Engineering multi-device user interfaces and architectures

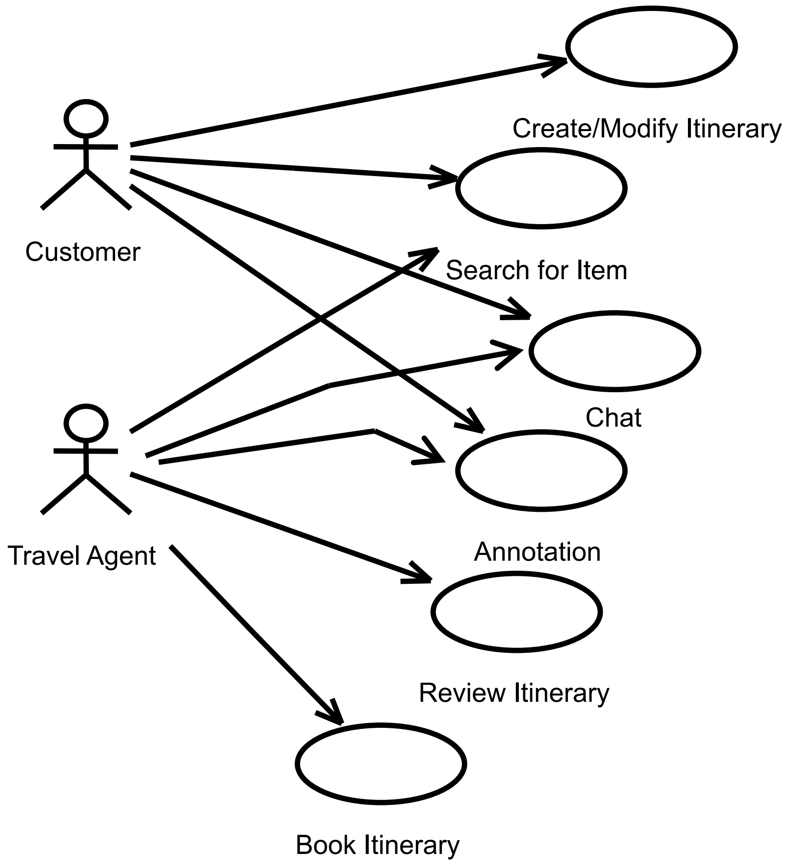
Professor John Grundy
Dept. Electrical and Computer Engineering and Dept. Computer Science
University of Auckland, New Zealand

Software Engineering
The University of Auckland

Outline

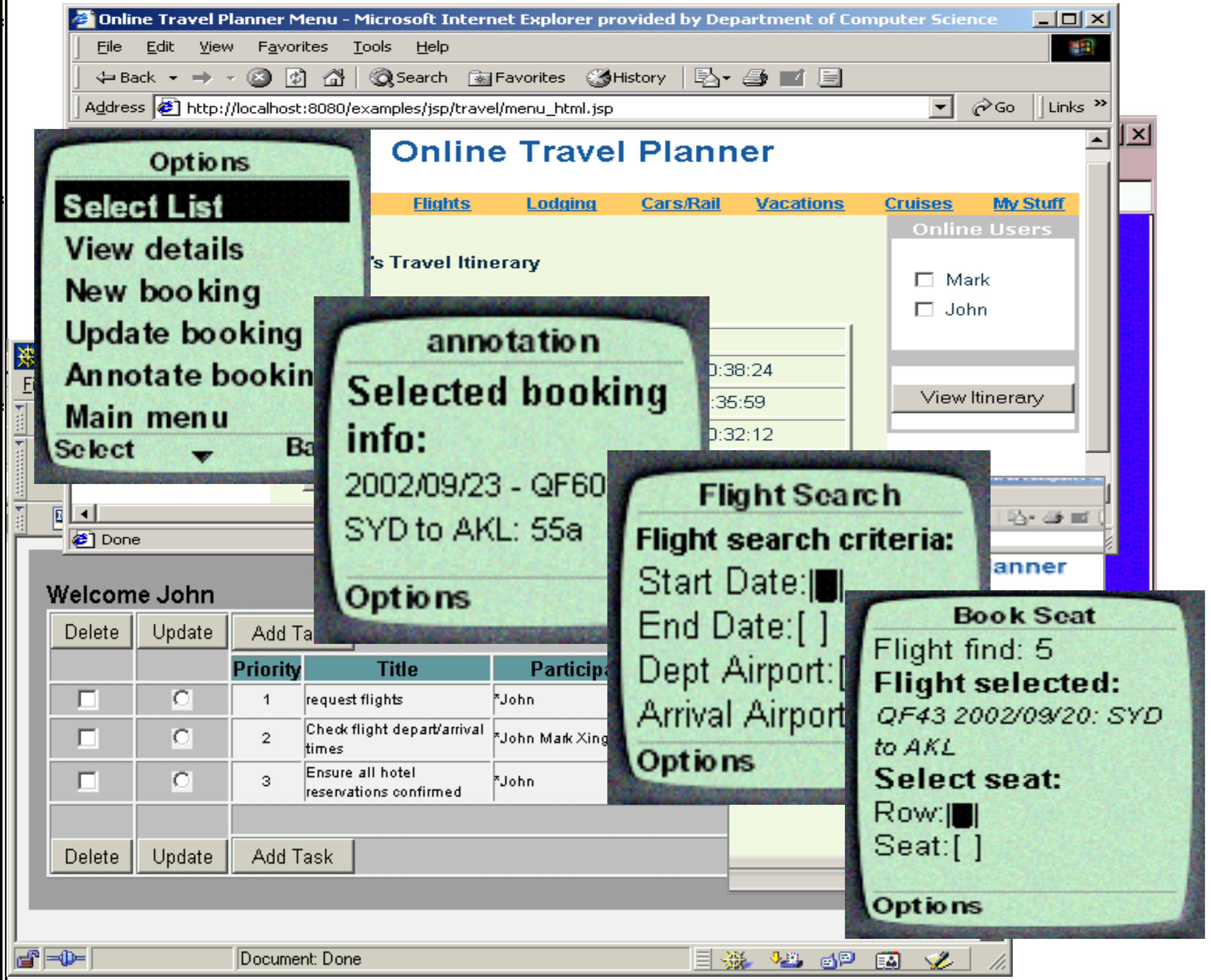
- What are Multi-device and Adaptive User Interfaces?
- Examples of MUIs
- Architectures for MUIs
- Design tools for MUIs
- Generating thin-client MUIs from thick-client UIs
- Evaluation
- Conclusions & Future Research

What are Multi-device & Adaptive User Interfaces?



- Example: collaborative travel planner
- Users might want to use *different devices* e.g. laptop vs PDA vs phone
- Might want to share same user interface between *different users*
- *Context of use* may change (task/role, device, connectivity, …)

Example #1: Travel Planner Application



Online Travel Planner Menu - Microsoft Internet Explorer provided by Department of Computer Science

File Edit View Favorites Tools Help

Address http://localhost:8080/examples/jsp/travel/menu_html.jsp

Online Travel Planner

Flights Lodging Cars/Rail Vacations Cruises My Stuff

Online Users
☐ Mark
☐ John
View Itinerary

's Travel Itinerary

:38:24
:35:59
:32:12

Done

Options
Select List
View details
New booking
Update booking
Annotate bookin
Main menu
Select Ba

annotation
Selected booking info:
2002/09/23 - QF60
SYD to AKL: 55a
Options

Flight Search
Flight search criteria:
Start Date:[ ]
End Date:[ ]
Dept Airport:[
Arrival Airport
Options

Book Seat
Flight find: 5
Flight selected:
QF43 2002/09/20: SYD to AKL
Select seat:
Row:[ ]
Seat:[ ]
Options

