

IT2755
Software Engineering

Use Case Models &
Use Case Descriptions

Learning Outcomes

To be able to:

1. explain the purpose of developing models
2. identify components in a use case diagram
3. construct a use case model
4. describe main flow and alternative flows (scenarios) for a use case using a use case description.



Models and Modeling

- **What are Models?**
 - Is a simplified representation to describe some selected aspects of the system being built.
- **Purpose of Modeling**
 - Models can be used to document different parts of a problem and/or solution during software development.
 - Analyst uses different types of models to show different levels of system details.
 - View the same problem from different perspectives for new requirements discovery.
 - Models are communication tools among stakeholders of a project.
Eg. Use case diagram, use case description, class diagrams.



Advantages of using models

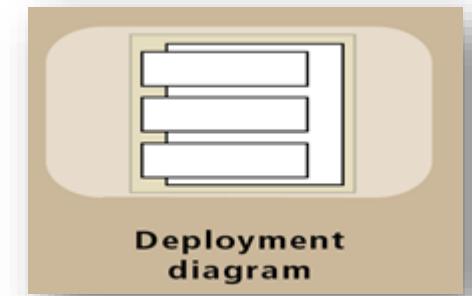
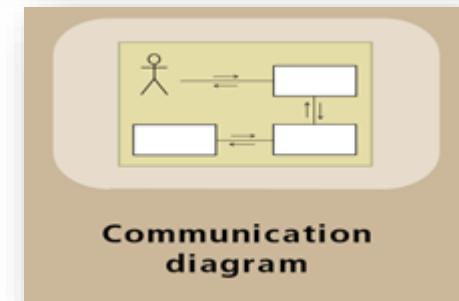
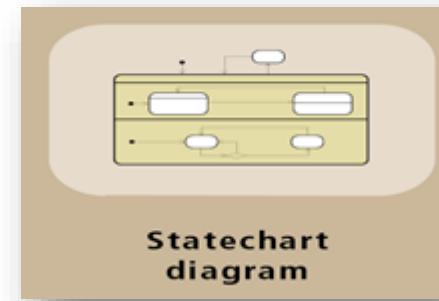
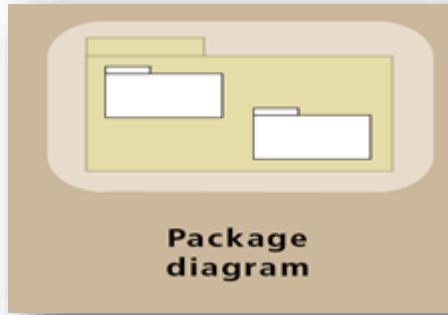
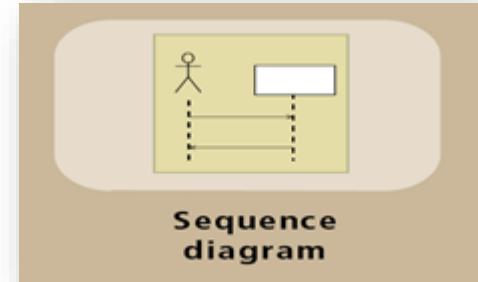
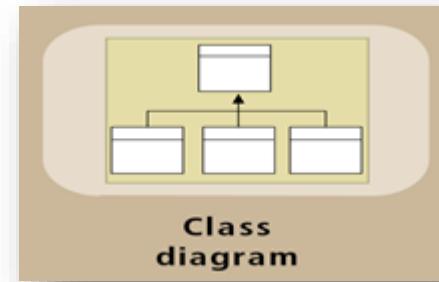
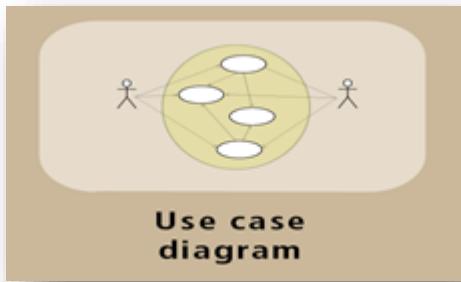
- **Manage complexity** of system to essential components.
- As a **visual cue** to convey information to users and get feedback.
- Allow analysts to **clarify and refine requirements** early in the development.
- Provides a way to **store information** as documentation for later use and reference.
- Serve as a platform for **effective teamwork** amongst project team members.
- Promotes **understanding**.

