



THE UNIVERSITY OF AUCKLAND
www.auckland.ac.nz

An environment for developing adaptive, multi- device user interfaces

John Grundy and Biao Yang



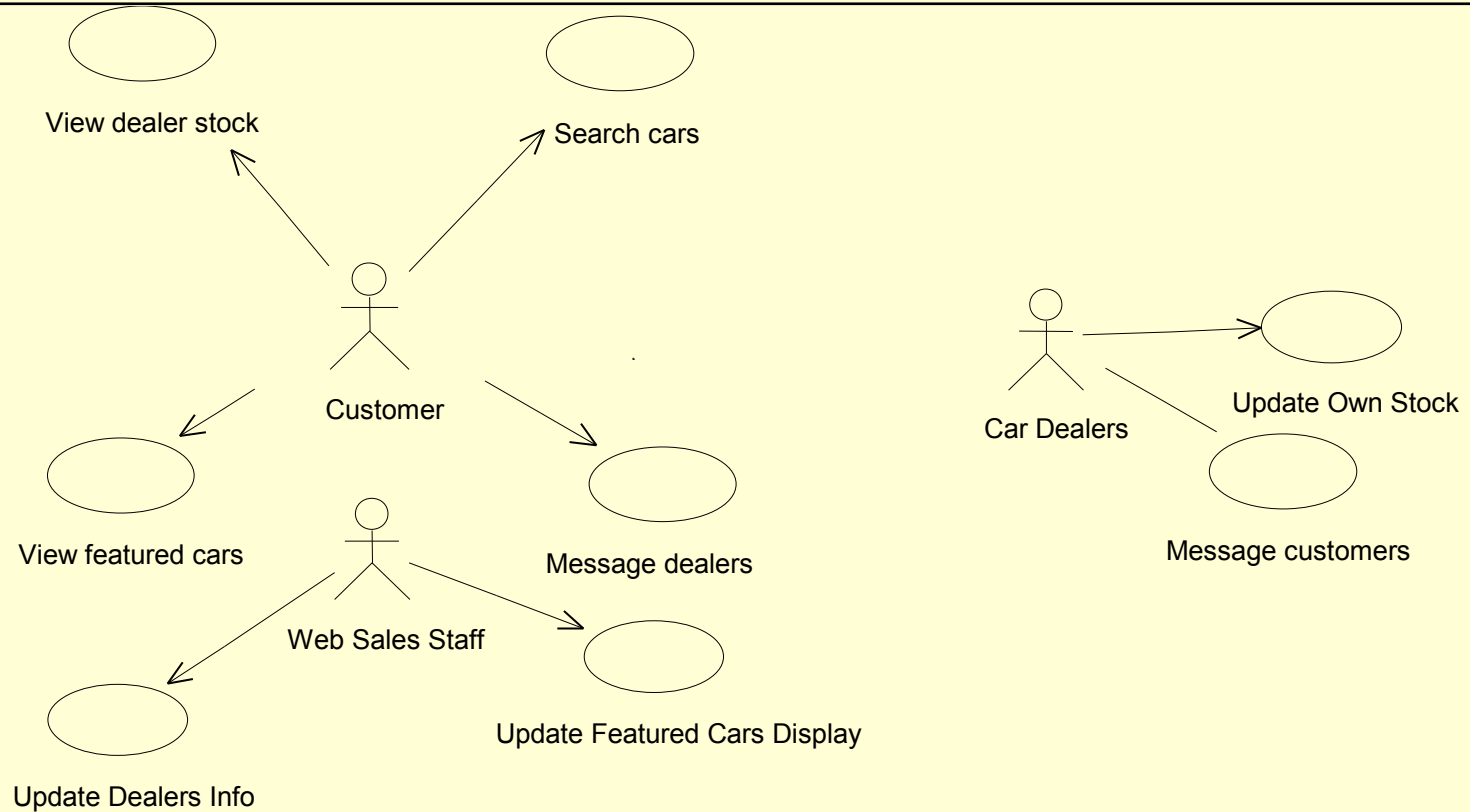
Outline

- ❖ Motivation
- ❖ Overview of Our Approach
- ❖ Tool architecture
- ❖ Examples of using design tool
- ❖ Tool implementation
- ❖ Tool evaluation
- ❖ Summary

