Use case

Lecture # 21, 22
18, 19 March

Rubab Jaffar
rubab.jaffar@nu.edu.pk

Introduction to Software Engineering

SE-110

Today’s Outline

- Use Cases
- Use Cases Notations
- Examples
- Exercises

Use Cases
Ivar Jacobson 1994
Use Case Model

"What will the System do?"

Use Cases

- What is a Use Case?
  - A scenario-based technique in the UML
  - A formal way of representing system functionality, the requirements of the system from the user's perspective.
  - representing how a system interacts with its environment
- Use Case diagram: that shows a set of use cases and actors and their relationships.

Use Case Diagram

– Guidelines & Caution

1. Use cases should ideally begin with a verb – i.e generate report. Use cases should NOT be open ended – i.e Register (instead should be named as Register New User)

2. Avoid showing communication between actors

3. Actors should be named as singular. i.e student and NOT students. NO names should be used – i.e John, Sam, etc.

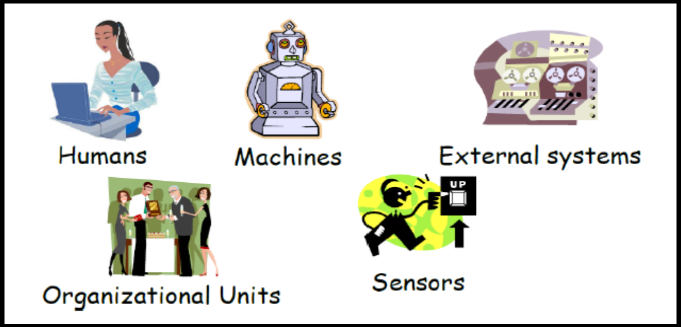
4. Do NOT show behaviour in a use case diagram; instead only depict only system functionality.

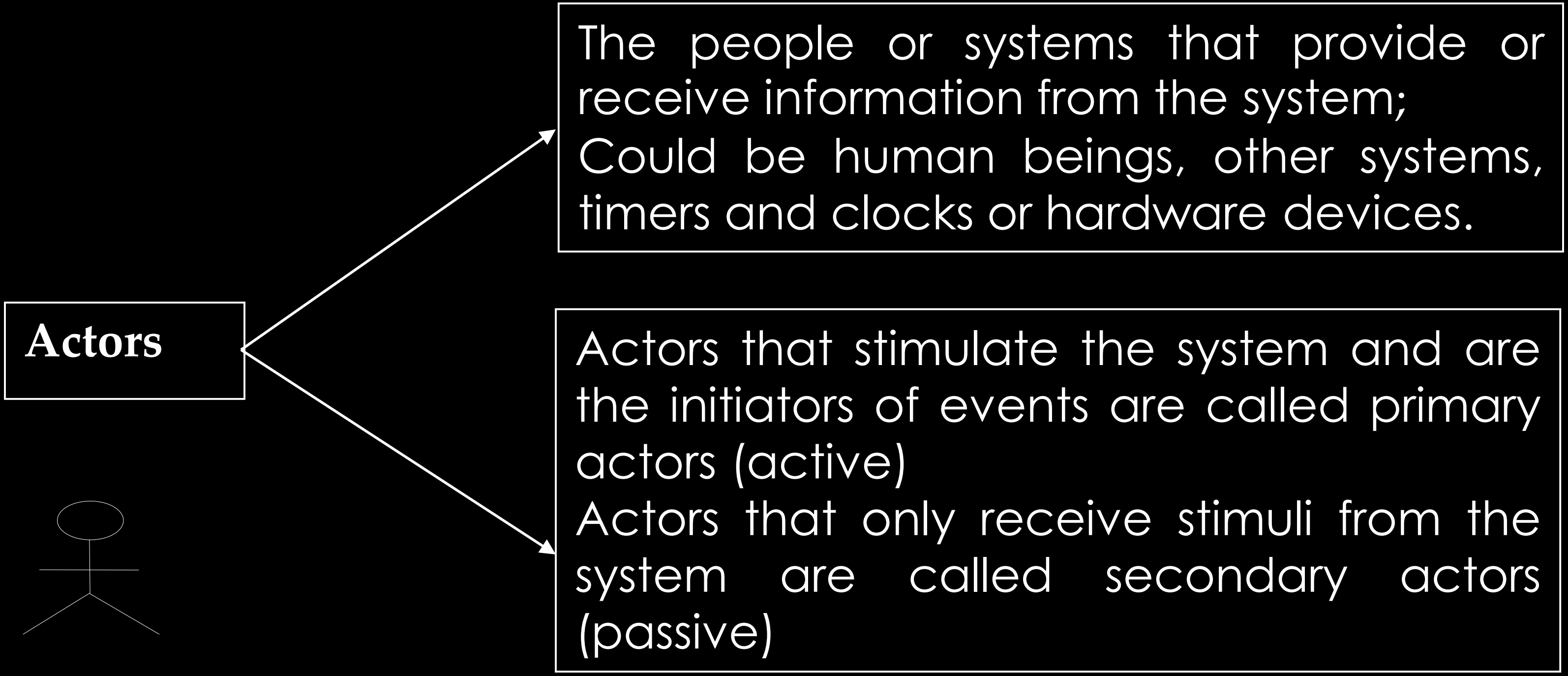
5. Use case diagram does not show sequence – unlike DFDs

