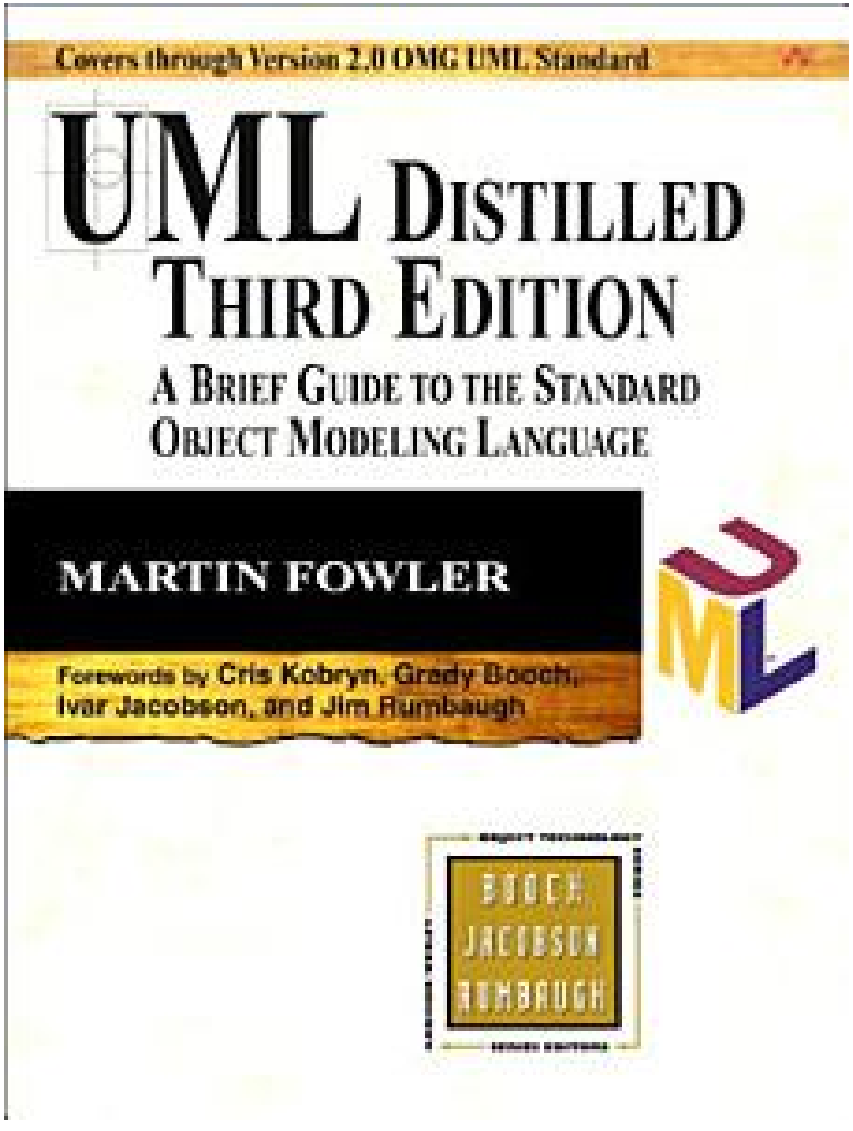
Use Cases

CSC 354, Software Engineering I

Chapter 9: Use Cases



Overview: Modeling with UML

- What is modeling?
- What is UML?
- Defining the Scope
- User Stories
- Use Case diagrams

What is modeling?

- Modeling consists of building an abstraction of reality.
- Abstractions are simplifications because:
  - The ignore irrelevant details, and
  - The only represent the relevant details
- What is relevant or irrelevant depends on the purpose of the model.

Why model software?

- Software is getting increasingly complex
  - Applications are on the order of millions of lines of code
  - A single programmer cannot manage this amount of code
- Code is not easily understandable by developers who did not write it (but need to maintain it.)
- We need simple representations for complex systems
- Modeling is a means for dealing with complexity

Application and Solution Domain

- First we need to understand the *what* and then the *how*
- Application domain (requirements analysis): the environment in which the system is going to operate in
- Solution domain (system design): The technologies to build the system

What is UML?

- UML: Unified Modeling Language
  - Standard for modeling software systems
  - Based on a convergence of notations used in Object-Oriented Analysis and Design (OOAD) methods
- The current UML version is 2.5.1
  - Link: https://www.omg.org/spec/UML/2.5.1/About-UML/

What is UML? (continued)

- Pareto principle: for many outcomes, approximately 80% of the consequences come from 20% of the causes.
  - 80% of your profits come from 20% of you customers
  - 80% of your complaints come from 20% of you customers
  - etc.
- UML: 80% of most problems can be modeled by 20% of UML
  - Plan: teach you that 20%

UML Context Diagram

- A context diagram defines what is inside the system and what is outside the system.
- Used during requirements elicitation and analysis to negotiate what is inside or outside the system.

UML Actors

- An actor is a model for an external entity that interacts with the system, for example:
  - User
  - External system
  - Physical environment
  - etc.
- An actor has a unique name and a description, for example:
  - Passenger: a person in the train
  - GPS: an external system that provides the system with coordinates

User Stories

- Once you define the context diagram, the next step is to generate user stories
- For each actor, define how they interact with the system
  - Brainstorm, then refine the list
  - Review with the client
- Example user story:
  - As a passenger I want to:
    - purchase a ticket
    - board the train
    - etc.

Use Case Diagrams

- A use case diagram represents the functionality of the system from *the users' point of view*
- User stories can help generate the high-level use cases
- A use case represents a class of functionality provided by the system
  - Each use case needs a use case description
- Use case model: the set of all cases the *completely* describes the functionality of the system

UML Use Case Descriptions

- Use cases are described textually with a focus on the event flow between actor and system
- The textual use case description consists of 6 parts:
  1. Unique name
  2. Participating actors
  3. Entry conditions (preconditions)
  4. Exit conditions (postconditions)
  5. Flow of events
  6. Special requirements

Textual Use Case Description Example

1. Name: Purchase ticket
2. Participating actor(s): Passenger
3. Entry conditions:
   - Passenger is at the ticket distributor
   - Passenger has sufficient funds to purchase the ticket
4. Exit conditions:
   - Passenger has ticket
5. Flow of events
   1. Passenger selects the number of zones of travel
   2. Ticket Distributor displays the amount due
   3. Passenger makes payment
   4. Ticket Distributor accepts payment
   5. Ticket Distributor issues ticket
6. Special requirements:
   - 4a if payment fails, do not issue the ticket and exit

Review the Steps

1. Define the Context Diagram showing the Actors that interact with the system
2. List the User Stories for each Actor
3. Identify the main Use Cases (from the *user's* perspective)
4. For each use case on the diagram, define the Textual Use Case Description
5. Review with the client at each step