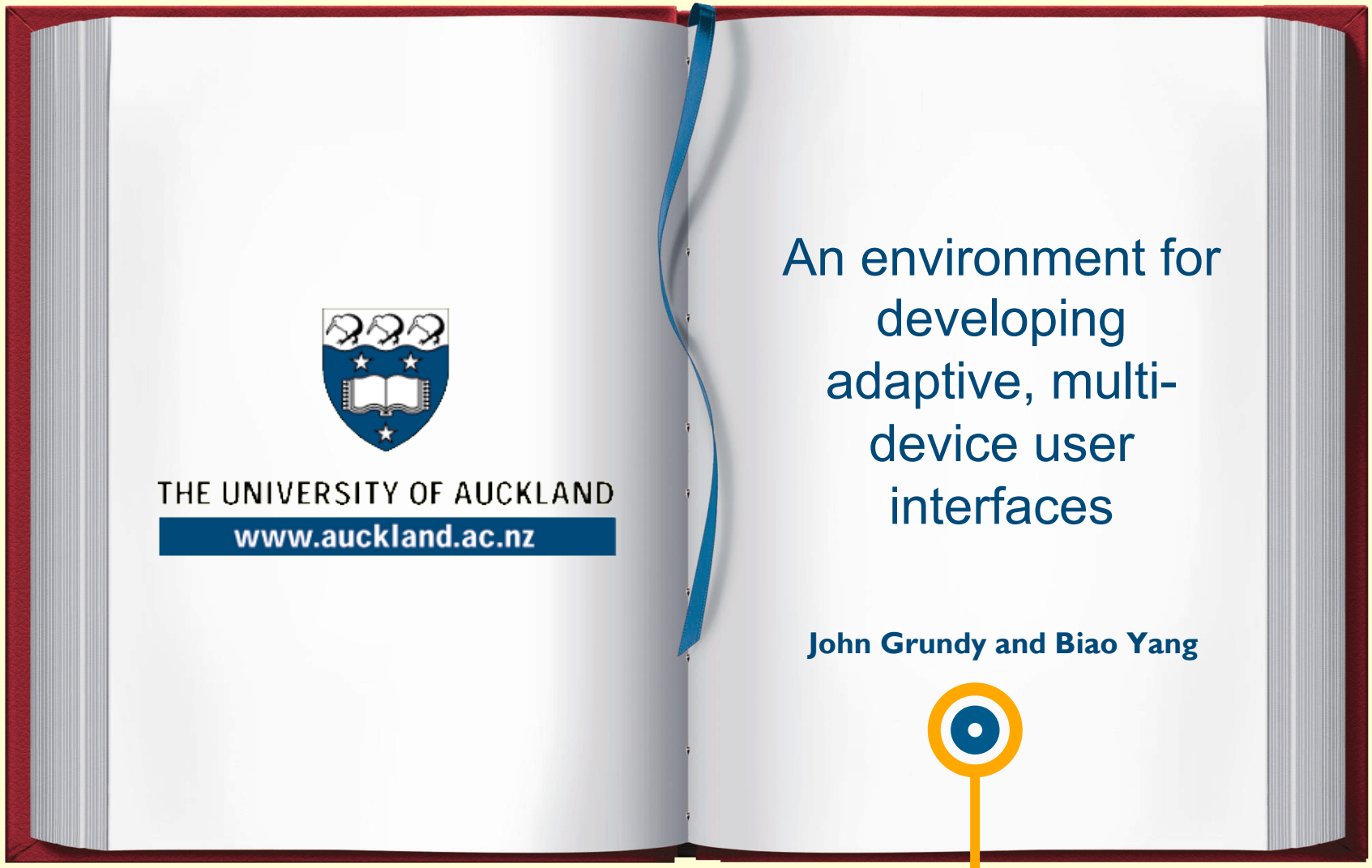


An environment for developing adaptive, multi-device user interfaces

John Grundy and Biao Yang

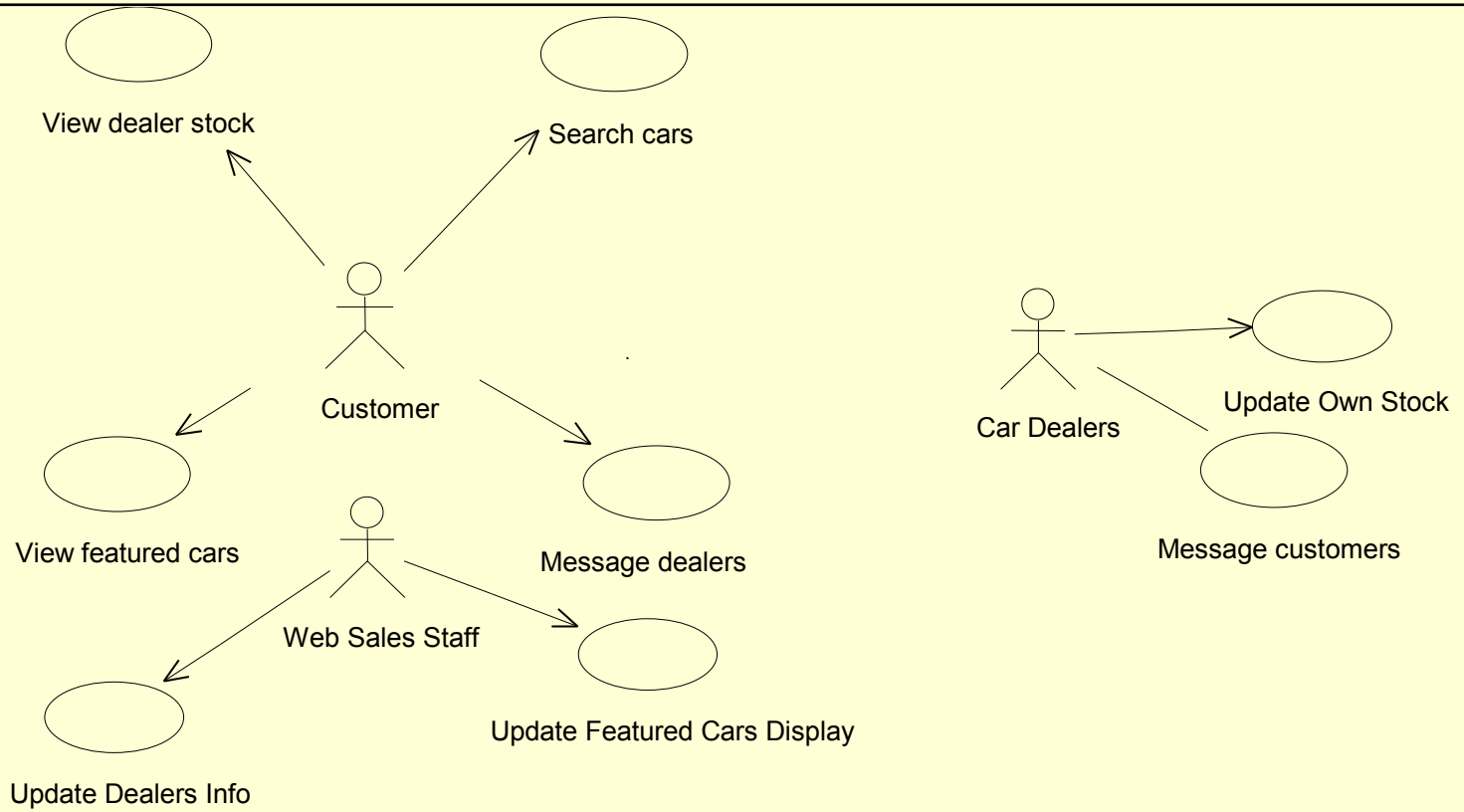
Outline

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

