

104KM Enterprise Information System

Topic: Object-Oriented Analysis and Design – Part 2

Learning outcomes for today

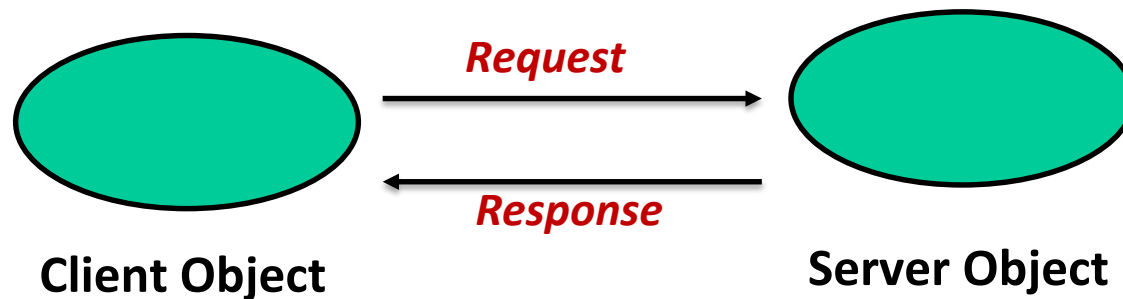
- Why use OO?
- UML Terminology
- UML Overview
- Use Case Diagram
 - Includes and Extends
 - Use Case Documentation

Benefits of OO Analysis and Design

- Localised changes.
- OO systems are more extensible and contribute to more rapid development.
- Suitable for distributed, parallel or sequential implementation.
- Correspond more closely to the entities of the real-world.
- Shared data areas are encapsulated.

Object-Oriented Analysis and Design

Data, processes and flows are modelled 'as one' allowing objects to be viewed as independent entities.



Modeling

- Describing a system at a high level of abstraction
 - A model of the system
 - Used for requirements and specifications
- Is it necessary to model software systems?

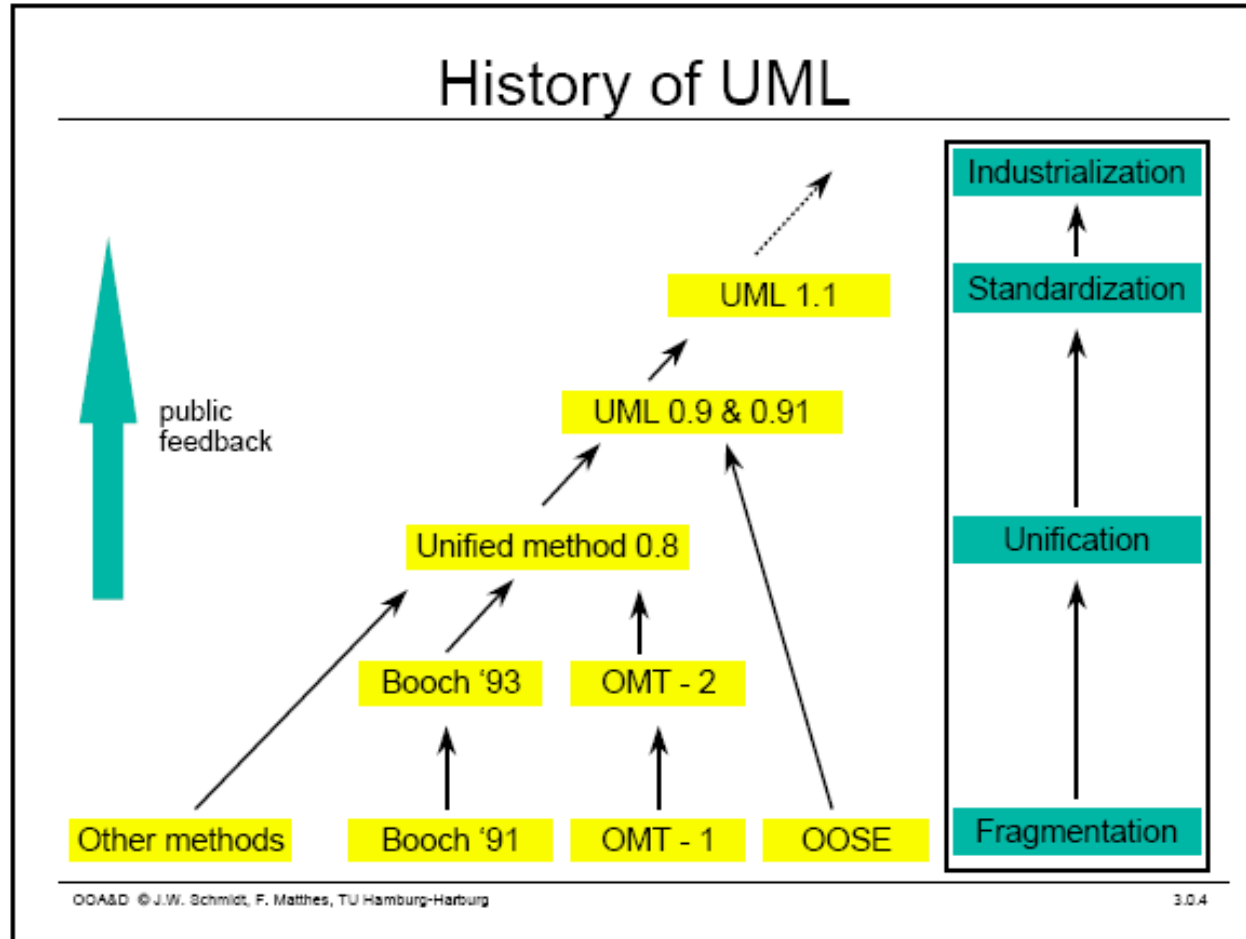
What is UML?

