Use Case Diagrams

Based on Chapter 6 of Bennett, McRobb and Farmer:

Object Oriented Systems Analysis and Design Using UML, (3rd Edition), McGraw Hill, 2005.
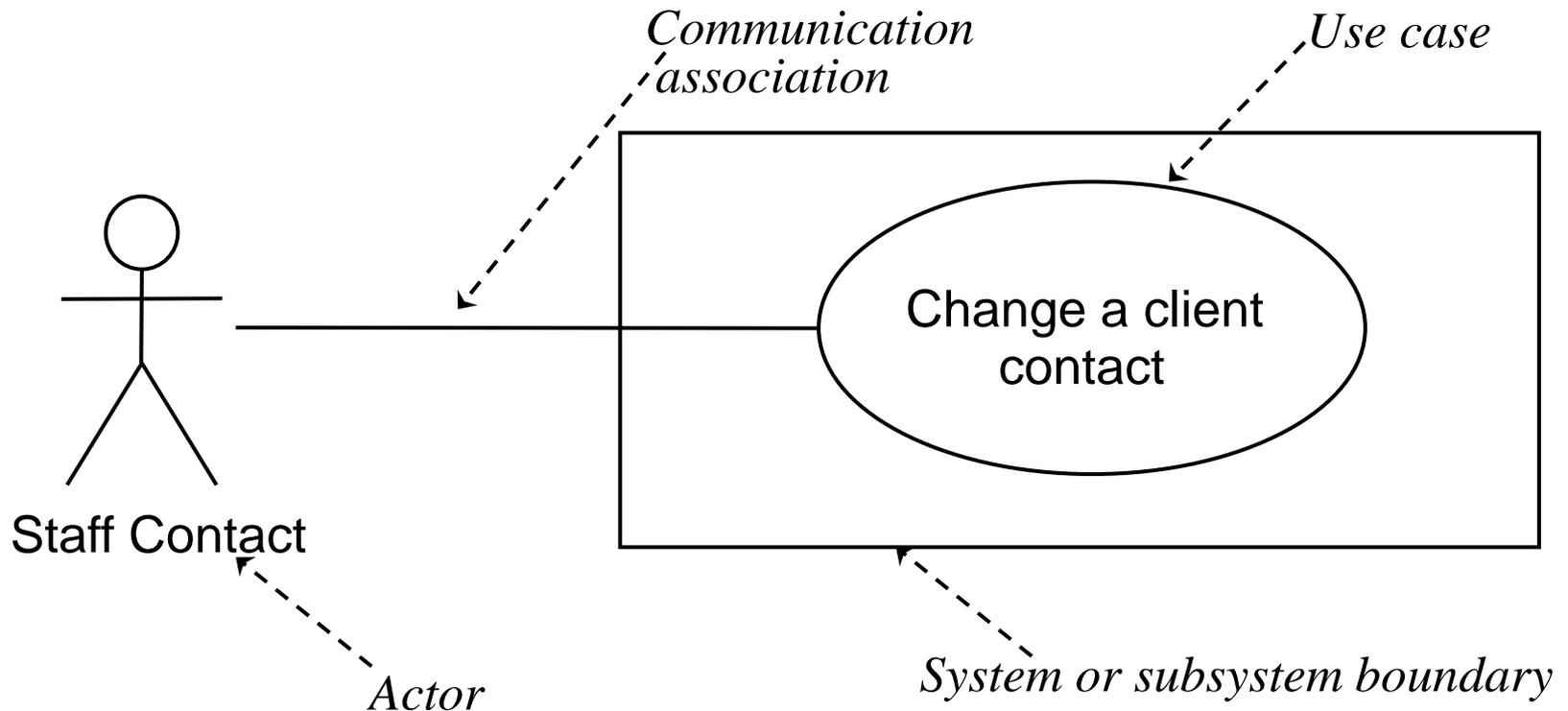
In This Lecture You Will Learn:

- 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

Drawing Use Case Diagrams

- Purpose
  - 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)