Motivation

- ❖ Want to be able to build “multi-device Uis” (MUIs) – same interface can run on PDA, pager, mobile phone, web browser etc
- ❖ Provide toolkit to developers to make this “easier”
- ❖ Toolkit not that easy to use...
- ❖ Want design environment support for developers
- ❖ Want to generate implementation from UI design diagrams

Example: Car Website



Example of User Interfaces

The screenshot shows a web browser window titled "search_result - Microsoft Internet Explorer" with the address "http://localhost:8080/". The main content area displays a search results page for cars. The page includes a table of car listings, a search form, and a "GUARANTEED MOTORS LTD" banner. Three callout boxes are overlaid on the page:

- Callout 1 (Top Left):** Titled "Find Cars", it lists several car models with their status, year, and mileage:
 - Toyota Crolla LX Ltd, 1994, 47561kms, 8995
 - Toyota Aristo, 1992, 73417kms, 17995
 - Honda Rafaga, 1995, 68830kms, 11995
 - Toyota Camry, 1995, 58900kms, 9995
 Below the list is a "Next Card" button.
- Callout 2 (Middle):** Titled "Find Cars", it shows a search form with the following fields:
 - Select Make: [toyota]
 - Select Model: [Camery]
 - Year From [1999] to [2000]
 - Price From [10000] to [20000]
 Below the form are "Edit" and "Prev" buttons.
- Callout 3 (Bottom Right):** Titled "Find Cars", it shows a search results card with the following fields:
 - Make [toyota]
 - Model [Camery]
 - Year [1999]
 Below the card are "Edit" and "Pre..." buttons.

The background table of car listings is as follows:

Status	Year	Make	Model	Mileage	Price	Update
For Sale	1997	Nissan	Wingroad JS 4WD Wagon	86865		
For Sale	1994	Nissan	Sunny Super Touring	76865		
For Sale	1994	Subaru	Legacy TX Wagon	108964		
For Sale	1991	Nissan	Pulsar X1R 5 Dr	49113		
For Sale	1991	Nissan	Pulsar X1R 3 Dr	92102	7995	Update
For Sale	1996	Mitsubishi	Mirage Asti Coupe	106530	9995	Update

Underlying implementation

```

<%@ taglib uri="/auit" prefix="auit" %> // page directive to access AUIT tags
<jsp:useBean id="car_manager" class="car.CarManager" /> // JavaBeans to use
...
<auit:screen name="car search result" /> // sets user/task/device information...
  <auit:heading level=2 value=" Car Search Result" />
  <auit:table width=60 border=0>
    <auit:row><auit:column><auit:label width=6
      value=" Num" /></auit:column>...
    <% cars = car_manager.selectCars(...); %>
    <auit:iterator name="car" data="cars" %>
      <auit:row height=1>
        <auit:column><auit:label width=6 value=
          '<% car.getCarID() %>' /></auit:column>
        <auit:column><auit:link width=20 name='<% car.getCarID() %>'
          href="car_details.jsp?task=detail&car=
            <% car.getCarID() %>" /></auit:column>
        <auit:column><auit:label width=30 value=
          '<% car.getMake() %>' /></auit:column>
      ...
    </auit:row>
  </auit:iterator>
</auit:table>
</auit:screen>

```

❖ “AUIT”