Welcome John

Delete Update Add Ta

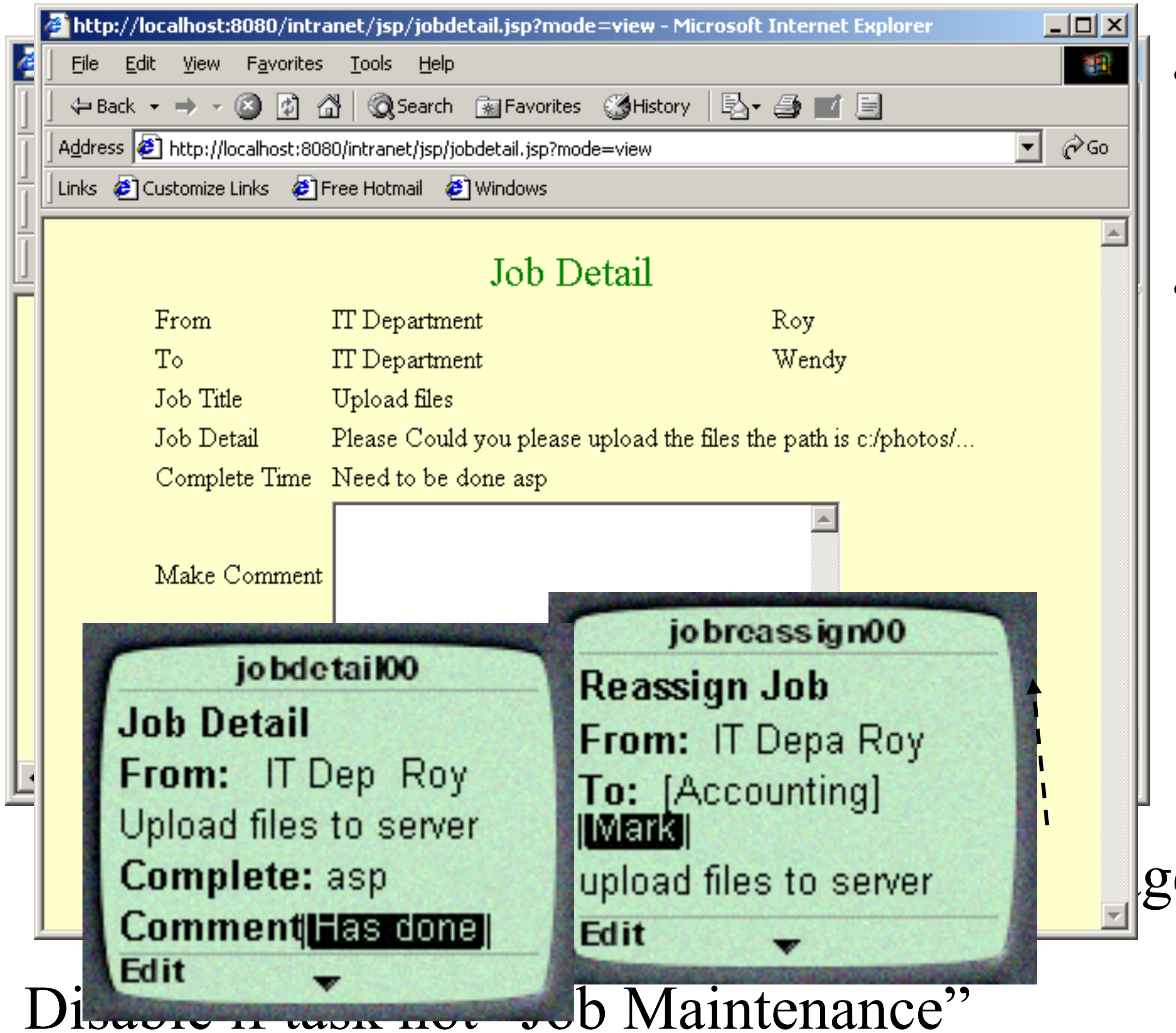
		Priority	Title	Particip
☐	○	1	request flights	*John
☐	○	2	Check flight depart/arrival times	*John Mark Xing
☐	○	3	Ensure all hotel reservations confirmed	*John

Delete Update Add Task

Document: Done

- Thick-client UIs
- Web-based UIs
- Mobile UIs
- Collaborative work support
- Different capabilities for different users/tasks…

Example #2: Job Management Application


http://localhost:8080/intranet/jsp/jobdetail.jsp?mode=view - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://localhost:8080/intranet/jsp/jobdetail.jsp?mode=view Go
Links Customize Links Free Hotmail Windows

Job Detail

From	IT Department	Roy
To	IT Department	Wendy
Job Title	Upload files	
Job Detail	Please Could you please upload the files the path is c:/photos/...	
Complete Time	Need to be done asp	
Make Comment		

jobdetail00

Job Detail
From: IT Dep Roy
Upload files to server
Complete: asp
Comment: Has done
Edit

jobreassign00

Reassign Job
From: IT Depa Roy
To: [Accounting]
[Mark]
upload files to server
Edit

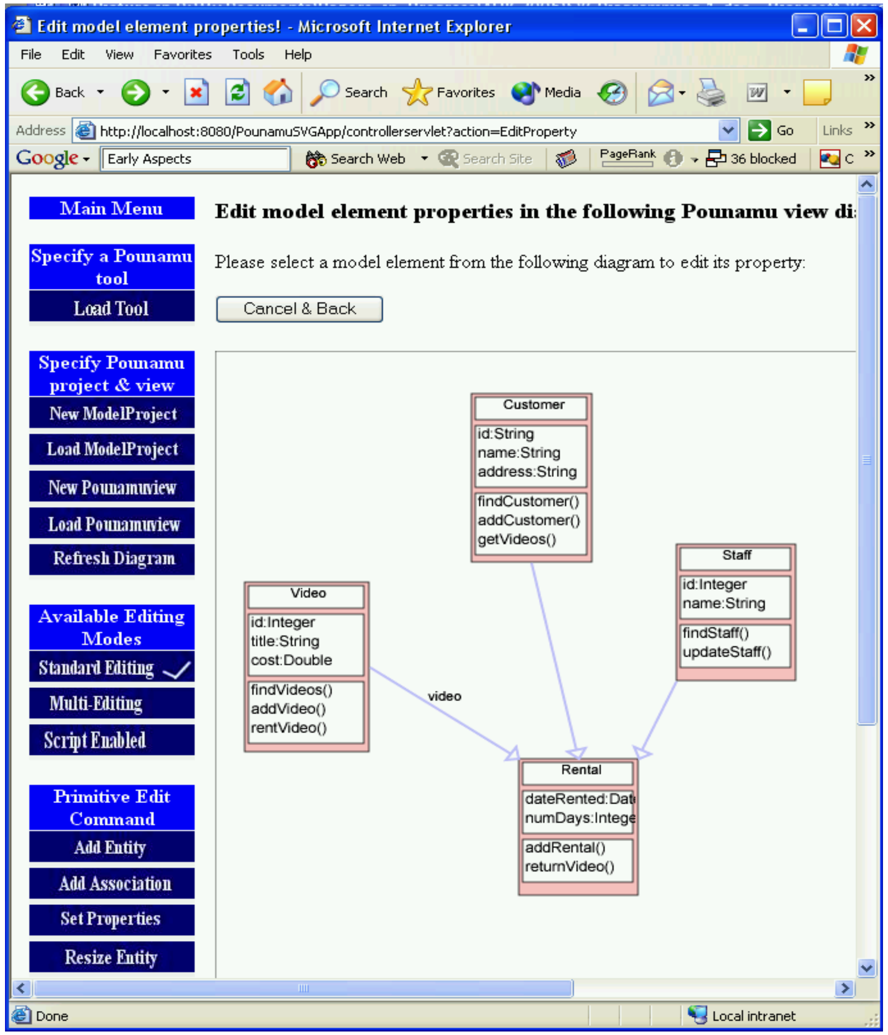
- Web and mobile-based user interfaces
- Need adaptive features – different content for different users e.g. Wendy vs John; manager vs staff; maintenance vs searching…

ger

Di… …b Maintenance”

