XML-Driven Device Independent User Interface

*Build Rich Client Applications Using XML*

(Final Presentation)

Tassanee Er-Jongmanee
244641

**Advisors**

Dr. Ralf Klamma
Dipl. Inform. Mohamed Amine Chatti
Lehrstuhl für Informatik V
Dipl. Ing. José M. de la Rosa
Dipl. Inform. Renate Stutz
Semantics GmbH

April 2005
Agenda

- Fundamentals
  - Problem scenario and motivation
  - Classification for multi-platform interface implementation

- Specification, Design and Implementation
  - Model-Based Design
  - XML technologies
  - System architecture
  - Evaluation

- Conclusions and future work
Problem Scenario and Motivation

- Accessing application via other devices
- How to implement multi-platform user interfaces considering productivity and usability?
- XML Driven with Model-Based Design for multi-platform UI
- Estimation of Implementation Effort

Devices and Platforms

Design and Development

How to develop User Interfaces?

Kind of users
“Booker” Prototype Application

- Add new button
- Query window

April 05 XML-Driven Device Independent User Interface
Classification of Multi-Platform Implementation [FIVa04]

- Specific design and interface for each platform
- Single design and interface for any platform
- Scalable technique based on the largest screen device

Create user interface in multiplatform

XML description languages with Model-Based UI Design (MBUID)

Single design for common part and extend for specific platform
# User Interface Modeling Tools

## Declarative Models available in Model-Based-IDEs

<table>
<thead>
<tr>
<th>Model</th>
<th>Task</th>
<th>User</th>
<th>Supported multi-platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collagen [DK02]</td>
<td>++</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>MASTERMIND [Szekely96]</td>
<td>++</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Mobi-D [Puerta96a]</td>
<td>++</td>
<td>++</td>
<td>--</td>
</tr>
<tr>
<td>ConcurTask Tree [Ctte03]</td>
<td>++</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>FUSE [Lonczewski96]</td>
<td>++</td>
<td>++</td>
<td>--</td>
</tr>
<tr>
<td>TADEUS [Elwert94,95]</td>
<td>++</td>
<td>++</td>
<td>--</td>
</tr>
</tbody>
</table>

++ support model
+  support by extending model
-- not support model
ConCurTask Trees

- Focus on Activities
- Hierarchical Structure
- Graphical Syntax
- Rich set of temporal operators
- Objects and task attributes

```xml
<Task Identifier="Login" Category="abstraction" Iterative="false">
  <Name>Login</Name>
  <Type>null</Type>
  <Description/> <Description>
  <Platforms>Pda</Platforms>
  <Platforms>Desktop</Platforms>
  <Platforms>Cellphone</Platforms>
  <Preconditions/></Preconditions>
  <TemporalOperator name="SequentialEnabling"/>
  <TimePerformance>
    <Max></Max>
    <Min></Min>
    <Average></Average>
    </TimePerformance>
  <Parent name="Booking Model"/>
  <SiblingRight name="Add list"/>
</Task>
```
Model User Interface with CTT

- Usable and Multiplatform UI
- Better requirement capturing, consistent and detailed interface design.

ConCurTask Tree (CTT)
Fabio Paternò[Ctte03]
XML-Based Model Design Approach

Task model (e.g., CTT) → Logical HCI design (XML with abstract description) → Physical HCI design (XML with concrete description) → Final Interface (Target language)

Platform → Transformation from Model to XML with device independence → Transformation to specific device → Rendering in target device

GUI PC → GUI PDA → GUI Handy

UI Design and Implementation [Uiml02]
## XML Description Languages

<table>
<thead>
<tr>
<th></th>
<th>TERESA [Tere04]</th>
<th>UIML3.0 [Uiml02]</th>
<th>AUIML [AzMR00]</th>
<th>XIML [XIM03]</th>
<th>XUL [Xul03]</th>
<th>XAML [Xaml04]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extensibilities</strong></td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Functionalities</strong></td>
<td>-</td>
<td>UIML</td>
<td>Java</td>
<td>XIML</td>
<td>Java Script</td>
<td>C#</td>
</tr>
<tr>
<td><strong>Script</strong></td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Portability</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Platform-Independent</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Desktop</td>
</tr>
<tr>
<td><strong>Interface Definition</strong></td>
<td>Tools available</td>
<td>Tools available</td>
<td>Tools available</td>
<td>Hand code</td>
<td>Tools available</td>
<td>Tools available</td>
</tr>
<tr>
<td><strong>Render</strong></td>
<td>Render to other form</td>
<td>Render to other form</td>
<td>Render to other form</td>
<td>Render to other form</td>
<td>Render embedded</td>
<td>Render embedded</td>
</tr>
</tbody>
</table>
Compared Criteria of Developing XML Description