❖ JSP custom tag library

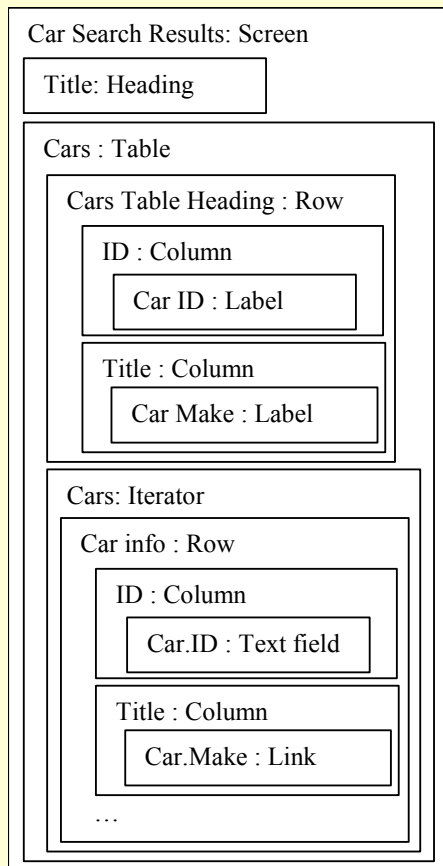
❖ Run-time adaptation

❖ Device, user and task adaptation supported...

Adaptive UI Design Issues

- ❖ Not like conventional tools e.g, VisualStudio, JBuilder – one UI design used to synthesise multiple UIs for different devices, users, user tasks, ...
 - ⇒ **Need abstract design that is used to create multiple concrete UIs...**
- ❖ Our work has shown structure-based design most useful for developers in such an environment
- ❖ Layout parts of (thin-client) adaptive UIs table-oriented
- ❖ Also want to give developer “feel” for resultant UI look and feel across devices

Example Design Sketch



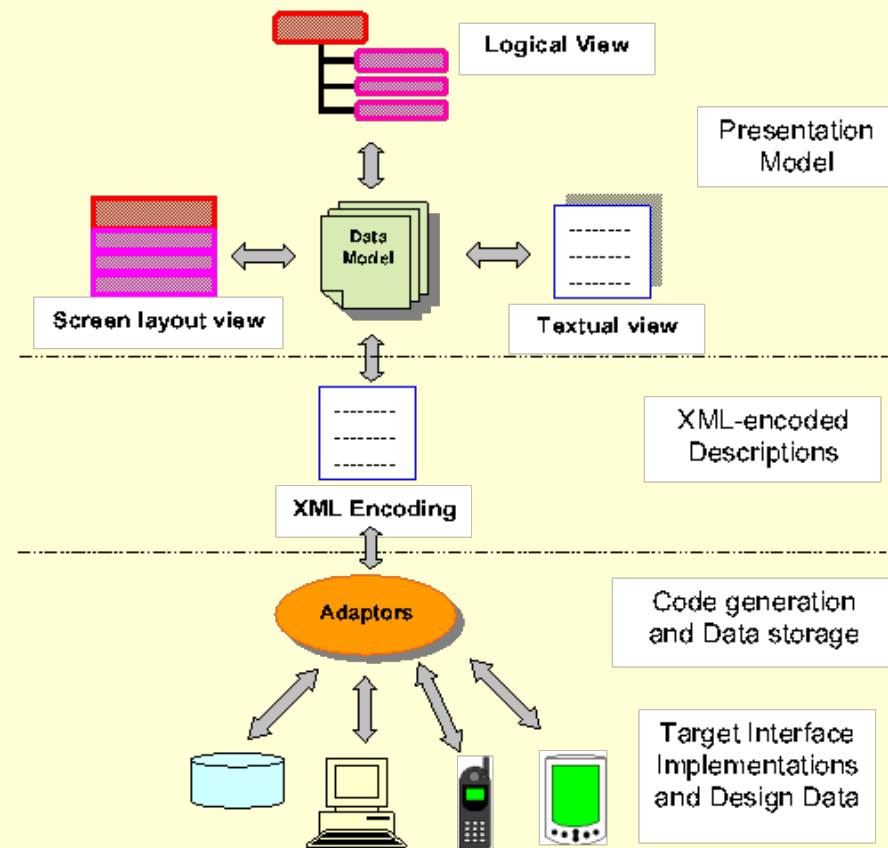
- ❖ Used to think about UI elements, composition, some idea of interaction
- ❖ UI structure-based
- ❖ AUIT synthesises MUI on-demand using requesting device characteristics, user profile/workflow (task) info...
- ❖ ...and yes, there are lots of limitations, esp. interaction, layout, simple complex->multiple simple UIs 😊