- UML stands for **Unified Modeling Language**.
- It is a industry-standard graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems.
- The UML uses mostly graphical notations to express the OO analysis and design of software projects.
- Simplifies the complex process of software design.

Why UML for Modeling?

- Use graphical notation to communicate more clearly.
- Help acquire an overall view of a system.
- UML is *not* dependent on any one language or technology.
- UML evolves from fragmentation to standardization.

UML



Note:

OMG – UML was created by Object Management Group

OMT – Object Management Technique

OOSE – Object-oriented Software Engineering

<https://www.omg.org/spec/UML/About-UML/>

UML

- UML or Unified Modeling Language comes from Rumbaugh, Booch, and Jacobson, who combined efforts to standardize on one modeling language.
- This is primarily a graphical communication mechanism for developers and customers.

UML

The main purpose of UML is to:

- support communication about the **analysis and design** of the system being developed,
- support the movement from the problem domain in the “real world” to the solution domain in the machine,
- present two views of the same system
 - ➔ *one view has diagrams*
 - ➔ *source code is another view.*

UML Defined by the Authors

The UML is a language for:

- visualizing
- specifying
- constructing
- documenting

the artifacts of a software intensive system.

Where Can the UML Be Used?

The UML is intended primarily for software intensive systems. It has been used effectively in:

- Enterprise Information Systems
- Banking and Financial Services
- Transportation
- Retail
- Medical Applications
- Scientific Applications
- Distributed Web-based Services

Key Components for Analysis

- **Use Case Diagram**

Models the business processes.

- **Use Case Documentation**

Support detailing business processes.

- **Class Diagram** with attributes and operations

Models static structure of a system rather than how it behaves.

What is a Use Case?

A depiction of the following requirement information:

- Basic functionality
- Any precondition for the functionality
- Flow of events (scenario) for the functionality
- Any post-condition for the functionality
- Any error condition and alternative flow

Requirement Analysis

For analysis or design of any system it is important to find out the following early on the systems development process.

Why is the system required?

What is it required to do?

When it is required to do it?

How is it required to do it?

Who will use the system?

THEREFORE.....