Components of Use Case Diagram

- Actors
- Use Case
- Relationship
- Boundary

Actors



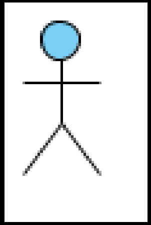


Actors

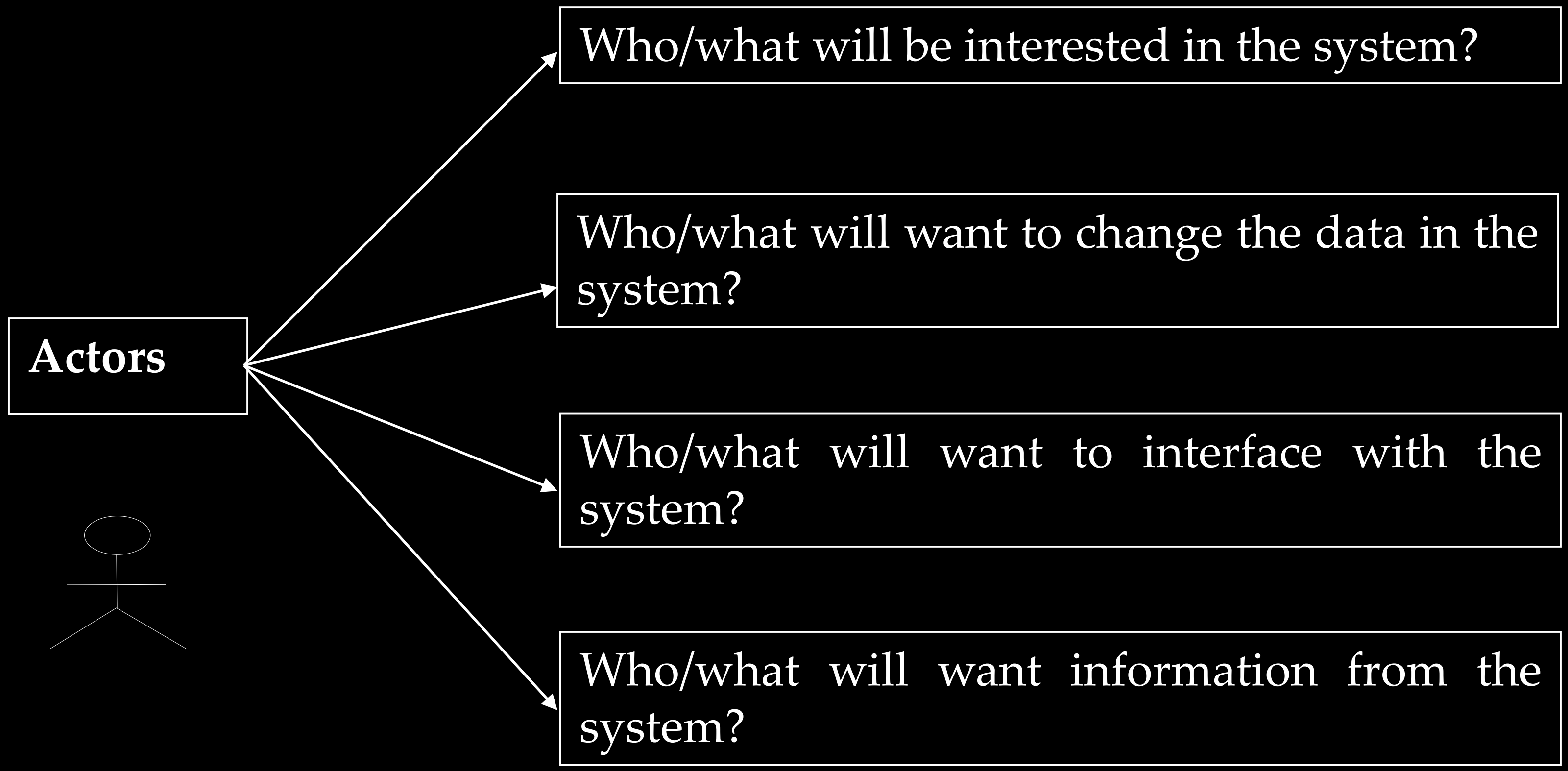
The people or systems that provide or receive information from the system; Could be human beings, other systems, timers and clocks or hardware devices.

Actors that stimulate the system and are the initiators of events are called primary actors (active)
Actors that only receive stimuli from the system are called secondary actors (passive)

Notation



Actors



Actors

Who/what will be interested in the system?

Who/what will want to change the data in the system?

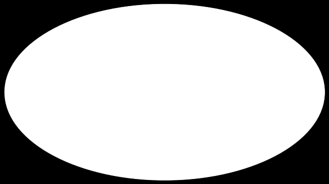
Who/what will want to interface with the system?

Who/what will want information from the system?

Components of Use Case Model

- Use Case
  - Define the functionality that is handled by the system.
  - Each use case specifies a complete functionality (from its initiation by an actor until it has performed the requested functionality).
  - Describes the interactions between various actors and the system.

- Notation



Use Case - Relationships and its Types

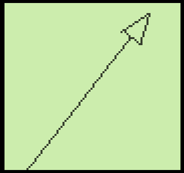
- Relationships
  - Represent communication between actor and use case
- 4 types of relationships
  - o Association relationship
  - o Generalization relationship
    - Generalization relationship between actors
    - Generalization relationship between use cases
  - o Include relationship between use cases
  - o Extend relationship between use cases

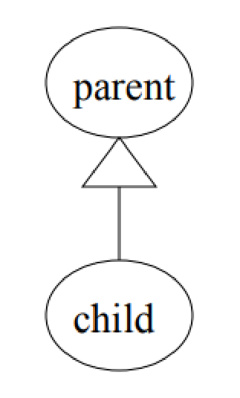
Use Case - Relationships and its Types

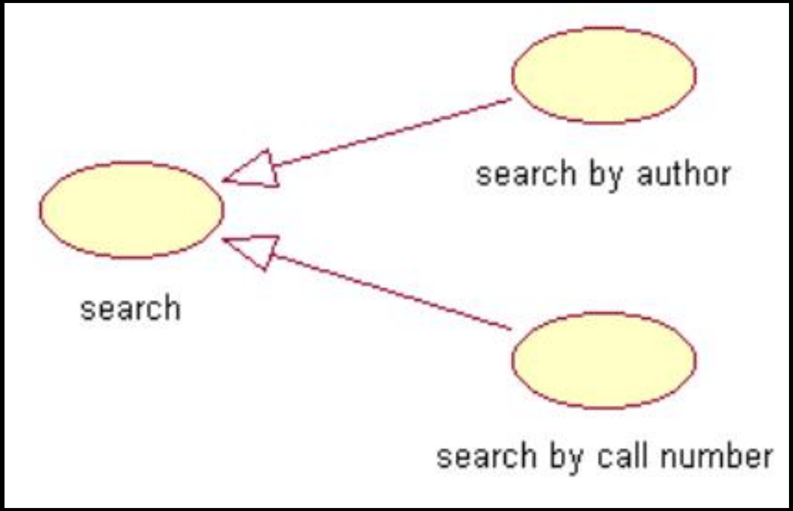
- Association relationship: Represent communication between actor and use case
- Often referred to as a communicate association
- use just a line to represent
- Notation