Our Approach

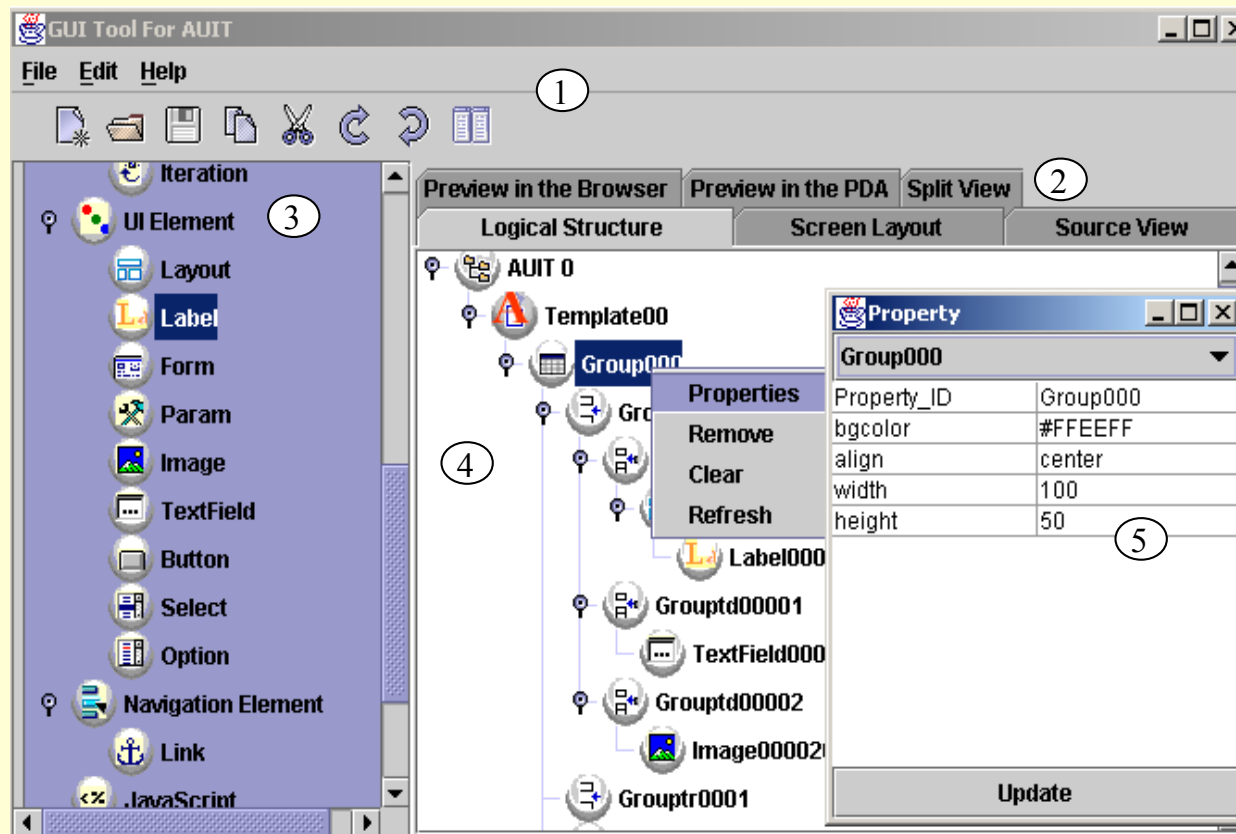
- ❖ Developed Adaptable User Interface Technology – custom tag library for JSPs to build such UIs
- ❖ Developed design environment allowing developers to specify adaptable user interfaces using three views:
 - UI element tree structure view
 - UI screen layout view
 - Textual view
- ❖ Generates AUIT (JSP) or Java Servlet implementations of the UI

Tool Architecture

- ❖ Three views share common data model
- ❖ Tree view-outline
- ❖ Screen layout-size, shape, positioning
- ❖ Textual-details
- ❖ XML encodes UI design
- ❖ Code generators produce multiple implementations of adaptive UI



Tool User Interface [I'm not brave enough for demo, sorry 😊]