- **Productivity**
  - Easy to create
  - Use less time
  - Extensibility

Comparing characteristics of selected approaches

- UIML
- XUL
- AUIML
Integration of UI into System Architecture

Client Devices

Network TCP/IP, HTTP

Request

Application server (JBOSS)

XML Files

UIML, XUL, AUIML

UIML, XUL, AUIML

Query & result

JDBC

Connection

MS SQL Server

Query data (session bean)

Schedule (entity bean)

Schedule detail (entity bean)

Customer

Schedule

Schedule Details

Booker Database

JBOSS Server

<datasources>
  <jndi-name>MSSQLDS</jndi-name>
  <connection-url>
    jdbc:jtds:sqlserver://localhost:1433/Booker
  </connection-url>
  <driver-class>net.sourceforge.jtds.jdbc.Driver</driver-class>
  <user-name>admin</user-name>
  <password>password</password>
  <metadata>
    <type-mapping>MS SQLSERVER2000</type-mapping>
  </metadata>
</datasources>
User Interface Markup Language (UIML)

<?xml version="1.0" encoding="UTF-8"?>
<uiml>
  <head> ... </head>
  <interface>
    <structure> ... </structure>
    <style> ... </style>
    <content> ... </content>
    <behavior> ... </behavior>
  </interface>
  <presentation> ... </presentation>
  <logic> ... </logic>
  <peers>
    <logic> ... </logic>
    <presentation> ... </presentation>
  </peers>
</uiml>

Widget level:<part class="JButton" id="submit"/>
Generic level:<part class="GenericButton" id="submit"/>
Abstract UI level:<part class="Login" id="submit"/>

for Java Swing
transformed automatically to
Java/Swing vs. Java/SWT vs. C++/GTK+
transformed automatically to combination of
GenericLabels & GenericButton

Platform widgets

Connect to target UI toolkit (e.g., Java Swing, HTML4)

Rules control behavior of interface

Connect to applications, services, content sources

Application API

Presentation style for each part (e.g., font size, color)

Content for each part (e.g., text, images, video)
**User Interface Markup Language (UIML)**

### Generic part elements in interface

```xml
<interface>
  <structure>
    <part id="LoginFrame" class="G:TopContainer">
      <part id="LoginTitle" class="G:Label"/>
      <part id="LoginField" class="G:TextField"/>
      <part id="Password" class="G:Label"/>
      <part id="PwdField" class="G:TextField"/>
      <part id="LoginButtonPanel" class="G:Area">
        ..........
      </part>
    </part>
  </structure>
  <style id="Java">
    <property name="rendering" part-name="LoginFrame">JFrame</property>
    <property name="title" part-name="LoginFrame">Booker Model Agency</property>
    ..........
  </style>
</interface>
```

### Style interface using Java, JFrame Window and Option attribute

### Define event and method for component reference by id

### Available UIML Vocabularies

<table>
<thead>
<tr>
<th>Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic Vocabulary</td>
</tr>
<tr>
<td>W3C's HTML v.4.01 with the frameset DTD and CSS Level 1</td>
</tr>
<tr>
<td>Java 2 SDK (J2SE) v1.3, specifying AWT and Swing toolkits</td>
</tr>
<tr>
<td>VoiceXML Forum's VoiceXML v1.0</td>
</tr>
<tr>
<td>WAP Forum's Wireless Markup Language (WML) v1.3</td>
</tr>
</tbody>
</table>
XML-Based User Interface Language (XUL)

- Interface Definition (XUL/JavaScript)
- Render with Mozilla browser

```
<window title="Booke Applicat" xmlns:html="http://www.w3.org/1999/xhtml" xmlns:
<script type="application/javascript" src="/global/content/dialog4overla">
<optgroup id="groupbox" orient="horizontal">
<caption id="caption" label="Login to Booker">
<ybox id="ybox1"/>
<hbox id="hbox1">
 <label id="label" value="Login:" control="login-text"/>
 <textbox id="login-text"/>
 </hbox>
<hbox id="hbox2">
 <label id="label" value="Password:" control="password-text"/>
 <textbox type="password" id="password-text"/>
 </hbox>
<hbox id="hbox3">
 <button id="xm:button1" label="OK" style="width: 90px; flex: 1"/>
 <button id="xm:button3" label="Cancel" style="width: 90px; flex: 1"/>
 </hbox>
</ybox>
</hbox>
</groupbox>
</window>
```
Luxor XUL – XUL to Java Project

- Construct XUL files
- Translate to Luxor class and mapping to action
- Call Luxor’s class and action
- Render to Java interface

