Patterns - Introduction

Background

- Gamma, Helm, Johnson, and Vlissides (the "Gang of Four") – Design Patterns, Elements of Reusable Object-Oriented Software
- This book solidified thinking about patterns and became the seminal Design Patterns text

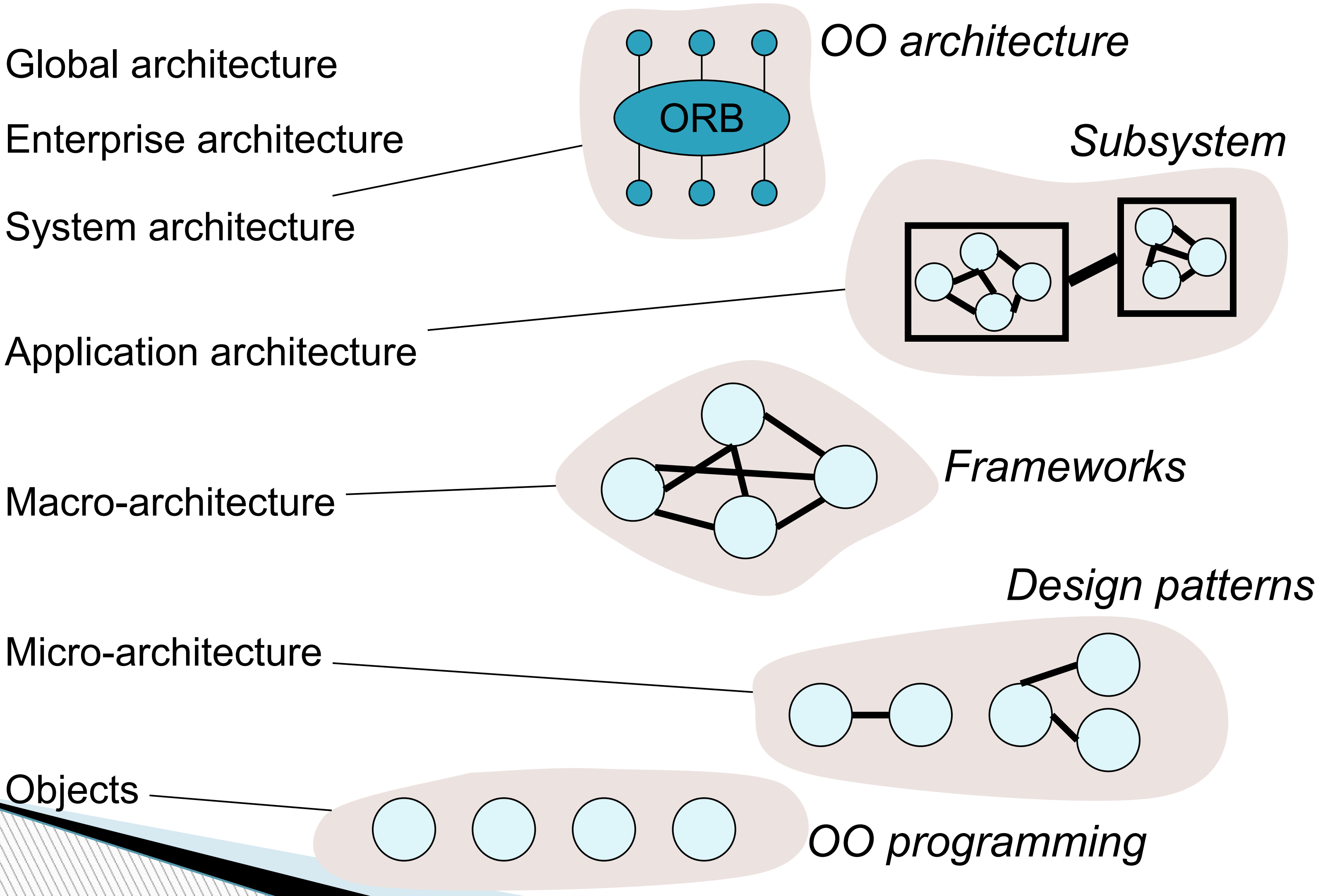
Purpose

- To capture *design expertise* –patterns are not created from thin air, but abstracted from *existing* design examples
- To *reuse* of design expertise
- To provide a *vocabulary* for talking about design
- To reduce the depth of Class hierarchy
- To develop high cohesive classes
- Studying design patterns is a way of studying how the “experts” do design

Why design patterns in SA?