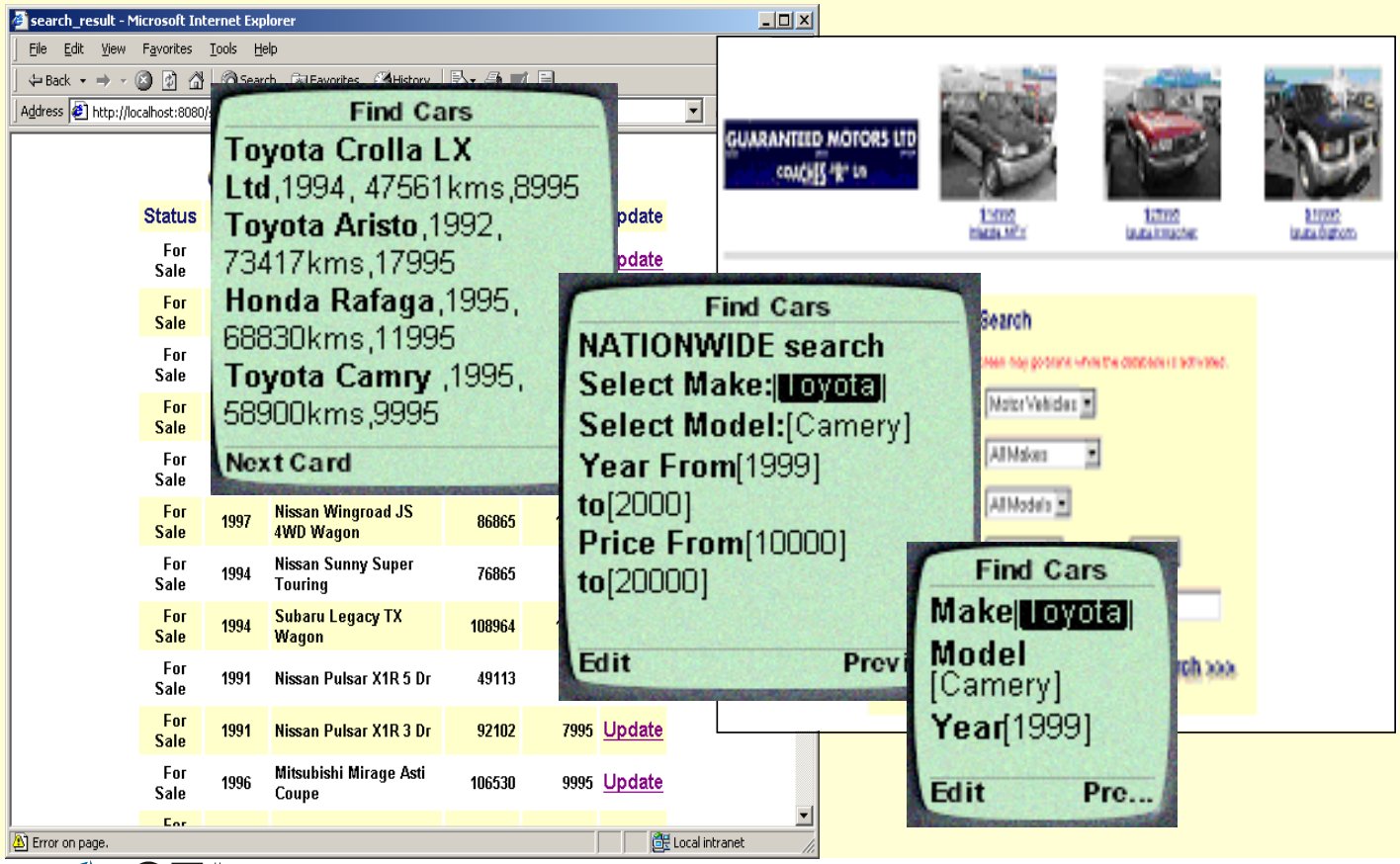
Motivation

- 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



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…

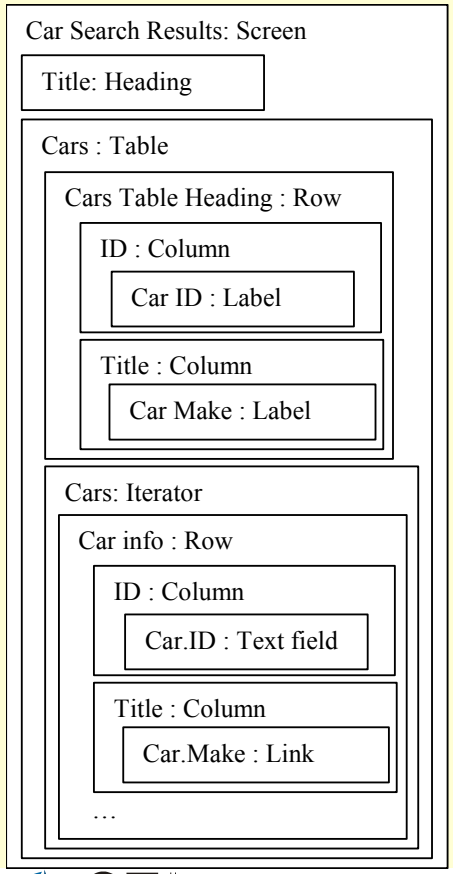
Adaptive UI Design Issues

