

# Use Case Diagrams

## Chapter 6B

## In this Lecture you will Learn:

- The purpose of use case diagrams
- The notation of use case diagrams
- How to draw use case diagrams
- How to write use case descriptions
- How prototyping can be used with use case modelling

## Purpose of Use Case Diagrams

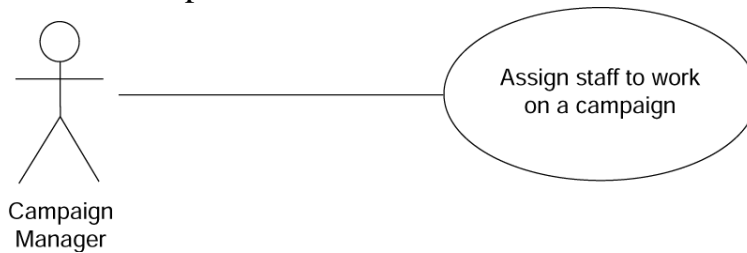
- Document the functionality of the system from the users' perspective
- Document the scope of the system
- Document the interaction between the users and the system using supporting use case descriptions (behaviour specifications)

## Class Exercise

- What is the purpose of producing use cases?

## Use Case Modeling

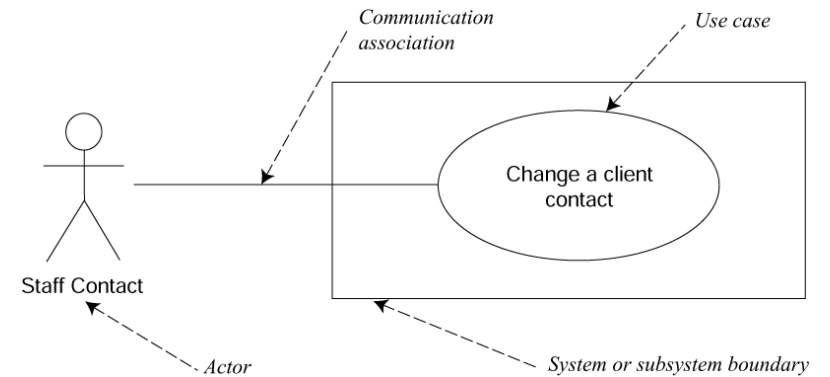
- A **Use Case** represents the steps in a specific business function or process.
- An external entity, called an **Actor**, initiates a use case by requesting the system to perform a function or process.



M8748 © Peter Lo 2007

5

## Notation of Use Case Diagrams



M8748 © Peter Lo 2007

6

## Notation of Use Case Diagrams

- Actors
  - ◆ Drawn as stick people with a name
  - ◆ The roles that people, other systems or devices take when communicating with a particular use case or use cases
  - ◆ Not the same as job titles or people
    - ◆ People with one job title may play the roles of several actors
    - ◆ One actor may represent several job titles

M8748 © Peter Lo 2007

7

## Notation of Use Case Diagrams

- Use Cases
  - ◆ Drawn as ellipses with a name in or below each ellipse
  - ◆ Describe a sequence of actions that the system performs to achieve an observable result of value to an actor
  - ◆ The name is usually an active verb and a noun phrase

M8748 © Peter Lo 2007

8

## Notation of Use Case Diagrams

- Communication Associations
  - ◆ Line drawn between an actor and a use case
  - ◆ Can have arrow heads to show where the communication is initiated (arrow points away from the initiator)
  - ◆ Represent communication link between an instance of the use case and an instance of the actor

## Notation of Use Case Diagrams

- System or Sub-systems Boundary
  - ◆ Drawn as a rectangle around a group of use cases that belong to the same sub-system
  - ◆ In a CASE tool, use cases for different sub-system are usually placed in separate use case diagrams, and the rectangle is redundant

