Software Design Patterns

Adam Brooks
COT4810

March 18, 2008

Design Patterns

- What are they?
- Why are they important?
- Software Design Pattern Template
- Some common OO patterns
  - Observer
  - Decorator

What are Design Patterns?

A **pattern** is a solution to a problem in a context.

What are Design Patterns?

A **pattern** is a solution to a problem in a context.

- (Context) A recurring situation
- (Problem) Goal within the context, plus any constraints
- (Solution) Design which resolves the goal and constraints

What are Design Patterns?

- They're broader than just Software Design
  - Architecture
    - Buildings, Towns
  - Business
    - Common interactions between businesses and customers
    - Business Processes and Organizational Layout

What are Design Patterns?

- In Computer Science
  - Application Patterns
    - System level architecture
  - Domain-Specific Patterns
    - Concurrent systems
    - Real-time systems
  - UI Patterns

Why Design Patterns?

- Simple, best practices implementations of recurring problems
- They aid developers in creating functional, elegant, reusable, and flexible software…
- Present a common vocabulary for talking about software design

Software Design Pattern Template

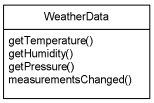
- The Gang of Four (GoF) Design Pattern Template:
  - Name
  - Classification
  - Intent
  - Motivation
  - Applicability
  - Structure
  - Participants
  - Collaborations
  - Consequences
  - Implementation
  - Sample Code
  - Known Uses
  - Related Patterns

OO Design Patterns

- The Observer Pattern
  - Defines a one-to-many relationship between objects so that when one object changes state, all of its dependents are notified and updated accordingly.
  - Publisher and Subscriber model for objects…

The Observer Pattern

- **Requirements/Problem:**
  - Implement three displays for the weather data (Current Conditions, Statistical, and Forecast)
  - Be expandable, allow outside developers to create additional displays and add to the application.
- **Assumptions:**
  - Use the provided WeatherData object…



The Observer Pattern

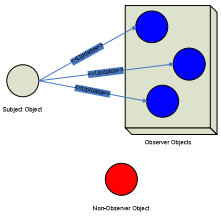
- One possible solution…

```
public class WeatherData {
   ...
   public voice measurementsChanged() {
         float temp = getTemperature();
         float humidity = getHumidity();
         float pressure = getPressure();

         currentConditionsDisplay.update(temp, humidity, pressure);
         statisticsDisplay.update(temp, humidity, pressure);
         forecastDisplay.update(temp, humidity, pressure);
   }
   ...
}
```

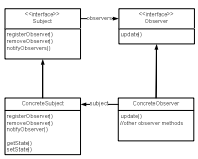
The Observer Pattern

- The Subscription Model…
  - Subject is the Publisher
  - Observers are the Subscribers

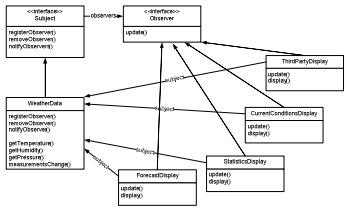


The Observer Pattern

- Class Diagram
  - Objects are Loosely Coupled



The Observer Pattern



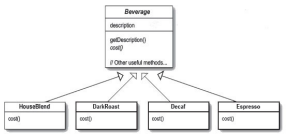
OO Design Patterns

- The Decorator Pattern
  - Attaches additional responsibilities to an object dynamically.
  - Provides a flexible alternative to subclassing for extending functionality.
  - A decorator adds its own behavior either before and/or after delegating to the object it decorates to proceed.

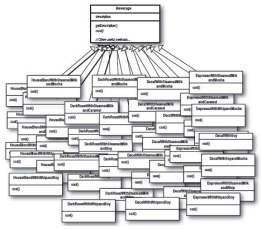
The Decorator Pattern

- **Requirements/Problem:**
  - Design a new system for ordering beverages from a Starbucks
  - The company is growing quickly, adding/removing drinks and updating pricing frequently.
  - Everything on the menu is ordered and priced A la carte.

The Decorator Pattern



The Decorator Pattern



The Decorator Pattern

- Inheritance isn't really a viable solution in this case…
- Enter the Decorator Pattern
  - Allows us to add new functionality by writing new code rather than changing existing code.
  - Features a "Open for Extension, but Closed for Modification" design principle.

The Decorator Pattern

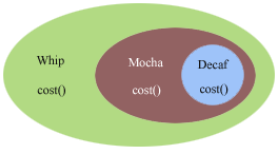
- Start with a Decaf

The Decorator Pattern

- Decorate it with a Mocha

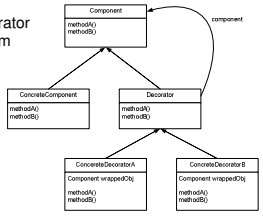
The Decorator Pattern

- Decorate again with a Whip

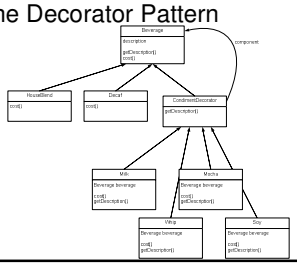


The Decorator Pattern

- Formal Decorator Class Diagram



The Decorator Pattern



Conclusions

- Patterns provide solutions with the benefit of being time tested by lots of other developers.
- They're not always the best solution.

Questions

- Who were the Gang of Four (GoF)?
- What's the name of a pattern for object notification (publish/subscribe)?

Sources

- Gamma, Helm, et al. Design Patterns: Elements of Reusable Object Oriented Software.
- Portland Patterns Repository. http://c2.com/
- Freeman, Eric & Elizabeth. Head First Design Patterns.
- Code Project. "Illustrated GOF Design Patterns…". http://www.codeproject.com