- 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



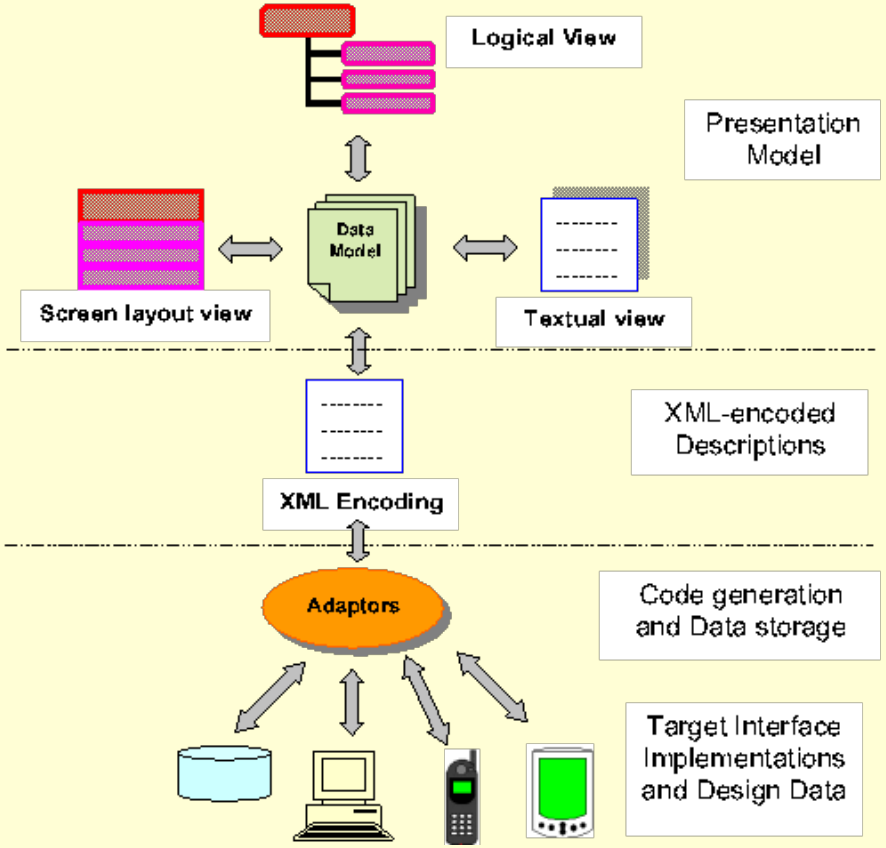
- Used to think about UI elements, composition, some idea of interaction
- UI structure-based
- AUIT synthesises MUI on-demand using requesting device characteristics, user profile/workflow (task) info…
- …and yes, there are lots of limitations, esp. interaction, layout, simple complex->multiple simple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

