Architectural Patterns

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo
An architectural pattern is a general, reusable \textbf{solution} to a recurring \textbf{problem} in software architecture.

- It documents knowledge about a common problem.
- It is supposed to be reused across applications.
Reference Architecture

A software system is unique
- However, several software systems may share similar architectures

A system may use an architectural pattern as a reference
- Architecture style and reference architecture are similar concepts
Choosing an Architecture

- Architectural patterns define ways to organize the system general structure
  - Each architectural pattern may favor specific system properties

- Therefore, it is important to know alternative architectural patterns to achieve particular software needs
Key Elements of a Pattern

- The choice of an architectural pattern is part of the problem solution

- Architectural patterns usually define
  - A set of components
  - Responsibility of the involved components (roles)
  - Relationships among components
Patterns in Software

- Architecture patterns represent general structures
  - Design patterns (detailed design)
  - Idioms (programming)

- Architecture patterns are defined the high level structure of a system
  - Design patterns and idioms are used in successive phases
All Patterns are Template

- (Architectural) Patterns **do not** define a complete solution
  - The partial solution should be refined

- Examples of refinements
  - To include program-specific components and relationships
  - To define design patterns and idioms to detail the solution
Non-Functional Requirements

- The choice of an architectural pattern is largely dependent on:
  - The type of system
  - Non-functional requirements

- Questions to be considered:
  - Is the system interactive?
  - Does it require frequent changes?
  - What non-functional requirements are important? Reliability? Performance?
Composition of Patterns

- Each architectural pattern focuses on specific non-functional requirements
  - There are also alternative patterns to address similar problems

- Complex systems may follow several architectural patterns
  - Similarly, a system may include several design patterns
Architectural Patterns

- From Mud to Structure
  - Layered Architecture
  - Blackboard
  - Pipes and Filters

- Distributed Systems
  - Client-Server
  - Broker

- Interactive Systems
  - Model-View-Controller (MVC)
  - Presentation-Abstraction-Control

- Adaptable Systems
  - Microkernel
  - Reflection
Architectural Patterns Books

- Pattern-Oriented Software Architecture: A System of Patterns (Vol. 1)
  - Chap. 2 Architectural Patterns

  - Chapter 6 Architectural Design