Use Cases

- One of the primary challenges is the ability to elicit the correct and necessary system requirements from the stakeholders and specify them in a manner understandable to them so those requirements can be verified and validated.

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is a difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other work so cripples the resulting system if done wrong. No other part is more difficult to rectify later. Fred Brooks

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 behavior in a use case diagram; instead only depict only system functionality.
5. Use case diagram does not show sequence – unlike DFDs.

System Concepts for Use-Case Modeling

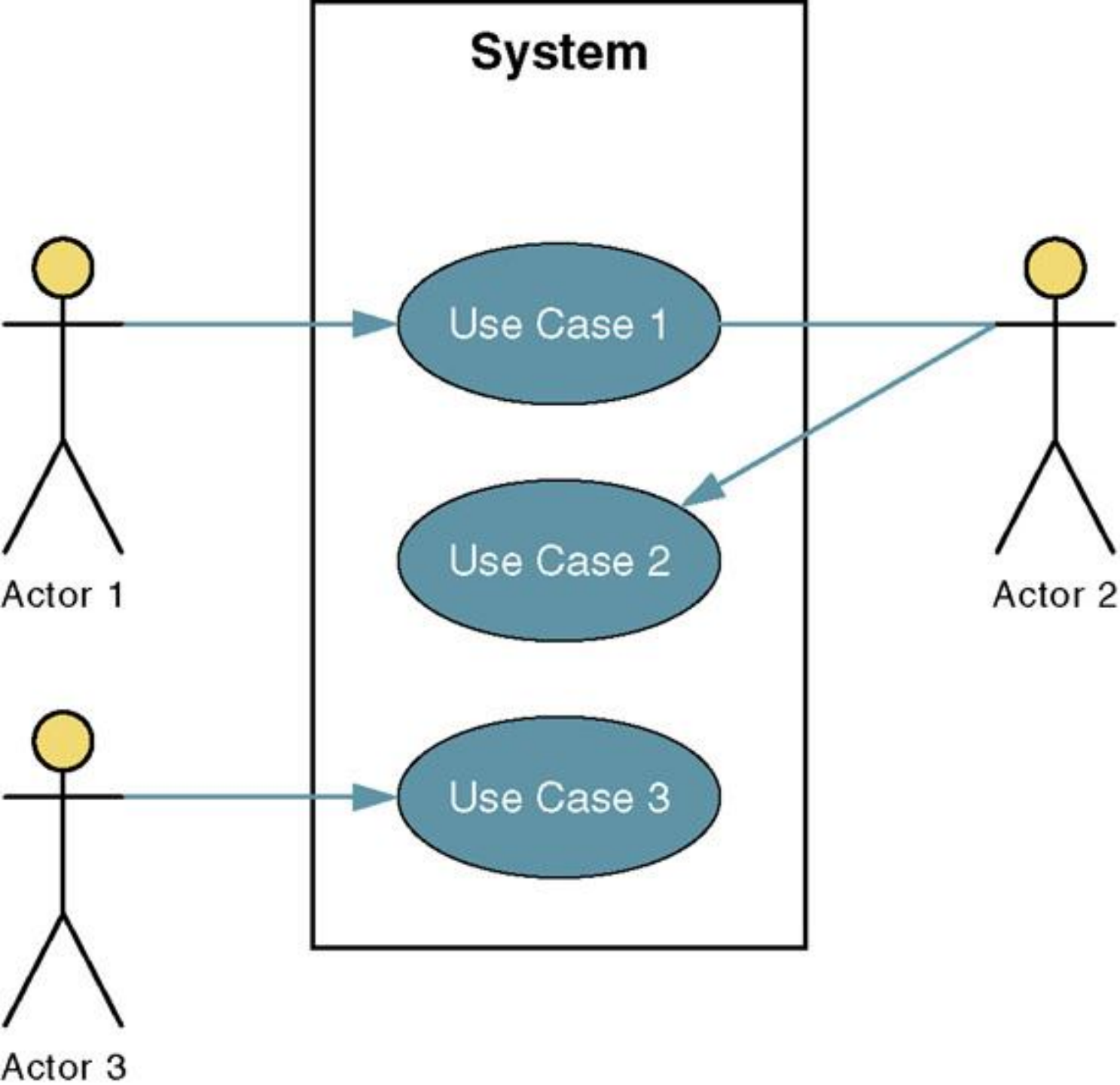
Use case – a behaviorally related sequence of steps (scenario), both automated and manual, for the purpose of completing a single task.

