Eclipse for Smalltalkers

Eric Clayberg
Sr. Vice President of Product Development
Instantiations, Inc.
April 22, 2002

clayberg@instantiations.com
http://www.instantiations.com
978-750-3621
Who Am I?

- First used Smalltalk in late ‘80s; full-time since 1991;
  First used Java in 1996; Eclipse in 2000
- Co-Founder of (the original) ObjectShare in 1992;
  V.P. of Product Development for ParcPlace-Digitalk 1996-97
  V.P. of Product Development for Instantiations since 1997
- Developer & Chief Architect of WindowBuilder Pro and over a dozen other
  commercial Smalltalk add-on products (VA Assist Pro, WidgetKits, etc.)

Instantiations

- Multi-faceted Software Company - founded in 1997
  (out of the ashes of ParcPlace, Digitalk and ObjectShare)
- Advanced Tier IBM Business Partner;
  Winner of the 2001 IBM Solutions Excellence Award for “Cool Tool”
  (for a product built with Smalltalk)
- VisualAge Smalltalk product line: VA Assist Pro, WindowBuilder Pro,
  WidgetKits, VSE to VAST Translation Tool, GF/ST; Also known for: JOVE,
  VA Assist/J, jFactor, jKits, CodePro Studio for WebSphere & JBuilder
- Built some of the very first Eclipse demos used by IBM
Agenda

- What is Eclipse?
- Eclipse vs. Smalltalk
- Eclipse Demo
What is Eclipse?

“An IDE for anything, and for nothing in particular”

Eclipse Platform Technical Overview, OTI 2001

The Eclipse Project is an open source software development project dedicated to providing a robust, full-featured, commercial-quality, industry platform for the development of highly integrated tools.

The Eclipse Platform is an open extensible IDE. “Eclipse” provides building blocks and a foundation for constructing and running integrated software development tools. It is composed of three primary sub-components: Platform, Java Development Tools (JDT), and Plug-in Development Environment (PDE).
What is WebSphere Studio Workbench?

IBM WebSphere Studio Workbench is based on the Eclipse platform and is fully supported by IBM. This is the basis for all of IBM’s Eclipse-based commercial tools.

IBM WebSphere Studio Application Developer is a J2EE application development tool optimized for the WebSphere software platform. Application Developer represents the next generation of IBM offerings and is the successor to VisualAge for Java Enterprise.
Eclipse Overview

- Created by OTI & IBM – the same folks who brought you ENVY and VisualAge Smalltalk
- The technology is organized around a workbench into which any number of tools can be installed.
- The workbench provides the run-time context in which these tools run.
- Tools generally operate on the tree of resources managed by the workbench.
- Resources are regular files and folders, arranged into larger units called projects.
- The desktop is built on top of the platform to provide a general-purpose graphical user interface (GUI) to the workbench and its resources.
- Tools integrate seamlessly into the platform by contributing buttons, menus, viewers, and other GUI elements to the desktop.
Key Components

- **Platform Services**
  - Resource Management
  - Team Programming Model
  - Interoperability with non-WebSphere Tools
  - Debugging
  - Extensibility Mechanism

- **Common Frameworks**
  - Standard Widget Toolkit (SWT)
  - User Interface Framework (JFace)
  - Source Editing Framework (SEF)

- **Platform Extensions**
  - Desktop
  - Java Tools
  - Web Tools
Eclipse vs. Smalltalk

- User’s Perspective
  - General Characteristics
  - Multi-Pane, MDI Interface
  - Standard IDE Features
  - Semantic Searching
  - Refactoring

- Developer’s Perspective
  - General Characteristics
  - Extension API
  - Workbench Extension Points
  - Other Extension Points
User’s Perspective

![Diagram of Eclipse IDE with labels for Workbench Window, Editor, Short cut bar, Tool bar, Menu bar, Status line, Page, Views.]

Eric Clayberg - Instantiations, Inc.   April 22, 2002   Eclipse for Smalltalkers - 9
## General Characteristics

- Eclipse is a great IDE…for Java development
  (or for any file-based language)

