The University of Iowa
CS:2820 (22C:22)
Object-Oriented Software Development
Spring 2015

Design Patterns
by
Cesare Tinelli
Sample UP Artifact Relationships

**Domain Model**

<table>
<thead>
<tr>
<th>Sale</th>
<th>1</th>
<th>1..*</th>
<th>Sales Linetem</th>
<th>1..*</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td></td>
<td></td>
<td>quantity</td>
<td></td>
</tr>
</tbody>
</table>

**Use-Case Model**

- **Process Sale**
  1. Customer arrives ...
  2. ...
  3. Cashier enters item identifier.

- **Operation Contracts**
  **Operation**: enterItem(...)  
  **Post-conditions**: ...

- **Use Case Diagram**

- **Use Case Text**

- **System Sequence Diagrams**

**Design Model**

- **Register**
  - makeNewSale()
  - enterItem(...)  

- **ProductCatalog**
  - getProductDescription(...)  

- **Sale**
  - getProductDescription(itemID)  
  - addLineItem( d, quantity )

**Supplementary Specification**

- **Functional requirements**
  - non-functional requirements
  - domain rules

**Glossary**

- item details, formats, validation

**Business Modeling**

- Requirements
  - inspiration for names of some software domain objects
  - starting events to design for, and detailed post-condition to satisfy

**Design**

- Design Model
Responsibility-Driven Design

Designing systems in terms of object responsibilities, roles, and collaborations

- Action responsibilities
- Knowledge responsibilities
Design Patterns

Named and well-known problem/solution pairs that can be applied in new contexts, with advice on how to apply them in novel situations

Design patterns help during responsibility assignment in RDD
Advantages of Patterns

• They support chunking and incorporating a concept into our understanding and memory

• They facilitate communication
Established Design Patterns

- Creator
- Information Expert
- Low Coupling
- Controller
- High Cohesion

- Polymorphism
- Pure Fabrication
- Indirection
- Protected Variation
- ...
Creator Pattern

P: Who creates a new instance of some class A?

S: Class B get the responsibility if:
   - B “contains” or compositely aggregates A,
   - B closely uses A, or
   - B has the initializing data for A instances
Monopoly Game

- Played-with
  - Die
    - faceValue
  - Monopoly Game
  - Board

- Played-on
  - Monopoly Game

- Plays
  - Player
    - name
  - Piece
    - name
  - Square
    - name

- Owns
  - Piece

- Is-on
  - Square
Who should create the squares?
Who should create the squares?
Information Expert Pattern

**P:** How to assign responsibilities to objects?

**S:** Assign a responsibility to the class that has the information needed to fulfill it
Who should return a square, given its name?
Applying the IE Pattern

\[ s = \text{get Square}(\text{name}) \]
Low Coupling Pattern

P: How to reduce the impact of change?

S:
- Assign responsibilities so that (unnecessary) coupling remains low
- Use this principle to evaluate alternatives
Not Applying Low Coupling

* Higher (more) coupling if Dog has getSquare!
Observation

- Low Coupling is one of the most important goals in design.
- It tends to reduce time, effort and defects in software evolution.
- It is supported by Information Expert.
Controller Pattern

**P:** Which object beyond the UI layer first receives and coordinates a system operation?

**S:** Assign the responsibility to an object representing:

- the overall system, a *root object*, (facade controller)
- a use case scenario within which the system operation occurs (session controller)
System Sequence Diagram

- Observer
  - initialize(numOfPlayers)
  - playGame

Loop [no winner]
- dice total, player, square
Who should be the controller of playGame?
Who should be the controller of playGame?
Who should be the controller of playGame?
High Cohesion Pattern

P: How to keep objects focused, manageable, and understandable?

S:

• Assign responsibilities so that the cohesion of an object remains high

• Use this principle to evaluate alternatives
Cohesion

Informally, a measure of

• how functionally related the operations of a software component are

• how much work a software component is doing
Cohesion

• Bad cohesion and high coupling are positively correlated

• All other things being equal, a design with higher cohesion is preferable
Who has higher cohesion?
Credits

Notes and figures adapted from