Design Patterns Introduction

Introduction into Software Engineering
Lecture 8

Bernd Bruegge
Applied Software Engineering
Technische Universitaet Muenchen

Outline of the Lecture

- What is a design pattern?
- Modifiable designs
- Example of a design Pattern
  - Observer: Provide publisher/subscribe mechanism.

Design pattern

A design pattern is…

…a template solution to a recurring design problem

- Look before re-inventing the wheel just one more time

…an example of *modifiable* design

- Learning to design starts by studying other designs

…reusable design knowledge

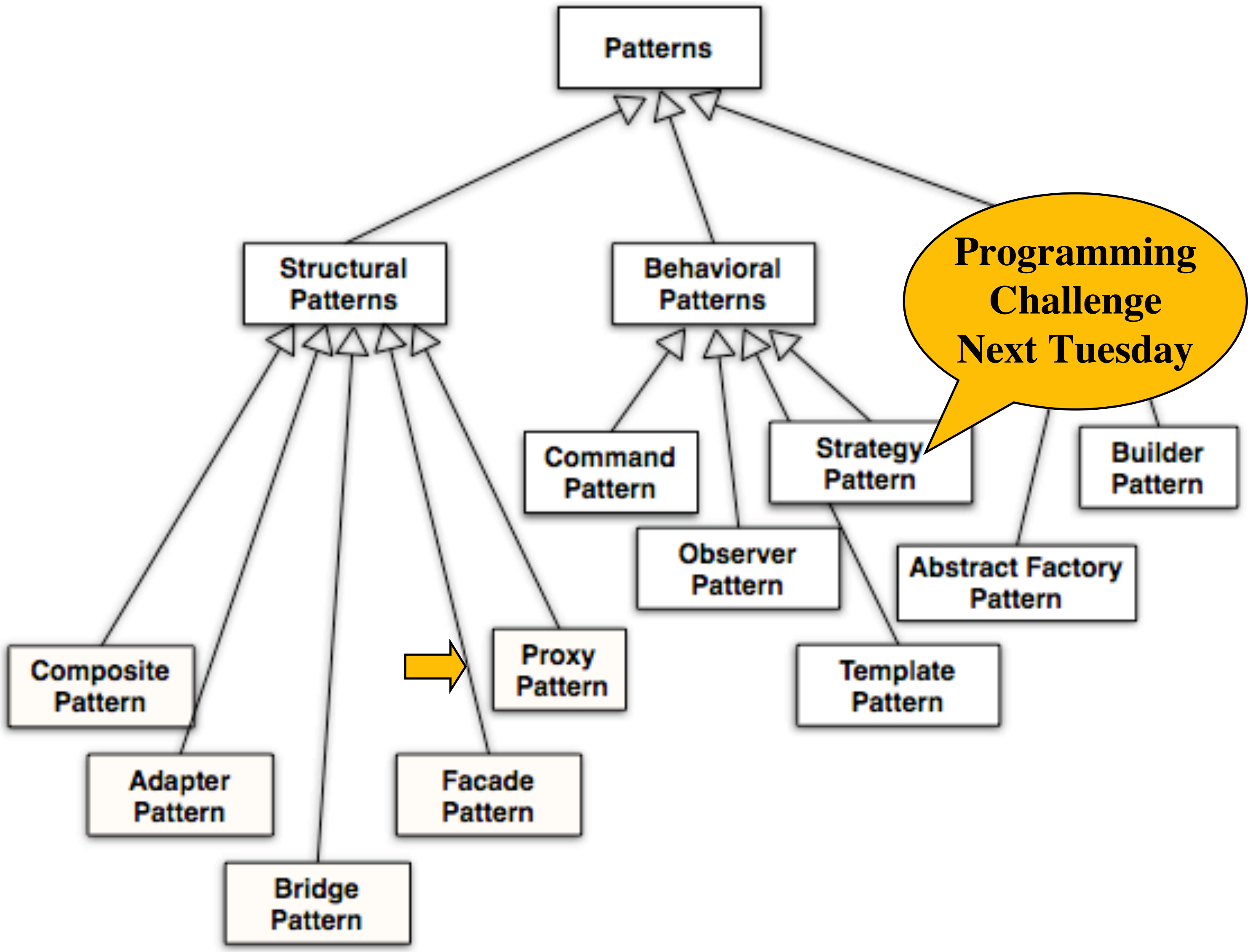
- 7+-2 classes and their associations
- Often actually more 5+-2 classes.

