UI Development Environments (UIDE) & Prototyping Tools

Lecture 8

The Problem

• Interface Programming is
  – Complex
  – Time-consuming
  – Error-Prone

• Results in
  – Low programming productivity
  – Untested products
  – Limited number of expert programmers who can write UI code

The Solution

• Abstraction
  – High-level programming languages tailored to UI
  – Object-oriented languages supporting event processing

• Environment
  – Specialized GuiBuilders

• Automation
  – UIMS
  – Geometry managers
  – Knowledge-based systems
    » UI expert systems
Tradeoffs with Abstraction

<table>
<thead>
<tr>
<th>High</th>
<th>Low</th>
<th>High</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Complexity</td>
<td>Generality &amp; Rigidity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X Window</td>
<td>Mac Toolbox</td>
<td>C++</td>
<td>JAVA</td>
</tr>
</tbody>
</table>

User Interface Implementation

- **Software Prototype only**
  - Primarily for developing and testing prototype
  - Examples
    - ??? (Macintosh)
    - Visual Basic (Microsoft Windows)
    - Tcl/Tk, QT, Suit, Garnet/Amulet (Cross-Platform)
- Index of UI Tools for Prototyping
  - [http://www.geocities.com/SiliconValley/Vista/7184/guitool.html](http://www.geocities.com/SiliconValley/Vista/7184/guitool.html)

- **Target Software Implementation**
  - Direct implementation in target language
  - Examples
    - Java, C++

Supportive Programming Environment

- **Strong support (GUI Builders)**
  - Drag and drop of widgets from inventory
  - Visual display and positioning of objects
  - Interactive sizing of object geometry
  - Programming by Demonstration
    - example: paths for animation
    - example: interactive selection of objects
- **Weak support**
  - Dialog boxes, menus
  - Structure editor
  - Debugging?
- **Specialized interface language**
- **Fast program/test cycle**
Demo of GUI Builder: Visual Tcl

- Download from
  - Available for numerous platforms

Attributes of a GUI Builder: Visual Tcl

- Strong Support
  - Selection of widgets from icon panel
  - Continuous display of constructed UI
  - Drag and drop to position widgets in window
  - Interactive sizing of object geometry
  - Display of constructed UI widget hierarchy
- Weak Support
  - Attribute editor for each widget
  - Click on widget to open command editor to bind actions to widget
- Fast program/test cycle
  - Button to change from "Edit" to "Test"
- Generates Tcl/Tk code

Visual Tcl Screen: Draw
Visual Tcl Screen: Hello world

Visual Tcl Program: Hello world

Goodbye window, button & message

Benefits and Limitations of GUI Builders

- **Benefits**
  - Improved productivity
  - Can often be used by non-programmers
  - Allow rapid program/test cycle to allow usability testing

- **Limitations**
  - Often rigid
    - Support limited types of widgets
    - Primitive semantics for actions
  - Single-pass
    - Can’t write new code into the program and run back through the GUI Builder
  - Generate very inefficient, undocumented code