Our Approach

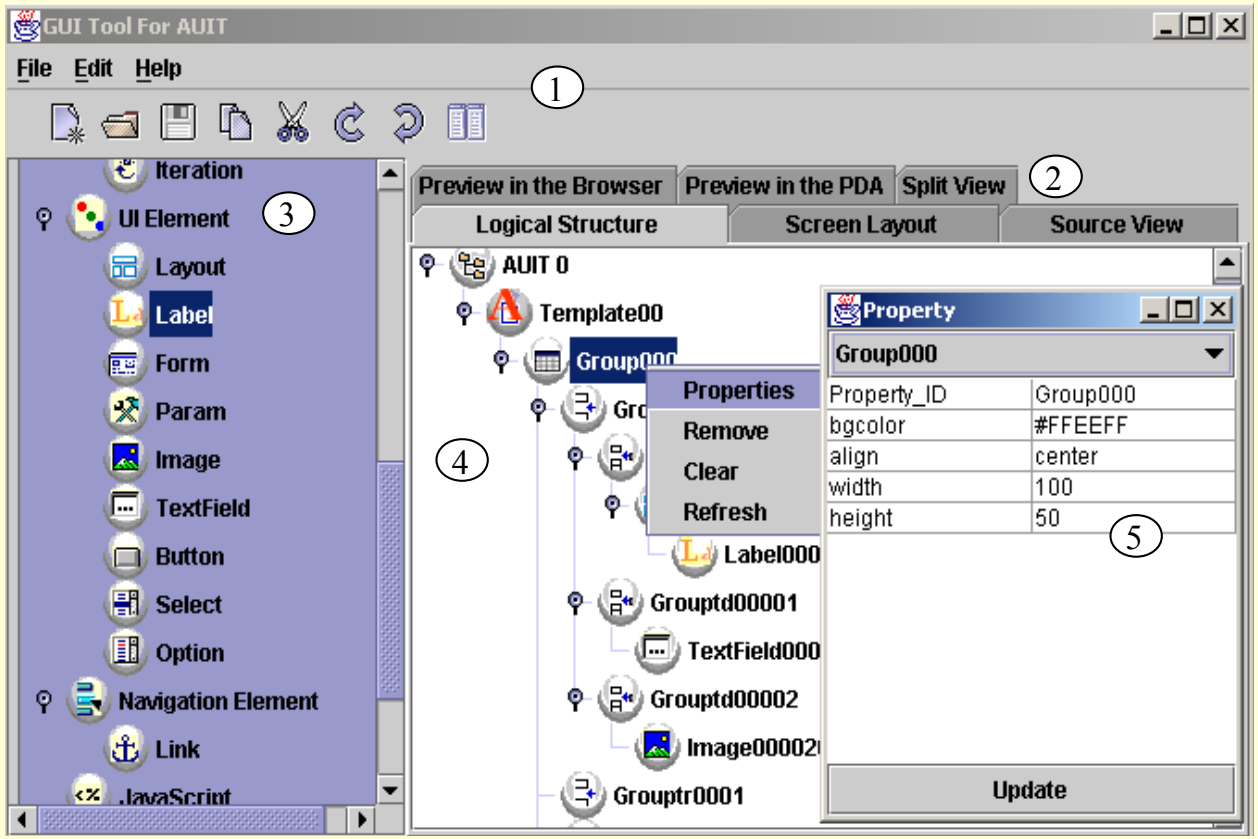
- Developed Adaptable User Interface Technology – custom tag library for JSPs to build such UIs
- Developed design environment allowing developers to specify adaptable user interfaces using three views:
  - UI element tree structure view
  - UI screen layout view
  - Textual view
- Generates AUIT (JSP)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