Use Case - Relationships and its Types

- Generalization:
- The child use case inherits the behaviour and meaning of the parent use case.
- The child may add to or override the behaviour of its parent.
- Notation:







Use Case - Relationships and its Types

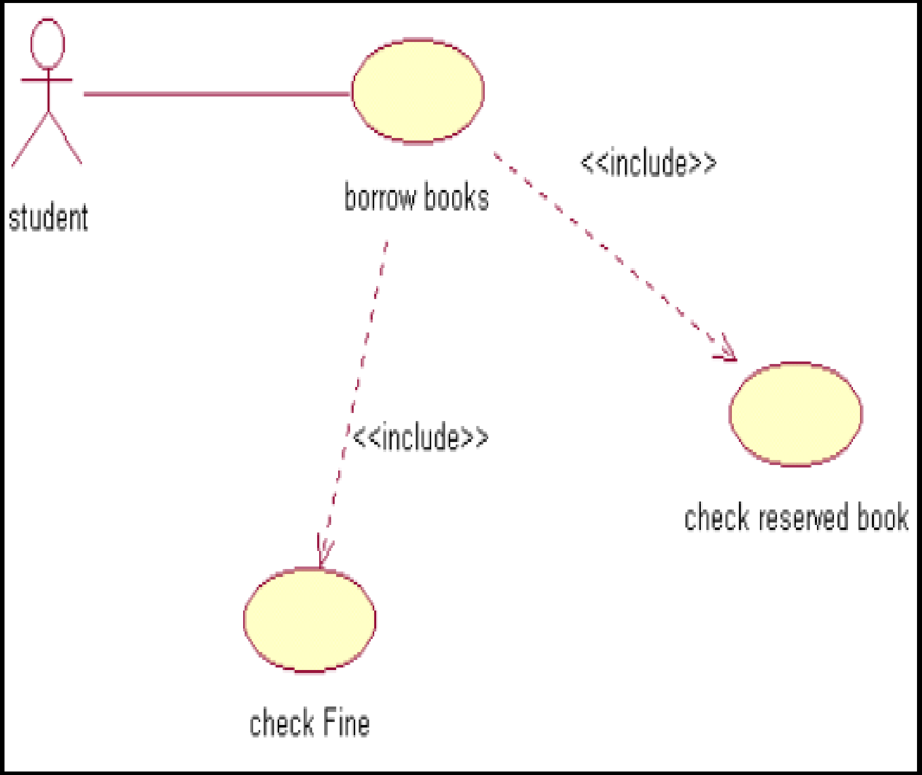
- **Include**
  - Specifies that the source use case explicitly incorporates the behavior of another use case at a location specified by the source
  - The include relationship adds additional functionality not specified in the base use case.
  - <<include>> is used to include common behaviour from an included use case into a base use case
  - Notation

<<include>>

---------------->

<<include>>

- An include relationship connects a base use case (i.e. borrow books) to an inclusion use case (i.e. check Fine).
- An include relationship specifies how behaviour in the inclusion use case is used by the base use case.



Use Case - Relationships and its Types

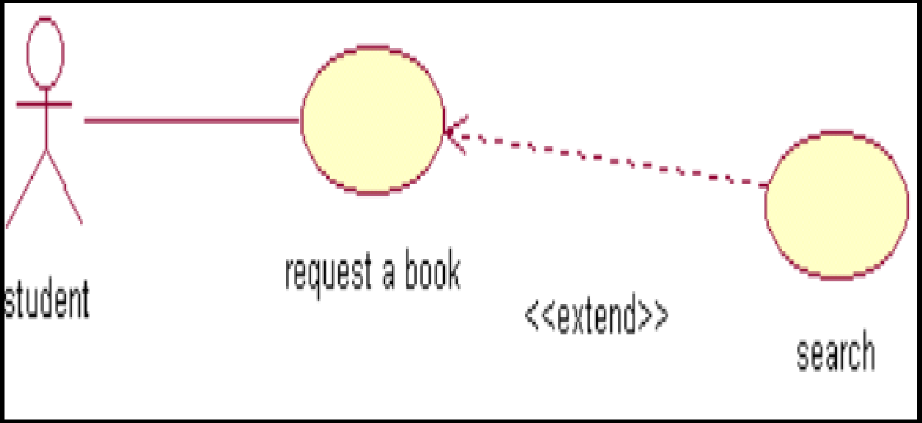
- **Extend**
  - o Specifies that the target use case extends the behavior of the source.
  - o The extend relationships shows optional functionality or system behaviour.
  - o <<extend>> is used to include optional behaviour from an extending use case in an extended use case.
  - o Notation