Example #3: Web-based diagramming

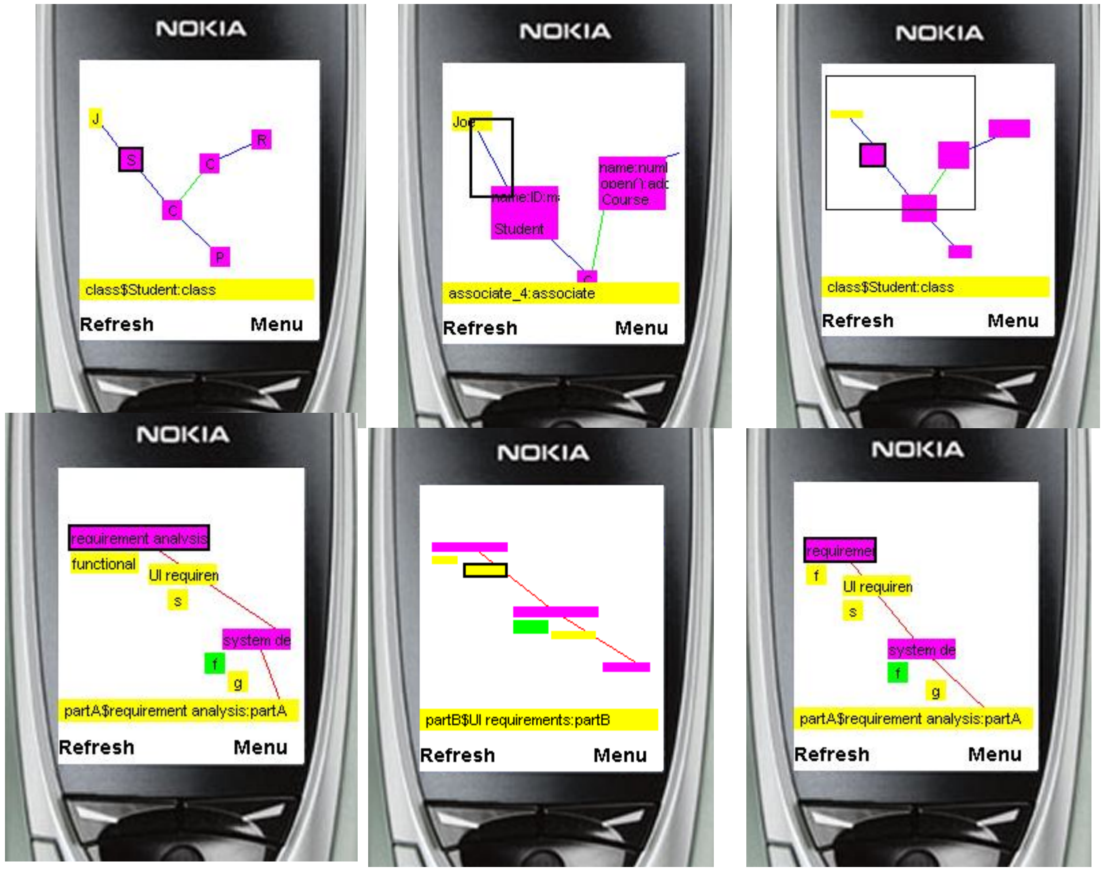
SE Software Engineering The University of Auckland



- Idea: want to access complex diagrammatic content via web browsers
- Avoids install on each machine, enables update only of server, can use web infrastructures to support collaborative work etc
- Three versions we have produced: GIF-based images, SVG-based images, VRML (3D) images – and editable!
- SVG & VRML plug-ins allow drag-and-drop in browser via ECMA client-side scripting
- Generated from meta-tool…

Example #4: Mobile device diagramming

SE Software Engineering
The University of Auckland



- Diagramming on mobile devices (PDAs, phones)?!
- Multi-level zoom, panning, use buttons to manipulate content
- Again, generated from meta-tool (how done – I'll talk about shortly…)

Software Engineering
The University of Auckland

Other Examples…

John
Rebecca

- Sketching-based user interfaces
- 3D rendering and interaction with complex data

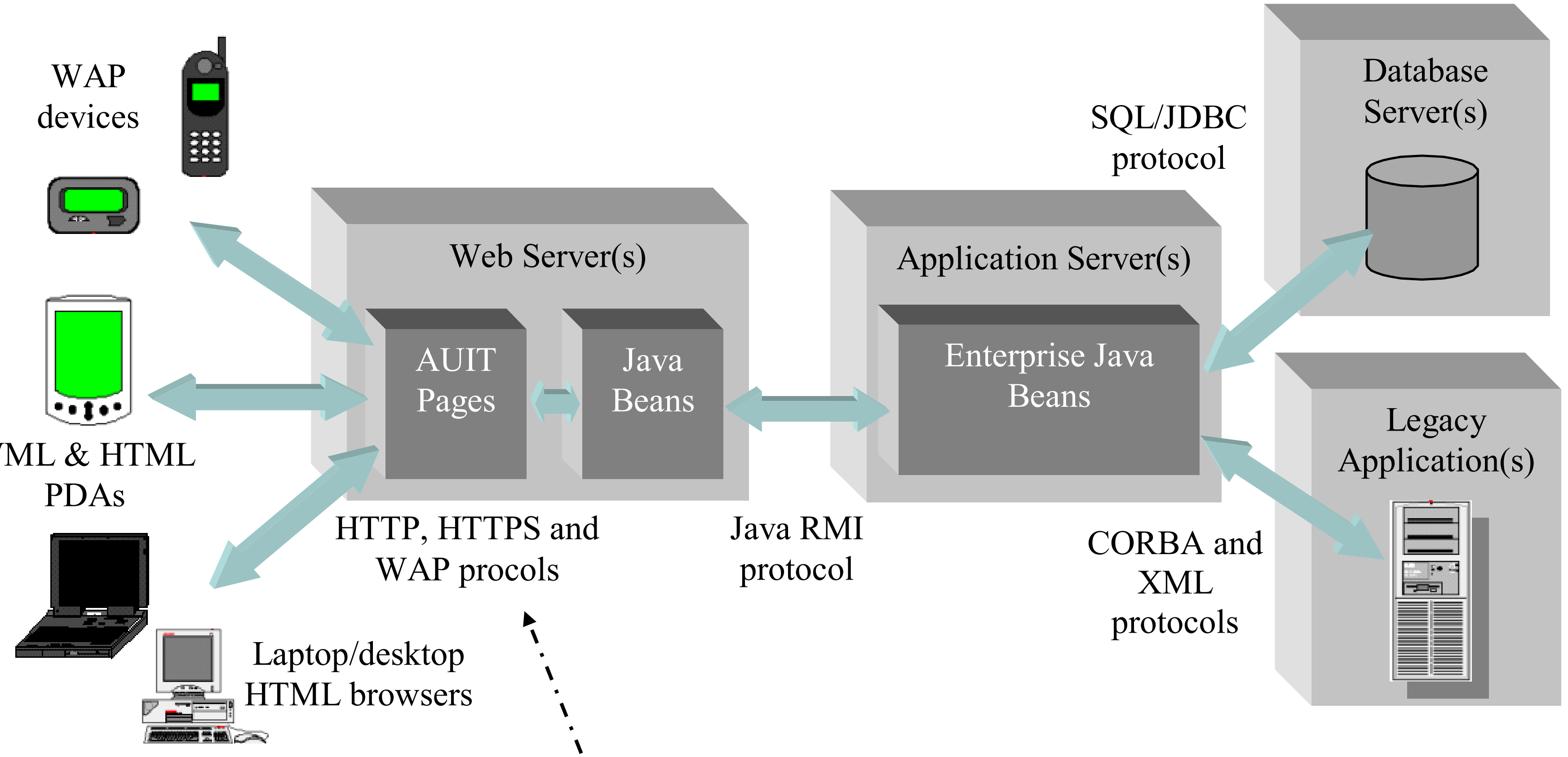
Architectures for Building MUIs

- How do we build such interfaces??
- Bespoke architectures
  - Most currently done this way
  - Lots of effort, difficult, limited adaptation etc
- Convert concrete format from e.g. HTML to WAP
  - Clipping e.g. Palm; page content and image transformation
  - Limited ability to translate as working on concrete UI content
- Generate interfaces from abstract specifications
  - Transcoding; WebML; various XML-based approaches
  - OK, but again limited ability to adapt interfaces
- **Generate from abstract specification, but also enrich with device/user/task information…**