Notation of Use Case Diagrams

Communication association

Use case

Change a client contact

Staff Contact

Actor

System or subsystem boundary

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

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

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

- Sub-systems
  - 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
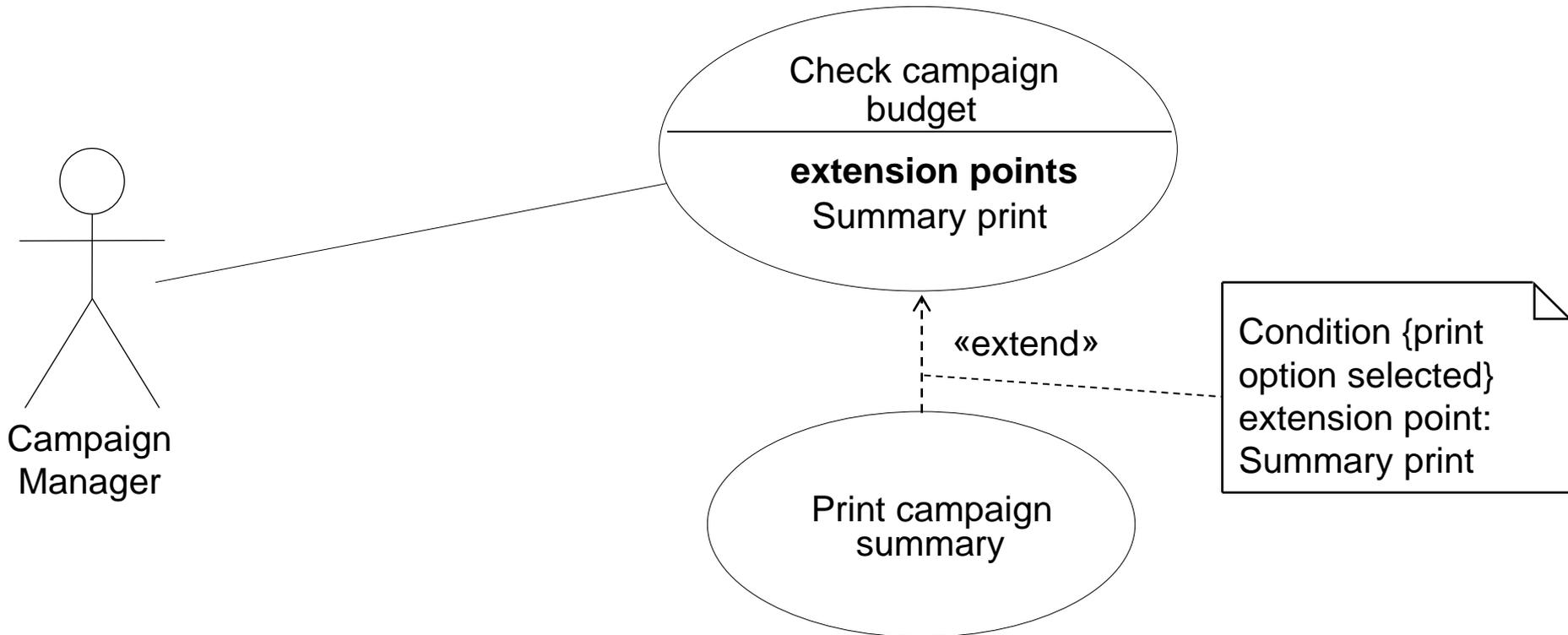
Notation of Use Case Diagrams

- 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 in a note joined to the dependency arrow (Note that it is not put in square brackets, unlike conditions in other diagrams.)

Extend relationship

Check campaign budget

extension points
Summary print

«extend»

Condition {print option selected}
extension point:
Summary print

Print campaign summary

Campaign Manager

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

Include Relationship

Campaign Manager → Assign staff to work on a campaign «include» → Find campaign
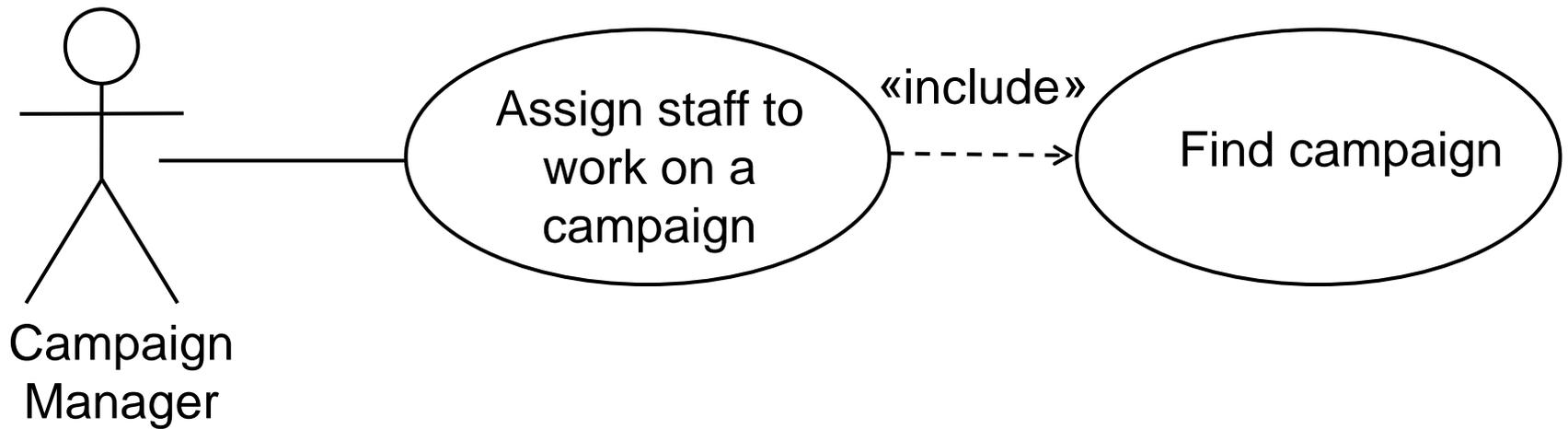
Notation of Use Case Diagrams

