Eclipse RCP

Jeff McAffer and Nick Edgar
IBM Rational Software
Ottawa, Canada
Eclipse Platform Committers
Overview

- The Story of RCP
- Example applications
- Why RCP?
- What’s new and cool in 3.1?
- Demos
- Summary
Eclipse RCP 2.1  Monster Garage
Eclipse RCP 3.0  What Not To Wear
Eclipse RCP 3.1  The Amazing Race

Eclipse is not just for tools anymore!
What is RCP?

While the Eclipse Platform is designed to serve as an open tools platform, it is architected so that its components can be used to build just about any client application. The minimal set of plug-ins needed to build a rich client application is collectively known as the **Rich Client Platform**.

- A subset of the full Platform
- There are other subsets
RCP is middleware

- End users don’t know about Eclipse, Java, plug-ins, …
- The implementation technology should be hidden
- RCP work effort == make this a reality for Eclipse applications
Can you tell?

RCP means you can’t tell
Example RCP apps
GumTree – Scientific Instrument Control
IBM Workplace Client Technology, Client Administrator
Maestro – NASA Space Mission Management
EclipseTrader
IBM Workplace Client Technology & Dassault
Some less conventional applications

- Kiosk – see demo
- Non-rectangular Hyperbola
- Hyperbola on PDA
- Nokia 9500 cell phone running eRCP
Non-rectangular Hyperbola
Hyperbola on a PDA
Why RCP? Our View

- Rich set of functionality
- Pervasive Plug-in architecture
- Extensible
  - Extensions and Extension Points
  - Workbench provides many extension points
- Scalable
  - Supports large products (like RAD, Lotus Workplace)
  - Scales down to embedded devices (eRCP, JCL/Foundation)
  - Progressive exposure to functionality (perspectives, activities)
  - Aggressive laziness
- Customizable (see examples)
- Dynamic
- Interoperable: COM/OLE, AWT/Swing

Middleware for Applications
Why RCP? Developer feedback : Pros

- “Slick GUI, well-thought out concepts and layouts”
- “Everyone loves the look of the products”
- “End result is more modularized and maintainable”
- “Re-using tried and true UI metaphors gets you 80% of the way”
- “Allow us to focus on the domain specific design while giving … a common style of interface in a set of diverse scenarios”
Why RCP? Developer feedback : Cons

- Fewer third party SWT libraries available
- API size/complexity – learning curve
- Poor performance on some platforms
- Security support
- Challenges using non-Eclipse libraries
## Base RCP components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSGi</td>
<td>Plug-in/bundle management, classloading, services, …</td>
</tr>
<tr>
<td>Runtime</td>
<td>Extension registry, jobs, preferences, content types,</td>
</tr>
<tr>
<td>SWT</td>
<td>Native widget toolkit, portable API</td>
</tr>
<tr>
<td>JFace</td>
<td>UI framework, viewers, wizards, dialogs, actions, preferences, …</td>
</tr>
<tr>
<td>Workbench</td>
<td>Extensible UI platform, extension points for views, editors, perspectives, action sets, wizards, preference pages, …</td>
</tr>
</tbody>
</table>
### Optional RCP components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help UI</td>
<td>Web-app-based Help UI</td>
</tr>
<tr>
<td>Update Manager</td>
<td>Discover and install new and updated versions of plug-ins</td>
</tr>
<tr>
<td>Text</td>
<td>Framework for high-function text editors</td>
</tr>
<tr>
<td>Forms</td>
<td>Forms-based control library</td>
</tr>
<tr>
<td>Welcome Page / Intro</td>
<td>Initial welcome experience and guided assistance</td>
</tr>
<tr>
<td>Cheat Sheets</td>
<td>Guides the user through a long-running, multi-step task such as a tutorial</td>
</tr>
</tbody>
</table>
### Optional RCP components (cont’d)

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical Editing Framework (GEF)</td>
<td>Includes Draw2D, a vector graphics framework</td>
</tr>
<tr>
<td>Eclipse Modeling Framework (EMF) and Service Data Objects (SDO)</td>
<td>EMF: Modeling framework and code generation facility based on a structured data model. SDO: Simplifies/unifies data application development in a service oriented architecture (SOA).</td>
</tr>
</tbody>
</table>

Lots of other components from Eclipse.org and elsewhere
What’s new (and cool) in 3.1? Tooling

- RCP application templates
- Product tooling
  - Wizard
  - Editor
  - Simplified deployment, now with launcher branding
- Delta pack of all platform-specific fragments
- Named targets
- Bundle tooling
  - Manifest validators
  - Improved classpath support
What’s new (and cool) in 3.1? UI

- SWT
  - New widgets (Spinner, Link)
  - Native multi-column Tree
  - Reorderable Table columns
  - Enhanced Browser support (scripting)
  - Advanced graphics

- Workbench
  - Advisor API refactored
  - Perspective lifecycle and shortcuts API
  - Wizard and view registry API
  - Custom backdrop
  - Undo/Redo framework
  - Nested parts

- Help
  - Integrated Help view
What’s new (and cool) in 3.1? Runtime

- Enhanced OSGi support
  - Bundle permissions
  - Improved modularity
- Java Web Start (JNLP) support
- Plug-ins as jars
- Security
  - Jar signing
Demo
Futures: Us

- RCP started by the community – continue to engage the community
- Security work in Equinox
- More tooling
- Workbench usability improvements
- More flexible layouts
- Further enable reusable components through improved decoupling
Futures: You

- Go forth and build…
- Let us know what features are important to you
  - enhancement requests, and voting in Bugzilla
- Participate in the community
  - eclipse.platform.rcp newsgroup
  - comment on enhancement requests and bug reports
  - provide patches
- Think about how to componentize your application
- Think Platform: Consider building not just a closed-form application, but a platform for your industry (and a community around it)
- Post your application info on the eclipse.org community page
  (eclipse.org > community > rcp applications)
RCP Outreach to other Eclipse.org projects

- Embedded RCP (eRCP)
- Eclipse Trust Framework (ETF)
- Eclipse Communication Framework (ECF)
- Business Intelligence and Reporting Tools (BIRT)
- Data Tools Project (DTP)
- Desktop Integration Project
- Other Eclipse.org projects

- Eclipse.org Healthcare initiative
Summary

- RCP is middleware for applications
- Middleware frees you to focus on your domain
- Everywhere
- RCP coming of age
- In use today
- Community driven

Eclipse is not just for tools anymore!
References

- eclipse plug-in central - [http://www.eclipseplugincentral.com](http://www.eclipseplugincentral.com)
- eclipsepowered - [http://www.eclipsepowered.org](http://www.eclipsepowered.org)