<<extend>>

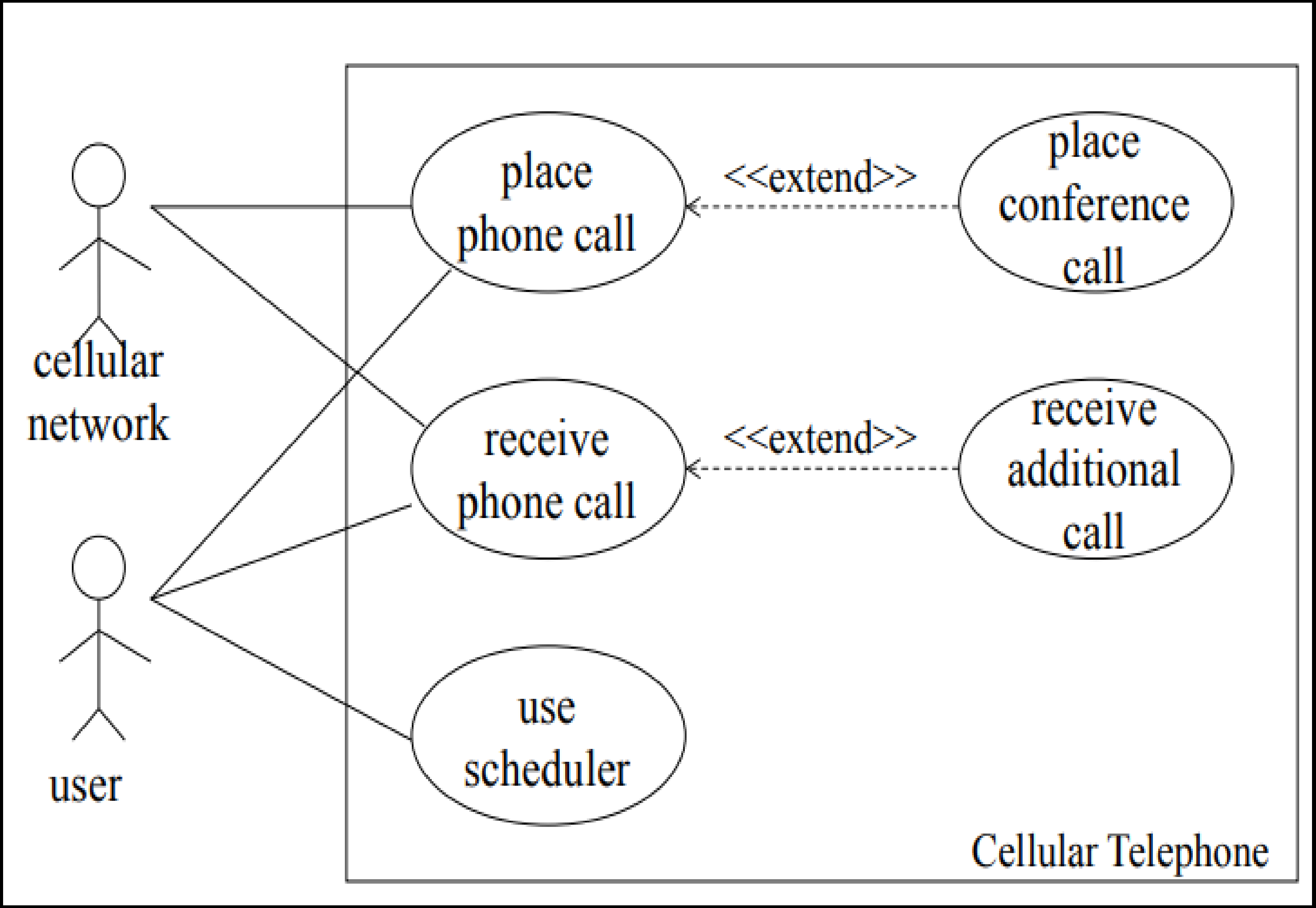
<----------------

<<extend>>

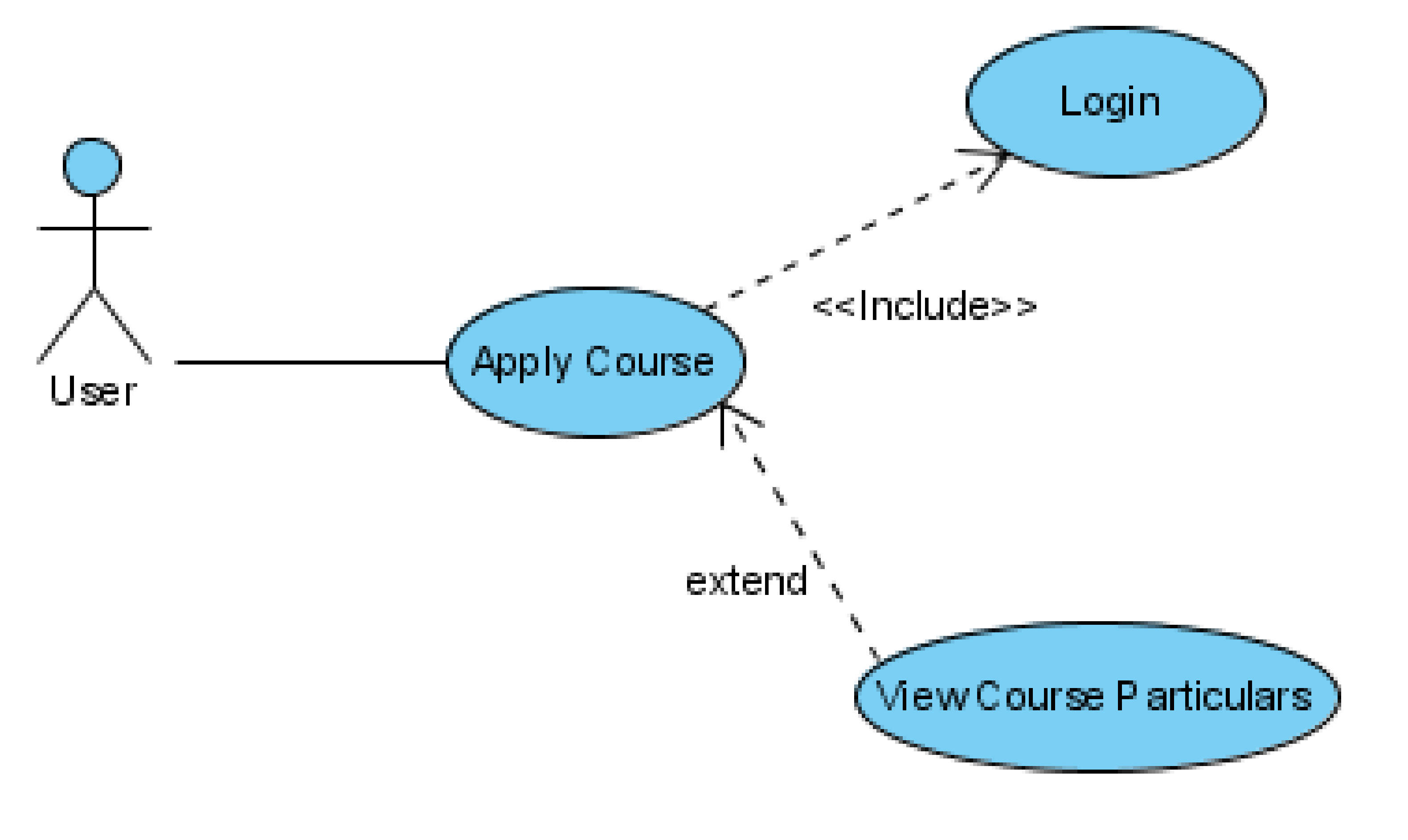
- The extend relationship is in between Request a book and Search.
- If the student desires, he/she can search the book through the system.
- However, the student may only Request a book through the system without searching the book if the student knows the call number.