- 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
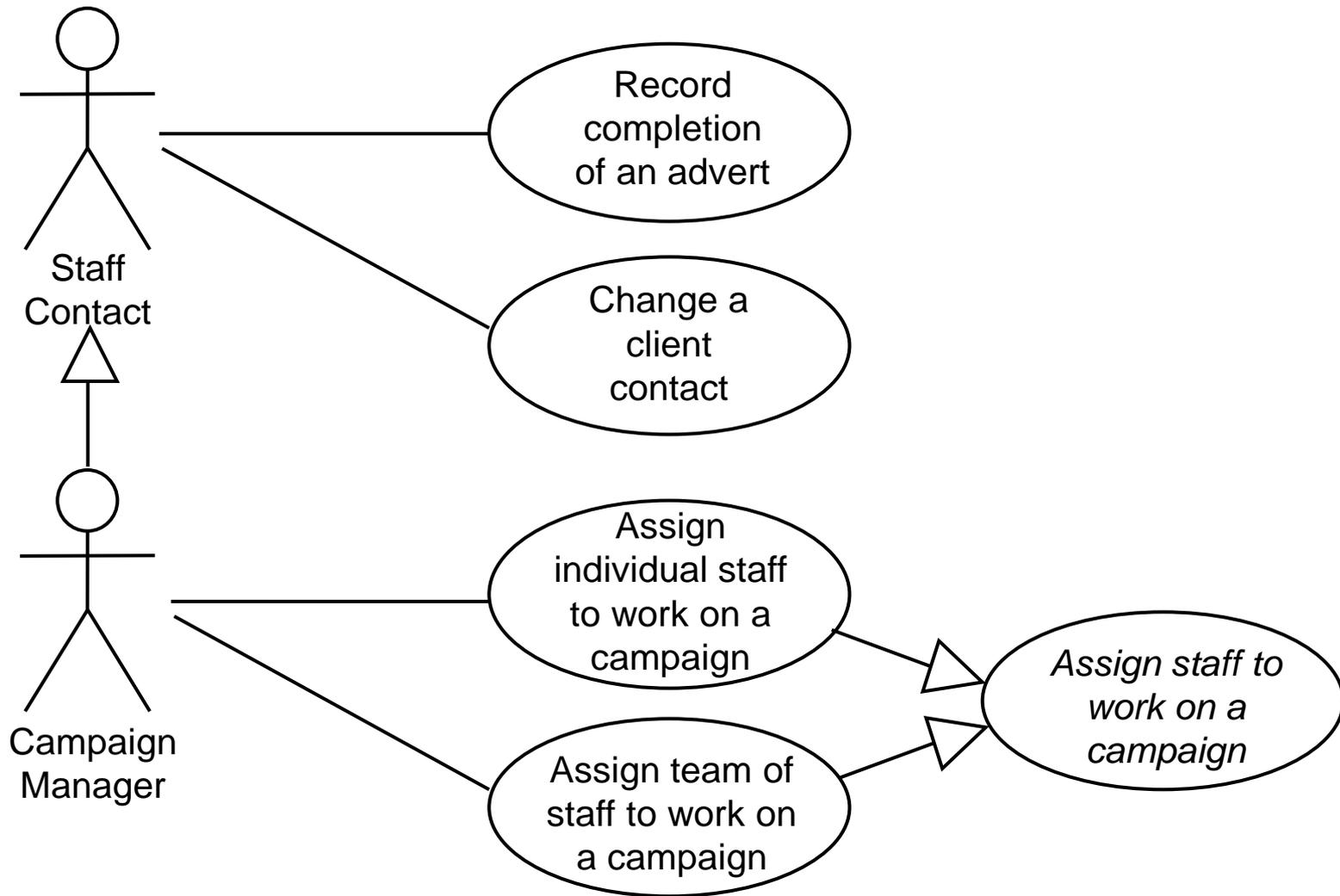
Use Case Descriptions