Capture Requirements

Elicitation techniques:

- Interview

- Questionnaires

- Observation

- Document sampling

Example - IT Online

IT Online is an Internet based IT company that sell computers, pre-build or customer can specify its configuration. Based on the following facts identify the key systems requirements and key people who are involved.

- Buying computers via the Internet.
- The customer can select a standard configuration or can build a desired configuration online.
- To purchase goods, the customer must register with the website during which process they are assigned a username and password.
- To place an order, the customer must:
 - firstly sign into website,
 - specify the product details.
- After which they must fill out the shipment and payment information.
- The customer can check online, at any time, the order status.
- The ordered configuration is shipped to the customer together with the invoice by the warehouse department.
- The invoice is prepared by the Accounting Department.

Requirements

No	Person/Entity Involved	Functionality
1	Customer	Display standard computer configuration
2	Customer	Order configured computer
3	Customer	Build computer configuration
4	Sales	Inform warehouse of order
5	Accounts	Print invoice
6	Customer	Register details
7	Customer	Sign into site
8	Sales	Update order status
9	Credit/Banking validation system	Verify and accept customer payment
10	Warehouse	Ship Order

Use Case Diagram

- A use case describes a set of flow or specification.
 - ➔ *Each scenario is a sequence of steps describing an interactions between a user and a system.*
 - ➔ *Each scenario corresponds to a single path or flow through a use case.*
- A use case captures the contract and “guarantees” that events will hold at the conclusion of the use case. Use cases are descriptions of:
 - ➔ *major functions the system will perform for its users*
 - ➔ *goals the system achieves along the way*

Use Case Diagram

- A visual representation of actors and use cases together with any additional definitions and specifications.
- Actors – represent whoever or whatever (person, machine or other system) that interacts with a use case or a part of the system.
- Goal oriented collaboration between a system and an actor.

Use Case Diagram

Use Case Diagram are helpful in three ways:

- Determine features (requirements) of the system
- Communicating with clients
- Generating test cases

Actor

- Actor is a role the user adopts.
- Actor can be a device or component as well as a person.
- Interact with or uses a system.
- Provides input to and receives information from the system.
- Is external to the system and has no control over the use cases.

Actors

Actors are discovered by examining:

Who directly uses the system.

Who is responsible for maintaining the system.

Other system that need to interact with the system.

The needs of the actor are used to develop use cases. This insures that the system will be what the user expected.

Use Cases

- In its simplest form, a use case can be described as a specific way of using the system from a user's (actor's) perspective.
- Pattern of behaviour the system exhibits.
- Sequence of related transactions performed by an actor and the system.
- Delivering something of value to the actor.

Use Case (Hint)

- Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.
- Since all the needs of the system typically cannot be covered in one use case, it is usual to have a collection of use cases. Together this use case collection specifies all the ways of using the system.

Use Case Diagram

Examples

Online C2C Shopping

System boundary:

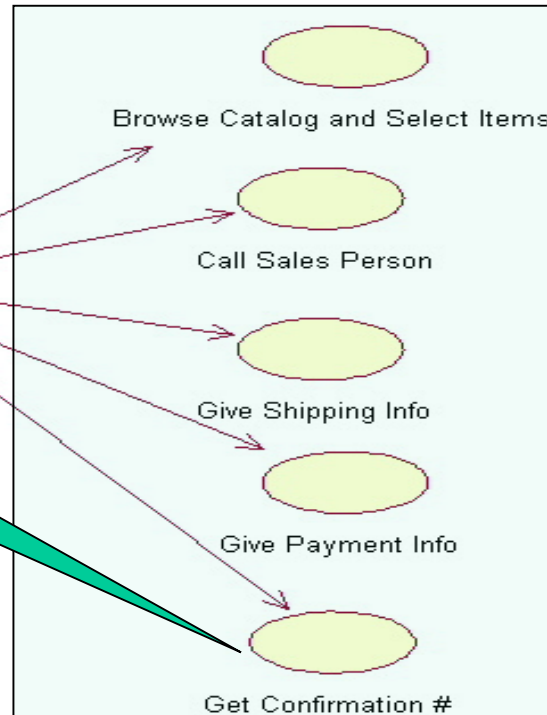
indicates the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not