- Description of system functions from the perspective of external users in terminology they understand.

Use-case diagram – a diagram that depicts the interactions between the system and external systems and users.

- graphically describes who will use the system and in what ways the user expects to interact with the system.

Use-case narrative – a textual description of the event and how the user will interact with the system to accomplish the task.



Sample Use-Case Model Diagram

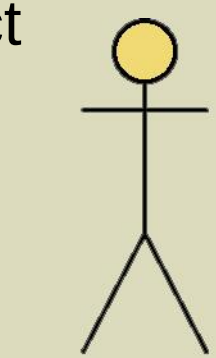
Basic Use-Case Symbols

Use case – subset of the overall system functionality

- Represented by a horizontal ellipse with name of use case above, below, or inside the ellipse.

Actor – anyone or anything that needs to interact with the system to exchange information.

- human, organization, another information system, external device, even time.



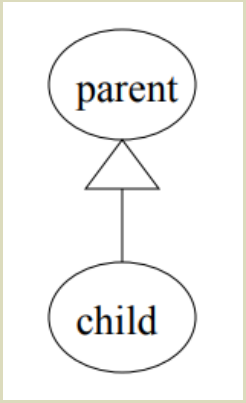
Use Case - Relationships and its Types

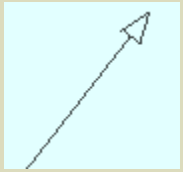
- Association relationship: Represent communication between actor and use case
- Often referred to as a communicate association
- use just a line to represent

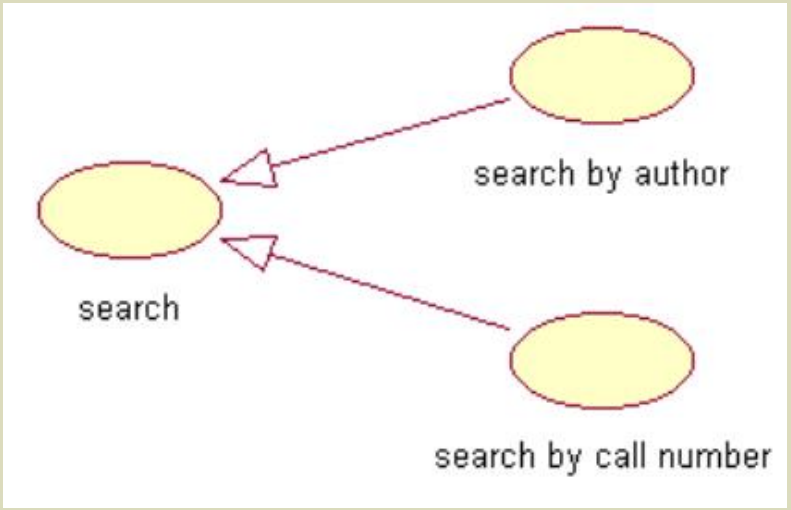
Use Case - Relationships and its Types

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:



