Design Patterns

宋　杰

Song Jie

东北大学 软件学院

Software College, Northeastern University

What are design patterns?

- Recurring solutions to design problems you see over and over. (Alpert et al, 1998)
- A set of rules describing how to accomplish certain tasks in the realm of software development. (Pree, 1994)
- Focus on reuse of recurring architectural design themes (Coplien and Schmidt, 1995)
- Address a recurring design problem that arises in a specific context and presents a solution to it (Buschmann. et al, 1996)
- Identify and specify abstractions that are above the level of single classes or instances, or of components. (GoF,1995)

Why study design patterns?

- Up to now in your degree you have been taught the OO basics and some higher level principles,
- BUT always building software from first principles is wasteful;
- Better if you can use some existing software or patterns produced and tested by experts;
- They help when you’re in the situation where you think: “Oh I’ve seen this problem before and I solved it by …

Why use design patterns?

- Reuse of design expertise;
- Support software that is flexible to change (extension).
- Improve communication between engineers: A common vocabulary;

Problems with design patterns?

- Trade-off: design can become a little more complicated.
- Inexperienced users often try to use more design patterns than they need to
  - You need to ask yourself why you’re using the design pattern.

Patterns History

- Software design patterns were inspired by the work of Christopher Alexander (architect).
  - □ Developed a pattern language for describing architectural features in buildings (1977).
- Seminal work of 23 software design patterns in:
  - □ E. Gamma, R. Helm, R. Johnson and J. Vlissades, Design patterns: Elements of reusable object-oriented software, Addison-Wesley, 1994
  - □ Informally known as the Gang of Four (GoF)

