



Eclipse Command Language

Cracow, Poland

Eclipse Day 2012, September 13

Bartosz Czerwiński

bczerwinski@luxoft.com

Agenda



- DSL
- Scripting in Eclipse
- ECL
 - Core Concepts
 - Architecture
 - Applicability
 - Future
- Live Demo



DSL



Domain Specific Language

- Most languages are general purpose languages
- Language Oriented Programming *
- Domain-Specific Language (noun): *a computer programming language of limited expressiveness focused on a particular domain* *
- Why do we need DSLs?

* Martin Fowler



Types of DSLs

- Internal (Embedded) – a particular idiom of writing a code in a host language
 - Fluent Interfaces (Method Chaining)
 - Flexible Syntax
 - Method Missing
 - Reflection (Annotations)
- External – a separate language or a set of resources that facilitate the way of how a user is interacting with its environment within a particular domain



Scripting in Eclipse Platform

■ EclipseShell

- <http://eclipse-shell.sourceforge.net/>

■ Eclipse Monkey

- Task automation and scripting, API exploration

- Based on Groovy Monkey Project

- <http://www.eclipse.org/proposals/eclipse-monkey/>

■ Eclipse Glimmer

- JRuby DSL for developing user interfaces

- <http://www.eclipse.org/proposals/glimmer/>



ECL

Eclipse Command Language



```
echo „Hello World”
```




Eclipse Command Language

- Scripting and Automation Framework
- Easily Extensible & Flexible
- Command-oriented
- Distributed
- Q7 Platform scripting language
- Entirely open-source
- Introduced by Xored Software at EclipseCon 2008



Core concepts

- Model Driven Engineering
- Functional Programming
- REST
- SOA
- TCL
- Microsoft® PowerShell™
- Unix pipes



What's under the hood?

Eclipse Modeling Framework

- Meta description of domain-specific model

Commands

- Actual implementation of domain-specific logic

Pipes

- Provide means of communication between commands

Script-lets

- Abstraction layer above commands, pipes (DSL specific)

