Java-based web-apps with the Rich Ajax Platform (RAP)

Elias Volanakis
evolanakis@innoopract.com
Eclipse Rich Ajax Platform Project (RAP)

Goal: develop rich Ajax-enabled web applications by:

- coding in Java
- using the full Java libraries
- using Java-based Widget Toolkit
- component-oriented development model (plug-ins / OSGi)
- extensible (xml-meta info + java-code)
- without hand-coding Ajax

- skill preservation – leverage existing Java / Eclipse know-how
- single sourcing of rich client and rich internet applications
Who is it for?

- Java Developers
- Eclipse Developers

(willing to learn some Eclipse concepts)
User's View of a RAP Application

![Diagram of RAP Startup Page - Mozilla Firefox](image)

- **Revenue (in Millions)**
  - Classic Cars: 3.85
  - Motorcycles: 1.12
  - Planes: 0.96
  - Ships: 1.80
  - Trains: 0.66
  - Trucks and Buses: 0.19
  - Vintage Cars: 1.02

- **Views**:
  - View I: Root, Locate in browser view, Parent 2, Child X - filter me!
  - View II: Selection View, Browser
  - View III: Root
  - View IV: Column 0, Column 1, Column 2, Column 3, Column 4, Column 5
  - View V: Item 0-0, Item 0-1, Item 0-2, Item 0-3, Item 0-4, Item 0-5, Item 1-0, Item 1-1, Item 1-2, Item 1-3, Item 1-4, Item 1-5, Item 2-0, Item 2-1, Item 2-2, Item 2-3, Item 2-4, Item 2-5, Item 3-0, Item 3-1, Item 3-2, Item 3-3, Item 3-4, Item 3-5

© 2006, 2007 Innoopract Inc – made available under the EPL 1.0
Developer's View of a RAP Application

- developer does not get in touch with Javascript, CSS, HTTP
Developer's View of a RAP Application

![Diagram of Eclipse RAP Extensions]

- All Extensions:
  - org.eclipse.rap.ui.workbench.entrypoint
  - org.eclipse.rap.ui.workbench.perspectives
  - org.eclipse.rap.ui.workbench.views
    - Explorer (view)
    - Map (view)
  - Search People (view)
  - Intro (view)
  - org.eclipse.equinox.http.registry.resources
  - org.eclipse.equinox.http.registry.resources
  - org.eclipse.rap.ui.workbench.resources
  - org.eclipse.rap.ui.workbench.phaselistener

- Extension Element Details:
  - id*: org.eclipse.rap.showcase.Map
  - name*: Map
  - class*: org.eclipse.rap.showcase.
    - Browse...
  - category:
  - icon:
    - icons/internal_browser.gif
    - Browse...
  - fastViewWidthRatio:
  - allowMultiple:
Developer's View of a RAP Application

```
invisibleRoot = new TreeParent( "" );
invisibleRoot.addChild( root );
```

```
public void createPartControl( final Composite parent ) {
    viewer = new TreeViewer( parent );
    viewer.setC
    viewer.set
    viewer.add
    getSite();
}
```

```
public void doubleClick(DoubleClickEvent event) {
    MessageDialog.openInformation(viewer.getTree().getShell(), "Treeviewer", "You doubleclicked on " + event.getSelection().toString());
```
How does it work?

- RAP Widget Toolkit (RWT) - port of the Standard Widget Toolkit (SWT)
- server-centric - logic resides on the server
- UI is rendered by emitting browser-side JavaScript at run-time
- user action triggers a state-update (example button was clicked)
- state is recomputed on server
- only the “delta” is transmitted (very small)
RAP Architecture Overview

- subset of SWT, JFace, Workbench APIs
- RWT uses qooxdoo java script library to render widgets on the client
- differences
  - multi-user environment
  - OSGi bundles are shared between sessions
RWT Controls

- Most basic widgets
- Trees
- Tables

- See Controls Demo
RWT Layouts

- Layouts are computed on the server
- All usual layouts
  - GridLayout
  - RowLayout
  - FillLayout
  - StackLayout
  - ...
- mostly 1:1 copies of SWT
- work exactly as in SWT
RWT Events and Listeners

- typed and untyped events and listeners
  - SelectionListener
  - ModifyListener
  - VerifyListener
  - MenuListener
  - ...

- some events not possible due to latency limitations of web environment
  - key-up/down, mouse-up/down, mouse-move

- modify events collect consecutive keystrokes
Hello World in RAP

```java
public class RWTHello implements IEntryPoint {

    public Display createUI() {
        Display result = new Display();
        final Shell shell = new Shell( result );
        RowLayout layout = new RowLayout();
        layout.justify = true;
        layout.pack = true;
        shell.setLayout( layout );
        Label label = new Label( shell, SWT.CENTER );
        label.setText( "Hello, World!" );
        shell.pack();
        shell.open();
        return result;
    }
}

<extension
    id="org.eclipse.rap.demo.demoentrypoint"
    point="org.eclipse.rap.ui.workbench.entrypoint">
<entrypoint
    class="org.eclipse.rap.tutorial.HelloWorld"
    parameter="default"/>
</extension>
```
The RWT Lifecycle Phases

- **PrepareUIRoot**: create / retrieve display
- **ReadData**: apply request data
- **Render**: create javascript output
- **ProcessAction**: trigger event handling

- at the end of the ReadData phase all widget attributes are in sync with the client
- these attributes are preserved for later comparison
- during the ProcessAction phase attribute changes may occur
- the Render phase compares the widget attributes with their preserved values and submits only the delta to the client
RWT Architecture