Example #1: Tree Viewer

The screenshot displays the GUI Tool For AUIT interface. On the left is a TagRepository with various UI elements. The central Structure pane shows a tree view of the application components, including useBean00, Template01, Group010, Grouptr0100, Grouptd01000, Form010000, Table0100000, Device0100000, Tr010000000, Td010000000, Layout0100000000, Label010000000, Tr01000001, Td010000010, Layout0100000100, Label01000001000, Td010000011, and TextField0100000110. A Property window is open for TextField0100000110, showing its properties. A Source Editor window is also open, displaying the AUIT source code for the selected component. A red box highlights the AUIT:Label tag in the source code, which corresponds to the selected component in the tree view.

Source Editor Content:

```

<AUIT:Template bgcolor="#FFFFCC" tag_id="01">
  <AUIT:Group height="500" bgcolor="" width="100" tag_id="010" align="cent
  <AUIT:Grouptr cellheight="3" tag_id="0100">
    <AUIT:Grouptd tag_id="01000" colspan="" cellwidth="100" align="">
      <AUIT:Form name="" action="joblist.jsp" tag_id="010000" method="p
      <AUIT:Table tag_id="0100000">
        <AUIT:Device device="html" value="" tag_id="01000000">
          <AUIT:Tr tag_id="010000000">
            <AUIT:Td tag_id="0100000000">
              <AUIT:Layout color="green" bold="" size="0" face="" text=""
              <AUIT:Label allow="" text="Login" tag_id="010000000"
            </AUIT:Label>
          </AUIT:Layout>
        </AUIT:Tr>
      </AUIT:Device>
    <AUIT:Tr tag_id="01000001">
      <AUIT:Td tag_id="010000010">
        <AUIT:Layout color="green" bold="" size="0" face="" text="" ta
        <AUIT:Label allow="" text="User Name" tag_id="0100000
      </AUIT:Label>
    </AUIT:Td>
  </AUIT:Tr>
</AUIT:Table>
  </AUIT:Form>
</AUIT:Grouptd>
</AUIT:Grouptr>
</AUIT:Group>
</AUIT:Template>
  
```

Example #2: Screen Layout View

GUI Tool For AUIT

File Edit Help

Structure Layout Source Browser PDA Split View

UI Control Element

- User
- Task
- Device
- Table
- Tr
- Td
- Iteration
- UI Element
- Layout
- Label
- Form
- Param
- Image
- TextField
- Button
- Select
- Option
- Break
- Navigation Element
- Link
- JavaScript

Row:

Size:

Add

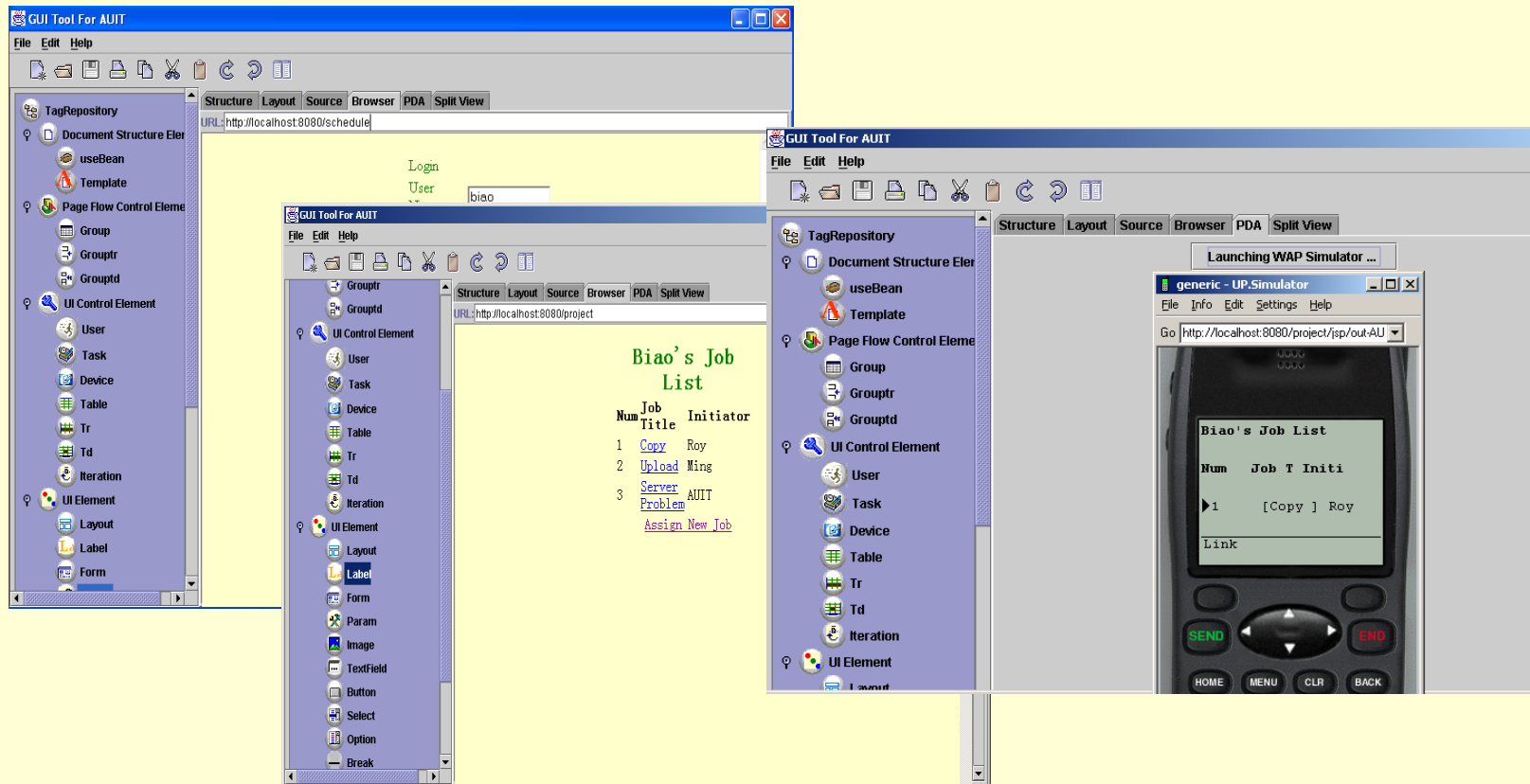
Delete

Column: Size: Add Delete

Show OK Cancel

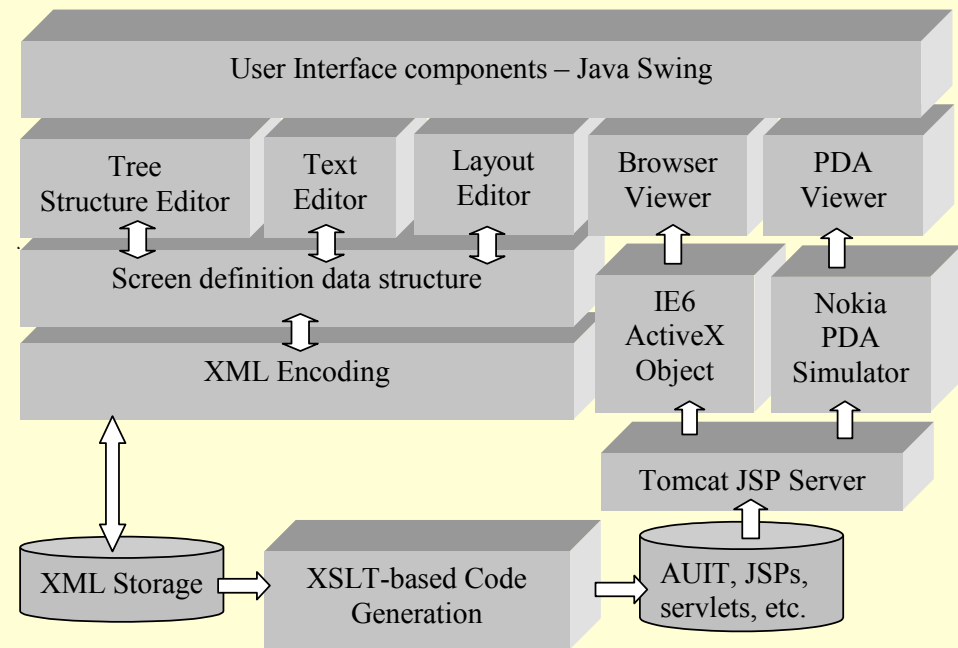
Num	Job Title	Initiator	Last Sender	Current Comment	Reassign To
1	Copy	Roy	James	Not done	Assign To
2	Upload	Ming	James	Done	Assign To
3	Server Problem	AUIT	James	Reassign	Assign To