What makes Design Patterns Good?

- They are generalizations of design knowledge from existing systems
- They provide a shared vocabulary to designers
- They provide examples of reusable designs
    - Inheritance (abstract classes)
    - Delegation (or aggregation)

Categorization of Design Patterns

- Structural Patterns
  - reduce coupling between two or more classes
  - introduce an abstract class to enable future extensions
  - encapsulate complex structures
- Behavioral Patterns
  - allow a choice between algorithms and the assignment of responsibilies to objects (“Who does what?”)
  - characterize complex control flows that are difficult to follow at runtime
- Creational Patterns
  - allow to abstract from complex instantiation processes
  - Make the system independent from the way its objects are created, composed and represented.


Patterns
Structural Patterns
Behavioral Patterns
Programming Challenge Next Tuesday
Command Pattern
Strategy Pattern
Builder Pattern
Observer Pattern
Abstract Factory Pattern
Composite Pattern
Proxy Pattern
Template Pattern
Adapter Pattern
Facade Pattern
Bridge Pattern

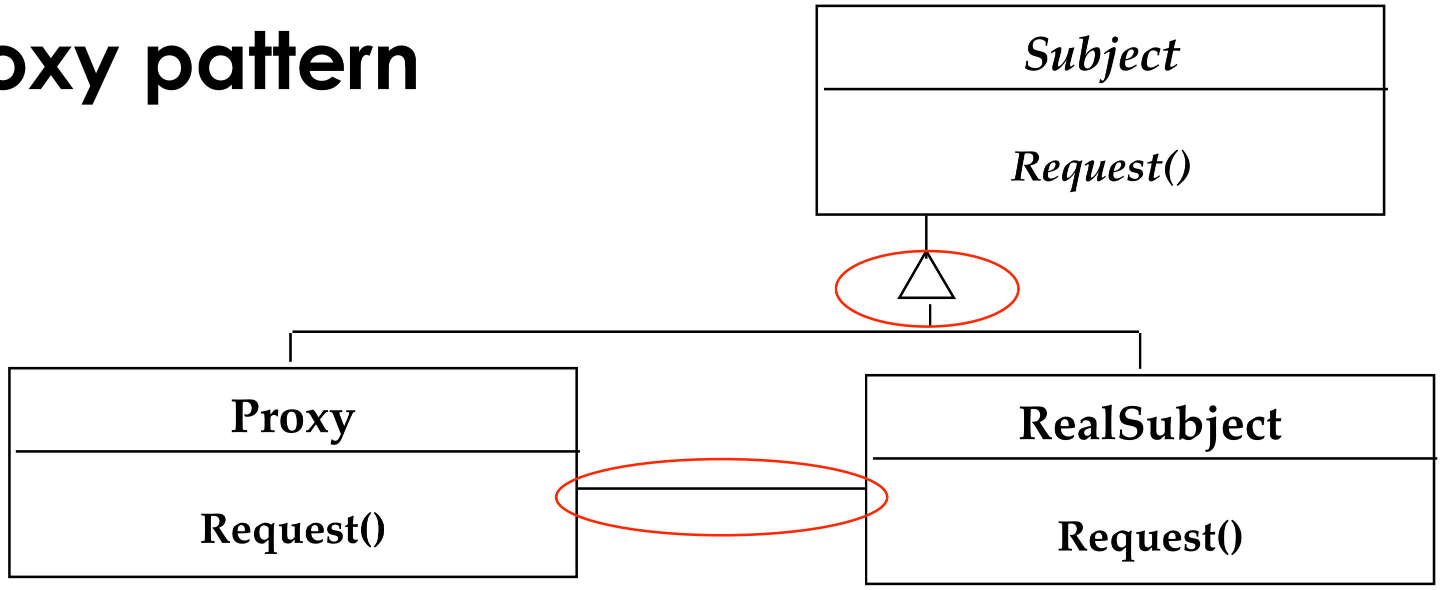
Proxy Pattern: Motivation

- I am sitting at my 768Kb DSL modem connection and try to retrieve a page during a busy time.
- I am getting 10 bits/sec.
- What can I do?