Tool Architecture

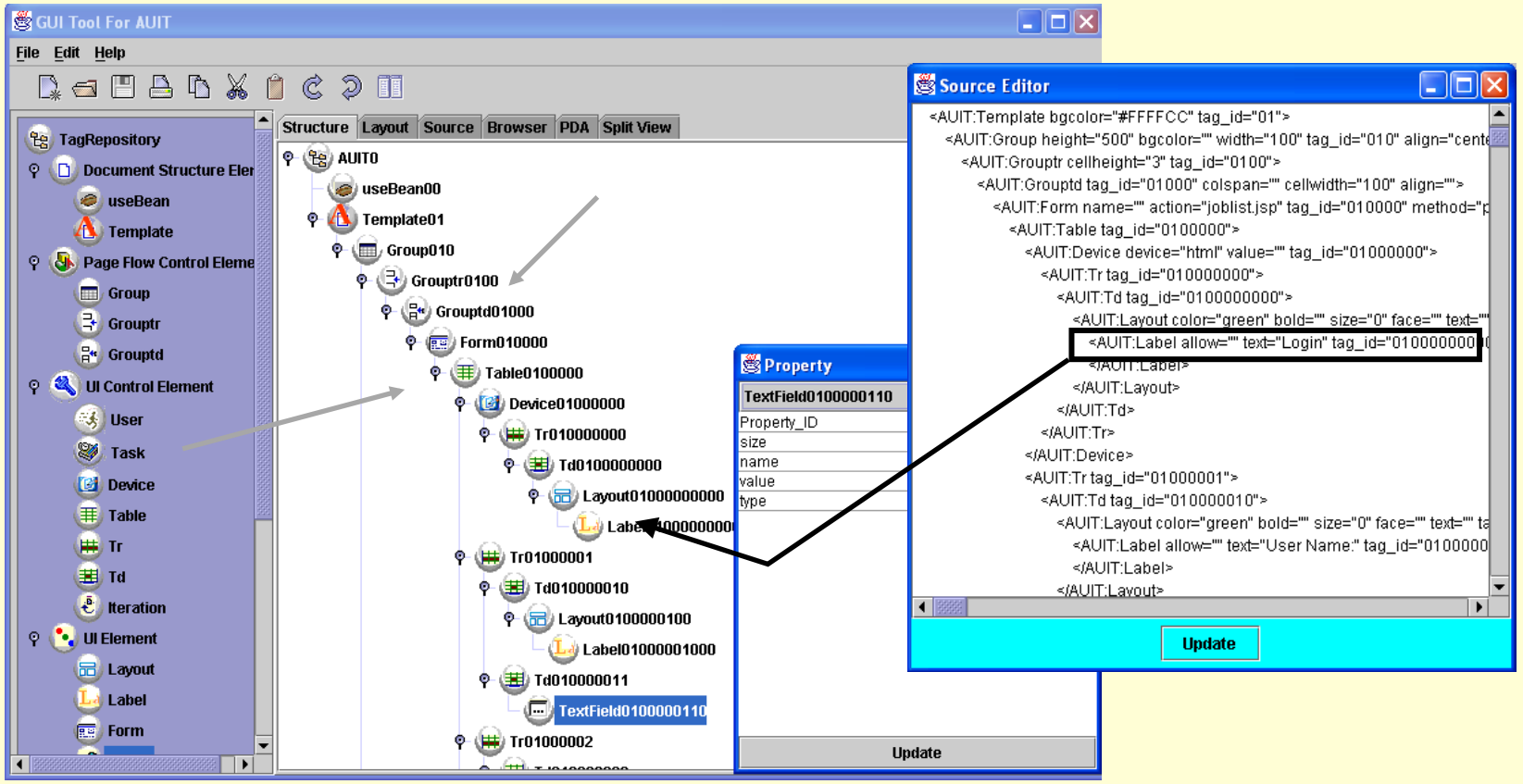
- Three views share common data model
- Tree view-outline
- Screen layout-size, shape, positioning
- Textual-details
- XML encodes UI design
- Code generators produce multiple implementations of adaptive UI



