Eclipse-based Applications: Java on the Desktop Revisited

EclipseCon - Rich Client Platform Track
Tuesday, February 3rd, 3:45pm-4:30pm

Todd Williams, Genuitec LLC
Paul May, AlterPoint Inc
Giovanni Farris, AlterPoint Inc
Agenda

• Part I – When and Why to Use Eclipse as an Application Framework

• Part II – Case Study: Building an Eclipse-based Product

• Part III – RCP: A Big Step Forward

• Part IV – Questions and Answers
Choosing a Client Type

• When to Use Thin Clients
  ▪ Large, Distributed User Base
    – No Installation Required
    – Functional Updates are Automatic
  ▪ No Control Over Runtime Environment
    – Browsers Run on All Business Platforms
  ▪ Need High Learnability – No Training Provided
    – Fill Out Forms, Click Links
    – Singular, Static View of Information
    – Desired Interaction Model Fits Browser Paradigm
    – Applications are for Occasional Use
Choosing a Client Type

• When to Use Rich Clients
  ▪ Usability is Key
    – High-demand Productivity / Data Entry Applications
    – Use Local Caching to Minimize Server Interaction
    – Background Server Operations to Hide Latency
    – Multiple, Simultaneous Views
  ▪ High Degree of User Acceptance
    – Behaves Like a Desktop Application (Drag and Drop)
    – Client Has Full Access to All Local Capabilities
  ▪ Extensibility and Flexibility are Important
    – Effectively Anything is Possible
Why a New Approach?

• Current Issues With Desktop Java
  ▪ User’s Don’t Embrace ‘Almost Native’ UI’s
    – Don’t adapt to new UI skins
  ▪ User Interaction Slower than with Native Apps
  ▪ Large Resource Requirements
  ▪ Developers Must Build Their Own Infrastructure
    – Development is Expensive and Time-Consuming
    – Dearth of Well-Supported Java Application Frameworks
Application Frameworks

• Advantages of Frameworks
  ▪ Accelerate Development
    – Provides an Implementation of an Architecture
    – Provides Blueprints to Follow
    – Allows Focus on Business Functionality
  ▪ Decrease Development Costs
    – Requires a Smaller Team
    – Infrastructure May Be 80% of the Total Effort
  ▪ Increase Reliability and Maintainability
    – Facilitates Uniform Development Across the Application
    – Hides Complexity from Developers
    – Decreases the Code You Must Test and Maintain
Advantages To Using Eclipse

- Built By Companies With Framework Experience
- Implements a Clear, Cohesive Architecture
- Support For All Major Desktop Platforms
- Native UI Look & Feel
- Good 3rd Party UI Design Tool Support
- Large Set of Standard and Extended Widgets
- Extensive Text Processing Support
- User Configuration and Preferences
- Platform-specific Features (OLE, ActiveX)
- Discovery and Installation of Updates
- Internationalization and Localization
- Integrated Help System
- Books and Training Readily Available
Part II
Case Study: Building an Eclipse-based Product
AlterPoint Overview

The leading provider and innovator of Multi-vendor, Network Configuration Management (NCM) solutions

- Tracking of who is performing which configuration changes.
- Verification that the appropriate changes were made.
- Rapid identification of an outage caused by an invalid configuration change.
- Automated workflow enforcing best practice configuration changes.
- Association of hardware, software & configuration changes.
The Network Engineer’s Desktop Today
The Resulting Business Problem

Manual, unintegrated process for activities like device updates and problem diagnosis
  • Error-prone
  • Inefficient

Corporate assets are decentralized (scripts, inventory lists, configuration files).
  • Security risk
  • Prevents knowledge sharing
Initial Goals of the Project

- **Introduce unified platform that integrates the network engineer’s tools for managing device configurations.**
- **Automate the network engineer’s most common and painful activities – e.g., making changes to devices.**
  
  Example – OS upgrade for certain Cisco devices:
  