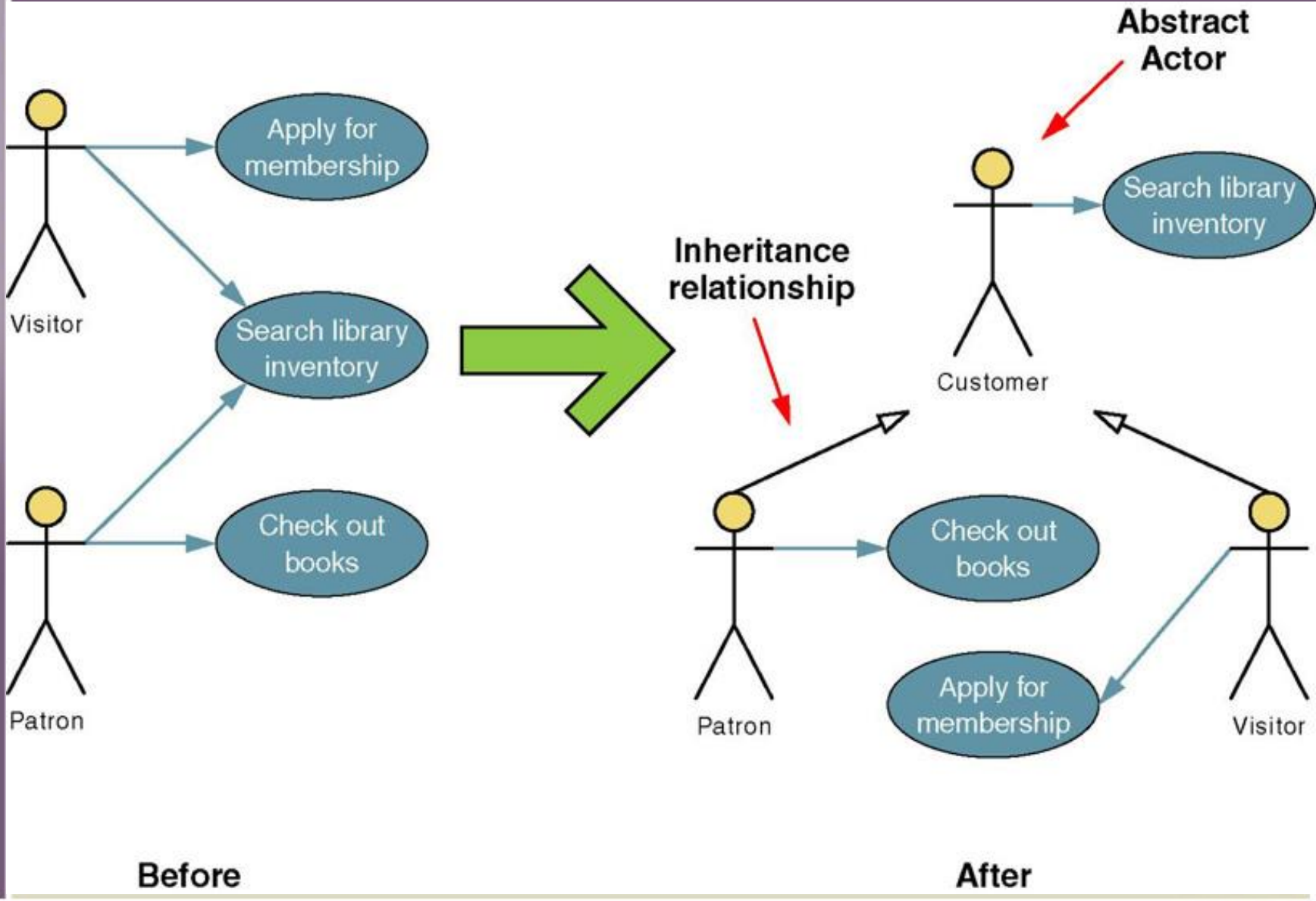




Use Case - Relationships and its Types

- Generalization relationship between actors
  - actor generalization refers to the relationship which can exist between two actors
- Generalization relationship between use cases
  - use case generalization refers to the relationship which can exist between two use cases