**Luxor Classes**
- Read XUL files
- Translate XUL description to Swing elements
- Action listener are setup

**Java Code**
- Call Luxor’s classes and interface for create Swing components
- Implement action associated to “action” tags
Abstract User Interface Markup Language (AUIML)

```auiml
<AUIML VERSION="AUIML 1.2">
  <DATA-GROUP NAME="Login">
    <CAPTION>
      <META-TEXT NL:SRC="resource://Login.TEXT"/>
    </CAPTION>
    <GROUP NAME="Group1">
      <CAPTION>
        <META-TEXT NL:SRC="resource://LoginGroup1.TEXT"/>
      </CAPTION>
      <STRING NAME="LoginName" BINDING="Client.auiml.LoginBean#LoginName">
        <CAPTION>
          <META-TEXT NL:SRC="resource://Login.LoginName.TEXT"/>
        </CAPTION>
      </STRING>
      <STRING NAME="Password" BINDING="Client.auiml.LoginBean#Password" OBLIGATED="TRUE">
        <CAPTION>
          <META-TEXT NL:SRC="resource://Login.Password.TEXT"/>
        </CAPTION>
      </STRING>
    </GROUP>
    <ACTION NAME="LoginOK">
      <CAPTION>
        <META-TEXT NL:SRC="resource://LoginLoginCK.TEXT"/>
      </CAPTION>
    </ACTION>
    <ACTION NAME="Cancel">
      <CAPTION>
        <META-TEXT NL:SRC="resource://Login.Cancel.TEXT"/>
      </CAPTION>
    </ACTION>
  </DATA-GROUP>
</AUIML>
```
Abstract User Interface Markup Language (AUIML)

- Define window as DataBean
- Invoke UserTaskManager
- Load AUIML file
- Render interface

```java
public void render(String mode) {
    String auimlFile = "Client.auiml.Booker";
    String group = "DeckPanePanel";
    try {
        ListDataBean lstBean = new ListDataBean();
        LoginBean logBean = new LoginBean();
        UpdateBean uptBean = new UpdateBean();
        lstBean.load();
        logBean.load();
        uptBean.load();
    }
}
```
Evaluation

- User test
- Questionnaire

Environment
Tools & Devices
Task
User
Context of use
Product
Efficiency
Qualitative Satisfaction
Usability measures

Use Satisfaction
Learnability, Efficiency, Memorability, Few errors, Satisfaction

Productivity, Extensibility, Portability

Number of components available

- AUIML has the best of user satisfaction and good feedback after testing a prototype

Comparison of three approaches by user satisfaction

- AUIML
  - Number of components supported (Java Swing Render)
  - Support tool: Under develop
  - Number of code lines: 2832
  - Man days: 30
  - Ratio (code lines / man days): 94.4

- XUL
  - Number of components supported
  - Support tool: XUL maker
  - Number of code lines: 2548
  - Man days: 23
  - Ratio (code lines / man days): 110.7

- UIML
  - Number of components supported
  - Support tool: Visual builder
  - Number of code lines: 1455
  - Man days: 15
  - Ratio (code lines / man days): 97
### Evaluation (cont.)

<table>
<thead>
<tr>
<th></th>
<th>Pro</th>
<th>Con</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIML</td>
<td>- provide abstract definition</td>
<td>- has to render to specific platform</td>
</tr>
<tr>
<td></td>
<td>- easy to add new interface definition</td>
<td>- emphasis only on abstract interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- editor tool is under developed</td>
</tr>
<tr>
<td>XUL (Luxor)</td>
<td>- has a good renderer and many projects support</td>
<td>- slower than AUIML</td>
</tr>
<tr>
<td></td>
<td>- rich components available</td>
<td></td>
</tr>
<tr>
<td>AUIML</td>
<td>- has a great tool support in developing UI</td>
<td>- has to render to specific platform</td>
</tr>
<tr>
<td></td>
<td>- performance of rendering interface is good</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- Classification of multiplatform development techniques
- Comparative study of the state of the art in XML description languages and model-based tools
  - Three XML description languages which are UIML, XUL (Luxor) and AUIML are studied
  - Survey of XML renderer and editor tools of each XML approach
- Multi-platform UI realized
  - Specification and Design
  - Model based
  - XML Driver Interface description
- Evaluation and comparison of XML description languages with model-based design
Future Work

Methodology
- Adapting this concept to other clients, such as browser in various devices
- Improve design process based on other devices, such as PDAs, mobiles
- Integrate the design tool and the implementation tools

Implementation
- Java is used in this work but the approach can be implemented with other languages, such as Python, .NET, etc.
- Using J2ME (Java Micro Edition) for less capable devices

Evaluation
- Measuring usability by users on various expertise level
- Testing the approach with other media, such as MPEG, audio, etc.