Architecture #1: Adaptive User Interface Technology (AUIT)

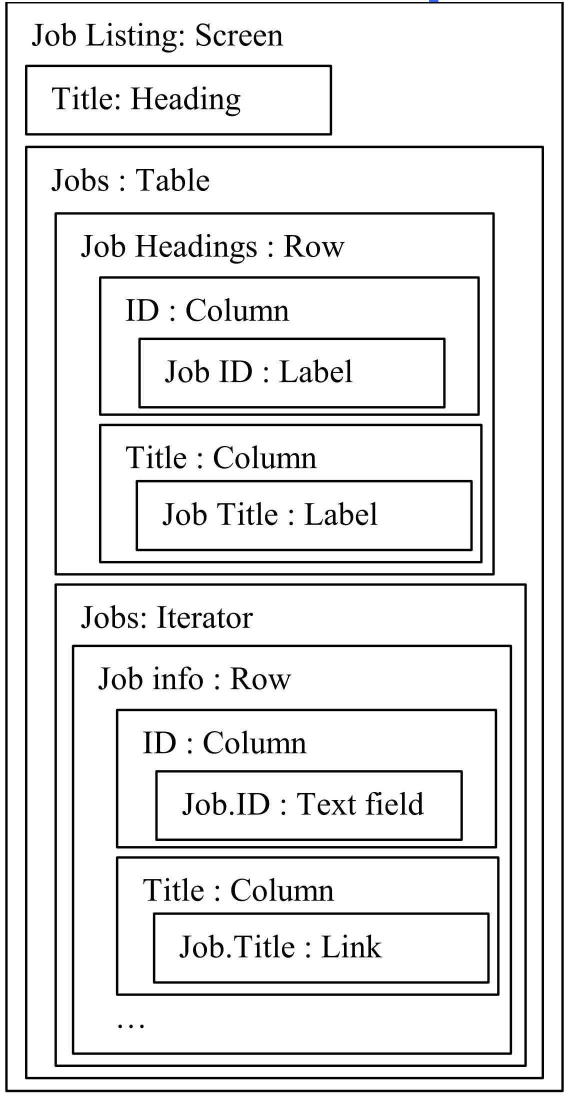
- Extension to Java Server Pages (JSPs) to specify one abstract, form-based, multi-device user interface
- Used to implement the travel planner & job management tools we saw earlier…
- Describe:
  - Elements of user interface with device-independent constructs
  - Composition of elements
  - What elements relevant/irrelevant to which users/user tasks
- At run-time, AUIT examines requesting device capabilities, device user and current user task
- Returns HTML or WAP encoding of user interface

Architecture



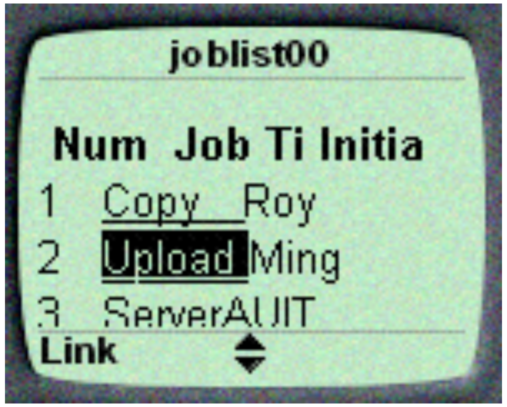
Extensions to J2EE web-tier components

Example of AUIT Page



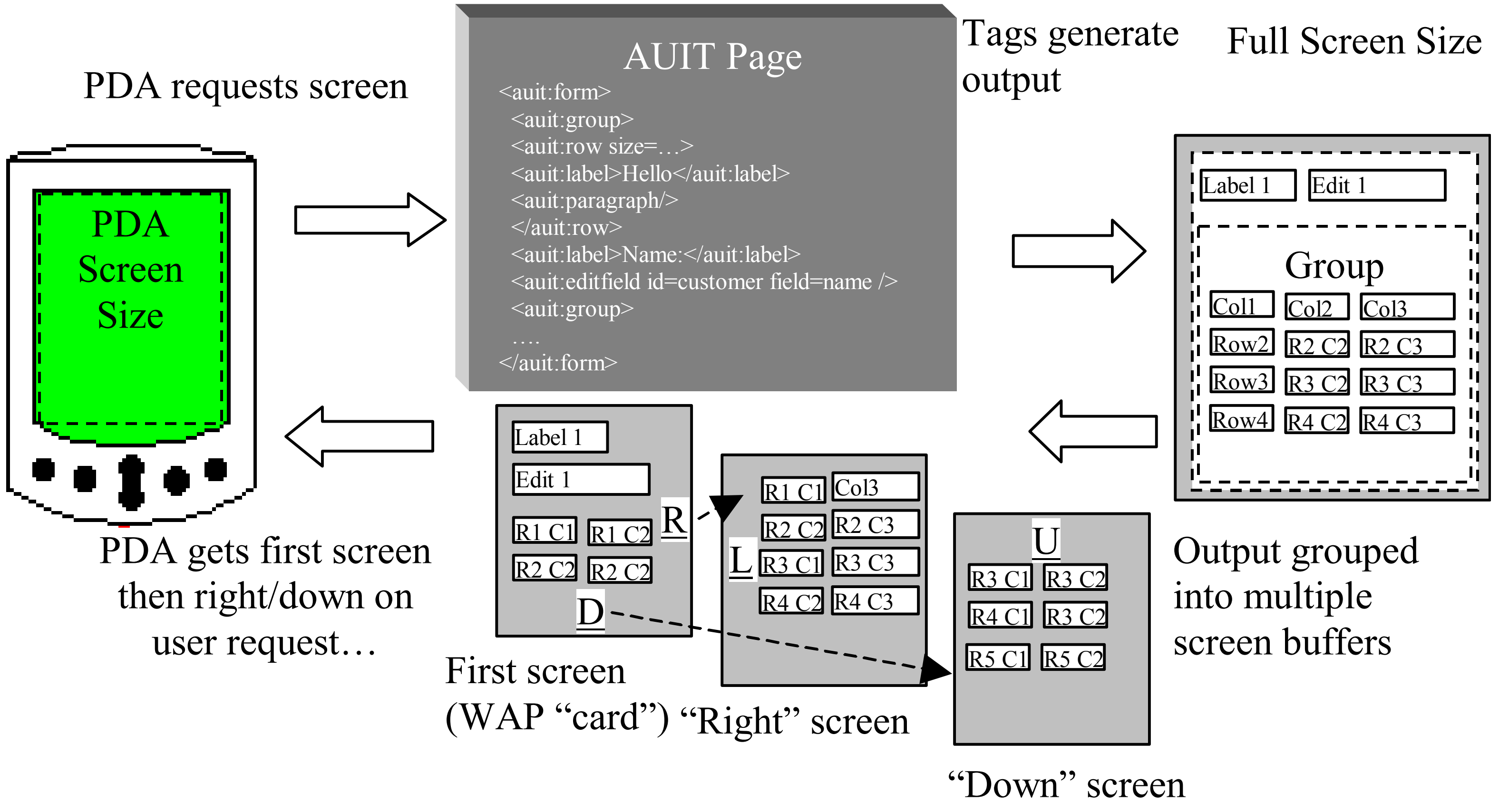
```
<%@ taglib uri="/auit" prefix="auit" %> // page directive to access AUIT tags
<jsp:useBean id='job_manager' class='jobs.JobManager /> // JavaBeans to use
…
<auit:screen name="job list"> // sets user/task/device information…
   <auit:heading level=2 value='<%= AUITUser.getUserName() %>'s Job List' />
   <auit:table width=60 border=0>
     <auit:row><auit:column><auit:label width=6 value='Num' /></auit:column>…
     <% jobs = job_manager.selectJobs(AUITUser.getUserName()); %>
     <auit:iterator name=job data=jobs %>
       <auit:row height=1>
         <auit:column><auit:label width=6 value=
                         '<% job.getJobNumber() %>' /></auit:column>
         <auit:column><auit:link width=20 name='<% job.getJobNumber() %>'
                         href='job_details.jsp?task=detail&job=
                                   <% job.getJobNumber() %>' /></auit:column>
         <auit:column><auit:label width=30 value=
                         '<% job.getInitiator() %>' /></auit:column>

         …
       </auit:row>
     </auit:iterator>
   </auit:table>
</auit:screen>
```