Example #3: Running Interfaces



Implementation

- ❖ Java, Java Swing used for design environment
- ❖ XML-encoding for UI designs
- ❖ XSLT generates AUIT, Servlet code
- ❖ Embedded viewing tools



Evaluation

- ❖ Evaluated both design tool and generated adaptive user interfaces
- ❖ UI developers evaluated design environment, servlets and AUIT custom tag library for building same UI
- ❖ Users evaluated hand-coded adaptive UIs and generated AUIT and Java Servlet UIs for usability
- ❖ Design environment and generated servlet implementations rated by far the best

Summary

- ❖ Designing adaptive UIs requires different support to conventional WYSIWYG UI development tools
- ❖ Our design tool provides three mutually consistent views for such design
- ❖ Provides generation of UI implementation from XML encoding of design
- ❖ Much quicker, more accurate, round-trip development of adaptive UIs supported
- ❖ Modifications needed to simplify UI design structure

References

- ❖ Grundy, J.C., Wang, X. and Hosking, J.G. Building Multi-Device, Component-Based, Thin-Client Groupware: Issues and Experience, In Proceedings of the 2002 Australasian User Interface Conference, Melbourne, Australia.
- ❖ Grundy, J.C. and Zhou, W. AUIT: Adaptable User Interface Technology, with Extended Java Server Pages, In Cross-Platform and Multi-device User Interfaces, Wiley, 2003.
- ❖ Grundy, J.C. and Yang, B. An environment for developing adaptive, multi-device user interfaces, In Proceedings of the 4th Australasian Conference on User Interfaces, Adelaide, Australia, February 3-7 2003.
- ❖ Grundy, J.C. and Jin, W. Experiences developing a thin-client, multi-device travel planning application, in Proceedings of 2002 New Zealand Conference on Computer-Human Interaction, July 12-13, Hamilton, New Zealand.
- ❖ Cao, S., Grundy, J.C., Stoeckle, H., Hosking, J.G., Tempero, E., Zhu, N. Experiences Generating Web-based User Interfaces for Diagramming Tools, In Proceedings of the 2005 Australasian User Interfaces Conference, Jan 31-Feb 3, 2005, Newcastle, Australia, Conferences in Research and Practice in Information Technology, Vol. 40.
- ❖ Grundy, J.C., Hosking, J.G., Cao, S., Zhao, D., Zhu, N., Tempero, E. and Stoeckle, H. Experiences developing architectures for realising thin-client diagram editing tools, Software – Practice and Experience, vol. 37, no.12, Wiley, October 2007, pp. 1245-1283.
- ❖ Zhao, D., Grundy, J.C. and Hosking, J.G. Generating mobile device user interfaces for diagram-based modelling tools, In Proceedings of the 2006 Australasian User Interface Conference, Hobart, Australia, January 2006.
- ❖ Cao, S. Grundy, J.C., Hosking, J.G., Stoeckle, H. and Tempero, E. An architecture for generating web-based, thin-client diagramming tools, In Proceedings of the 2004 IEEE International Conference on Automated Software Engineering, Linz, Austria, September 20-24, IEEE CS Press, pp. 270-273.
- ❖ Abizer Khambati, John Grundy, John Hosking, and Jim Warren, Model-driven Development of Mobile Personal Health Care Applications, In Proceedings of the 2008 IEEE/ACM International Conference on Automated Software Engineering, L'Aquila, Italy, 15-19 September 2008, IEEE CS Press.