Domain Specific Languages and Modeling Technologies for Eclipse Scripting & Automation

ECLIPSE COMMAND LANGUAGE

Andrey Platov

Santa Clara, California
March 18th, 2008
Scripting attracts minds of Eclipse engineers

- org.eclipse.scripting (inactive) plug-in – since 2001
- Eclipse Monkey project
- Scripting sessions each EclipseCon
- Glimmer project proposal - JRuby DSL
What is ECL?

- ECL is about using modeling technologies, Java, scripting languages, and DSLs for scripting Eclipse and for DSL orchestration.
- Moreover, ECL is a style of software architecture inspired by different concepts, technologies, and products like REST, SOA, MDD, and…
  Microsoft® PowerShell™
Purpose of the ECL

- Console Applications
- Web Applications
- Distributed Applications
- Mash-up Applications

ECL
ECL Program Model

```
void service(IScriptletService init)
```

xored software, Inc. Made available under EPL 1.0
March 17, 2008
Pipes and Objects

Scriptlet → PIPE
- EMF objects
- SDO graph
- JSON
- XML

Scriptlet
Scriptlet communication

- Host
- Scriptlet
- In-process communication

TCP/IP
Sample Scriptlets

- GetBundleScriptlet
  - name: String

- SortObjectScriptlet
  - property: String
  - top: int
  - ascending: boolean

xored software, Inc. Made available under EPL 1.0
March 17, 2008
Sample ECL Command

```
shell> get-bundle | sort-object -property id
```
Demo I

- Scriptlet chains
- Code injection
- Scripting languages supported by JVM
- Eat our own dogfood
Pipe content associations

- Object of type A
- Object of type B
- Object of type C
Defaul groovy template

```groovy
<TableFragment xmlns="http://www.xored.com/ecl/table.ecore">
<%
    caption = false;
    for (r in resource) {
        print "<rows>";
        if (!caption) {
            for (attr in r.eClass().getEAttributes()) {
                label = attr.getName();
                print "<columns style='border: bottom' text='"+label.toUpperCase()+"'/>";
            }
            print "</rows><rows>";
        }
        caption = true;
    }
    for (attr in r.eClass().getEAttributes()) {
        Object value = r.eGet(attr);
        if (value != null)
            print "<columns text='"+xmlEncoder.escapeXML(value)+"'/>";
        else
            print "<columns text='"/>";
    }
    println "</rows>";
}>
</TableFragment>
```
Demo II (Web & Mash-ups)

- Rendering to HTML
- Rendering Eclipse UI with UI DSLs
- Legacy code support - RSS Feed Reader
Thank you!

- Please contact us if interested
- E-mail: andrey@xored.com
- Source code will be available at http://www.xored.com/ecl
Why scripting languages attract developer minds?

- Productivity at first
  - Python or TCL programs are often 2 times shorter (LOC) than Java or C++ equivalents. They usually takes less time (hours) to develop.

- Simplicity and Developers availability
  - There are 2.5 millions PHP programmers.
  - Most of web sites done with PHP.

- And this is true, so it looks like many of us will develop plug-ins with scripting languages; Q: Who are developing plug-ins with them?
Domain Specific Languages

- There are several XML-based DSLs exist for building Eclipse UI
  - XSWT: [http://www.coconut-palm-software.com/the_visual_editor/?page_id=58](http://www.coconut-palm-software.com/the_visual_editor/?page_id=58)
- Some of these DSLs pairing with scripting languages like DHTML/JavaScript
- The truth is that declarative UI significantly increases productivity for UI developers, Q: Who are developing eclipse UI with declarative languages (besides authors of mentioned projects)?
Right tools for right tasks

- If I need to develop a simple database browsing application I’ll use PHP, do not you?
- Simplified deployment and short development cycle using declarative UI (HTML) with script embedded
- Unfortunately all of above is true until complexity of the system increased: at some point you shall expect integration problems, deployment issues, and maintain huge test-base to keep project in health
- Conclusion: We need to have clear boundaries for script responsibility
ECL – What developer wants?

- **Productivity of DSLs**
  - Perl initially designed as a DSL for textual processing
  - Declarative UI languages reduce time to develop UIs in times
  - Custom DSLs increasing productivity in specific areas

- **Simplicity**
  - There is a lack of experienced engineers on the market

- **ECL is about to have all of them: to make any scripting language and DSLs work together**