Overview

- Collaborative Object Modeling and architectural design mETHod (COMET)
  - Object Oriented Analysis and Design Method
  - Uses UML (Unified Modeling Language) notation
    - Standard approach for describing a software design
    - COMET = UML + Method
- Provides steps and guidelines for
  - Software Modeling and Design
  - From Use Case Models to Software Architecture
Object-Oriented Software Life Cycle

Requirements Modeling

- Requirements Modeling
- Use Case Modeling
  - Define software functional requirements in terms of use cases and actors

Figure 5.1 COMET software life cycle model

Copyright © 2011 Hassan Gomaa
Object-Oriented Software Life Cycle

Analysis Modeling

- Analysis Modeling consists of
  - Static Modeling
  - Dynamic Modeling
    - State Machine modeling using statecharts
    - Object interaction modeling

Figure 5.1 COMET object-oriented software life cycle model

Object-Oriented Software Life Cycle

Analysis Modeling

- Static Modeling
  - Define structural relationships between classes
  - Depict classes and their relationships on class diagrams

Figure 21.3 Banking System class context diagram
Object-Oriented Software Life Cycle

Analysis Modeling

- Static Modeling
  - Define structural relationships between classes
  - Depict classes and their relationships on class diagrams

- Dynamic Modeling
  - Define statecharts for state dependent control objects
Object-Oriented Software Life Cycle

Analysis Modeling

- **Dynamic Modeling**
  - Defines how objects participate in use cases using communication diagrams or sequence diagrams

---

**Steps in Using COMET/UML**

1. Develop Software Requirements Model
2. Develop Software Analysis Model
3. **Develop Software Design Model**
   - Design Overall Software Architecture (Chapter 12, 13)
   - Design Distributed Component-based Subsystems (Chapter 13, 15)
   - Structure Subsystems into Concurrent Tasks (Chapter 18)
   - Design Information Hiding Classes (Chapter 14)
   - Develop Detailed Software Design

---

Copyright © 2011 Hassan Gomaa
Object-Oriented Software Life Cycle

Design Modeling

- Design Overall Software Architecture (Chapter 12, 13)
- Design Distributed Component-based Subsystems (Chapter 13, 15)
- Structure Subsystems into Concurrent Tasks (Chapter 18)
- Design Information Hiding Classes (Chapter 14)
- Develop Detailed Software Design

Figure 21.27 Subsystem design—high level communication diagram for Banking System

Figure 21.28 Subsystem interfaces – high level communication diagram for Banking System
Object-Oriented Software Life Cycle

Design Modeling

- Design Overall Software Architecture (Chapter 12, 13)
- Design Distributed Component-based Subsystems (Chapter 13,15)
- Structure Subsystems into Concurrent Tasks (Chapter 18)
- Design Information Hiding Classes (Chapter 14)
- Develop Detailed Software Design

Copyright © 2011 Hassan Gomaa
Object-Oriented Software Life Cycle

Design Modeling

- Design Overall Software Architecture (Chapter 12, 13)
- Design Distributed Component-based Subsystems (Chapter 13, 15)
- Structure Subsystems into Concurrent Tasks (Chapter 18)
- Design Information Hiding Classes (Chapter 14)
- Develop Detailed Software Design

Steps in Using COMET/UML

1. Develop Software Requirements Model
   - Develop Use Case Model (Chapter 6)

2. Develop Software Analysis Model
   - Develop static model of problem domain (Chapter 7)
   - Structure system into objects (Chapter 8)
   - Develop statecharts for state dependent objects (Chapter 10)
   - Develop object interaction diagrams for each use case (Chapter 9, 11)

3. Develop Software Design Model
   - Design Overall Software Architecture (Chapter 12, 13)
   - Design Distributed Component-based Subsystems (Chapter 13, 15)
   - Structure Subsystems into Concurrent Tasks (Chapter 18)
   - Design Information Hiding Classes (Chapter 14)
   - Develop Detailed Software Design