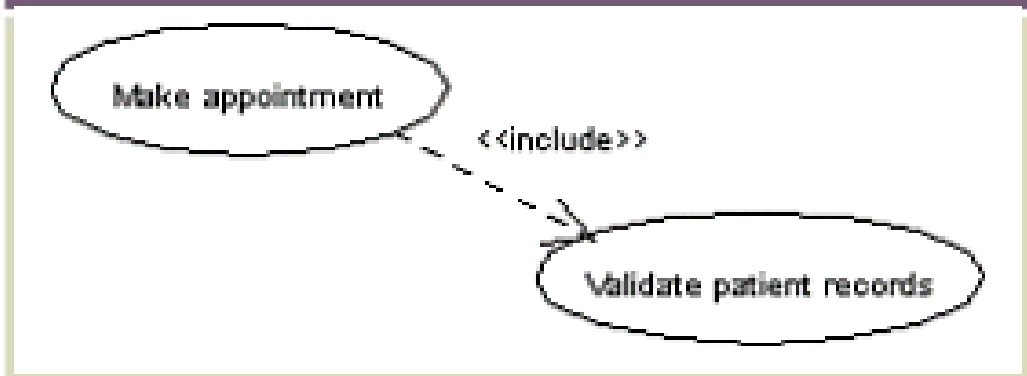
Use Case Inheritance Relationship



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 behavior from an included use case into a base use case in order to support re-use of common behavior.
  - Notation <<include>>

<<include>>

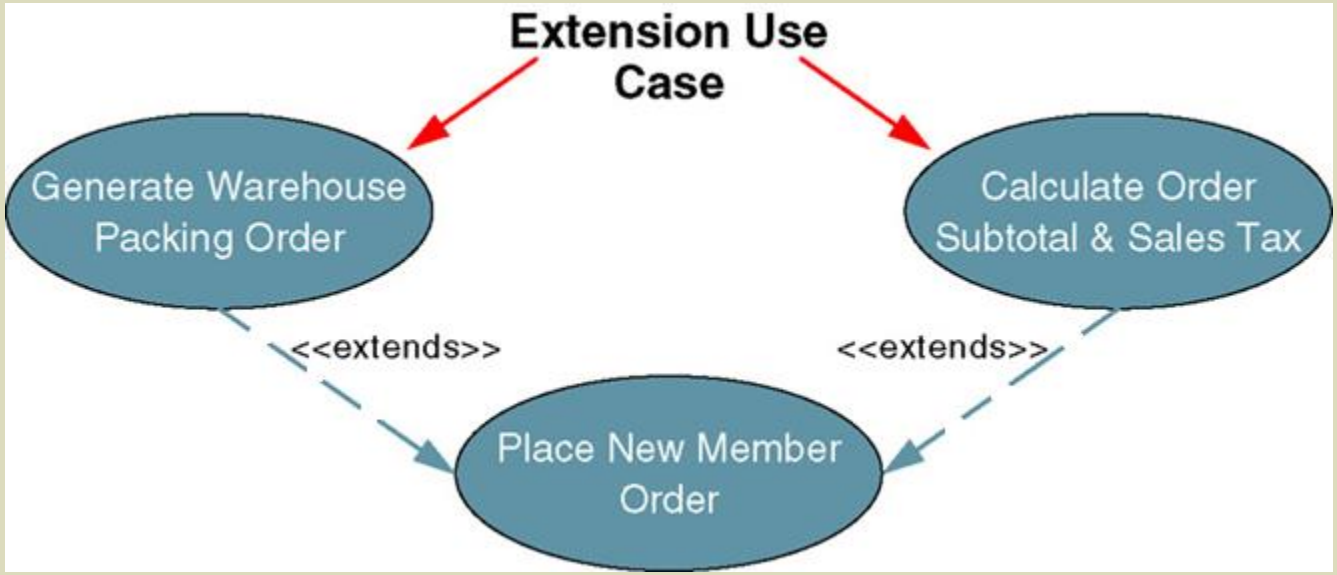


In Figure, the functionality defined by the "Validate patient records" use case is contained within the "Make appointment" use case.
Hence, whenever the "Make appointment" use case executes, the steps defined in the "Validate patient records" use case are also executed.

