Builders
How MOPs Make Life Easy

Dr Russel Winder

Partner, Concertant LLP

russel.winder@concertant.com

Director, It'z Interactive Ltd

russel@itzinteractive.com

Aims and Objectives of the Session

- Inform people as to what meta-object protocols (MOPs) are.
- Show that MOPs lead to Builders.
- Show that MOPs and Builder are good things for application development.

Structure of the Session

- Introduce meta-object protocols (MOPs):
  - What they are, and why they are useful.
- Introduce domain specific languages (DSLs):
  - Internal, and external
- Examples of use of MOPs – Builders.
- Summarize the affect of MOPs on system development.
- Argue that maybe C++ should have a MOP.

Meta-object Protocol (MOP)

- Definition of a class is managed by a metaclass.
- Semantics of execution are controlled by the metaclass.

A MOP is an Extension of Dynamic Binding

- The 3 Is:
  - Inspection
  - Introspection
  - Intercession
- It's all about binding:
  - MOP allows more general activity than any other binding method.

A MOP is an Extension of Overloading

- The meaning of all symbols in an object is determined by the metaobject of the object.

Who has a MOP?

- Languages with MOPs:
  - CLOS (Common Lisp)
  - Smalltalk
  - Python
  - Ruby
  - Groovy
  - *all dynamic programming languages*

Moot point

Arguably

MOPs lead to DSLs

- Overloading, especially of operators, allows internal domain specific languages (DSLs)
- Redefining the semantics of function call is a trick that leads to Builders.

Affect of a MOP

- Language itself is fluid.
- Application design is language design.
- Remove separation of language, library and application in terms of the definition of the ‘alphabet’ of the application.
- Programming is about DSLs.

Ludwig Wittgenstein:
Logisch-Philosophische Abhandlung
(Tractus Logico-Philosphicus)
Philosophische Untersuchungen
(Philosophical Investigations)

Design Patterns

- Create a language for discussion of design possibilities.
- Enable communication at a higher level than the programming language.

Christopher Alexander, Sarah Ishikawa and Murray Silverstein:

The Timeless Way of Building
A Pattern Language: Towns, Building, Construction

The 'Gang of Four' – Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides:

Design Patterns: Elements of Reusable Software

Not to be confused with the Gang of Four: Jiang Qing, Zhang Chunqiao, Yao Wenyuan and Wang Hongwen

The Groovy MOP

- Every object has a metaboject
- The metaclass (which is the class of the metaobject) has an invokeMethod method

Look at MOP_Experiment

Builders

- A class can have invokeMethod as well.

Look at Builder_Experiment

SwingBuilder

- Build Swing interfaces without the pain.

Look at SwingBuilder and Presentation

MarkupBuilder

- MarkupBuilder, StreamingMarkupBuilder
- DOMBuilder, StreamingDOMBuilder
- SAXBuilder, StreamingSAXBuilder

AntBuilder

- Ant task scripting without the XML.
- AntBuilder is the infrastructure . . .
- . . . **Gant** is the tool !

Look at AntBuilder, Gant and some scripts.

Gant

- A way of working with Ant tasks using Groovy scripting rather than XML.
- Replaces Ant but uses Ant.
- Using programming languages is the future for build technology:
  - SCons
  - Waf
  - Rant

Gant

Summary

- If it's hierarchical, Builders are your friend.
- Builders in Groovy really rock.
- The above is patently true as Ruby and JRuby are copying the whole thing.
- Programming with MOPs is all about designing a language to express easily the solution to the problem, and then using it to do exactly that.
- DSLs are where programming is at.

The Future of Programming

- Programming with MOPs is all about designing a language to express easily the solution to the problem, and then using it to do exactly that.
- This is the future of programming.