System boundary

Actor

Customer

Use case



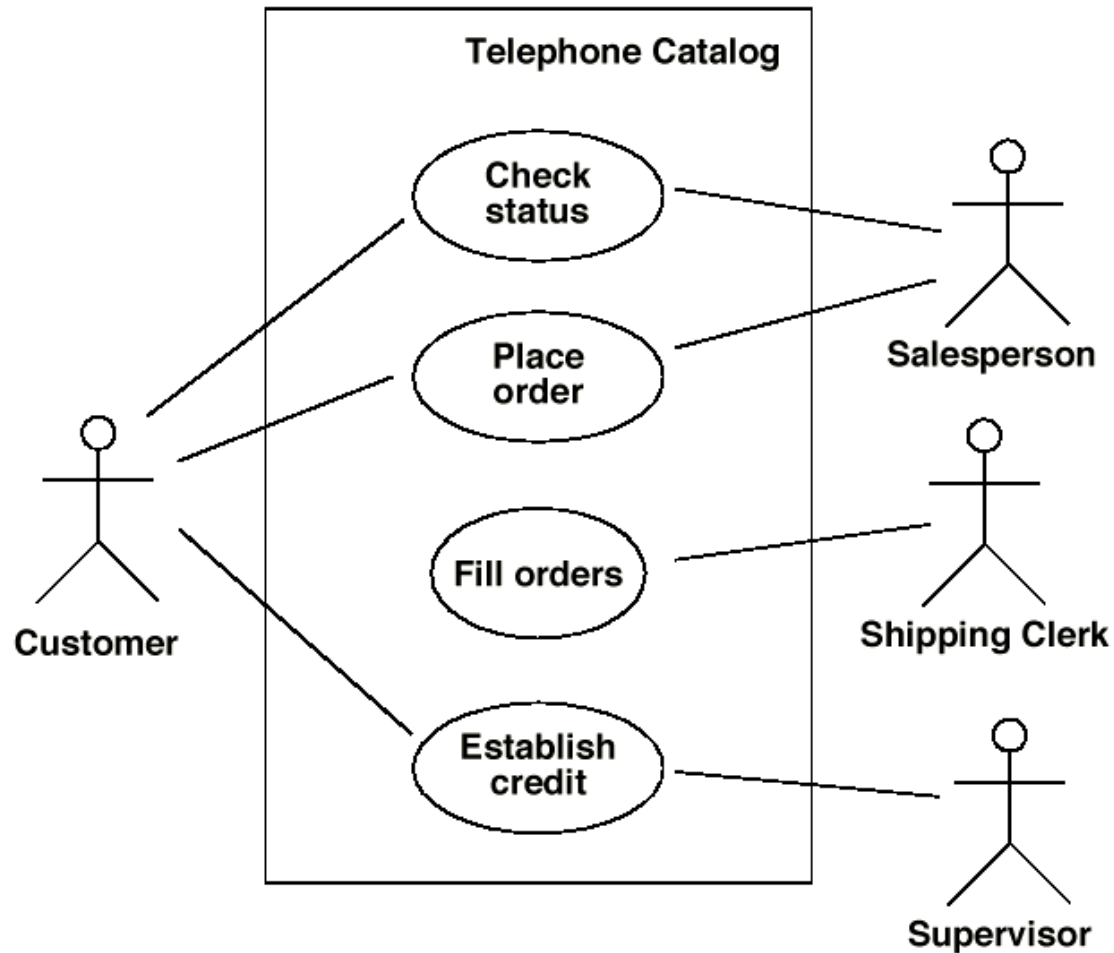
Actor:

An actor is a person, organization, or external system that plays a role in one or more interactions with your system