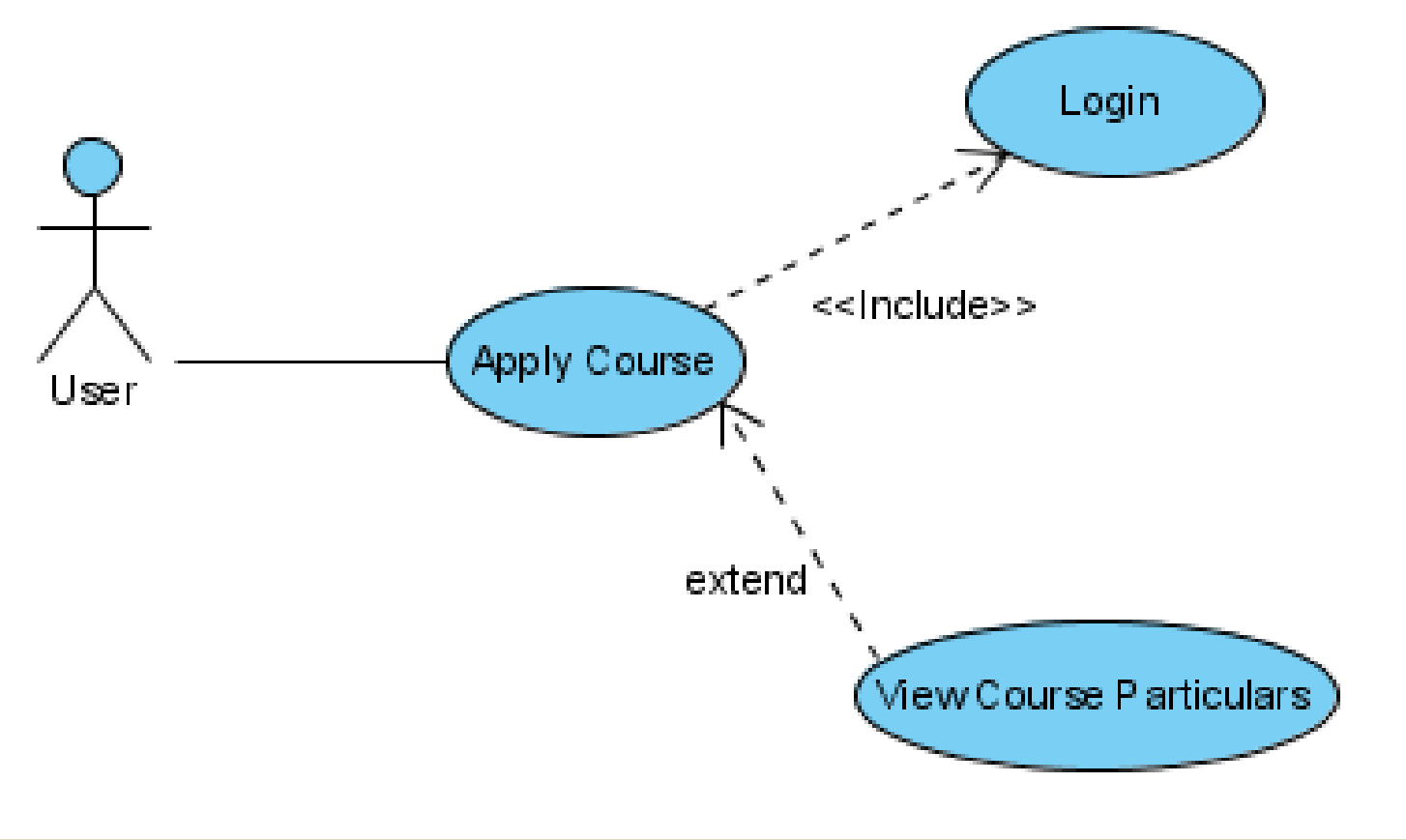
Use Case - Relationships and its Types

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

<<extend>>


Extension Use Case
Generate Warehouse Packing Order
Calculate Order Subtotal & Sales Tax
<<extends>>
<<extends>>
Place New Member Order