UML

- **Unified Modeling Language**
- Not a programming language like C++, C# or Java
- UML is a pictorial language used to make software blue prints.
- A standardized, general-purpose visual modeling “language” used in software engineering to visualize, specify, construct and document software system.
- Includes a set of graphic notation techniques to create visual models of object-oriented software systems.



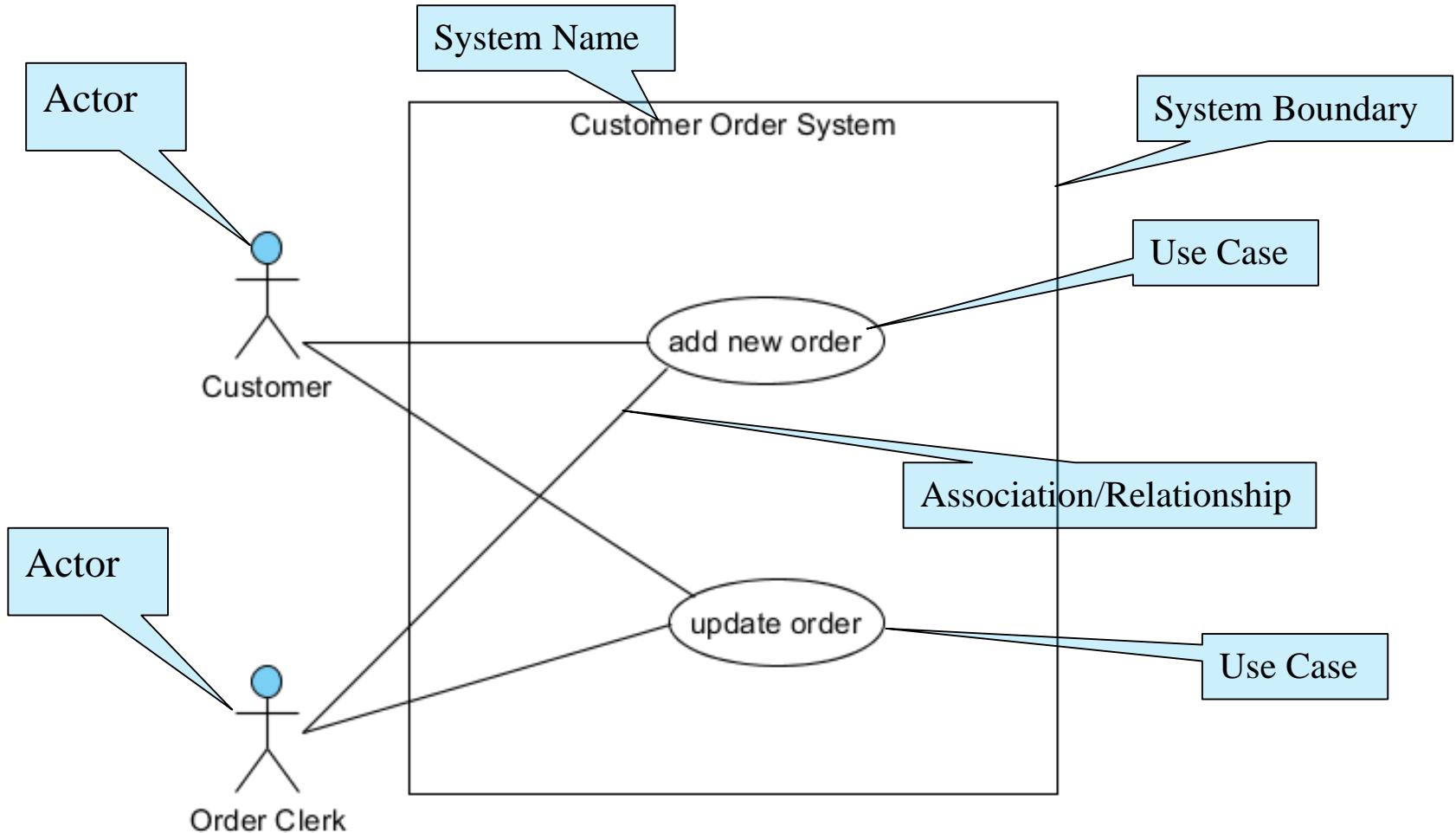
UML Diagrams used for Modeling



UML Use Case Diagram

- Provides an overview of the requirements for a system.
 - Focus in the **functional requirements of system** used by users.
- Components of a Use case diagrams:
 - **Use cases**
 - Actors
 - Associations
 - System boundary