- SWT-like API
- composition of widgets into a component tree
- AJAX engine – based on qooxdoo
- lifecycle management of the request
- rendering kits (life cycle adapter – LCA)
- user-defined components
Cool Stuff - Workbench

- Views
  - Outline
  - Properties
- Editors
  - Multi-page
- Perspectives
  - Perspective Switcher
- Menus
- Framework
Cool Stuff – Viewers

- MVC like approach for managing Table and Tree Widgets
- Viewer:
  - manages Widget
  - has Input (your model)
  - has LabelProvider – generates Strings and Icons
  - has ContentProvider – generates structure
Cool Stuff – Wizard Framework

- WizardDialog shows a Wizard
- Wizard has
  - one to n pages
  - maintains state (current page, can switch, can finish)
- WizardPage
  - UI for one page
  - State for one page (isComplete)
Cool Stuff – Jobs / Progress

- General support of background jobs
- Support for long-running tasks in the UI
- Progress View
Cool Stuff - Forms

- Eclipse UI Forms
  - 2D / flat widgets
  - SWT widgets with customized look and feel
  - RAP does not yet completely mimic original implementation
Cool Stuff – Theming

- Allow for a fresh look of web applications
- Predefined widget properties can be customized
- Format: simple Java .properties file
Example - Themed Application

Web Standard Tools (WST) Project
- Project runtime
- Note from Yoxos:
  - Eclipse Web Services Toolset (WST)
- See also:
  - Eclipse.org

Eclipse C/C++ Development Tools
- Project runtime
- Note from Yoxos:
  - C/C++ Development Tool (CDT)
- See also:
  - Eclipse.org

Eclipse Platform
- Project runtime
- Note from Yoxos:
  - Eclipse Platform
- See also:
  - Eclipse.org

J2EE Standard Tools (JST) Project
- Project runtime
- Note from Yoxos:
  - JST - J2EE Standard Tools
- See also:
  - Eclipse.org

Feature Tags
- Database
- Desktop
- Development
- Essential
- Europa
- GEF
- Java
- JST
- Languages
- Mobile
- Modeling
- NLS
- Parser
- PeDe
- PHP
- Plugin
- PTP
- Python
- Quality
- Rel-0.0.2-0
- Reporting
- Ruby
- Runtime
- Science
- SCM
- SDK
- Sources
- Tar
- Telnet
- Terminal
- Testing
- TM
- Tools
- Web
- WebpageEditor
- Weiguang
- XLC
- Yellow
- Yoxos
- ZIP

Select the platform for your download:
- Windows
- Linux
- Mac OS X

Legend:
- Included in your download
- Only available when logged in
- Not available for the selected platform

Start Download

Building your download...: (78%)
Cool Stuff – Custom Widgets

- Like SWT this requires good knowledge of the platform
- Component Developer needs Javascript, Qooxdoo and RAP knowledge
- Application Developer simply uses Java API

```java
GMap map = new GMap( shell, SWT.NONE );
map.setAddress( "..." );
```
- Tutorial in RAP Help
Scalability and Performance

- RAP has a per-session memory requirement on the server
- Based on optimization experience from more than 5 years of working on W4T (RAPs ancestor)

- Web Workbench performance tests
  - Core Duo CPU, 500 MB of heap
  - 500 concurrent users, one request every five seconds
  - Average response times below 20ms
  - Approx. 100 MB heap space

- W4T-based Yoxos on Demand server several thousand users every day
RAP Deployment

- Complies with JEE servlet standards 2.3, 2.4 and 2.5
- Applications can be deployed as standard web archives (.war)
- Eclipse is running inside a web app
  - Servlet interaction through Equinox “Servlet Bridge”
  - Eclipse runs once per Web-App
- Alternatively a web service (i.e. jetty) can be started as a plug-in
Who's contributing to development?

- **Innoopract**
  - leads RAP development
  - Eclipse specialist – RAP / Eclipse Consulting, Training
  - makers of the Yoxos Eclipse Distribution

- **CAS**
  - leading expert for customer relationship management (CRM) in the SME sector

- **1and1**
  - ISP - world's largest web host by known servers
Releases

- Version 1.0 - released this week

- Version 1.1 - targeting February 2008
  - adding a couple of things that didn't make it into 1.0
    - internalization
    - drag & drop
    - ...

- Version 1.1 - targeting the June 2008 release train
What others are saying...

- RAP is hot…RAP is pretty innovative and definitely scores very high on the “wicked cool” scale. If you haven’t already, you should take a look. - Wayne Beaton

- Web Development is so 90’s…the Rich Ajax Platform (RAP) project … is doing stuff that is so 2010 - Chris Aniszcyk

- "RAP is very easy if you have skills in Eclipse / RCP technology. Even if you have developed Java desktop applications, RAP has a lot of similar concepts" - Roberto Sanchez Custodio, Autonomind

- “Using the OSGi component architecture for our Browser-based applications allows us to quickly create and deploy customizable applications” - Ludwig Neer, CAS Software

RAP Resources

Give it a try!

- Project Homepage: [www.eclipse.org/rap/](http://www.eclipse.org/rap/)
- qooxdoo: [qooxdoo.org](http://qooxdoo.org)

Thanks for attending!

- Contact Info:

  Elias Volanakis  
evolanakis@innoopract.com  

  Innoopract Inc  
  351 NW 12th Avenue  
  Portland, OR 97209