Cellular telephone



Example – Include and Extend

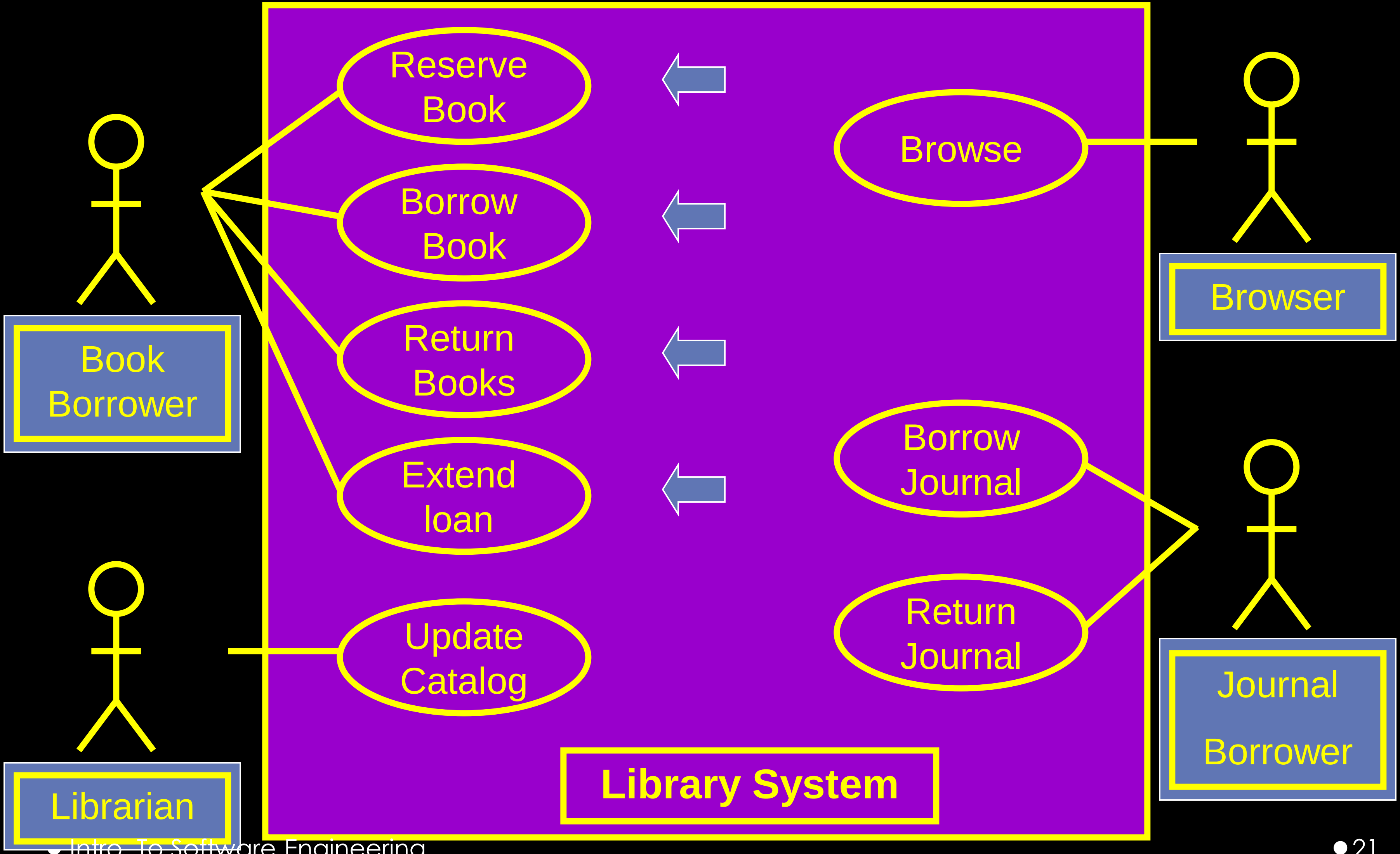


The use of include and extend is discouraged simply because they add unnecessary complexity to a Use Case diagram.

Use Case - Boundary

- Boundary
  - A boundary rectangle is placed around the perimeter of the system to show how the actors communicate with the system.
  - A system boundary of a use case diagram defines the limits of the system.

Use Diagram for a Library System



Use Cases

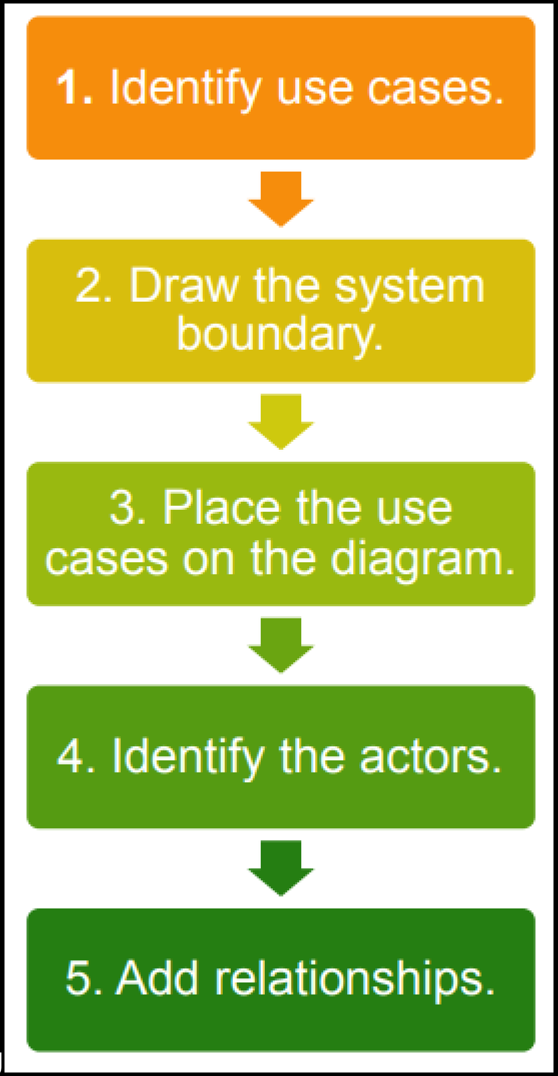
- Here is a scenario for a purchasing items.
- “Customer arrives at checkout with items to purchase in cash. Cashier records the items and takes cash payment. On completion, customer leaves with items. "
- We want to write a use case for this scenario.
- Remember: A **use case** is a summary of scenarios for a single task or goal.