Logical structure encoded via special mark-up language


joblist00

Num Job Ti Initia
1 Copy Roy
2 Upload Ming
3 ServerAUIT
Link

Page-splitting

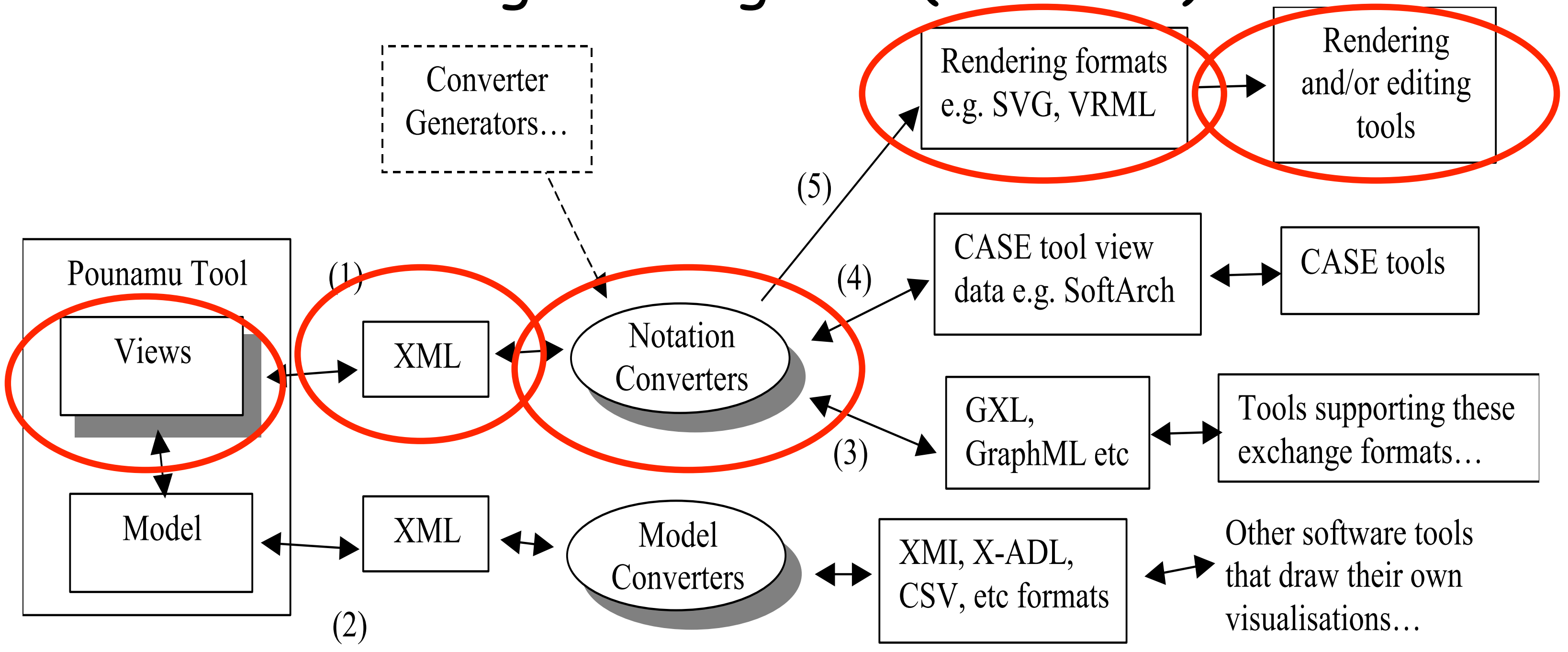


```
<auit:form>
  <auit:group>
  <auit:row size=…>
  <auit:label>Hello</auit:label>
  <auit:paragraph/>
  </auit:row>
  <auit:label>Name:</auit:label>
  <auit:editfield id=customer field=name />
  <auit:group>
  ….
</auit:form>
```

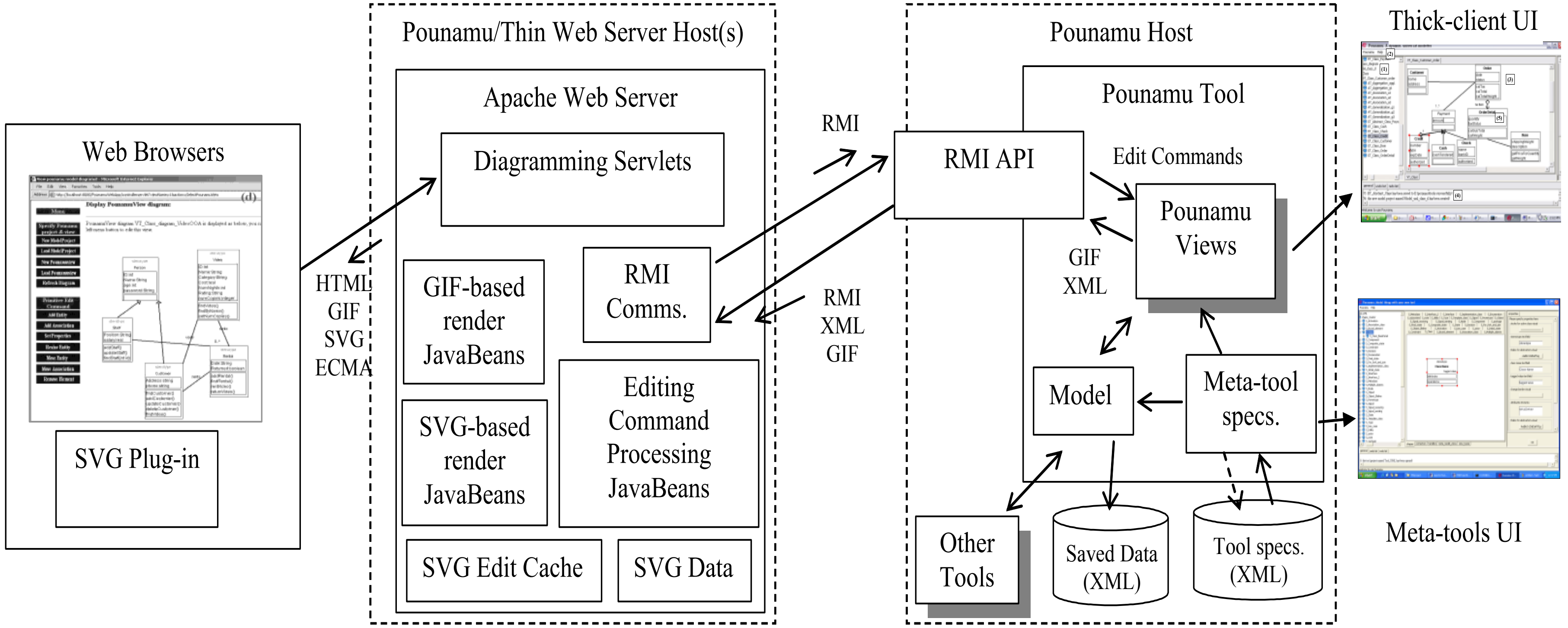
SE Software Engineering The University of Auckland