  - Search for all Cisco devices with a specific OS version and add them to target list
  - Automatically check for pre-requisites on target list
  - Execute the OS update
  - View real-time interactive feedback to ensure success of the change

- **Make the platform open and extensible so tools can be easily snapped in.**
Integrated Network Environment (INE)
Configuration Search

Search

Devices  | Configurations  | Help Search

Containing text:

access-list 117

(* = any string, ? = any character, "" = phrase,
AND, OR, NOT = boolean operators)

Configuration Type

- Current
- Draft

Inventory Scope

Inventory Source:

- All Categories
- Category

Customize...  | Search  | Cancel

Configure. Manage. Control.
Creating a Change Script with AutoScriptor
Creating an Update Template
Real-Time Feedback on Job Results
Lessons Learned – What Worked

- The IDE to INE paradigm.
- Extensive functionality: Decorators, contribution model, workspace, search, perspectives, rich-editor support, preferences, property sheets, etc.
- Plug-in model for building a platform of integrated tools for the network practitioner.
- Usability and responsiveness of native platform widgets.
- Multi-platform support.
- Partnership with Genuitec.
Lessons Learned...Challenges We Faced

- Platform has strong ties to the file system. Problem for remote Java objects accessed from an app server.
- Assumes your building an IDE. Required removal of some base Eclipse plugins (e.g., Build menu)
- Significant learning curve due to size of the platform.
- Difficulty in testing an integrated set of tools as opposed to a specific application workflow.
- Market education – “Why should we used a rich-client (non-web) interface?”
Next Steps

Add additional applications on top of the platform

- Enforcement
- Grammar/syntax checking

Extend the ecosystem through integrations with third party solutions

Examples:

- Fault/performance monitoring (MIB browsing)
- Vulnerability remediation
- Configuration analysis (security, routing, etc.)

Integrate with other inventory sources
Part III
RCP: A Big Step Forward
What’s Easier With 3.0 RCP

• Enhancements Over Eclipse 2.1
  ▪ IDE Decoupled from Workbench Platform
    – No Code Edits to Excise IDE Functionality
    – Still Based On Perspectives, Views, Editors
  ▪ Enhanced UI Configurability
    – Policy Support: i.e. Toolbar for Workbench Window?
    – Menu Management is Fully Dynamic
    – More Branding Support
  ▪ SWT/Swing Interoperability (Enhancement #37724)
    – Portions in Place, More Coming
  ▪ Tabular & Tree Data (Enhancement #37998)
    – Greatly Increased Capacity and Capability
  ▪ Much More – See the Other RCP Track Presentations
Eclipse As A Framework Now

- Remaining Challenges
  - Eclipse Frameworks are Still Complex
    - Steep Learning Curve
    - Books, Training, and Consulting are Available
  - Physical Resource Requirements ([Enhancement #37723](#))
    - No Support for Editors/Markers on Logical Resources
  - No Built-in Security Model ([Enhancement #37692](#))
    - User / Group / Capability Management
    - Hard to Tailor UI Presentation By Role
  - No SWT Charting Capabilities
    - But Embedded Browser Can Be Used for Graphics
  - Work In Progress Until June 2004
Part IV
Questions and Answers
### Additional Information

<table>
<thead>
<tr>
<th>Todd Williams</th>
<th>Paul May</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP – Technology,</td>
<td>Director of Business Development,</td>
</tr>
<tr>
<td>Genuitec LLC</td>
<td>AlterPoint Inc</td>
</tr>
<tr>
<td><a href="http://www.genuitec.com">todd@genuitec.com</a></td>
<td><a href="http://www.alterpoint.com">pmay@alterpoint.com</a></td>
</tr>
<tr>
<td>972-691-5717</td>
<td>512-536-8318</td>
</tr>
</tbody>
</table>

### References

*The Case For Using Eclipse Technology in General Purpose Applications*, Todd Williams & Marc Erickson, July 2002