Use Cases

- Step 1 Identify the actors
- As we read the scenario, define those people or systems that are going to interact with the scenario.

- Customer arrives at checkout with items to purchase in cash. Cashier records the items and takes cash payment. On completion, customer leaves with items.

Steps involved in creating use cases



1. Identify use cases.

2. Draw the system boundary.

3. Place the use cases on the diagram.

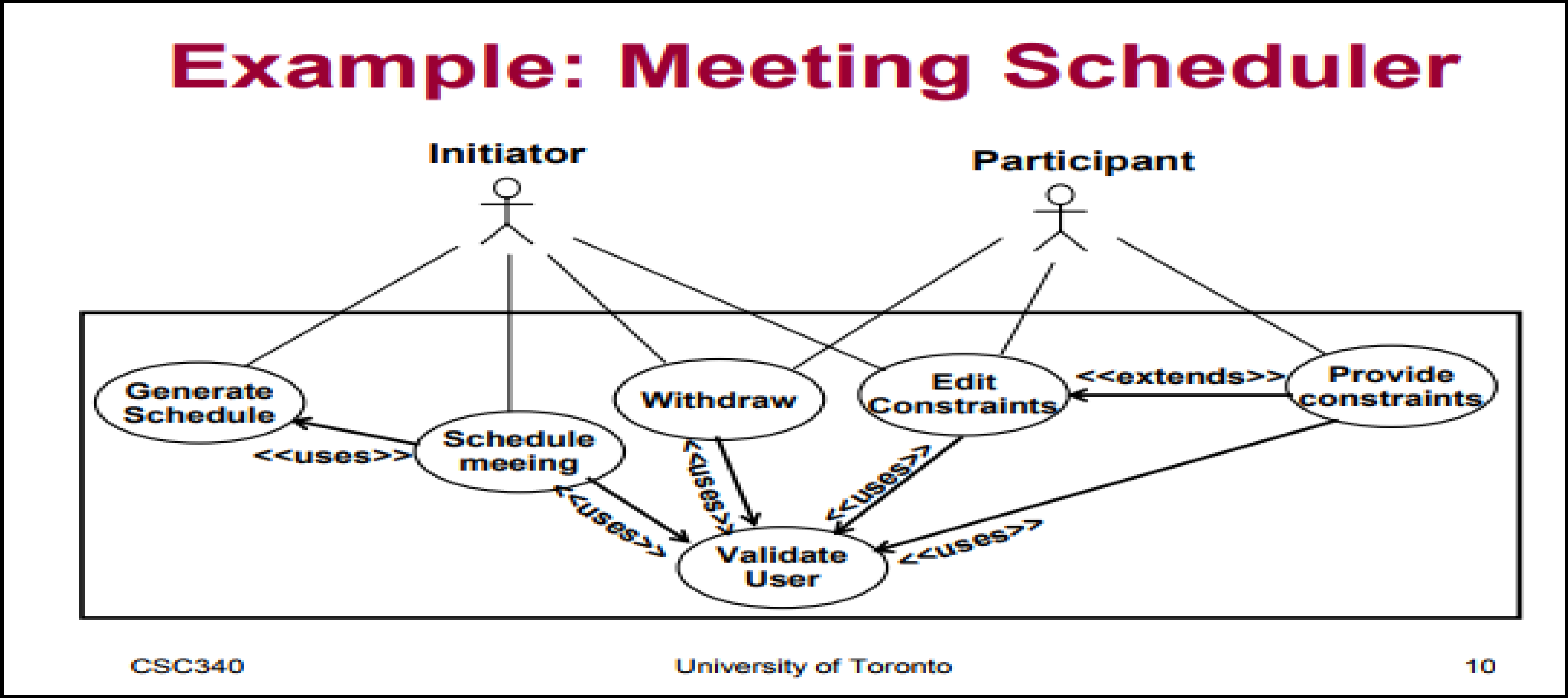
4. Identify the actors.

5. Add relationships.

<<uses>>

One use case invokes another (like a procedure call)