Proxy Pattern

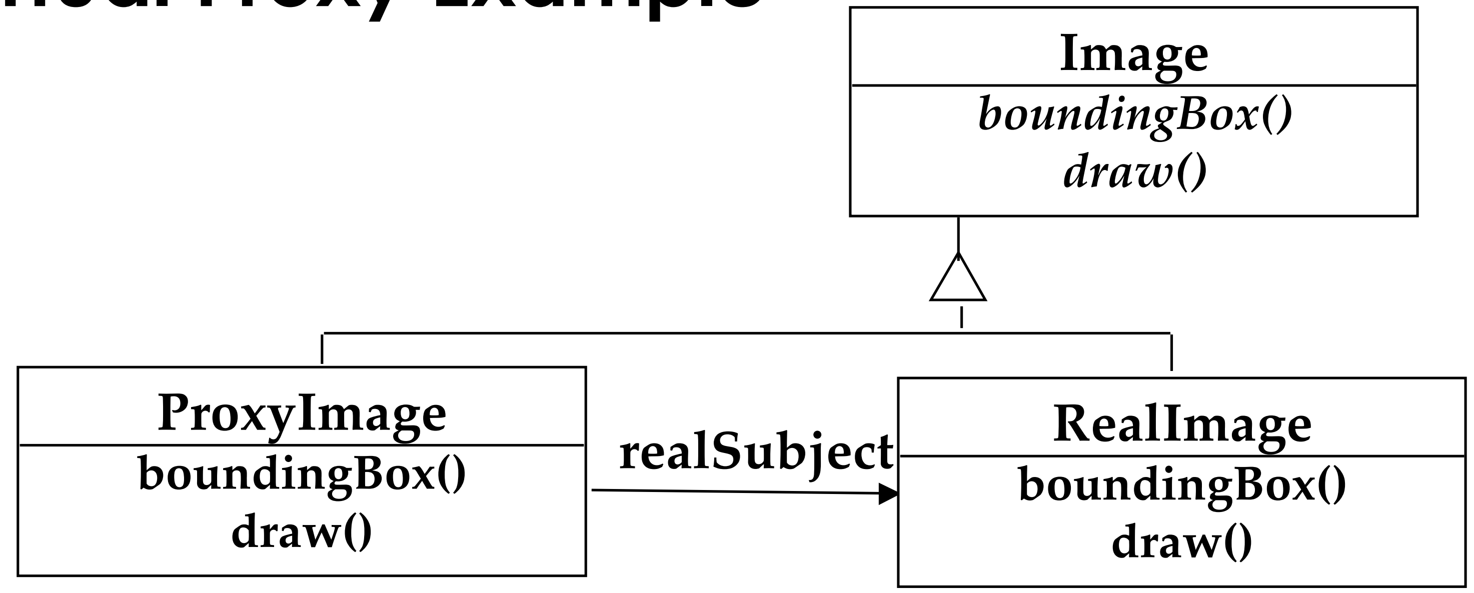
- Design Problem: What is particularly expensive in object-oriented systems?
  - Object creation
  - Object initialization
- Solution:
  - Defer object creation and object initialization to the time you need the object
- Proxy pattern:
  - Reduces the cost of accessing objects
  - Uses another object (“the proxy”) that acts as a stand-in for the real object
  - The proxy creates the real object only if the user asks for it.

Proxy pattern



- Interface inheritance is used to specify the interface shared by Proxy and RealSubject
- Delegation is used by Proxy to forward any accesses to the RealSubject (if desired).

Virtual Proxy Example



- The RealImage is stored and loaded separately
- If the RealImage is not loaded, a ProxyImage draws a grey rectangle in place of the image
- The class user of Image cannot tell, if it is dealing with ProxyImage instead of RealImage.

"Interim Summary"

- Design Patterns are collections of design knowledge
- They focus on reusability and extensibility
- They are useful especially when the system requirements are changing
- Become a master of design patterns!
  - The programming challenge next week and the exercises focus on design patterns
- If you want to be prepared:
  - Study the strategy pattern (p. 704 in the text book).