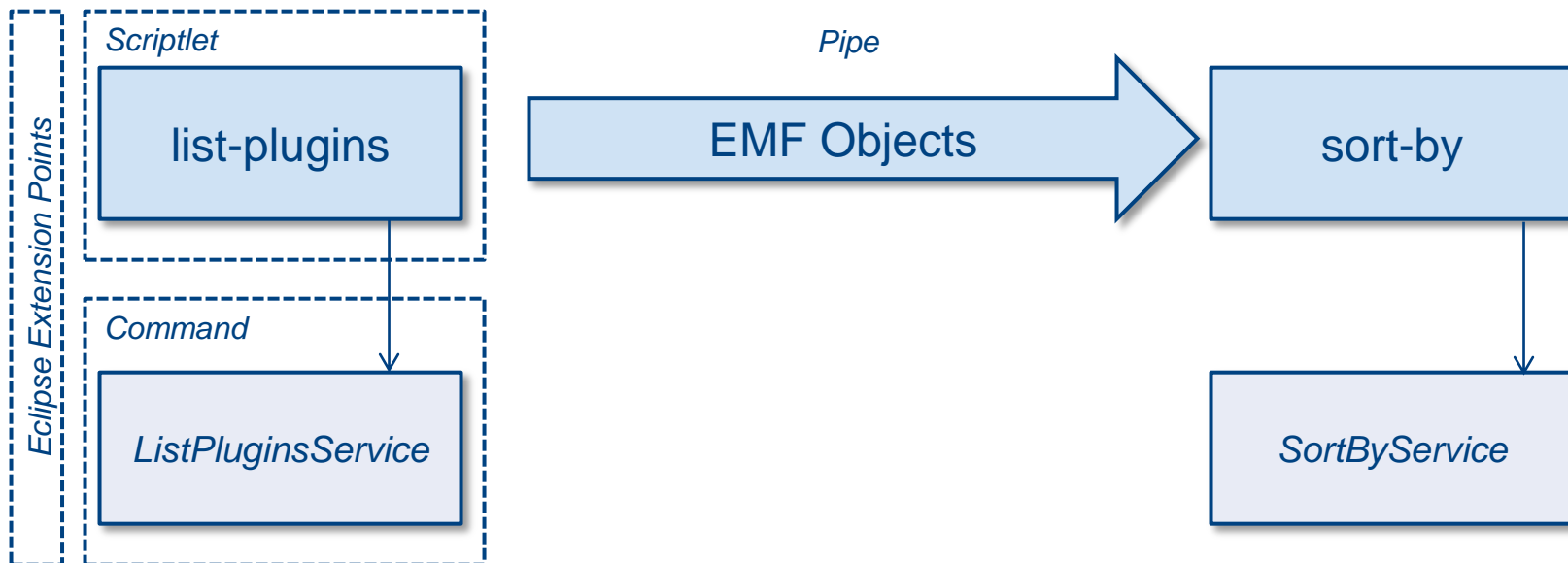
Eclipse Extension Points

- Effectively associate *Scriptlets* with commands

Scriptlet



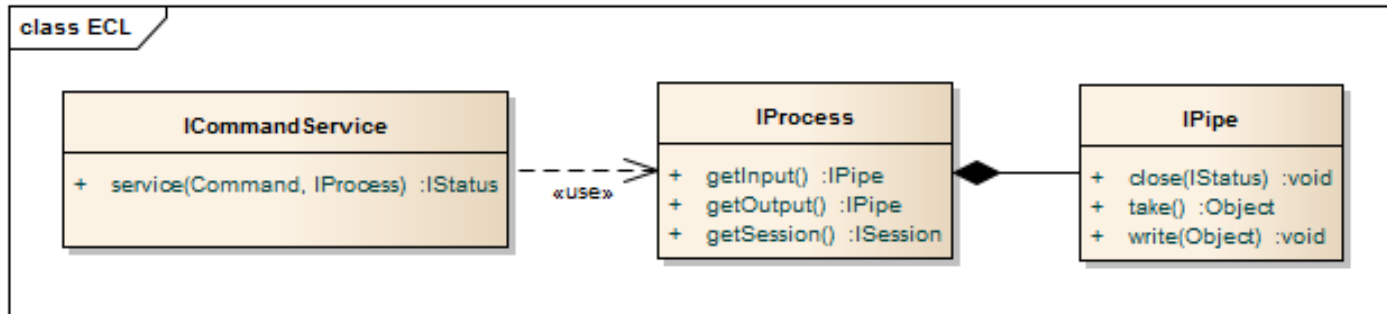
- An independent unit (Actor)
- Stateless
- Pipes are (conceptually) immutable



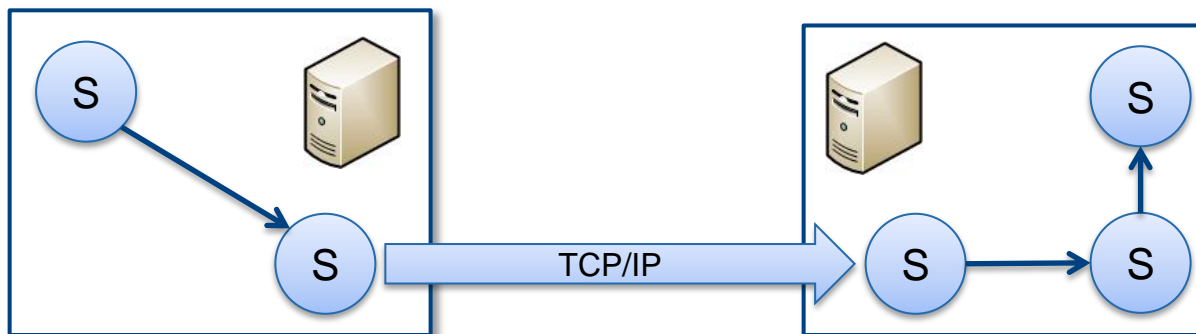
Scriptlet c.d.



■ Domain Model



■ Distributed communication



Applicability



Desktop Applications

- Scripting & Automation
- Instrumentation
- Productivity
- Issue Tracking
- Issue Investigation
- DSL Toolkits

Web Applications

- Scripting & Automation
- Mash-ups

Distributed Systems

- Scripting & Automation
- Remote Control
- Remote Support
- DSL Admin Toolkits



Demo 1

Telnet Console



- Core features
- Interacting remotely with Eclipse UI



Demo 2

Q7



- Functional and UI testing platform
- Handcrafted for Eclipse
- Runtime Intelligence
- State Management
- Utilizes ECL as its core scripting language

Future



■ Comprehensive DSL orchestration

```
... | get-dom-nodes „a” | foreach js:{set-attr target „_top”}
```

```
... | get-list | script python:{sorted( $_, key=itemgetter(1) )}
```

■ Truly distributed

```
get-data-sets | host1:remove-duplicates \  
                | host2:get-graphviz-layout twopi
```

■ Code Injection

```
get-data-set calls | get-node -query „self.duration < 13”
```

■ ECL core in other languages (Python, Scala, Ruby, ...) and their interoperability



Thank you!

bczerwinski@luxoft.com

Special thanks to Andrey Platov and Ivan Inozemtsev
from Xored Software Inc.



Resources

- Sources and Extensions

- <https://github.com/xored/ecl/>

- <https://github.com/xored/q7.extensions.ecl>

- API (<http://download.xored.com/q7/docs/ecl-api/1.2.4.html>)

- New Command Guide

- (<http://help.xored.com/display/ECL/New+Command+Guide>)