Producing Production Quality Software

Lecture 11: Design Patterns

Prof. Arthur P. Goldberg

Fall, 2004

Design Patterns

- The Gang of Four (GoF) book: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns*, 395 pages, Addison-Wesley Pub Co; 1st edition (January 15, 1995), ISBN: 0201633612
  - Design Patterns "describe simple and elegant solutions to specific problems in OO design"
  - "Descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context".
  - http://st-www.cs.uiuc.edu/users/patterns/DPBook/DPBook.html

Objectives

- Reuse
- Flexibility
- Modularity
- Comprehensibility

OO Design is Difficult

- Systems usually require some redesign
- Reuse helps
- Design Patterns facilitate

An example: Model/View/Controller (MVC)

- History: Smalltalk-80 interfaces
- MVC
  - Model: the application
  - View: screen presentation
  - Controller: controls the UI
- Figure

Another Example: Composite

The Catalog

- All patterns have been used multiple times
- Part of OO ‘folklore’ or elements of some successful systems
- Incomplete: there are hundreds of patterns

A Pattern Summary: A Catalog Entry

- Pattern name
  - Problem the pattern solves
- An abstract description
  - The solution the pattern offers
- The pattern's general arrangement of elements (classes and objects)
  - The consequences of using the pattern
- Impact on reuse, portability and flexibility

A Pattern's Details: A Full Catalog Entry

- Pattern name and classification
- Intent
- AKA
- Motivation
- Applicability
- Structure
- Participants
- Collaborations
- Consequences
- Implementation
- Sample code
- Known uses
- Related patterns
- An example pattern description

Design Pattern Space

(GoF Table 1.1, simplest and most common patterns in *italics*)

		Purpose		
		Creational	**Structural**	**Behavioral**
Scope	Class	*Factory Method*	Adapter (class)	Interpreter *Template Method*
	Object	*Abstract Factory* Builder Prototype Singleton	*Adapter (object)* Bridge *Composite* *Decorator* Façade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento *Observer* State *Strategy* Visitor

Pattern Purposes

- Creational: create objects
- Structural: compose classes or objects into larger structures
- Behavioral: help define the communication between objects in the system and how the flow is controlled in a complex program

References

- Books
  - James W. Cooper, *Java Design Patterns: A Tutorial*, Addison-Wesley Pub Co, 2000, ISBN: 0201485397
  - Steven John Metsker, *Design Patterns Java Workbook*, 496 pages, Addison-Wesley Pub Co; 1st edition (2002), ISBN: 0201743973 (see www.oozinoz.com)
  - Mark Grand, *Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML*, Volume 1, John Wiley & Sons; 2nd edition, 2002, ISBN: 0471227293
- Articles
  - Gamma, E., *Applying Design Patterns in Java*, in Java Gems, SIGS Reference Library, 1997
- Lists of patterns
  - http://hillside.net/patterns/onlinepatterncatalog.htm
  - http://patterndigest.com/

Pattern Intent (from Metsker)

Interfaces

- Adapter, façade, composite, bridge

Responsibility

- Singleton, Observer, Mediator, Proxy, Chain of Responsibility, Flyweight

Construction

- Builder, Factory Method, Abstract Factory, Prototype, Memento

Operations

- Template Method, State, Strategy, Command, Interpreter

Extensions

- Decorator, Iterator, Visitor

Advice

Program to an interface and not to an implementation.

Favor object composition over inheritance.