Architecture #2: Pounamu meta-tool translators and plug-ins

- Used to produce thin-client diagramming tools (web browser – SVG, VRML; mobile phone – MUPE)
- Synthesizes content (image+scripting+HTML) from thick-client diagramming tool (Pounamu) content…

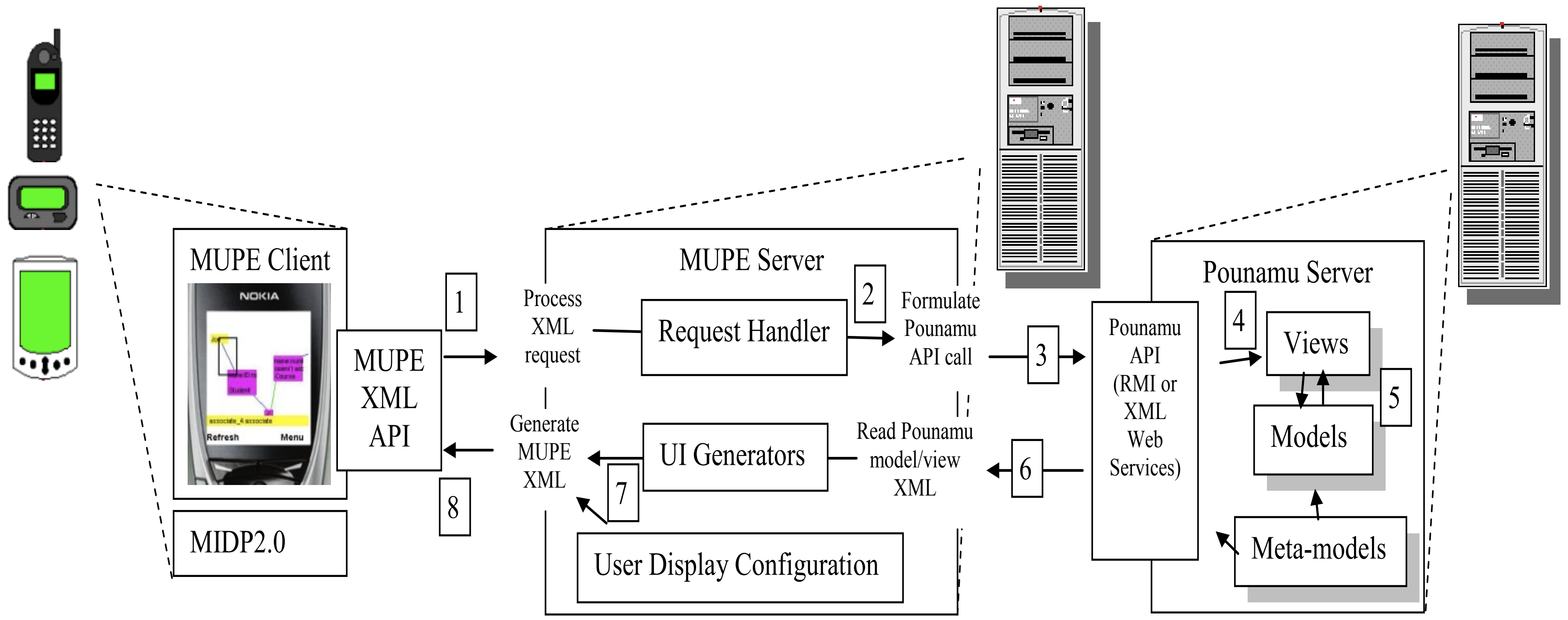

Converter Generators…
Rendering formats e.g. SVG, VRML
Rendering and/or editing tools
(5)
Pounamu Tool
Views
Model
(1)
XML
Notation Converters
(4)
CASE tool view data e.g. SoftArch
CASE tools
GXL, GraphML etc
Tools supporting these exchange formats…
(3)
XML
Model Converters
XMI, X-ADL, CSV, etc formats
Other software tools that draw their own visualisations…
(2)