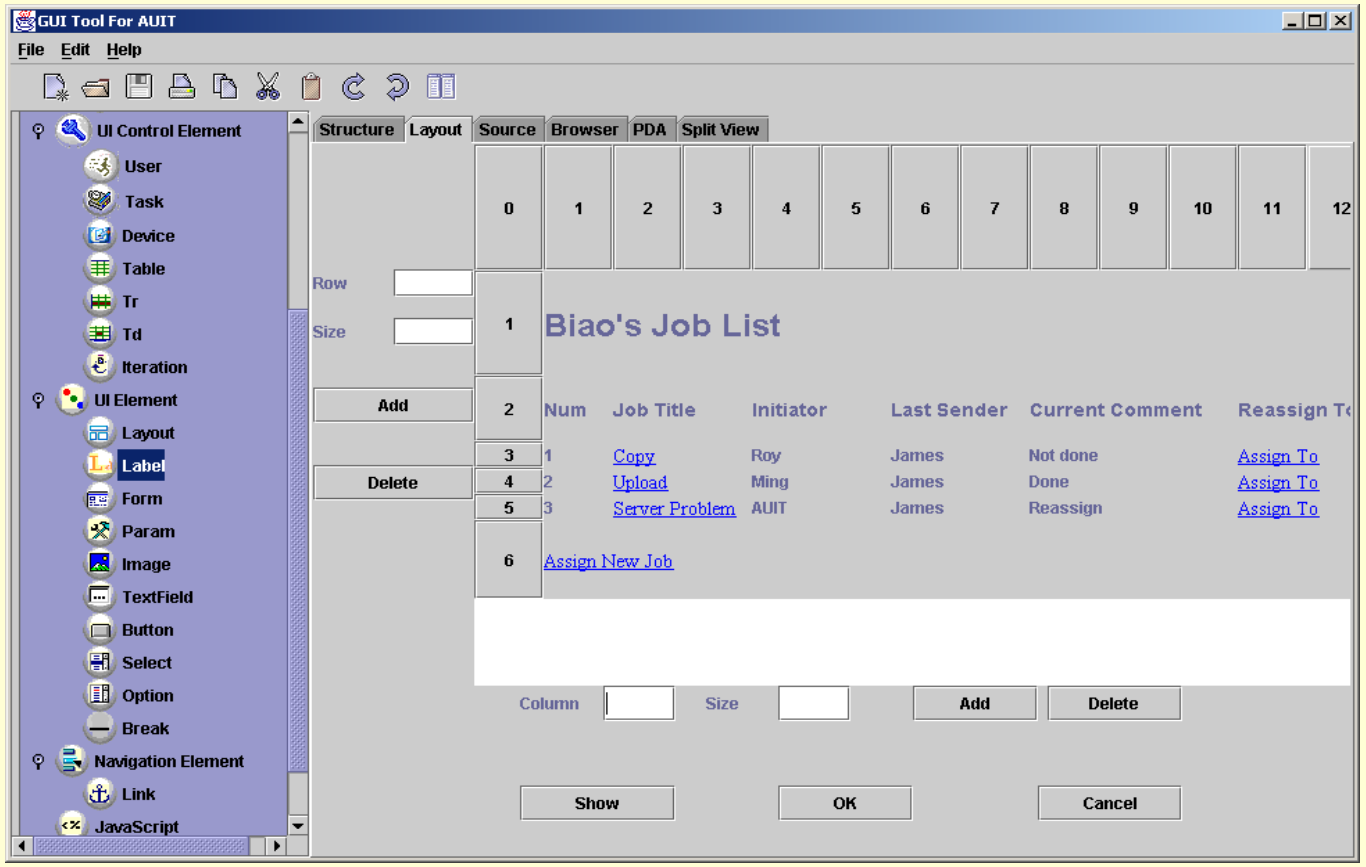
Tool User Interface [I'm not brave enough for demo, sorry ☺]