<table>
<thead>
<tr>
<th></th>
<th>Eclipse</th>
<th>Smalltalk IDE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Languages</strong></td>
<td>Java, HTML, XML, C++, C#, etc.</td>
<td>Smalltalk</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td>CVS, PVCS, ClearCase, etc.</td>
<td>ENVY, StORE, etc.</td>
</tr>
<tr>
<td><strong>Platforms</strong></td>
<td>Windows, Linux, Mac, Solaris, QNX, AIX</td>
<td>Everything</td>
</tr>
<tr>
<td><strong>Navigation</strong></td>
<td>File &amp; Directory based, some OO views</td>
<td>Classes &amp; Methods, very OO</td>
</tr>
<tr>
<td><strong>UI Style</strong></td>
<td>Single window, MDI-like</td>
<td>Multi-window</td>
</tr>
<tr>
<td><strong>Change Tracking</strong></td>
<td>Configurable “local history”</td>
<td>Every change saved forever (ENVY)</td>
</tr>
<tr>
<td><strong>Compilation</strong></td>
<td>Incremental or batch, limited hot code</td>
<td>Incremental with hot code</td>
</tr>
<tr>
<td></td>
<td>replacement with JDK 1.4</td>
<td>replacement</td>
</tr>
<tr>
<td><strong>Debugger</strong></td>
<td>OK</td>
<td>Excellent</td>
</tr>
<tr>
<td><strong>Customization</strong></td>
<td>Limited to exposed preferences and plugin</td>
<td>Virtually unlimited for a</td>
</tr>
<tr>
<td></td>
<td>points</td>
<td>moderately skilled developer</td>
</tr>
<tr>
<td><strong>Deployment</strong></td>
<td>Easy, well defined</td>
<td>Problematic (IDE dependent)</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td>JUnit</td>
<td>SUnit</td>
</tr>
</tbody>
</table>
Multi-Pane, MDI Interface
Standard IDE Features

- Eclipse includes most “modern” IDE features like drag/drop, syntax coloring, code assist, etc. Does not include a GUI builder.
- Eclipse also includes most “standard” Smalltalk IDE features like true semantic searching (e.g., senders & implementors with various scoping options)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Eclipse</th>
<th>Smalltalk IDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drag &amp; Drop</td>
<td>Yes</td>
<td>Yes (IDE dependent)</td>
</tr>
<tr>
<td>Syntax Coloring</td>
<td>Yes</td>
<td>Yes (IDE dependent)</td>
</tr>
<tr>
<td>Formatter</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Code Assist</td>
<td>Yes</td>
<td>Limited</td>
</tr>
<tr>
<td>Macros</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Workspace/Scrapbook</td>
<td>Limited, scoping problems</td>
<td>Just “do it” anywhere</td>
</tr>
<tr>
<td>GUI Builder</td>
<td>Not yet</td>
<td>Great choices</td>
</tr>
<tr>
<td>Searching</td>
<td>Very powerful, semantic-based, limited to exposed capabilities</td>
<td>Very powerful, semantic-based, easily scripted by user</td>
</tr>
</tbody>
</table>
Semantic Searching

- Declaration and Reference searches for types, methods and fields
- Additional Read and Write Access searches for fields
Refactoring

- Eclipse includes extensive built-in refactoring capabilities with a comprehensive API.
- The Refactoring Browser is available for most Smalltalk dialects and includes a very nice scriptable replacement engine.

<table>
<thead>
<tr>
<th>Refactoring Feature</th>
<th>Eclipse</th>
<th>Smalltalk IDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename Class/InstVar/Method</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>Modify Method Parameters</td>
<td>Rename,</td>
<td>Rename, Add</td>
</tr>
<tr>
<td></td>
<td>Rearrange</td>
<td></td>
</tr>
<tr>
<td>Extract/Inline Method</td>
<td>Extract Only</td>
<td>Both</td>
</tr>
<tr>
<td>Inline/Extract Local Variable</td>
<td>Both</td>
<td>Both</td>
</tr>
<tr>
<td>Pull Up/Push Down InstVar/Method</td>
<td>Pull Up Only</td>
<td>Both</td>
</tr>
<tr>
<td>Encapsulate/Inline InstVar</td>
<td>Encapsulate Only</td>
<td>Both</td>
</tr>
</tbody>
</table>
Demo – User’s Perspective
General Characteristics

- Eclipse is a well-designed, well-factored, OO environment
- Most of its weaknesses are due to Java language restrictions (had it been implemented in Smalltalk, it would have been amazing!)
- Based on the concept of layered, dependent plugins

<table>
<thead>
<tr>
<th></th>
<th>Eclipse</th>
<th>Smalltalk IDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>File-Based</td>
<td>Image-Based</td>
</tr>
<tr>
<td>Source Access</td>
<td>Full Source</td>
<td>Full Source</td>
</tr>
<tr>
<td>Modularity</td>
<td>Highly Modular, dynamically built on available plugins</td>
<td>Tends to be monolithic</td>
</tr>
<tr>
<td>Plugin Coupling</td>
<td>Very little, good isolation provided, no chance of collision</td>
<td>Highly coupled, easy for one plugin to kill entire IDE, easy for tools to collide</td>
</tr>
<tr>
<td>Visibility</td>
<td>Controlled by plugins and special class loaders</td>
<td>All classes are visible</td>
</tr>
<tr>
<td>Reflection</td>
<td>Weak</td>
<td>Highly Capable</td>
</tr>
<tr>
<td>Experimentation</td>
<td>Hard, must build complete plugin</td>
<td>Just “do it” anywhere</td>
</tr>
</tbody>
</table>
## Extension API