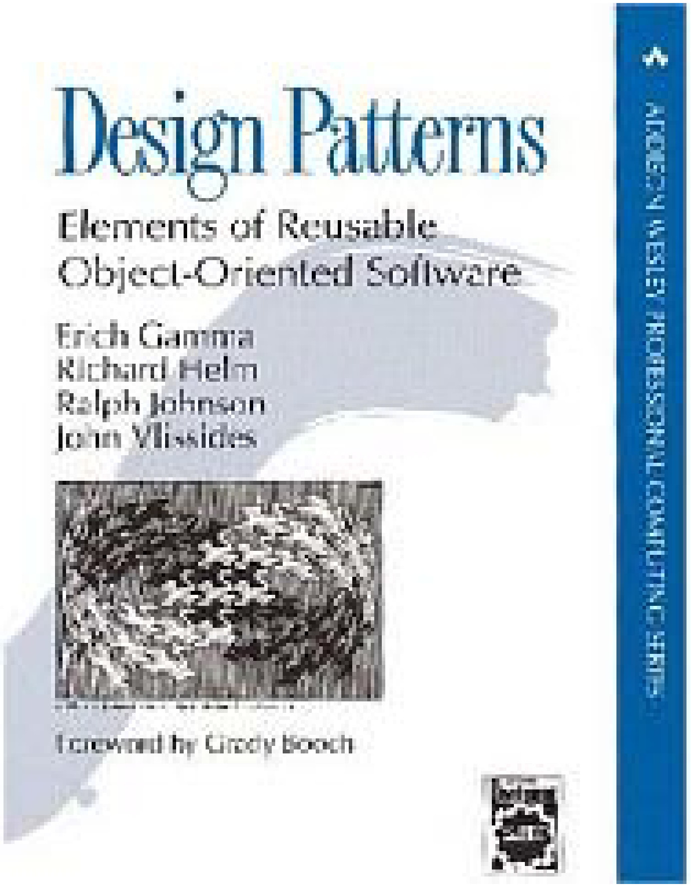
Pattern List – Three types of pattern

- Creational Patterns 创建型
  - □ Factory Method (工厂方法)
  - □ Abstract Factory (抽象工厂)
  - □ Singleton (单例)
  - □ Builder (创建)
  - □ Prototype (原型)
- Structural Patterns 结构型
  - □ Adapter (适配器)
  - □ Decorator(装饰器)
  - □ Composite(合成)
  - □ Facade(外观)
  - □ Flyweight(享元)
  - □ Proxy(代理)
  - □ Bridge(桥梁)
- Behavioral Patterns 行为型
  - □ Interpreter(解释器)
  - □ Template Method(模板方法)
  - □ Chain of Responsibility(责任链)
  - □ Command(命令)
  - □ Iterator(迭代器)
  - □ Mediator(调停者)
  - □ Memento(备忘录)
  - □ Observer(观察者)
  - □ State(状态)
  - □ Strategy (策略)
  - □ Visitor(访问者模式)

How to introduce a pattern

- Intent
- Structure
- Participants
- Collaborations
- Consequences
- Applicability
- Implementation
- Sample Code
- Examples
- Variation
- Extension
- Related Patterns

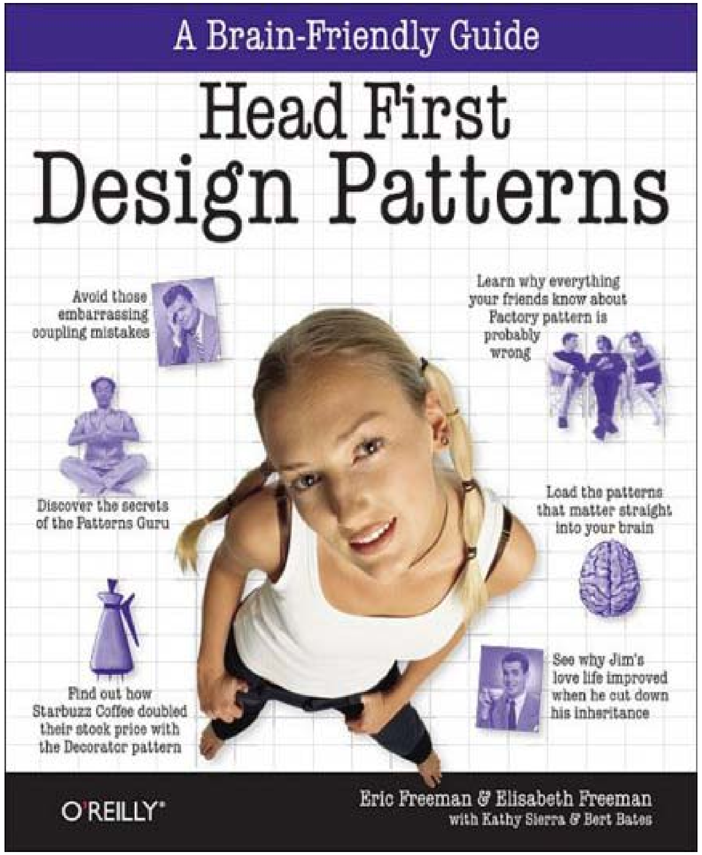
Referenced Book



Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON WESLEY PROFESSIONAL COMPUTING SERIES

- **Authors (GoF):**
  - □ Erich Gamma
  - □ Richard Helm
  - □ Ralph Johnson
  - □ John M. Vlissides
- **Published by:**
  - □ Addison Wesley 1994

Referenced Book


A Brain-Friendly Guide
Head First
Design Patterns
Avoid those embarrassing coupling mistakes
Learn why everything your friends know about Factory pattern is probably wrong
Discover the secrets of the Patterns Guru
Load the patterns that matter straight into your brain
Find out how Starbuzz Coffee doubled their stock price with the Decorator pattern
See why Jim's love life improved when he cut down his inheritance
O'REILLY
Eric Freeman & Elisabeth Freeman
with Kathy Sierra & Bert Bates

- **Authors:**
  - Eric Freeman & Elisabeth Freeman with Sierra & Bert Bates
- **Published by:**
  - O'REILLY 2005

Let’s go to next…