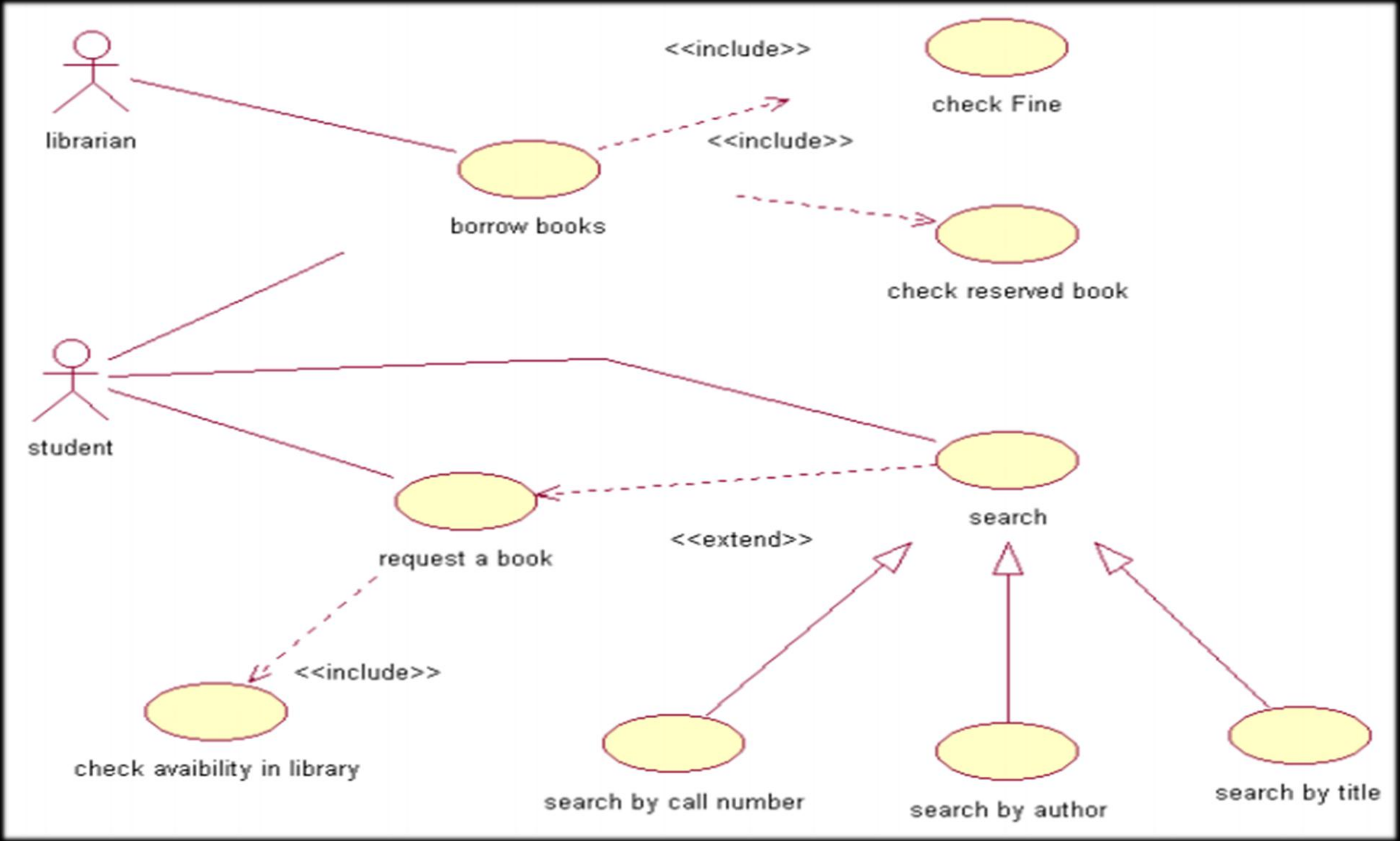
- used to avoid describing the same flow of events several times
- puts the common behaviour in a use case of its own.