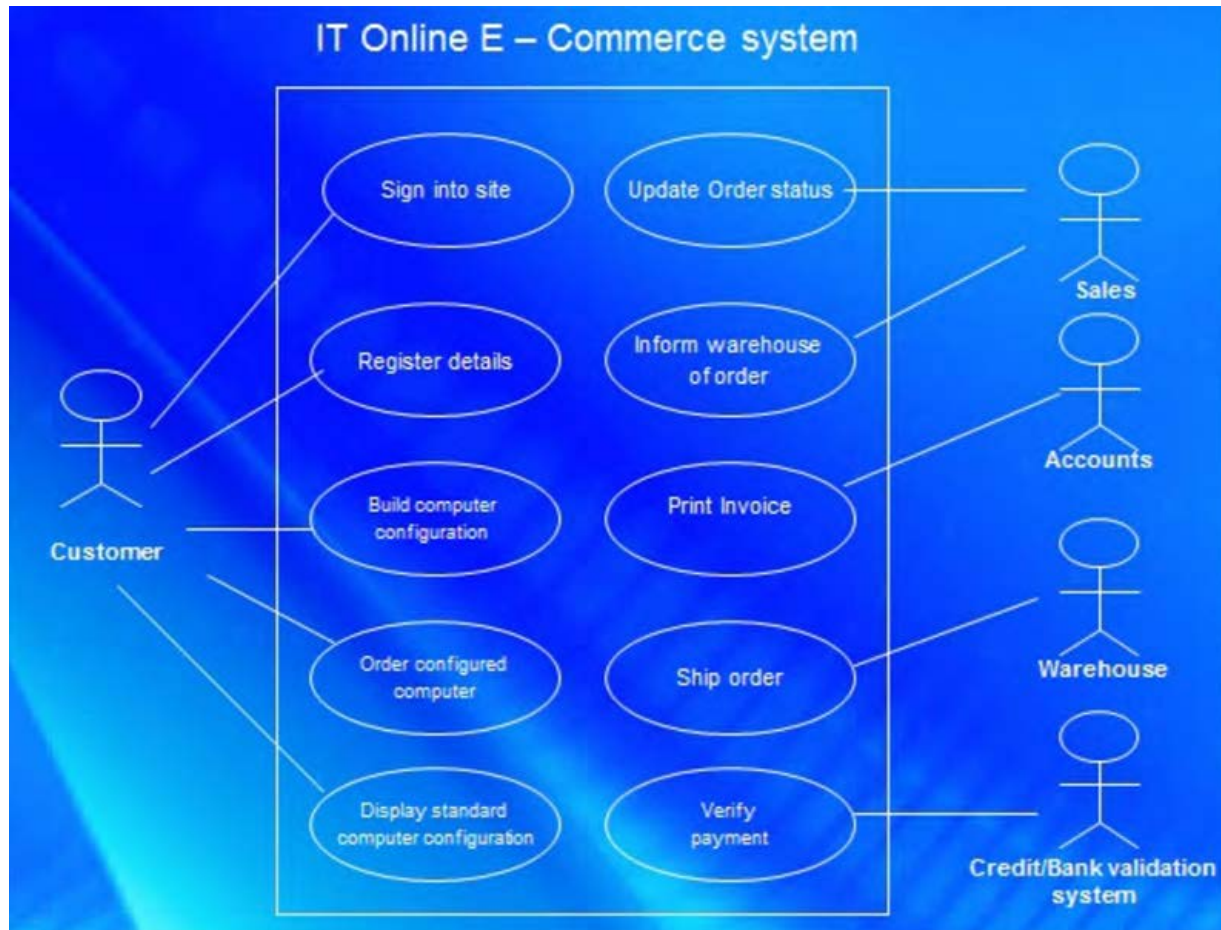
Use case:

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse

Use Case Diagram with Multiple Actors



IT Online



Dependencies

A dependency relationship between two modeling elements where a change to one will probably require change to the other because the one is dependent in some way on the other.