A Use Case Diagram for an Order System



Components of a Use Case Diagram

- Use cases name
 - describes a **goal** (system function) that the user wants to achieve by using the system.
- Actors
 - a person, organization, or external system that directly (or indirectly) interacts with the system with a specific role.
- Associations
 - shows interactions between use cases and actors.

Identifying Actors

- **Actor**
 - Person or thing that interacts with the system.
 - Primary actor – initiates or **triggers** the use cases to achieve the actor's goals.
 - Secondary actors – participates in the use cases to assist in the completion of a use case.
 - Normally described by nouns.
 - **Example:** A *online bookstore system* allows customer to buy books and payment is handled by an external payment system (e.g. Paypal or eNets).
 - *Customer* – Primary actor
 - *External payment system* – Secondary actors

Identifying Use Case

- **Use Case**
 - Something that the actors wants the system to do in order to achieve a goal.
 - Always triggered by primary actor(s)
 - Identified from the point of view of the actor
 - Actions phrases or verbs

Eg: A online bookstore allows customer to buy books and payment is done through an external payment system (e.g. Paypal or eNets).
- **Use Case Name:** use a short verb phrase
 - E.g. Add orders, Cancel membership, Print reports

Developing a Use Case Diagram

- Developing Use Case is an **Iterative process:**
 - [1] Identify the actors and roles.
 - [2] Identify the functions offered by the system.
 - [3] For each actor map the relationship to functions
 - **Sometimes you will uncover new system capabilities*
 - [4] If required, map the relationship to the use case with the secondary actor. (actor needed to support the use case but do not trigger the use case)
 - **Sometimes you will uncover new actors*
 - [5] Iteratively refine the model till it stabilizes.



Identifying Actors

Perform a Textual Analysis to Identify Actors and Use Cases

- Hints to identify Actors:
 - Looks for nouns that indicates:
 - Who will use the system?
 - Who needs the system to support daily work?
 - Who will maintain, administrate and operate the system?
 - Will it interact with other systems?
 - What is the scope of the system?
 - If functions are outside of the scope do I include them?



Identifying Use Cases

- Hints to identify Use Case:
- Looks for verbs describing the following:
 - What function does the actor wants the system to perform.
 - What goals or intentions does the actor wants the system to achieve.