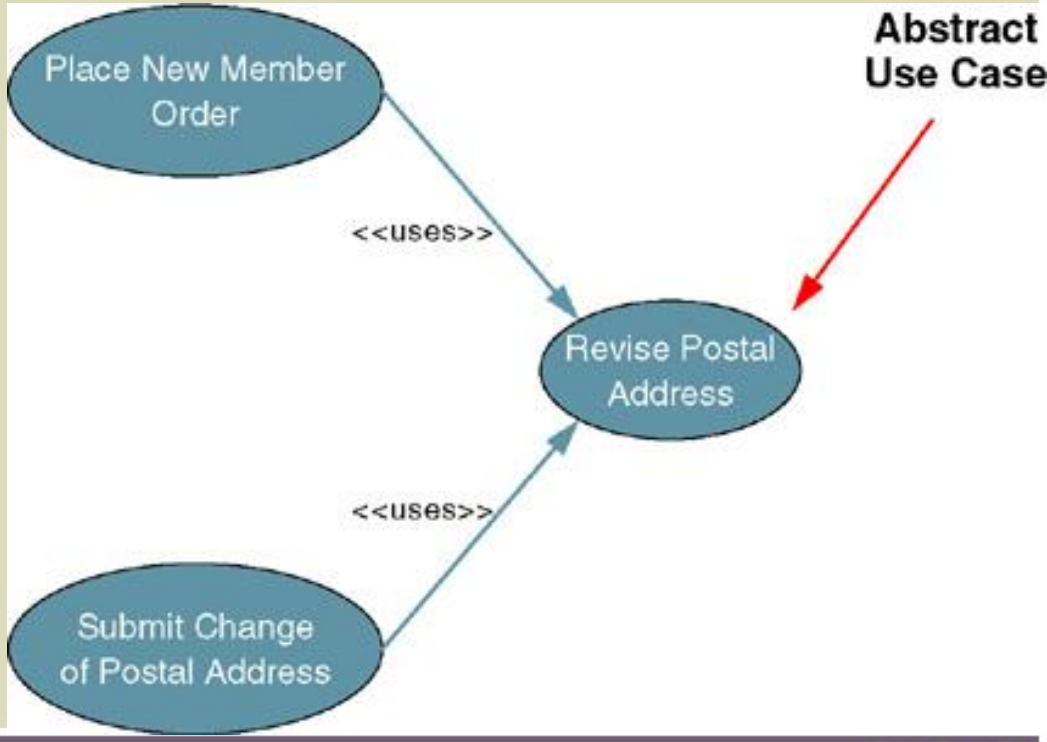
Example – Include and Extend



Use Case Uses Relationship

Abstract use case – use case that reduces redundancy in two or more other use cases by combining common steps found in both.

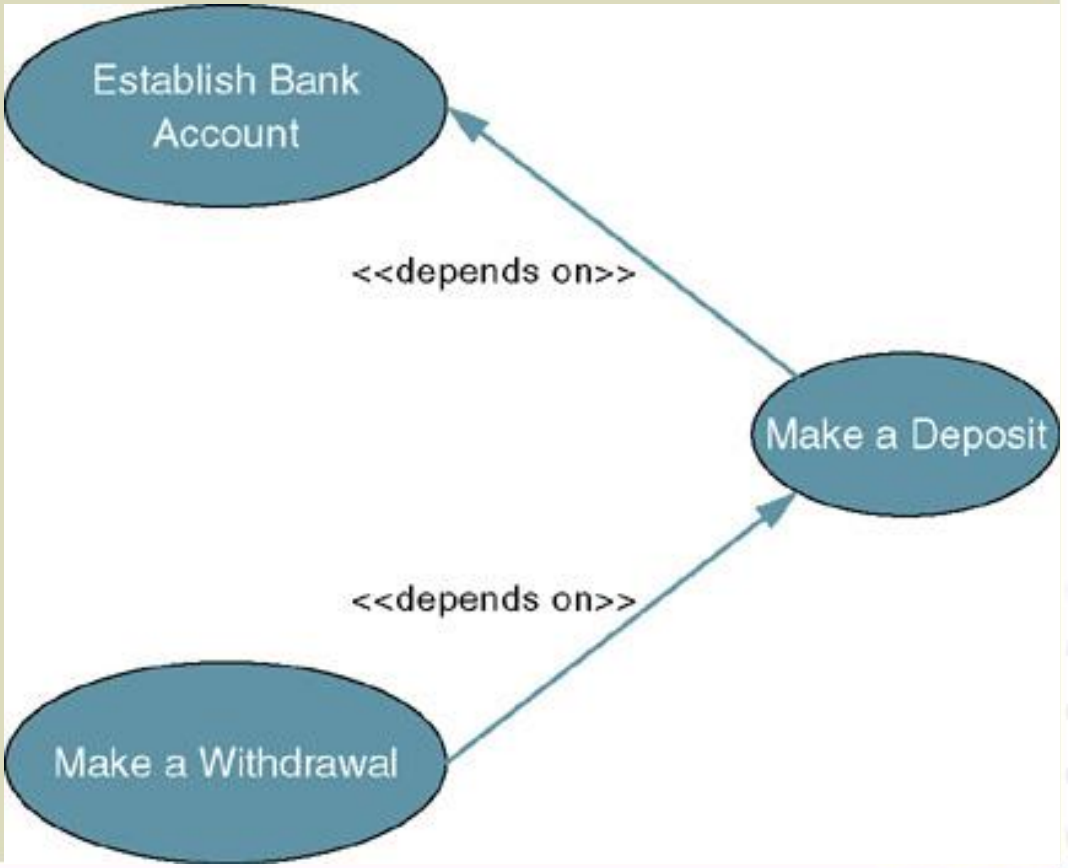
- Available by any other use case that requires its functionality.
- Generally not identified in requirements phase
- Relationship between abstract use case and use case that uses it is called a *uses* (or *includes*) relationship.
- Depicted as arrow beginning at original use case and pointing to use case it is using.
- Labeled <<uses>>.