Thin-client web plug-in

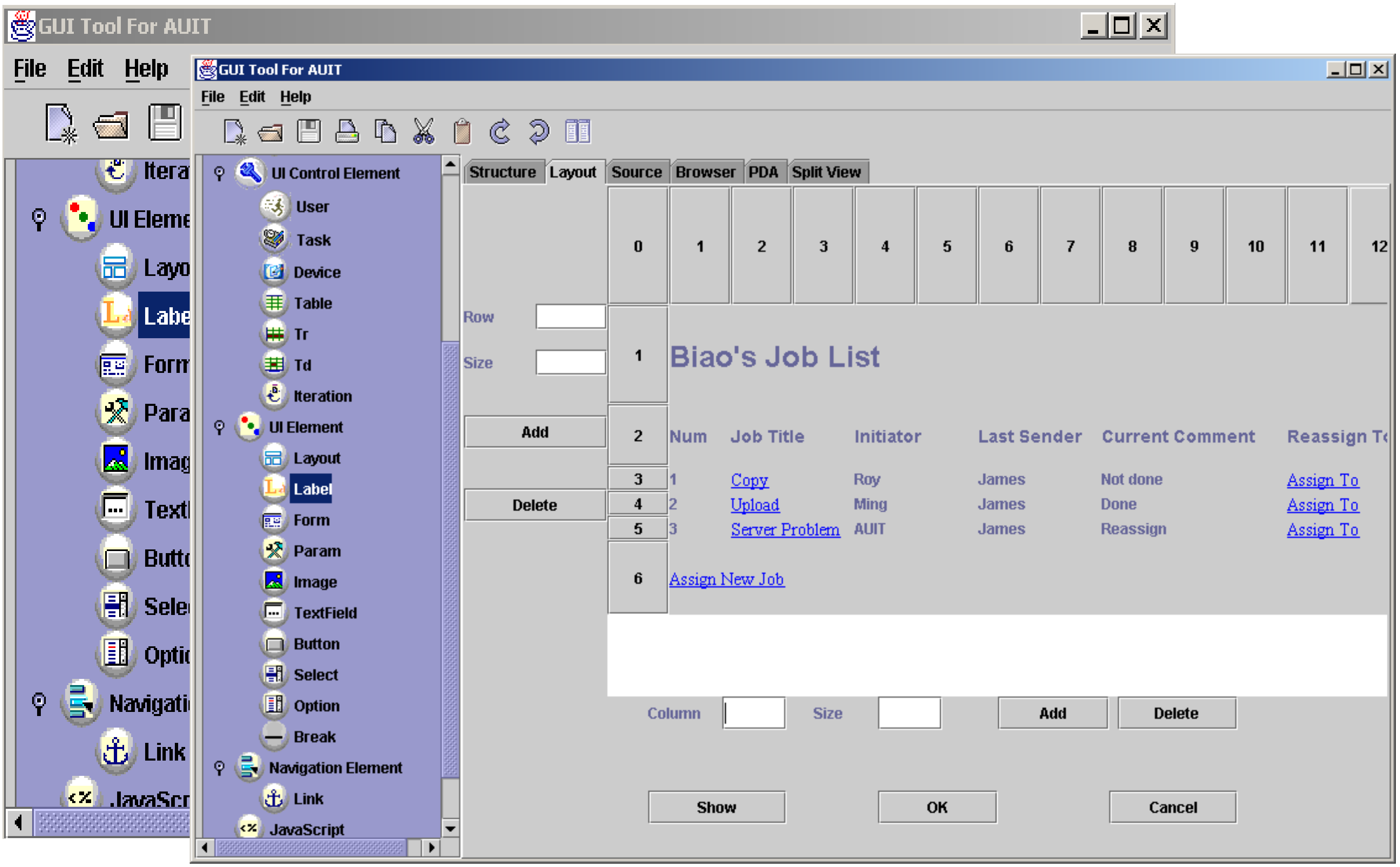


MUPE Mobile device plug-in

SE Software Engineering
The University of Auckland



Design tools: iView



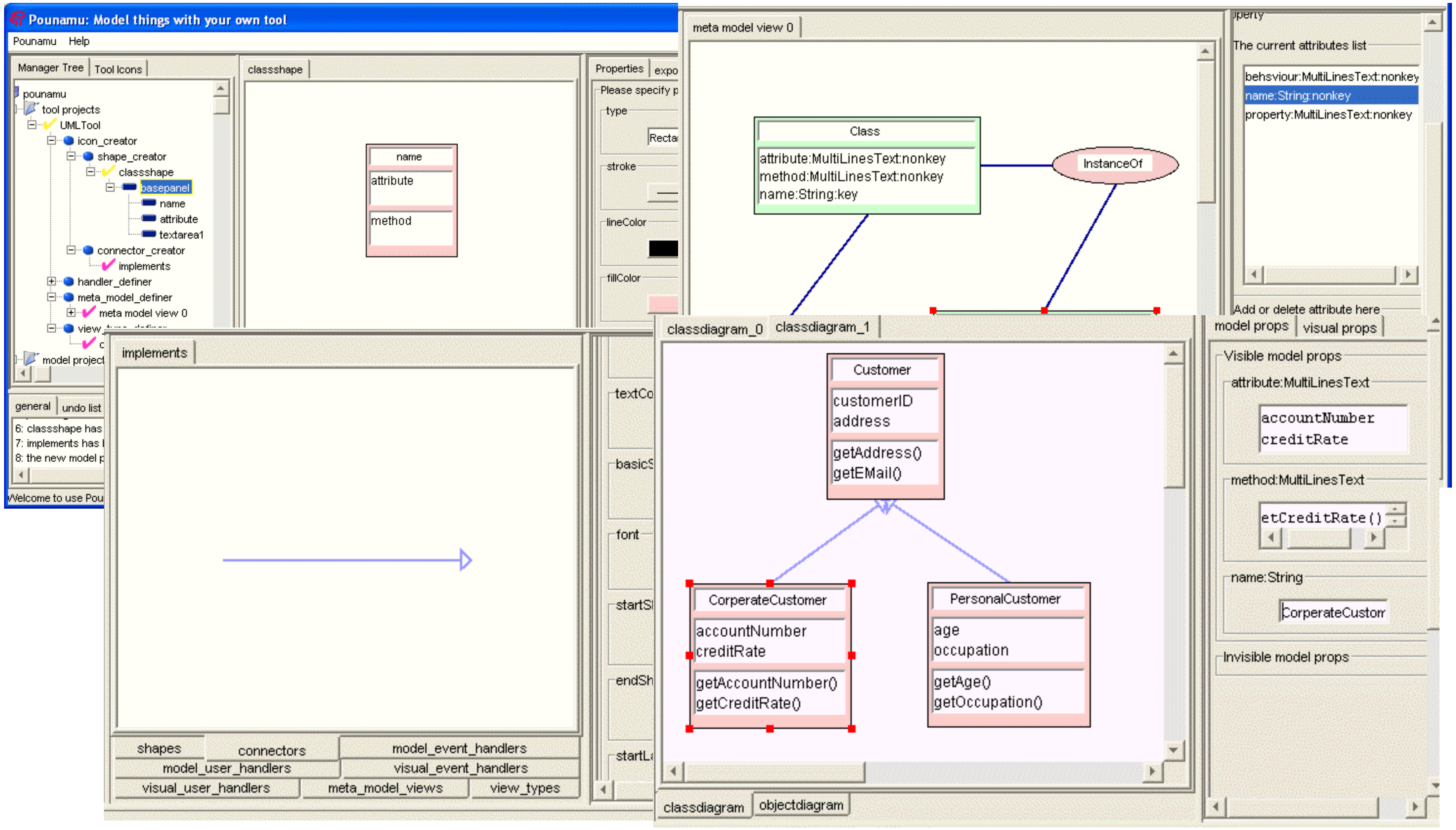
Biao's Job List

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

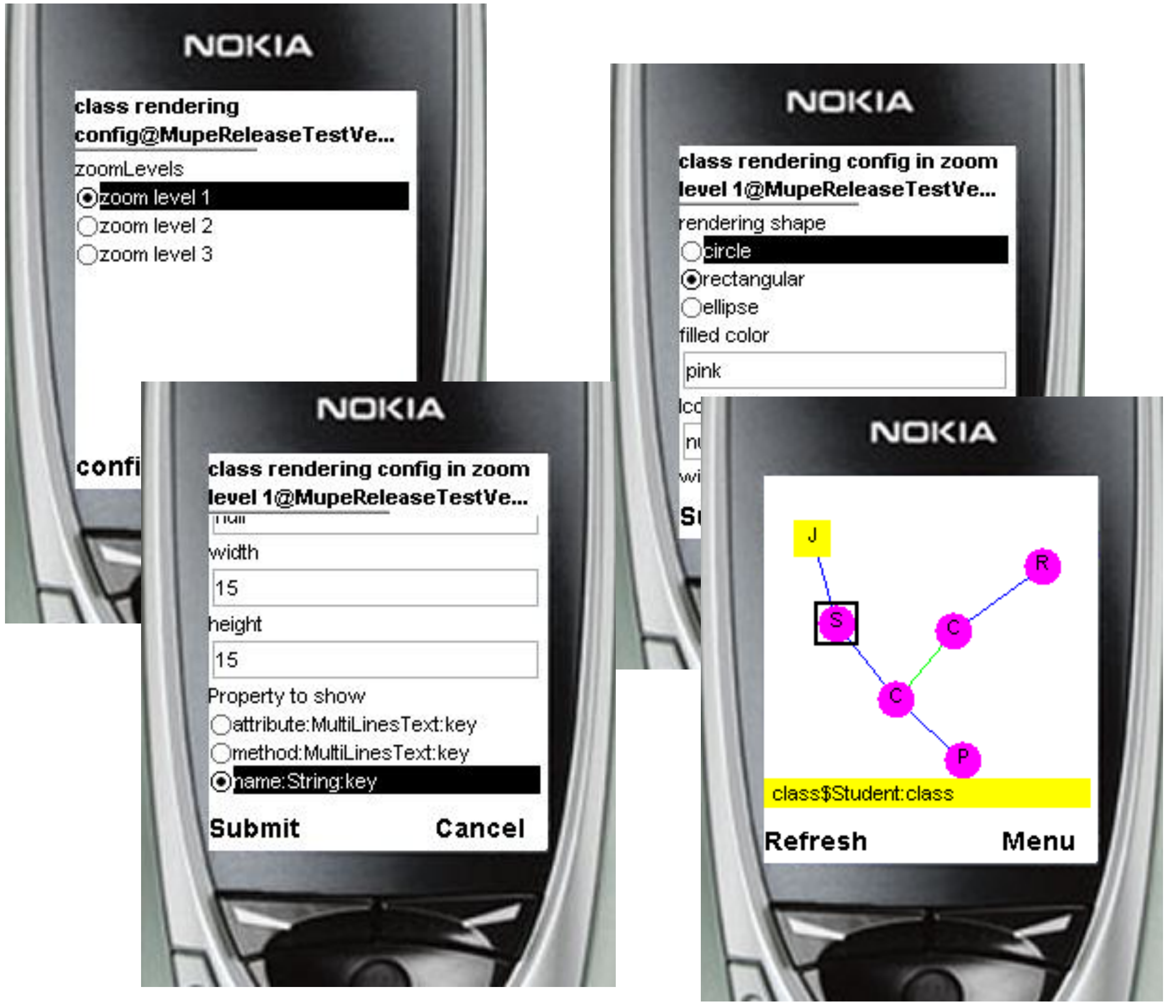
Assign New Job

Design tools: Pounamu

SE Software Engineering
The University of Auckland



Pounamu/Mobile User Preferences

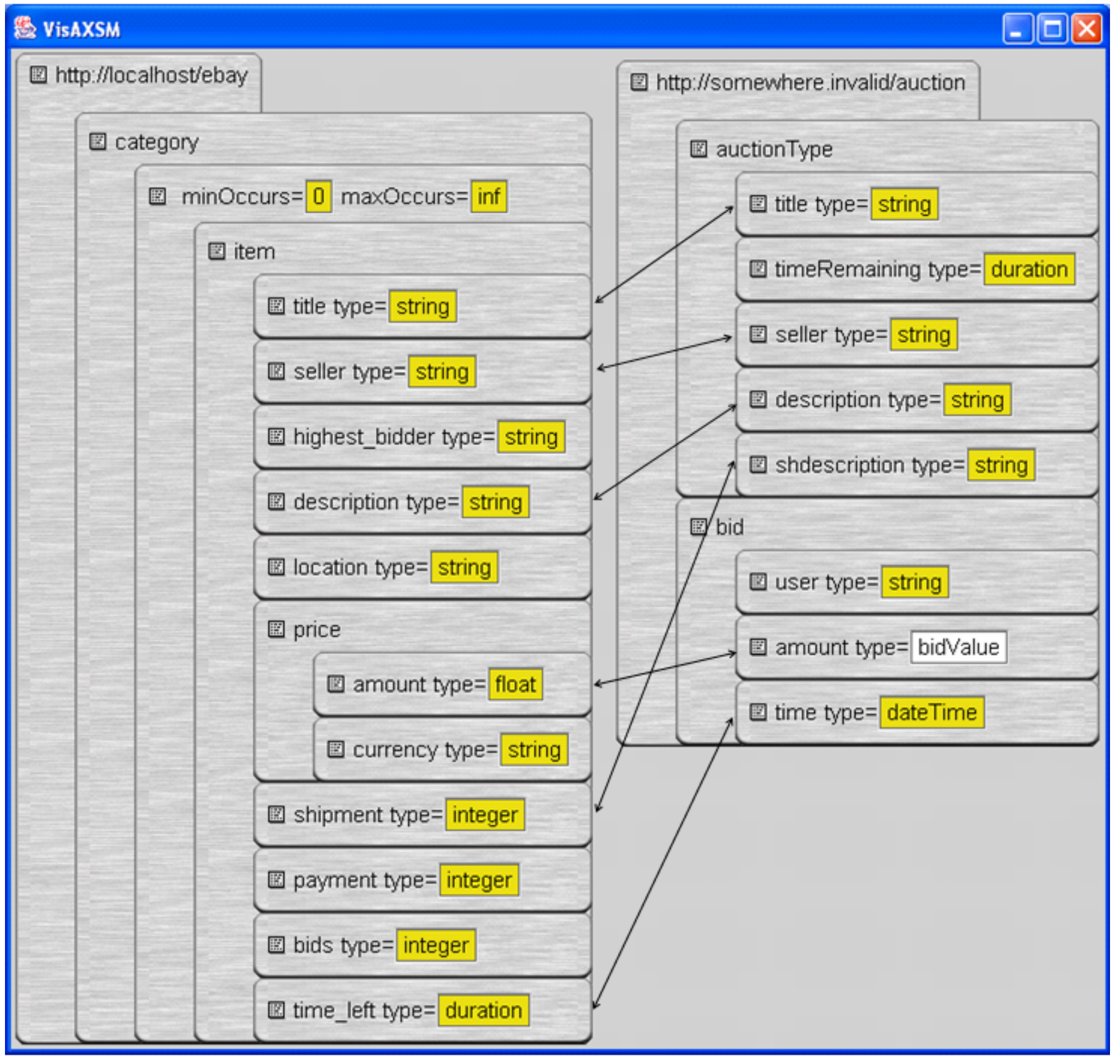


class rendering config@MupeReleaseTestVe...
zoomLevels
[✓ zoom level 1]
zoom level 2
zoom level 3
confi

class rendering config in zoom level 1@MupeReleaseTestVe...
rendering shape
circle
[✓ rectangular]
ellipse
filled color
pink
Ico
nu
wi
Su

class rendering config in zoom level 1@MupeReleaseTestVe...
width
15
height
15
Property to show
attribute:MultiLinesText:key
method:MultiLinesText:key
[✓ name:String:key]
Submit Cancel

VisAXSM: Converter Generator



Evaluation

SE Software Engineering The University of Auckland

- AUIT:
  - Industry web developers + academics
  - Both used web-based and mobile PDA-based UIs and developed UIs with AUIT
  - iView tool for AUIT: both used & developed UIs
  - Users preferred AUIT/iView-generated UIs over hard-coded, device-specific UIs for our example problem domain… (!)
- Pounamu/Thin:
  - Industry UML designers + academics
  - Single user tasks: build/refine UML designs
  - Multi-user tasks: review/modify UML designs
  - SVG+scripting preferred; multi-user tasks didn't work well
- Pounamu/Mobile:
  - Initial results with project management tool promising…
- VisAXSM:
  - Applied to data translation problems (XML -> XML, code etc)
  - Now applying to generating UI content for browser via XML…

Conclusions & Future Research

Software Engineering
The University of Auckland

- Multi-device, adaptive user interfaces challenging to design and build
