Material and some slide content from:
- Software Architecture: Foundations, Theory, and Practice
- Krzysztof Czarnecki
- John Vlissides
- GoF Design Patterns Book
- Atif Khan

Design Introduction

Reid Holmes

Lecture 7 - Tuesday October 4 2011.

Why architecture?

HL7v3

Object-oriented design

References:

Design Patterns: Elements of Reusable Object-oriented Software by Gamma, Helm, Johnson, and Vlissides

Design of Design. by Fred P. Brooks Jr.

Analysis vs. design

- Analysis
  - Domain-level modelling of “real world” objects.
    - Addresses functional challenges.
    - Captures what a system does.
    - Provides implementation guidance.
- Design
  - Modelling of implementation objects.
    - Addresses implementation challenges.
    - Captures how the system realizes its OOA.
    - Assigning responsibilities to objects.

Design process

Christopher Jones

"If, as is likely, the act of tracing out the intermediate steps exposes some unforeseen difficulties... the designers are thrown back to square one."

Motivation

- OOD methods emphasize design notations.
  - But... these notations must be expressible in code.
- The importance of experience in OOD cannot be overemphasized.
- Design-level reuse is valuable.
  - Matches problems to design experience.
  - Avoid difficulties encountered before.

Concept

- OO systems exploit recurring design structures that promote:
  - Abstraction
  - Flexibility
  - Modularity
  - Elegance
- Therein lies valuable design knowledge
- Capturing, communicating, and applying this knowledge is problematic
- Must fight same issues as architecure

Abstraction

- The removal of detail while retaining essential properties of its structure
- Plays a central role in the design process:
  - Enables the designer to focus on the key issues without being distracted by implementation
- It can be easy for developers to be distracted by implementation minutiae
- Different abstractions are appropriate for different applications and needs

Static vs. Dynamic

- Is it enough that the code compiles?
- Is it sufficient that it meets its specified structure?
- Behaviour matters. Static relationships are only a subset of a complete system
- Behaviour is inherently dynamic
  - The code alone may not be sufficient
  - Debugging only gives glimpses in time
- Increased abstraction == decreased control

Activity