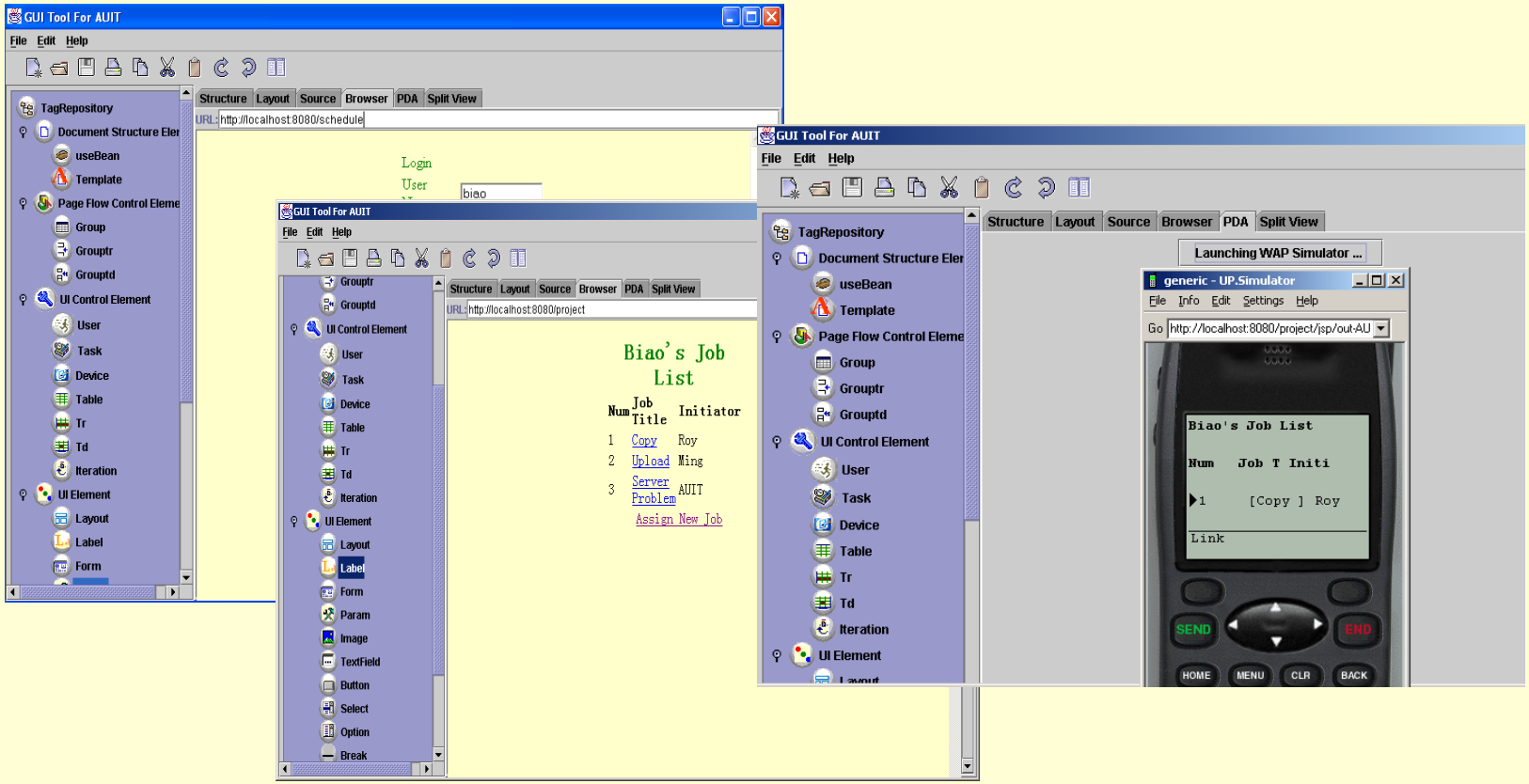
Example #1: Tree Viewer



Example #2: Screen Layout View

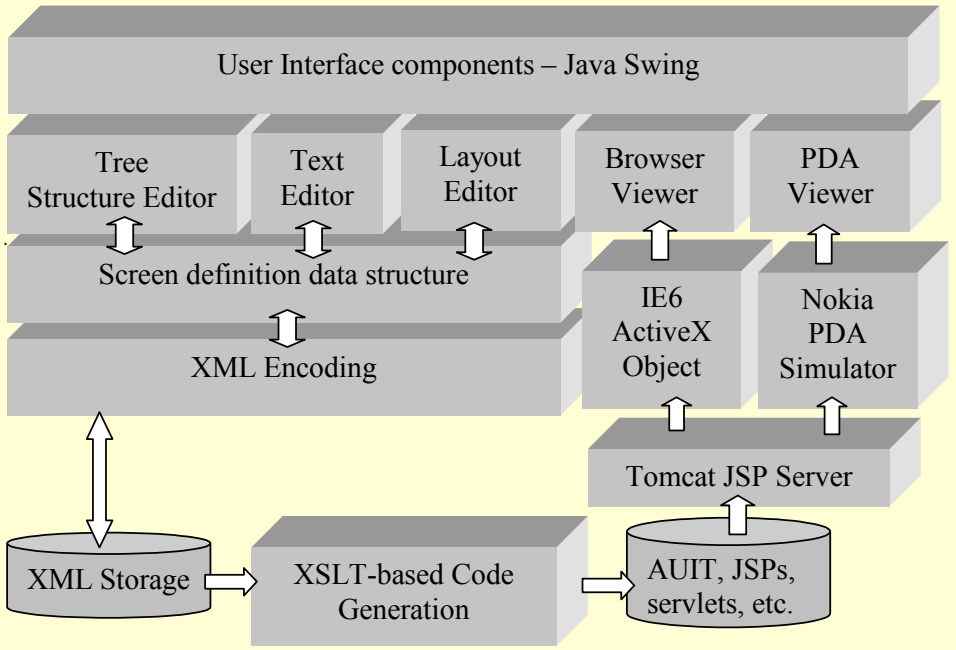


Example #3: Running Interfaces



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 accurarte, 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'Aquilla, Italy, 15-19 September 2008, IEEE CS Press.