Includes and Extends

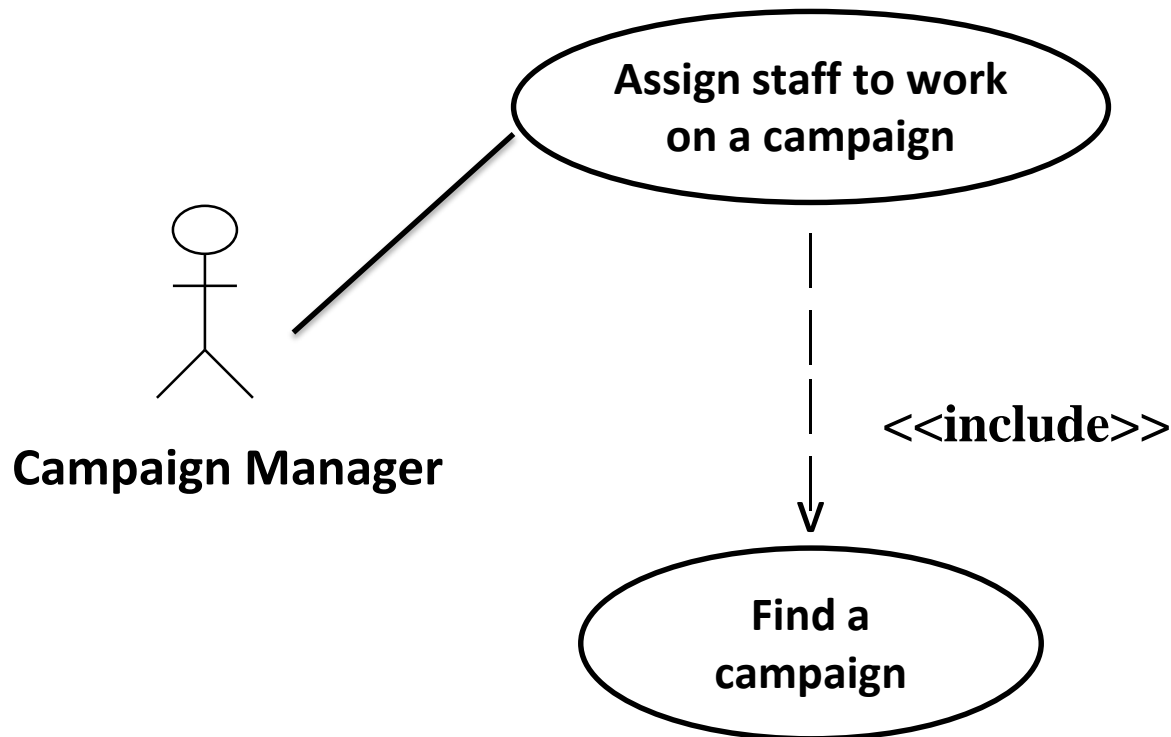
- Distinguish by stereotypes
- Behavioral constraints on the model element
- Shown using keywords:

<<include>>

<<extend>>

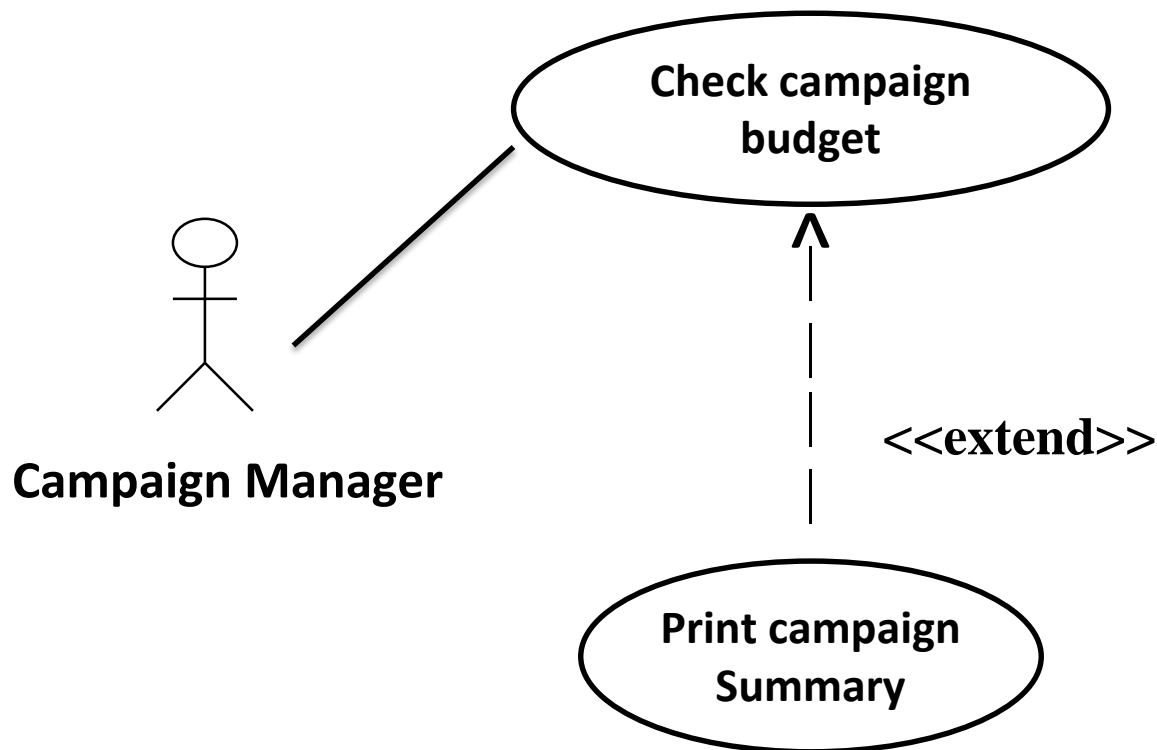
<<include>>

Applies when there is a sequence of behavior that is used **frequently** in a number of use cases.



<<extend>>

Used when you wish to show that a use case provides **additional functionality that may be** required in another use case.



Part 2

Use Case Documentation

Use Case Documentation

- What is Use Case Documentation?
- Structure of Use Case Documentation.
- How to produce Use Case description?

Use Case Documentation

Purpose?

- ➔ The use case diagram itself maybe ambiguous.
- ➔ The diagram on its own does not explain alternative scenarios.
- ➔ Helps to identify duplicate use case.
- ➔ Helps to identify contradictory use cases.
- ➔ It is a good practice to document as you go along.
- ➔ Helps to list all requirements.

Structure

Title

Brief Description

Actors

Pre-Conditions

Flow of Events

Post-Conditions

Alternative Scenarios

Documenting Use Cases

Title – Name of use case as per use case diagram.

Brief Description – A summary outlining the purpose of the particular use case.

Actors – To identify which actor interacts with the use case.

Documenting Use Cases

Pre-Conditions – What is necessary for the use case to start.

Flow of Events – A flow of events. This is normal course of events.

Alternative Scenarios – To define exceptional situations.

Post-Conditions – Define the state of the system after the use case ends.

Documenting Use Cases – *An Example*

Title

Order Computer

Brief Description

Allows customer to enter a purchase order.

Actors

Customer

Pre-Conditions

- Customer has signed in.
- Order entry webpage is displayed.
- Order item has been selected.
- Stock details are up-to-date.

Documenting Use Cases – *An Example*

Flow of Events

- Customer selects 'Buy' button.
- System requests item quantity.
- Customer enters item quantity.
- System requests delivery details.
- Customer enters delivery address.
- System accepts details.
- System request payment details – includes verify Payment details.

Documenting Use Cases – *An Example*

Post-Conditions

System generates order number.

Alternative Scenarios

- Item not in stock.
- Customer not signed in.
- Payment not accepted.

Summary

- *RUP is a Software Development Process.*
- *UML is a Modelling Language.*

Next Week

More UML

- *Class Diagram*
- *Sequence Diagram*