CSE 403
Lecture 14

Design Patterns

Today's educational objective

- Understand the basics of design patterns
- Be able to distinguish them from design approaches such as information hiding and layering
- Be able to find patterns that meet specific needs
- Know what the Gang of Four is

Experts vs. Novices

- Experience
- Higher level thought
  - Chunking
  - Idioms
  - Techniques
  - Examples

Examples of expertise

- Chess playing
  - Experts view pieces in groups
- Mathematics
  - Integration by trigonometric substitution
- Programming

```
for (int i = 0; i < n; i++)
    a[i] = b[i];
```

Design patterns in Architecture

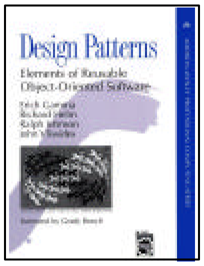
- Alexander: "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to the problem. In such a way that you can use this solution a million times over, without ever doing it the same way twice."

Design Pattern

- Pattern name: Strip mall.
- Problem: Make small commercial establishments and franchises accessible to car driving customers.
- Solution: Parking area with store fronts facing parking. Uniform construction.
- Consequences: Traffic flow, congestion, parking availability, building rents.

Gang of Four

- Gamma, Helm, Johnson, Vlissides
- Catalog of design patterns for software



Design Patterns
Elements of Reusable Object-Oriented Software

Case study

- Lexi Editor (Calder)
  - Document structure
  - Formatting
  - Embellishing UI
  - Multiple look and feel standards
  - Multiple window systems
  - User operations
  - Spelling checking and hyphenation

Document structure

- Characters, pictures, lines, words, paragraphs, columns, tables, . . .
- Represent uniformly
- Recursive solution
- Glyph – display object

Composite pattern

- Intent
  - Tree structures
- Structure

Composite Pattern

- Participants
  - Component (Graphic)
  - Leaf
  - Composite
  - Client

Composite Pattern

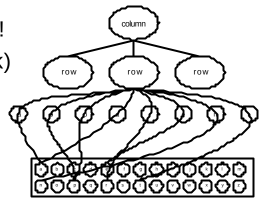
- Consequences
  - Simple client
  - Easy to extend
  - Runtime check required to restrict components

Composite Pattern

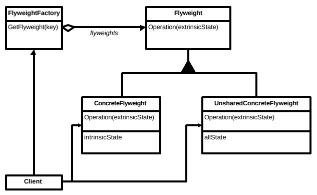
- Implementation issues
  - Explicit parent references
  - Sharing components
  - Child ordering
  - Responsibility for deletion (in non-GC language)

Document structure

- An object allocated for every character!
- Solution (trick, hack)
- Table of character objects
- Reference into the table



Flyweight pattern


FlyweightFactory
GetFlyweight(key)
flyweights
Flyweight
Operation(extrinsicState)
ConcreteFlyweight
Operation(extrinsicState)
intrinsicState
UnsharedConcreteFlyweight
Operation(extrinsicState)
allState
Client

Formatting

- Break text into lines
- Approach – insert row glyphs to break text into lines
- Want to allow different algorithms for formatting
- Compositor class – formatting algorithm
  - Composition glyph has a compositor
  - Compositor is responsible for formatting children

Strategy pattern

- Context, strategy pair
- Specific algorithms subclass strategy
  - ConcreteStrategy

UI Embellishment

- Add border or scrollbar to component
- MonoGlyph extends Glyph
- Border extends MonoGlyph
- ScrollBar extends MonoGlyph
- Decorator Pattern

Multiple look and feel standards

- Motif menus, Mac menus
- GuiFactory guiFactory = new MotifFactory();
- ScrollBar sb = guiFactory.CreateScrollBar();
- Button bu = guiFactory.CreateButton();

- Abstract Factory Pattern

Supporting Multiple Window Systems

- Window Class Hierarchy
- WindowImp Class Hierarchy
  - Extend WindowImp for each different system
  - Avoid polluting Window Class with system dependencies
- Bridge Pattern
  - Link between Window and WindowImp

User commands and spell check/hyphenation

- User commands
- Command Pattern
  - Includes Undo functionality
- Spell check and hyphenation
  - Iterate over words of document
  - Iterator Pattern and Visitor pattern

Classification of patterns

- Creational
  - Abstract factory, builder, factory method, prototype, singleton
- Structural
  - Adapter, bridge, composite, decorator, façade, flyweight, proxy
- Behavioral
  - Chain of responsibility, command, interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor

Original GoF patterns