- If you're a software engineer, you should know about them anyway
- There are many architectural patterns published, and the GoF Design Patterns is a pre-requisite to understand these:
    - Mowbray and Malveau – CORBA Design Patterns
    - Schmidt et al – Pattern-Oriented Software Architecture
- Design Patterns help you *break out* of first-generation OO thought patterns

The seven layers of architecture*



Global architecture

Enterprise architecture

System architecture

Application architecture

Macro-architecture

Micro-architecture

Objects

OO architecture

Subsystem

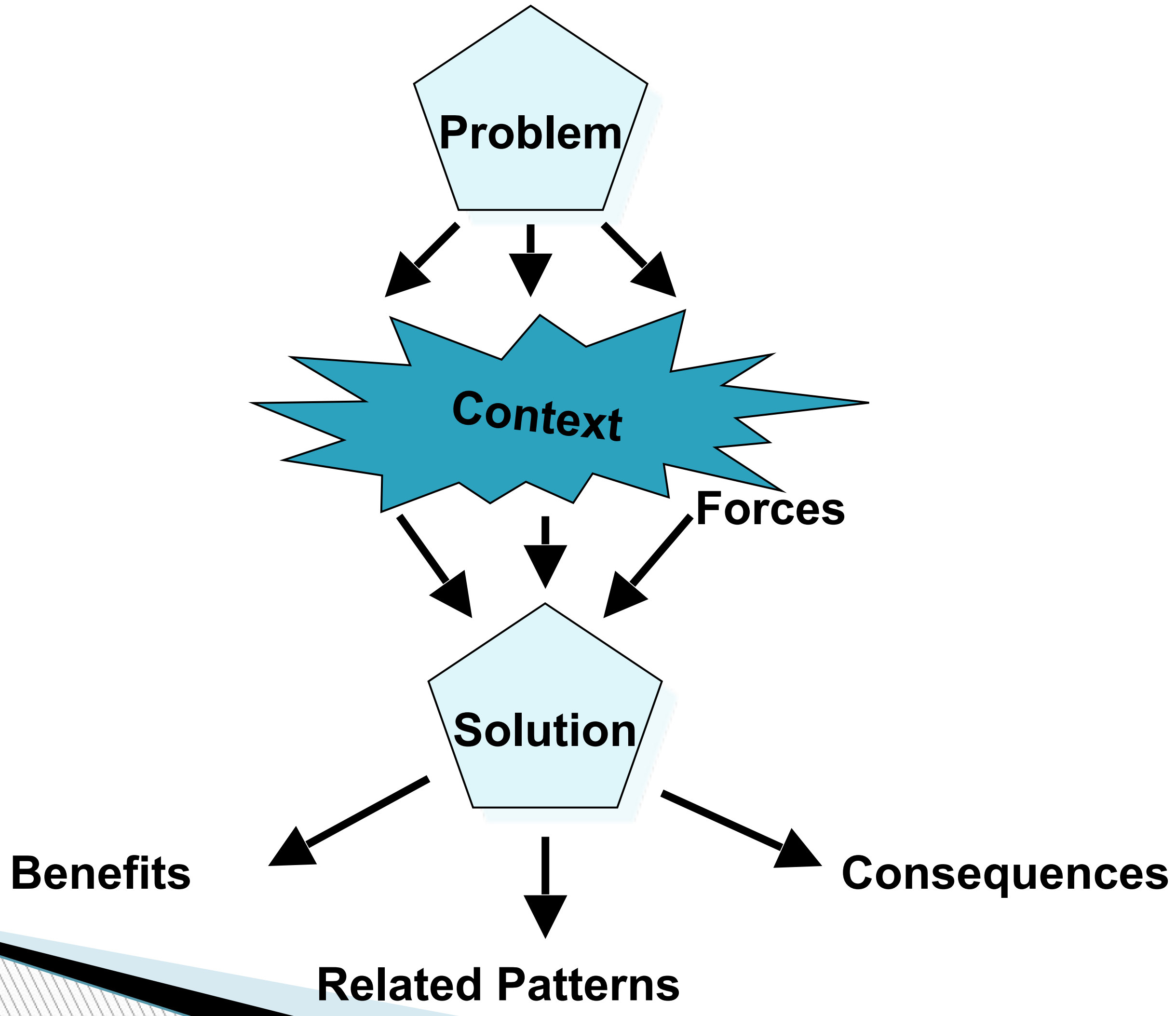
Frameworks

Design patterns

OO programming

* Mowbray and Malveau

How patterns arise



Structure of a pattern

- Name
- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Implementation
- Known Uses
- Related Patterns

Patterns vs "Design"

- Patterns *are* design
  - But: patterns transcend the "identify classes and associations" approach to design
  - Instead: learn to recognize patterns in the *problem* space and translate to the solution
- Patterns can capture OO design principles within a specific domain
- Patterns provide structure to "design"

Patterns vs Frameworks

- Patterns are lower-level than frameworks
- Frameworks typically employ many patterns:
  - Factory
  - Strategy
  - Composite
  - Observer
- Done well, patterns are the "plumbing" of a framework

Patterns vs Architecture

- Design Patterns (GoF) represent a lower level of system structure than “architecture” (cf: seven levels of A)
- Patterns can be applied to architecture:
  - Mowbray and Malveau
  - Buschmann *et al*
  - Schmidt *et al*
- Architectural patterns tend to be focussed on middleware. They are good at capturing:
  - Concurrency
  - Distribution
  - Synchronization

Concluding remarks

- Design Patterns (GoF) provide a foundation for further understanding of:
  - Object-Oriented design
  - Software Architecture
- Understanding patterns can take some time
  - Re-reading them over time helps
  - As does applying them in your own designs!