- Eclipse provides a full-featured API
- Plugins are written in Java and integrated via XML
- Plugins aren’t activated until actually needed

### Comparison Table

<table>
<thead>
<tr>
<th></th>
<th>Eclipse</th>
<th>Smalltalk IDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extension Language</td>
<td>Java, XML</td>
<td>Smalltalk</td>
</tr>
<tr>
<td>Extensibility</td>
<td>Rigid, Formal, Well defined, Limited</td>
<td>Easy, Ad-hoc, Unlimited</td>
</tr>
<tr>
<td>API Access</td>
<td>Limited access (much is marked private or internal)</td>
<td><strong>Unlimited access, fully public</strong></td>
</tr>
<tr>
<td>UI Library</td>
<td>Native, SWT, JFace, ActiveX, no LCD</td>
<td>Native or Emulated (Dialect dependent), no LCD</td>
</tr>
<tr>
<td>Plugin Activation</td>
<td>Lazy (no startup hook)</td>
<td><strong>Startup hook available</strong></td>
</tr>
</tbody>
</table>

```xml
<objectContribution
  objectClass="org.eclipse.core.resources.IFile"
  nameFilter="*.java" id="com.foo.openMyJavaEditor">
  <action label="Edit File" icon="images/my_editor.gif"
    id="com.foo.openMyEditorAction"
    class="com.foo.OpenMyEditorActionDelegate"
    enablesFor="1" menuBarPath="additions"
    helpContextId="my_editor_action_context">
  </action>
</objectContribution>
```
Workbench Extension Points

- Workbench
  - **Action Sets** - used to add menus, menu items and toolbar buttons to the common areas in the workbench window
  - **Editors** - used to add new editors to the workbench
  - **Editor Actions** - used to add actions to the menu and toolbar for editors
  - **Export Wizards** - used to register export wizard extensions
  - **Import Wizards** - used to register import wizard extensions
  - **New Wizards** - used to register resource creation wizard extensions
  - **Perspectives** - used to add perspective factories to the workbench
  - **Perspective Extensions** - used to extend perspectives
  - **Popup Menus** - used to add new actions to pop-up menus
  - **Preference Pages** - used to add pages to the preference dialog box
  - **Property Pages** - used to add additional property page for objects of a given type
  - **Views** – used to define additional views for the workbench
  - **View Actions** - used to add actions to the menu and toolbar for views
Other Extension Points

- Workspace
  - Builders - The job of a builder is to process a set of resource changes
  - Markers - A marker is a kind of metadata (similar to properties) which can be used to tag resources with user information
  - Natures - A nature associates lifecycle behaviour with a project

- Compare – content viewers, merge viewers

- Ant – objects, tasks, types

- Targets (Repositories)

- Help
Demo – Developer’s Perspective
Conclusion

- Smalltalk is still the best platform for building tools (especially IDEs)
- Smalltalk is still the better language for building highly reflexive and extensible tools
- Java, the language, is probably Eclipse’s weakest link
- The Java IDE world is catching up fast and is better in many regards
- Eclipse is a solid, well designed system representing the state-of-the-art in Java IDE design
- Eclipse is “good enough” to support the creation of a wide variety of IDE tools and facilities
- High quality semantic searching and refactoring capabilities are no longer unique to the Smalltalk world
- Smalltalk vendors should not rest on their laurels and assume that Smalltalk IDEs will always be better…
Eclipse Resources

- Eclipse & WebSphere Studio
  - Eclipse.org Home Page
  - Downloads
  - Technical Articles
    [http://www.eclipsecorner.org/articles/](http://www.eclipsecorner.org/articles/)
  - Mailing Lists
    [http://www.eclipsecorner.org/mail/](http://www.eclipsecorner.org/mail/)
  - FAQ
  - IBM WebSphere Studio Family
  - Eclipse Plugins
    [http://eclipse-plugins.2y.net/](http://eclipse-plugins.2y.net/)
  - Newsgroups
    news://www.eclipse.org/eclipse.tools
    news://news.software.ibm.com/ibm.software.websphere.studio.application-site-developer

- Other
  - Instantiations’ Web Site
  - Me ;-) email to clayberg@instantiations.com