- 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	System Response
1. The actor enters the client name.	2. Lists all campaigns for that client.
3. Selects the relevant campaign.	4. Displays a list of all staff members not already allocated to this campaign.
5. Highlights the staff members to be assigned 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

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 Communication Diagram, a Sequence Diagram, a State Machine or more than one of these

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 subsystems
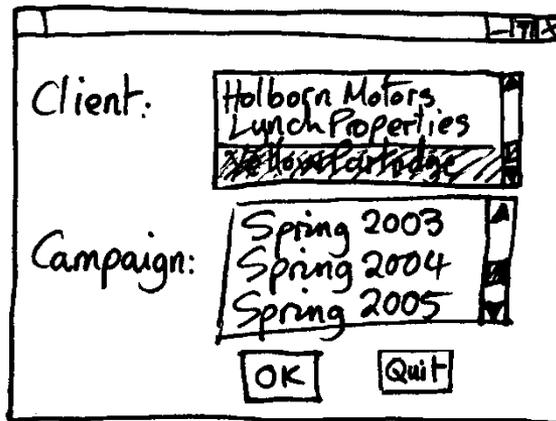
Prototyping

- 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
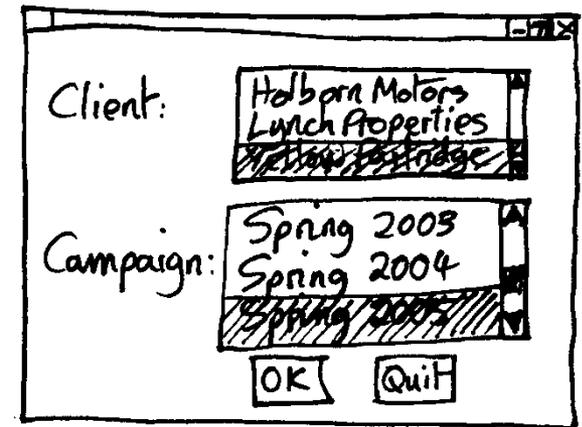
Prototyping

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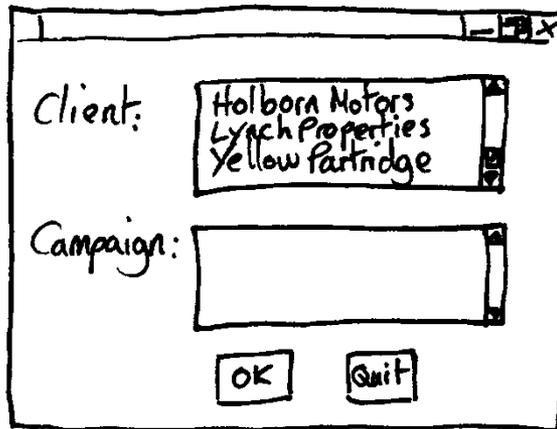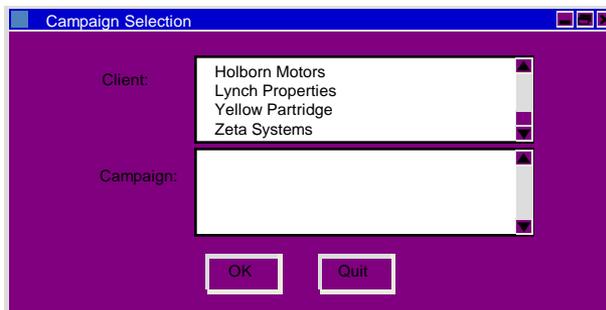

Dialogue initialized.

User selects Client. Campaigns listed.

User selects Campaign.

Prototyping

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

Dialogue initialized.

User selects Client. Campaigns listed.

User selects Campaign.

Summary

In this lecture you have learned about:

- 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

References

- Jacobson et al. (1992)
- Rosenberg and Scott (1999)
- Cockburn (2000)

(For full bibliographic details, see Bennett, McRobb and Farmer)