Object-Oriented Analysis and Design
Lecture 4: Analysis and Design Overview
Objectives: Analysis and Design Overview

- Review the key Analysis and Design terms and concepts
- Introduce the Analysis and Design process, including roles, artifacts and workflow
- Explain the difference between Analysis and Design
The purposes of Analysis and Design are to:

- Transform the requirements into a design of the system-to-be.
- Evolve a robust architecture for the system.
- Adapt the design to match the implementation environment, designing it for performance.
Analysis and Design Overview

Use-Case Model

Analysis and Design

Glossary

Supplementary Specification

Design Model

Architecture Document

Data Model
Analysis & Design Overview Topics

★ ◆ Key Concepts
◆ Analysis and Design Workflow
Analysis Versus Design

- **Analysis**
  - Focus on understanding the problem
  - Idealized design
  - Behavior
  - System structure
  - Functional requirements
  - A small model

- **Design**
  - Focus on understanding the solution
  - Operations and attributes
  - Performance
  - Close to real code
  - Object lifecycles
  - Nonfunctional requirements
  - A large model
Analysis and Design Are Not Top-Down or Bottom-Up

- Use Cases (Define a middle level)
- Analysis Classes
- Design Classes
- Subsystems

- Top Down
- Bottom up
What Is Architecture?

- Software architecture encompasses a set of significant decisions about the organization of a software system.
  - Selection of the structural elements and their interfaces by which a system is composed
  - Behavior as specified in collaborations among those elements
  - Composition of these structural and behavioral elements into larger subsystems
  - Architectural style that guides this organization

Grady Booch, Philippe Kruchten, Rich Reitman, Kurt Bittner; Rational (derived from Mary Shaw)
Architecture involves a set of strategic design decisions, rules or patterns that constrain design and construction. Architecture decisions are the most fundamental decisions, and changing them will have significant effects.
Software Architecture: The “4+1 View” Model

- **Logical View**: End-user Functionality
  - Analysts/Designers: Structure
  - System integrators: Performance, Scalability, Throughput

- **Use-Case View**: User Functionality
  - Programmers: Software management
  - System engineering: System topology, Delivery, installation, communication

- **Process View**: End-user Functionality
  - Analysts/Designers: Structure
  - System integrators: Performance, Scalability, Throughput

- **Implementation View**: End-user Functionality
  - Programmers: Software management
  - System engineering: System topology, Delivery, installation, communication

- **Deployment View**: End-user Functionality
  - Programmers: Software management
  - System engineering: System topology, Delivery, installation, communication
Analysis & Design Overview Topics

- Key Concepts

★★✧ Analysis and Design Workflow
Analysis and Design Workflow

Analysis

[Early Elaboration Iteration] → Define a Candidate Architecture → Analyze Behavior

[Inception Iteration (Optional)] → Perform Architectural Synthesis → Refine the Architecture

Design

Define Components → Design the Database (Optional)

Refine the Architecture
Analysis and Design Activity Overview

Architect
- Assess Viability of Architectural Proof-of-Concept
- Construct Architectural Proof-of-Concept
- Architectural Analysis
- Incorporate Existing Design Elements
- Describe the Run-time Architecture
- Describe Distribution
- Identify Design Mechanisms
- Identify Design Elements
- Capsule Design
- Capsule Designer

Designer
- Use-Case Analysis
- Use-Case Design
- Subsystem Design
- Class Design
- Design Test Classes and Packages
- Database Design
- Architecture Reviewer
- Review the Architecture
- Design Reviewer
- Review the Design

Software Architect
The Software Architect leads and coordinates technical activities and artifacts.
Designer’s Responsibilities

- The designer must know use-case modeling techniques, system requirements, and software design techniques.
Use cases defined for a system are the basis for the entire development process. Benefits of use cases:

- Concise, simple, and understandable by a wide range of stakeholders.
- Help synchronize the content of different models.
What Is a Use-Case Realization?

Use-Case Model

Use Case

<table>
<thead>
<tr>
<th>Use Case Model</th>
<th>Design Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case</td>
<td>Use-Case Realization</td>
</tr>
</tbody>
</table>

Sequence Diagrams

Collaboration Diagrams

Class Diagrams

Use Case
Analysis and Design in an Iterative Process

Start of iteration

Use Case A
Scenarios 1 & 2

Use-Case Realization A

Iteration n

End of iteration

Use Case B
Scenario 1

Use Case A
Scenario 3

Use-Case Realization A

Use-Case Realization B

Iteration n + 1
Review: Analysis and Design Overview

- What is the purpose of the Analysis and Design Discipline?
- What are the input and output artifacts?
- Name and briefly describe the 4+1 Views of Architecture.
- What is the difference between Analysis and Design?
- What is architecture?