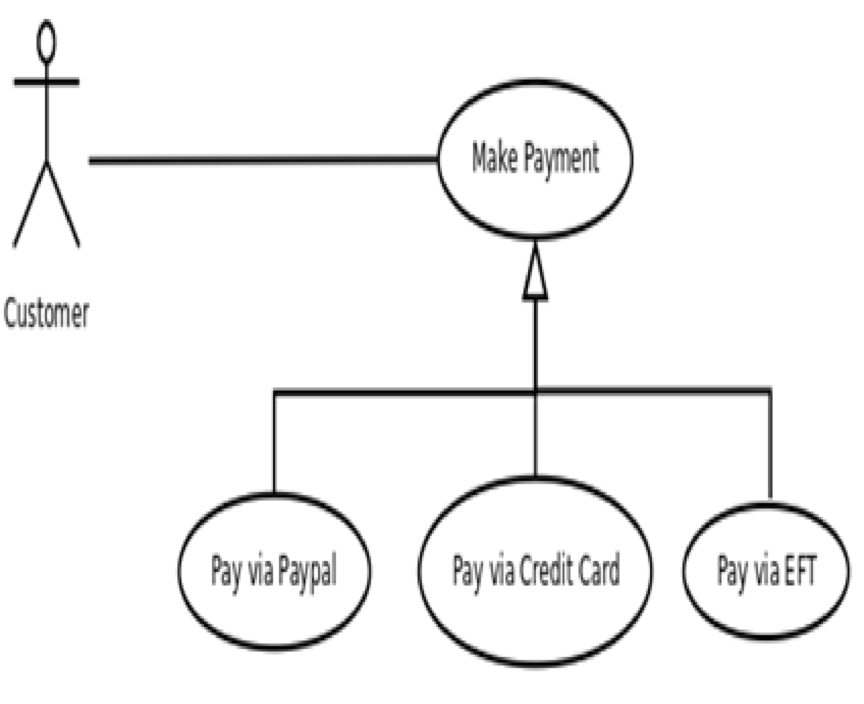
Example: Meeting Scheduler

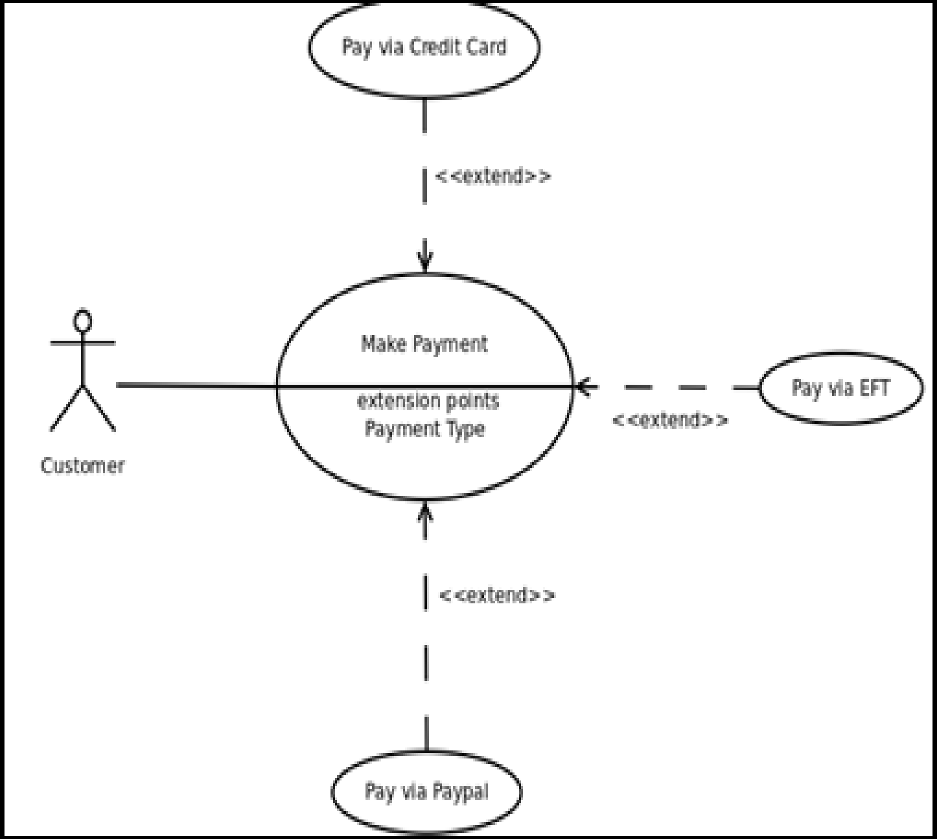
CSC340 University of Toronto 10

Difference between Generalization and Extend



Difference between Generalization and Extend





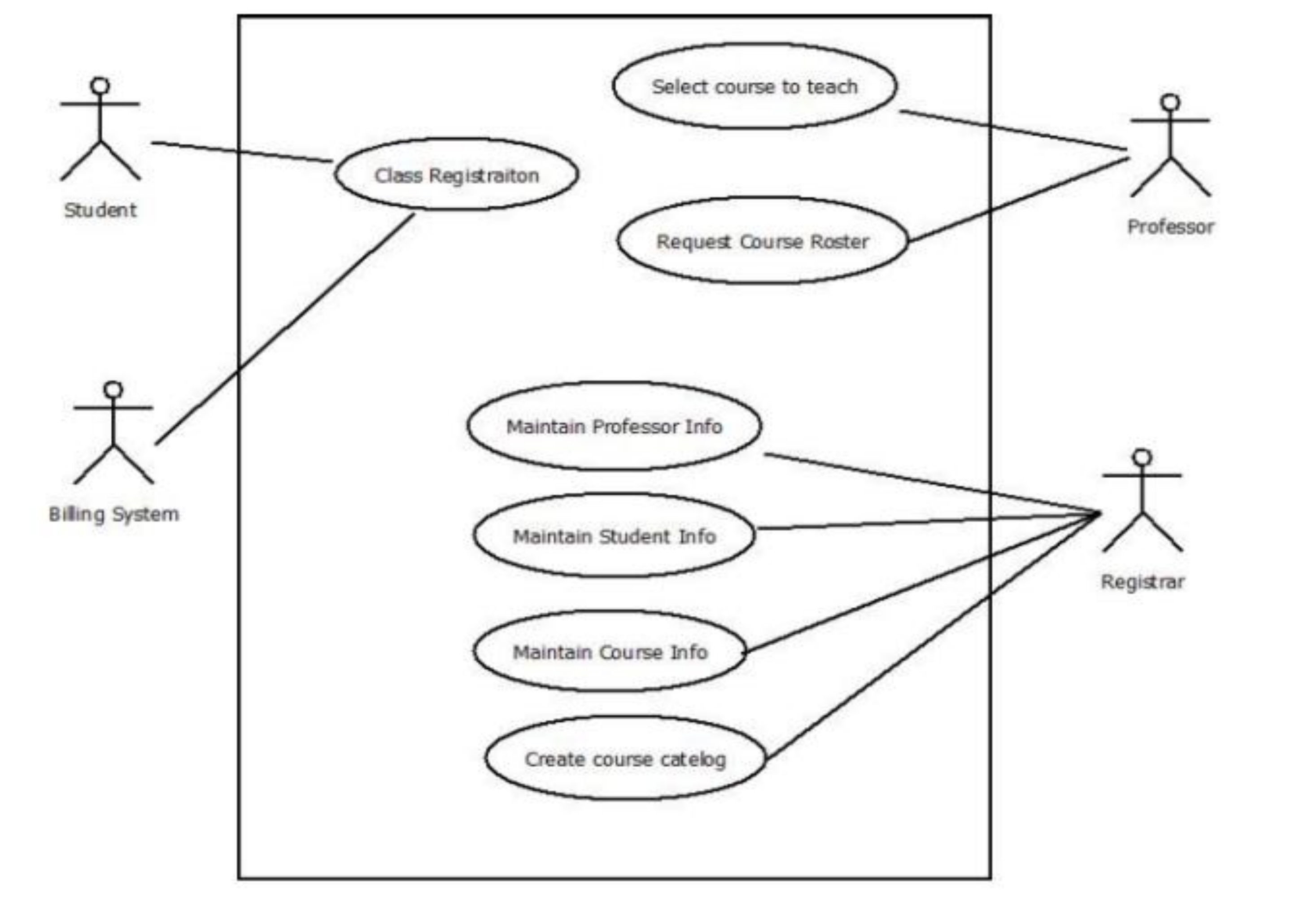
Examples

Example

- A user placing an order with a sales company might follow these steps :
- Browse catalog and select items.
- Call sales representative.
- Supply shipping information.
- Supply payment information.
- Receive conformation number from salesperson.

Altered State University (ASU) Registration System

- Professors indicate which courses they will teach on-line.
- A course catalog can be printed which is created by Registrar.
- Allow students to select on-line courses for upcoming semester.
- When the registration is completed, the system sends information to the billing system.
- Professors can obtain course rosters on-line.
- Students can add or drop classes on-line.
- Registrar maintains all the information about student, course and professor.


Student
Class Registraiton
Billing System
Select course to teach
Request Course Roster
Professor
Maintain Professor Info
Maintain Student Info
Maintain Course Info
Create course catelog
Registrar

That is all