Use Case Description



Use case description

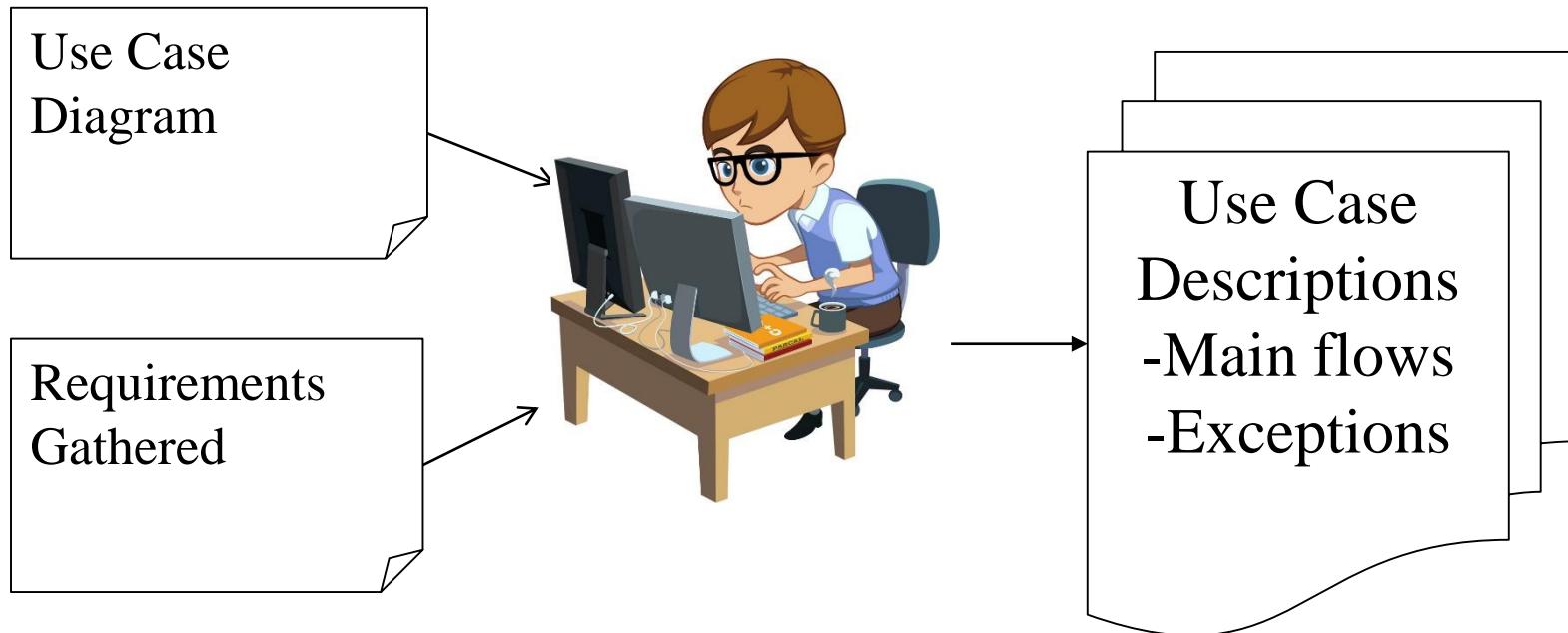
- Use case diagram is a model suitable for non-technical people to appreciate the functions of the proposed system in a holistic manner.
- When we need to develop the system, some extra details would be needed to handle details such as what to do when user cannot remember the login password. So we use another model to manage these details.
- We need “Use case description” to specify/define the many possible situations (scenarios) that the system is expected to handle.

Use Case Descriptions

- Use cases have **internal complexity**:
 1. Sequence of steps to execute business process to achieve the user's goal
 2. Several variations may exist within single use case - eg: “place order” steps varies : is it customer a registered member entitled to privileges?
 3. Each variation is called a scenario
 - a. Represents a possible situation that may arise
 4. Important to identify all possible scenarios
 - a) Good system must take care of all identified scenarios
 - b) Should we take care of every possible scenarios?

Use Case Descriptions

- Each use case is accompanied with a detailed use case description



Use Case Description Template

Use Case ID:	
Use Case Name:	
Created By:	
Date Created:	
Description:	
Primary Actor:	
Secondary Actor:	
Preconditions:	1. 2.
Postconditions:	1. 2.
Main Flow:	1. 2. 3.
Alternative Flows:	2a ScenarioName1 1. 2. 4a ScenarioName2 1. 2.

Use Case Description Components

1. Primary Actor
 - Actors that triggers/initiates the use case (initiator)
2. Secondary Actor
 - Actors that interact with the use case after it is triggered (participator)
3. Main flow
 - List the user actions and system responses that will take place during execution of the use case under normal, expected conditions.
 - The sequence of actions and responses will ultimately lead to accomplishing the goal stated in the use case name and description

Use Case Description Components

4. Alternative flows

- Document other, legitimate usage scenarios that can take place within the use case
- Each flow should be preceded with a scenario name
- List the user actions and system responses for the alternate scenario

Guidelines for use case description

- UML does not provide standard rules for writing detail use case descriptions
- There are **guidelines** that one can adhere to
- Use Simple Grammar, eg:
 - Librarian enters book ID
 - System displays book details
- Show clearly who is in control at point in time
 - Always start with:
 - The *actor* (replace actor with the actor's name in the use case)
 - The system ...

Guidelines for use case description

- Eg:
 - The customer enters quantity to purchase
 - The system adds items into order list
- Do not impose constraints on user interface
 - User ~~clicks on OK button~~ to submit personal details X
 - User submits personal details to system ✓
- User Input1 (by entering data):
 1. The actor enters
 2. The actor submits ...
 3. The system validates
 4. The system verifies ...