## Notation of Use Case Diagrams

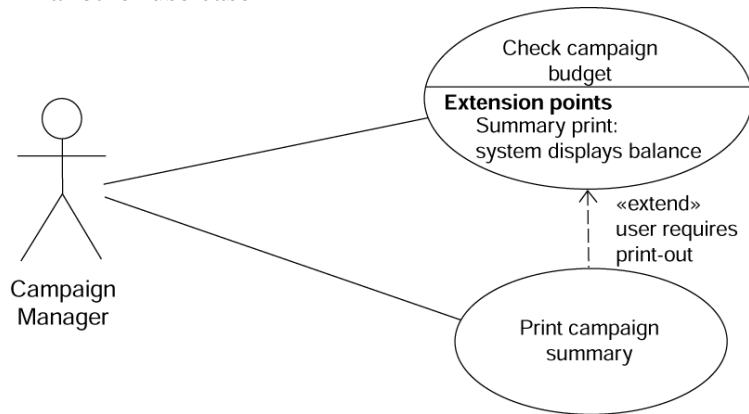
- Dependencies
  - ◆ **Extend** and **Include** relationships between use cases
  - ◆ Shown as stereotyped dependencies
  - ◆ Stereotypes are written as text strings in guillemets: «extend» and «include»

## Notation of Use Case Diagrams

- Extend Relationship
  - ◆ One use case provides additional functionality that **may** be required in another use case
  - ◆ There may be multiple ways of extending a use case, which represent variations in the way that actors interact with the use case
  - ◆ Extension points show when the extension occurs
  - ◆ A condition can be placed next to the dependency arrow (note that it is not put in square brackets, unlike conditions in activity diagrams.)

## Use Case Diagram showing «extend»

- <<extend>> is used when you wish to show that a use case provides additional functionality that may be required in another use case



13

## Notation of Use Case Diagrams

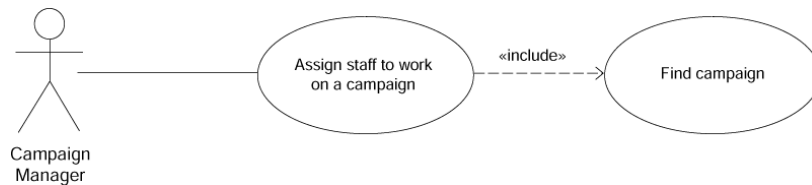
- Include Relationship
  - ◆ One use case **always** includes the functionality of another use case
  - ◆ A use case may include more than one other
  - ◆ Can be used to separate out a sequence of behaviour that is used in many use cases
  - ◆ Should not be used to create a hierarchical functional decomposition of the system

M8748 © Peter Lo 2007

14

## Use Case Diagram showing «include»

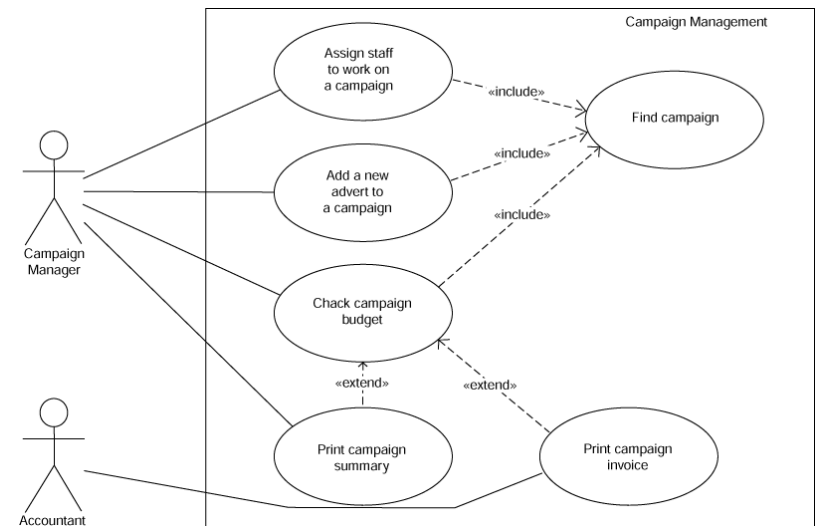
- <<include>> applies when there is a sequence of behavior that is used frequently in a number of use cases, and you want to avoid copying the same description of it into each use case in which it is used.