- We have developed several proof-of-concept approaches to building
- Domains include software design, project management, web-based information systems (travel planning, vehicle and house purchase, on-line trading)

- Working on 3D visualisations using VisAXSM – translate Marama XML into VRML and Games Engines (like Pounamu/Thin->SVG)
- Developing translation specification tools (VisAXSM)
- Developing better UI descriptions/user tailoring support
- Developing web services UI description to auto-generate UIs

References

- 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.
- Grundy, J.C. and Zou, W. Building multi-device, adaptive thin-client web user interfaces with Extended Java Server Pages, In Cross-Platform and Multi-device User Interfaces, Wiley, 2005.
- 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
- 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., 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 Zou, W. An architecture for building multi-device thin-client web user interfaces, In Proceedings of the 14th Conference on Advanced Information Systems Engineering, Toronto, Canada, May 29-31 2002, Lecture Notes in Computer Science.
- Grundy, J.C. and Hosking, J.G. Engineering plug-in software components to support collaborative work, Software - Practice and Experience, Vol. 32, No. 10, August 2002, Wiley, 983-1013.
- Mehra, A., Grundy, J.C. and Hosking, J.G. A generic approach to supporting diagram differencing and merging for collaborative design, In Proceedings of the 2005 ACM/IEEE International Conference on Automated Software Engineering, Long Beach, California, Nov 7-11 2005, IEEE Press, pp. 204-213
- Mehra, A., Grundy, J.C. and Hosking, J.G., Adding Group Awareness to Design Tools Using a Plug-in, Web Service-based Approach, In Proceedings of the Sixth International Workshop on Collaborative Editing Systems, CSCW 2004, Chicago, November 6, 2004.