>

## Requirements: Summary (Platform)

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Different Libraries</td>
<td>Java + UI Compatibility Layer</td>
</tr>
<tr>
<td>Different Input Mechanisms</td>
<td>Java + UI Compatibility Layer</td>
</tr>
<tr>
<td>Different Output Devices</td>
<td>CSS, XSL transformations, or automated content adaptation</td>
</tr>
<tr>
<td>Different Capabilities</td>
<td>CSS, XSL transformations, or automated content adaptation</td>
</tr>
</tbody>
</table>
Implementation: SMIL

SMIL player divided into three layers:
- Core Logic handles timing, layout, hyperlinks, etc.
  - GUI independent
- Viewer Layer draws the user interface
  - Swing, AWT, HAVi GUI versions
- Media Players play media files
  - text, images have custom players, audio, video by JMF
- SMIL Player also integrated into X-Smiles
  - Download it at: www.xsmiles.org
A Device Independent XML User Agent for Multimedia Terminals

Implementation: XForms

XForms engine divided into three layers

- **XForms model:**
  - XML parsers (Xalan, Xerces), validation and calculation engine

- **Meta UI:**
  - Implementation of repeating user interface constructs and switching parts dynamically

- **User Interface:** Implementation of form controls
  - High Level: logic implementation, not aware of the actual details
  - Low Level: actual implementation (using a component factory)

- **XForms also integrated into X-Smiles**
  - Download it at: [www.xsmiles.org](http://www.xsmiles.org)

---

**Diagram:**

```
User interface
  Swing  AWT  Havi
  ComponentFactory

Meta UI
  Form Controls  Adaptive Controls
  Actions  Switch  Repeat

XForms model
  Calculation Engine  Validation
  Xalan Xpath  Xerces Schema
```

pcesar@tml.hut.fi
December 13th 2004
Implementation: Component Factory

- Generic user interface API
  - Defines abstract widgets (e.g., XInput)
  - Maps those widgets with its actual implementation depending on the device at hand
- Currently three back-ends: AWT (Personal Java), Swing (PC), and HAVi (Digital TV Receiver)
- Benefits:
  - Application is not aware of the specific user interface library in used
  - Developing new back-ends is easy
### Implementation: Component Factory

<table>
<thead>
<tr>
<th>XForms Control</th>
<th>AWT Widget</th>
<th>Swing Widget</th>
<th>HAVi Widget</th>
</tr>
</thead>
<tbody>
<tr>
<td>select1 &amp; select</td>
<td>List</td>
<td>JList</td>
<td>HListGroup</td>
</tr>
<tr>
<td>trigger</td>
<td>Button</td>
<td>JButton</td>
<td>HTToggleButton</td>
</tr>
<tr>
<td>submit</td>
<td>Button</td>
<td>JButton</td>
<td>HTToggleButton</td>
</tr>
<tr>
<td>label</td>
<td>TextField</td>
<td>JTextField</td>
<td>HStaticText</td>
</tr>
<tr>
<td>textarea</td>
<td>TextArea</td>
<td>JTextArea</td>
<td>HText</td>
</tr>
<tr>
<td>input + xsd:string</td>
<td>TextField</td>
<td>JTextField</td>
<td>HSingleLineEntry</td>
</tr>
<tr>
<td>input + xsd:date</td>
<td>TextField</td>
<td>JCalendar</td>
<td>HSingleLineEntry</td>
</tr>
<tr>
<td>input + xsd:boolean</td>
<td>CheckBox</td>
<td>JCheckBox</td>
<td>HToggleButton</td>
</tr>
</tbody>
</table>

### Implementation: Platform - Ubik

- **Platform Layer (Ubik)**
  - XForms + SMIL Support
  - Component Factory
  - HAVi (FTV)
  - Basic AWT
  - Underlying Software

- **XML User Agent**
  - MHP

- **Application Layer**
  - Application 1
    - Doc
    - Objects
  - Application 2
    - Doc
    - Objects

- **Platform Layer**
  - XML User Agent
    - XML Language Profile Implementation
    - UI Compatibility Layer
    - Digital TV configuration
      - HAVi
      - Basic AWT
      - Underlying Software
    - PC configuration
      - Swing
      - AWT
      - Underlying Software
    - Handheld configuration
      - MIDP
      - Underlying Software
Implementation: Platform - Ubik

- A Linux based prototype system for configurable digital television receivers
- Based on MHP standard, hence it allows us the study of new ideas for interactive television
- Study how different programming languages should be used for different purposes:  
  - Native: 3D graphics demanding applications  
  - Java: complex services  
  - XML: information services + simple multimedia applications

Implementation: Platform – Ubik (HAVi)

- 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)
Case Study: Distance Education Portal

- Author
  - Non-programmer
  - Very easy to produce the content
- User
  - Student accessing from a remote terminal
- Content
  - Attractive multimedia content
- Synchronisation:
  - Internal: video content of the lectures
  - External: slides changing as the lecturer talks
- User Interaction:
  - Navigation: Normal links
  - Exam: submission and validation controls

Case Study: Screenshots

- MITA
  - Please Select a topic:
    - Web Services
    - Quality of Service

- Mid-Term Exam
  - 10 secs
  - Options:
    - Which is the correct VLAN protocol?
    - 1. 802.1g
    - 2. 802.11
    - 3. 802.11b
    - 4. 802.11a
    - Wi-Fi
    - What is the range of a Wi-Fi network?
Case Study: Evaluation

- Includes synchronisation mechanism
  - Cannot be achieved using only HTML
- Includes validation and calculation mechanisms of user input
  - Cannot be achieved using only SMIL
- Application completely developed using declarative languages (SMIL+XForms)
  - Easy to author: no scripting is used
  - Quick to author: number of lines for the whole application is around 256

Conclusions

- Device Independent XML based user interface model
  - Presentation (SMIL): separates the synchronisation logic from user interface
  - User Interaction (XForms): separates the widgets from their look and feel
  - Component Factory: cross-platform user interface support
- Case Study shows the benefit of the approach
  - easy and fast to develop interactive multimedia applications
- Ubik Environment: prototype of configurable digital television receivers
  - DTV stream: audio, video, composite graphics (transparencies)
  - 3D Graphics support (OpenGL, Java OpenGL): games, basic 3D applications
  - MHP support: services such as Teletext or Navigator
  - X-smiles: support for XML based languages such as XHTML 2.0, SMIL, XForms
- X-smiles have been ported to commercial device:
  - Nokia communicators as a high end handheld device

pcesar@tml.hut.fi
December 13th 2004
Future Work

- SMIL; temporal dimension can be included using Timesheets
  - Similar to CSS, but time
  - Easier to implement
  - Less troublesome
  - Separated time from layout (could use flow layout)
  - Embedded into XHTML
- Some XForms functionality could not be implemented (e.g., switch and repeat)
- In depth study of how the system should react against:
  - Different capabilities (J. V. Ossenbruggen et al.)
  - Different output (J. Smith et al.)
- Study of MIDP (too different from AWT!!)

Questions

Thank you very much!

Questions?

Comments?