M8748 © Peter Lo 2007

15

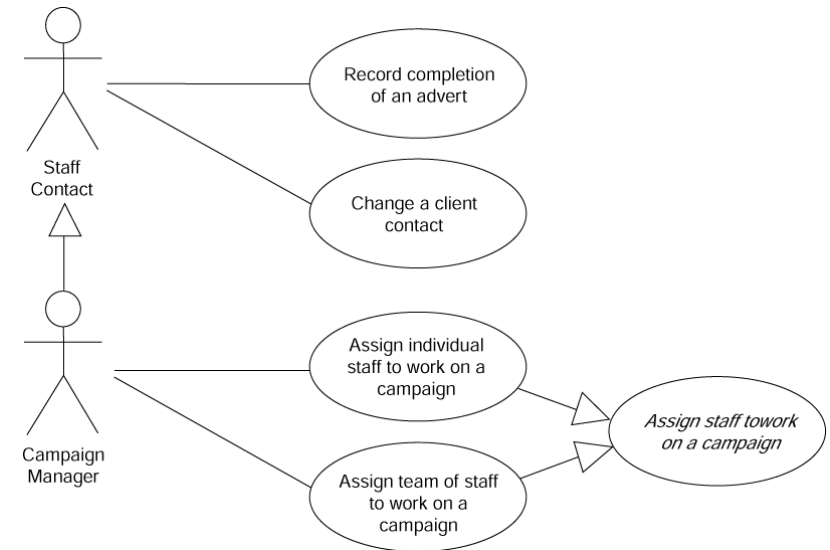
## Use Case diagram showing both «include» and «extend»



## Notation of Use Case Diagrams

- Generalization
  - ◆ Shows that one use case provides all the functionality of the more specific use case and some additional functionality
  - ◆ Shows that one actor can participate in all the associations with use cases that the more specific actor can plus some additional use cases

## Generalization of Actors and Cases



## Drawing Use Case Diagrams

- Identify the actors and the use cases
- Prioritize the use cases
- Develop each use case, starting with the priority ones, writing a description for each
- Add structure to the use case model: generalization, include and extend relationships and sub-systems

## Class Exercise

- Describe in your own words the difference between the «extend» and «include» relationships in use case diagrams.

## Class Exercise

- What is the difference between an ‘essential’ and a ‘real’ use case?

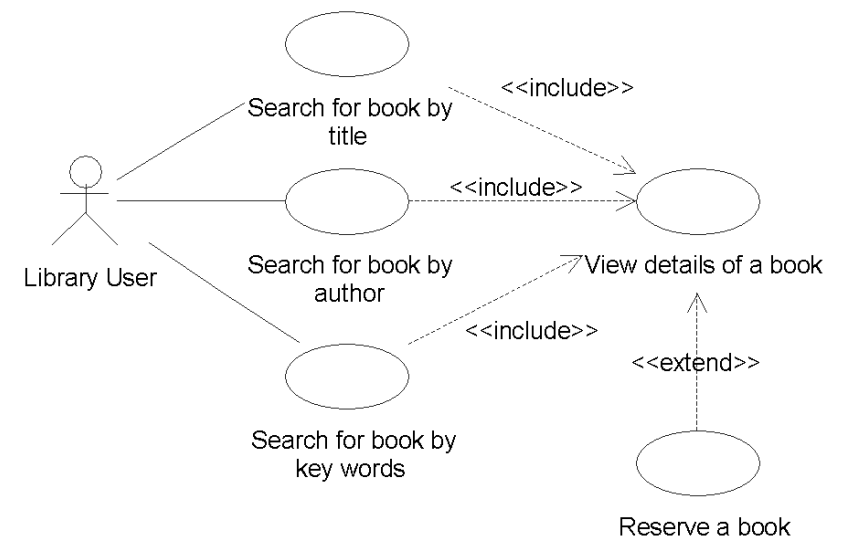
## A Sample Use Case Example

- Think of the different possible uses you could make of a library computer system and draw a use case diagram to represent these use cases.
- List some non-functional requirements a library computer system that you would not model using use cases.

## A Sample Use Case Example

- Typical use cases for a library system from the point of view of the user are:
  - ◆ Search for a book by title.
  - ◆ Search for a book by author.
  - ◆ Search for a book by key words.
  - ◆ View details of a book.
  - ◆ Reserve a book.

## A Sample Use Case Example



## A Sample Use Case Example

- The system must be capable of holding details of 20,000 users and 500,000 books (or whatever figures you think might be appropriate).
- The system will begin to respond to all user search enquiries within 5 seconds.
- Personal details of users will only be accessible to authorized staff and will be protected using a password system.

## Use Case Description

- For each use case, you also develop a use case description in the form of a table.
- A use case description documents the name of the use case, the actor, a description of the use case, a step-by-step list of the tasks and actions required for successful completion, a description of alternative courses of action, pre-conditions, post-conditions, and assumptions.