Guidelines for use case description

- User Input2 (by selecting data):
 1. The system retrieves list of ...
 2. The system displays list of ...
 3. The actor selects ...
 4. The actor submits
 5. The system validates
- System responses:
 - The system computes ...
 - The system prints ...
 - The system triggers ...
 - The system :-
 - inserts, creates, updates, deletes

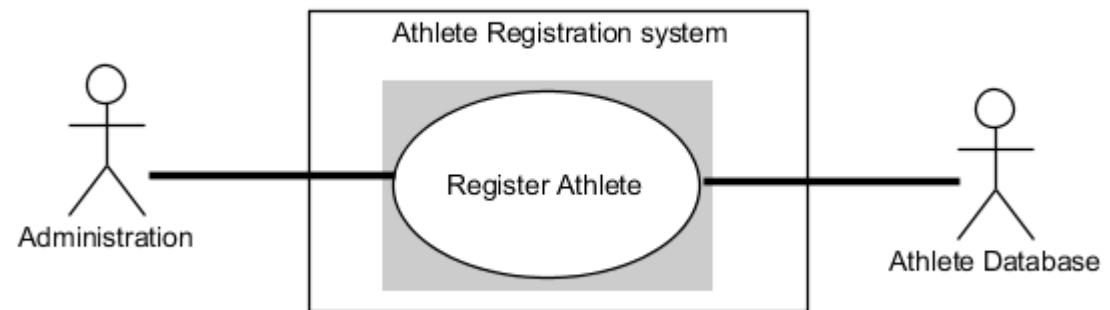
Use Case Description Example - login

Main Flow:	<ol style="list-style-type: none">1. The user enters the staffID and password2. The user submits the staffID and password3. The system validates the staffID and password4. The system verifies the staffID and password5. The system displays the user's homepage6. The use case ends
Alternative Flows:	<p>3a Missing staffID and/or password</p> <ol style="list-style-type: none">1. The system prompts for staffID and password2. Use case resumes at main flow step 1 <p>3b Maximum 3 attempts exceeded</p> <ol style="list-style-type: none">1. The system displays "Maximum attempts exceeded, contact administrator" message2. The system locks user account3. The use case ends <p>4a Invalid staffID and/or password</p> <ol style="list-style-type: none">1. The system displays "Invalid staffID and/or password" message2. The system prompts for staffID and password3. Use case resumes at main flow step 1

Use case description example

- **Requirements gathered:**

A use case is used by the athlete/official administrator to register the athlete / official. The user has to key in the personal information such as name, date of birth, gender, address, country, passport number, type of participation and contact. Passport number and contact must be only in numbers.



Use case description – First attempt

Use Case Name: Register Athletes

Actor: Administrator

Basic flow:

1. Administrator Choose **Create**
2. Administrator Key in **information**
3. The System validates the **information** entered.
4. The System **check** for duplicate record from the data base.
5. System **Creates** the Athletes record successfully.

Alternate flow

- 3.1) **Error message** displayed. The system prompts Administrator to key in the information correctly.
- 4.1) **Error message** displayed. The System prompts Administrator that there is an Athlete with the same information.

Use case description example (improved)

Basic Flow:

1. Administrator selects create athlete record option.
2. System prompts administrator for details .
3. Administrator enters athlete/official's details and submits.
4. System validates user personal details
5. System verifies the passport number for duplicate records.
6. System displays personal details, and prompts for confirmation.
7. Administrator confirms details entered.
8. System saves the personal details of the athlete/official in DB.
9. System displays success message.

Alternate Flows:

- 4A. Invalid personal details. Prompt user to re-enter.
- 5A. Duplicated records detected. Prompt user to re-enter.
- 8A. Record is not saved successfully.

Summary

- Different UML models are used to model **business events** and **objects**.
- Models are used to **document system requirements**
- Use case diagram is used to model functional requirements for users to appreciate what the system can do. Ie the Goal the user wants the system to achieve.
- Use case description is used to supplement a use case by defining the main flow (steps to achieve the goals) and alternate flows (exceptions) that the use case is expected to handle.