Uiml.net: An Open Uiml Renderer for the .Net Framework

Kris Luyten and Karin Coninx
{kris.luyten,karin.coninx}@luc.ac.be
Limburgs Universitair Centrum
Expertise Centre for Digital Media

IUI/CADUI'2004, Funchal, Madeira, Portugal
Overview

- Introduction
- Uiml
- What's new
- .Net
- Uiml.net properties
- Rendering with Reflection
- Application Logic
- Multi-Platform User Interfaces
- Conclusions and Current Status
- Questions
Introduction

- Dygimes: our Multi-Device User Interface Creation framework
- Uses simple XML for describing form-based Uis (presentation model)
- Moving towards more pervasive environments
- A more powerful UIDL is necessary
  - Uiml is a suitable candidate
  - How flexible/reusable is it?
Uiml: User Interface Markup Language

- “A declarative, XML-compliant meta-language for describing User Interfaces...”
- Open Standard (OASIS, towards W3C Recommendation)
- One language to describe User Interface for Multiple Devices (≠ One description for multiple devices)
Uiml: User Interface Markup Language (2)

```xml
<uiml>
  <interface>
    <structure> ... </structure>
    <style> ... </style>
    <behavior> ... </behavior>
    <content> ... </content>
  </interface>
  <peers> ... </peers>
</uiml>
```
Uiml.net: What's New?

- No other current implementation of UIML 3.0
- New widget set(s) on new environment (.Net)
- As reusable and flexible as possible
- Integrates with binary application logic
- Very loosely coupled with application logic
- Free Software
.NET

- Common Language Runtime + Class Library
- Comparable to Java, but for multiple languages (> 20 languages)
- Microsoft Initiative
- Standardised through ECMA
- Multiple widget sets available
Uiml.Net Properties

- Renderer (vs. code generation)
- Reads mappings from vocabularies
  - UIML document: Abstract Interaction Objects
  - Vocabulary: Concrete Interaction Objects
- Current vocabularies: Gtk# and Wx.NET
  - Have a common subset of widgets
  - Redesign of Uis can be minimized
  - Changing output widget set = changing vocabulary
- Fast prototyping with different widget sets
Uiml.Net Properties (2)
Rendering with Reflection

- Rendering core based on reflection
- Rendering core does not know what widget set it is creating
- Depends on the mappings provided in the vocabulary
- Result: Highly reusable renderer
- Support for API Evolution
  - Only the vocabulary has to be updated
  - Vocabularies can be partial automatically created
Binding with Application logic

- UIML provides *Rules* = (condition, action)
- E.g.: `<behavior>
  <rule>
  <condition>
    <event class="ButtonPressed" part-name="do-it"/>
  </condition>
  <action>
    <call name="Math.sin">
      <param name="in">
        <property part-name="input" name="text"/>
      </param>
    </call>
  </action>
  </rule>
</behavior>`
Binding with Application logic (2)

- Two kinds
  - Standard .NET functionality
  - Other functionality specified within uiml document

- UIML 3.0 specification is unclear
  - How to bind with object instances
  - How UIML can be used from within the program code

- User Interface can be developed completely independent of application logic
  - As long as the same interface is being used
  - “Lazy linking” with the appropriate functions
Multi-platform User Interfaces

- **Generic Vocabularies**
  - Easier migration to other widget sets
  - Common set of Interactors (Button, Text, List,...)
  - Specific widgets also available

- **Common layout management**
  - Form-based
  - Traditionally specified as properties of UI parts
  - Vertical and Horizontal Boxes
  - Not flexible enough
  - Constraint-based requires complex software
Conclusion and Current Status

- Rendering Backends:
  - Gtk#: Most widgets implemented, no Tree Control
  - More complete Wx.Net backend on its way; only basic widgets now
- Binding with Application Logic
  - Direct Method Invocation
  - Remote Method Invocation and Web Services on the way
- Further integration with MBUID
- Feel free to download, try out and give feedback!
Announcement

First Workshop on
Developing User Interfaces with XML:
Advances on User Interface Description Languages

http://www.edm.luc.ac.be/uixml2004/
25th of May

in conjunction with AVI'2004, May 25-28, Gallipoli, Italy

Organizers: M. Abrams, J. Vanderdonckt, Q. Limbourg, K. Luyten
Questions, Remarks,...