## Use Case Descriptions

- Can be a simple paragraph

### **Assign staff to work on a campaign**

*The campaign manager wishes to record which staff are working on a particular campaign. This information is used to validate timesheets and to calculate staff year-end bonuses.*

## Use Case Descriptions

- Can be a step-by-step breakdown of interaction between actor and system

### **Assign staff to work on a campaign**

#### **Actor Action**

1. The actor enters the client name.
3. Selects the relevant campaign.
5. Highlights the staff members to be assigned to this campaign.

#### **System Response**

2. Lists all campaigns for that client.
4. Displays a list of all staff members not already allocated to this campaign.
6. Presents a message confirming that staff have been allocated.


#### **Alternative Courses**

Steps 1–3. The actor knows the campaign name and enters it directly.

## Use Case Descriptions

- Many projects use templates
  - ◆ Name of use case
  - ◆ Pre-conditions
  - ◆ Post-conditions
  - ◆ Purpose
  - ◆ Description
  - ◆ Alternative courses
  - ◆ Errors

## Use Case Description Example

	
<b>Add New Student Use Case</b>	
<b>Name:</b>	Add New Student
<b>Actor:</b>	Student/Manager
<b>Description:</b>	Describes the process used to add a student to a fitness-class
<b>Successful completion:</b>	<ol style="list-style-type: none"><li>1. Manager checks FITNESS-CLASS SCHEDULE object for availability</li><li>2. Manager notifies student</li><li>3. Fitness-class is open and student pays fee</li><li>4. Manager registers student</li></ol>
<b>Alternative:</b>	<ol style="list-style-type: none"><li>1. Manager checks FITNESS-CLASS SCHEDULE object for availability</li><li>2. Fitness-class is full</li><li>3. Manager notifies student</li></ol>
<b>Precondition:</b>	Student requests fitness-class
<b>Postcondition:</b>	Student is enrolled in fitness-class and fees have been paid
<b>Assumptions:</b>	None

## Behaviour Specifications

- Rather than (or as well as) using text, a use case can be linked to another diagram that specifies its behaviour
- Typically a Collaboration Diagram, a Sequence Diagram or a Statechart Diagram

## Supporting Use Case with Prototyping

- Use case modelling can be supported with prototyping
- Prototypes can be used to help elicit requirements
- Prototypes can be used to test out system architectures based on the use cases in order to meet the non-functional requirements



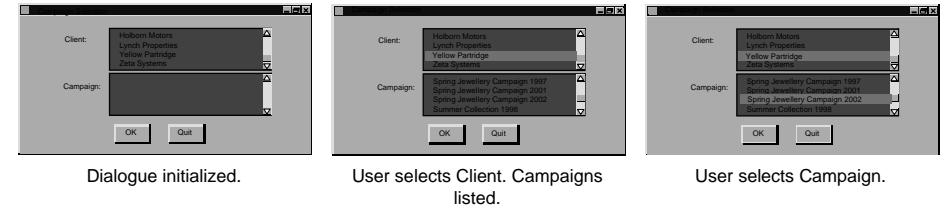
# Prototyping

- For user interface prototypes, storyboarding can be used with hand-drawn designs



# Prototyping

- User interface prototypes can be implemented using languages other than the one that the system will be developed in

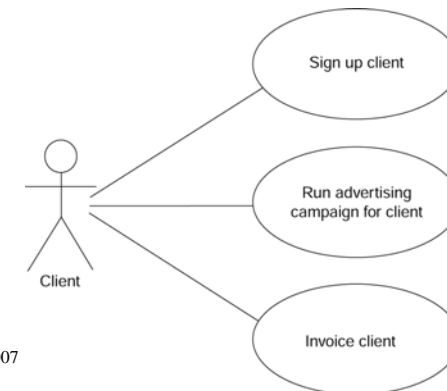


# Business Modeling with Use Case Diagram

- Use case can be used earlier in the life of a project to model an organization and how it operates.
- Business modeling is sometimes used when a new business is being set up, when an existing business is being “reengineered”, or in a complex project to ensure that the business operation is correctly understood before starting to elicit the requirement.

# Example of Business Modeling with Use Case

- In business modeling, the actors are the people and organizations outside the company, interacting with functions within the company.



## Class Exercise

- In what way are use case diagrams different when used for business modeling?