Use Case Depends On Relationship

Depends On – use case relationship that specifies which other use cases must be performed before the current use case.

- Can help determine sequence in which use cases need to be developed.
- Depicted as arrow beginning at one use case and pointing to use case it depends on.
- Labeled <<depends on>>.



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.

Questions for Identifying People Actors

- Who is interested in the scenario/system?
- Where in the organization is the scenario/system be used?
- Who will benefit from the use of the scenario/system?
- Who will supply the scenario/system with this information, use this information, and remove this information?
- Does one person play several different roles?
- Do several people play the same role?

Questions for Identifying Other Actors

- What other entity is interested in the scenario/system?
- What other entity will supply the scenario/system with this information, use this information, and remove this information?
- Does the system use an external resource?
- Does the system interact with a legacy system?

Courseware Management System

- Course administrators manages topics and courses that make up a course, additionally he also can view courses.
- Course administrators manages Tutor who teach courses
- Course administrators who mange the assignment of the courses to tutors
- Calendar or Course Schedule is generated as a result of the Students who refer to the Course schedule or Calendar to decide which courses for which they wish to take.

User Placing an Order

- A customer placing an order with a sales company might follow these steps :

1. Browse catalog and select items.
2. Call sales representative.
3. Supply shipping information.
4. Supply payment information.
5. Receive conformation number from salesperson.

Vending Machine

1. A customer buys a product
2. The supplier restocks the machine
3. The supplier collects money from the machine

Altered State University (ASU) Registration System

1. Professors indicate which courses they will teach on-line.
2. A registrar create and print the course catalog.
3. Allow students to select on-line four courses for upcoming semester.
4. No course may have more than 10 students or less than 3 students.
5. When the registration is completed, the system sends information to the billing system.
6. Professors can obtain course rosters on-line.
7. Students can add or drop classes on-line.
8. It is the duty of registrar to maintain the course, professor and student information on-line.

Vehicles Sale System.

- Draw a use case diagram for the vehicle sales system. Customer makes offer for the vehicle. Customer can be new customer or old customer. New and old customer can make their own offers. For every individual they have to get registered. System can update the existing customer information as well. Customer make payment if his/ her offer is accepted. Management has right to accept or reject the offer by managing the offer. Sales person records the sales contract of the accepted offer.

Home Work
Hospital Management System

- Draw a use case diagram for the hospital reception system. In this system, receptionist can schedule patient appointment and patient hospital admission after the patient registration. Both types of patients i.e. outpatient and inpatient can be admitted in the hospital. Receptionist also checks the insurance and claim forms and put them in file. Patient medical report is also filed by the receptionist.