Generating Web-based User Interfaces for Diagramming Tools

Shuping Cao\textsuperscript{1}, John Grundy\textsuperscript{1, 2}, John Hosking\textsuperscript{1}, Hermann Stoeckle\textsuperscript{1}, Ewan Tempero\textsuperscript{1} and Nianping Zhu\textsuperscript{1}

\textsuperscript{1}Dept of Computer Science and \textsuperscript{2}Dept of Electrical and Computer Engineering
University of Auckland, New Zealand
Outline

• Why need web-based diagramming tools??
• Overview of our approach
• Examples of using web-based diagramming:
  - GIF-based diagram editing
  - SVG server-side editing
  - SVG multi-update caching
  - SVG browser-side drag-and-drop
• Evaluation
• Conclusions and Future Research
Why Web-based Diagramming?

- Traditional CASE and other diagramming tools use “thick client” UI:
  1. hierarchy
  2. menus
  3. drawing pane
  4. messages
  5. pop-up menus, drag-and-drop editing

- Must install on all users’ PCs/port to different OS etc
- Must update on all PCs
- Heavy-weight support for collaboration
Our Motivation

- Wanted to see if we could realise visual design tools in web browser.
- Potential application areas for web-based diagramming:
  - E-learning, process modelling, data visualisation
  - Collaborative design
  - ASP model of design tool delivery
- Wanted to develop suitable architecture for these tools
- Wanted to explore usability of these tools
- Wanted to identify alternative editing approaches to thick-client e.g. page-based editing to fit web browser metaphor etc
- Wanted to support collaborative design using web-based architecture
- It was a fun challenge... 😊
Our Approach

- Add plug-in to existing thick-client meta-tool
- Web server to provide GIF and SVG diagram rendering
- Web browsers to render/post or with SVG plug-in support (limited) client-side scripting for drag-and-drop diagram editing
- Currently support diagram editing but not meta-facilities...
Example of Pounamu/Thin

- Page with diagram & controls via buttons
- Groups of controls - view management; editing style; editing modes
- Diagram in either GIF or SVG image formats
1. Server-side Diagram Editing

- **Browsers**
  - Diagram

- **Web Server**
  - Pounamu/Thin Servlets
    - Generate Update Message
  - Regenerate HTML/GIF/SVG

- **Shared Pounamu Server**
  - Process Update
  - Get Diagram Data
  - Shared Diagrams

1. Get Diagram Data
2. Process Update
3. Shared Diagrams
4. Regenerate HTML/GIF/SVG
5. Generate Update Message
Example

Highlight the target model element

The model element whose properties need to be set is shown below. However, at all:

Model element property setting:

You have selected the model element class Video under the PounamuView Diagram 2 for property setting:

<table>
<thead>
<tr>
<th>PropertyName</th>
<th>PropertyType</th>
<th>PropertyValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Integer</td>
<td>123</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>Video</td>
</tr>
<tr>
<td>cost</td>
<td>Double</td>
<td>10.5</td>
</tr>
</tbody>
</table>

SubmitChange  CancelChange

Visual Properties

<table>
<thead>
<tr>
<th>PropertyName</th>
<th>PropertyType</th>
<th>PropertyValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Video</td>
</tr>
</tbody>
</table>
Moving a shape...
2. Multi-cached edits

Diagram

Browsers

Web Server

Pounamu/Thin Servlets

Update servlet’s diagram model

SVG Diagram Model

Get Diagram Data

Process Update

Shared Diagrams

Shared Pounamu Server

(1)

(2)

(3)

(4)

(5)

(6)

(7)
Example

Move the entity element to a new location of the Ponnannu view diagram:

You are going to move the entity shape element class $Rental$ to a new location (361,456) in the Ponnannu view diagram class diagram.
3. Client-side drag-and-drop
Example

Video
- id: Integer
- title: String
- cost: Double
- fnv::Video()
- addVideo()
- rentVideo()

Rental
- dateRented: Date
- numDays: Integer
- addRental()
- returnVideo()

Staff
- name: String
- addStaff()
- findStaff()
- updateStaff()
Architecture

Pounamu/Thin Web Server Host(s)

Apache Web Server
Diagramming Servlets

GIF-based render JavaBeans
SVG-based render JavaBeans
SVG Edit Cache
SVG Data

RMI Comms.
Editing Command Processing JavaBeans

RMI API

Pounamu Host

Pounamu Tool

Edit Commands

GIF XML

RMI XML GIF

Meta-tool specs.

Model

Saved Data (XML)

Tool specs. (XML)

Other Tools

SVG Plug-in

Web Browsers

SVG-based render JavaBeans

GIF-based render JavaBeans

ECMA

HTML

GIF

SVG

JAVA

ECMA

JAVA

GIF

SVG

JAVA

ECMA

JAVA

GIF

SVG

JAVA
Evaluation

- 9 users - 4 experienced industry UML designers + 5 post-grads & academics
- 3 groups of 3; set of single and multi-user design tasks
- Each group performed these using different versions (server-side, buffered edits, client-side edits)
- Single user tasks:
  - Generally positive feedback on tools; client-side edits definitely preferred; some limitations of tools themselves were problematic for survey
- Multi-user tasks:
  - Feedback was not so positive, in part due to lack of awareness capabilities in prototype
Conclusions and Future Research

• Possible to add plug-in to existing (meta-)design tool to support web-based diagramming
• Users like some “conventional” facilities like client-side drag-and-drop, as well as web-based page metaphor
• Ideally want meta-capabilities in web browser too

• Adding other rendering support e.g. VRML for 3D navigation and interaction
• Developed mobile device version using Nokia MUPE MIDP2.0 client handset plug-in
• Want to improve collaboration support - adapting set of thick-client editing plug-ins to support this
References