Use Case Diagram

Use Case Diagrams

❖ Use case diagrams are used to **visualize, specify, construct, and document the (intended) behavior of the system**, during requirements capture and analysis.

❖ Provide a way for developers, domain experts and end-users to Communicate.

❖ Serve as basis for testing.

❖ Use case diagrams contain use cases, actors, and their relationships.

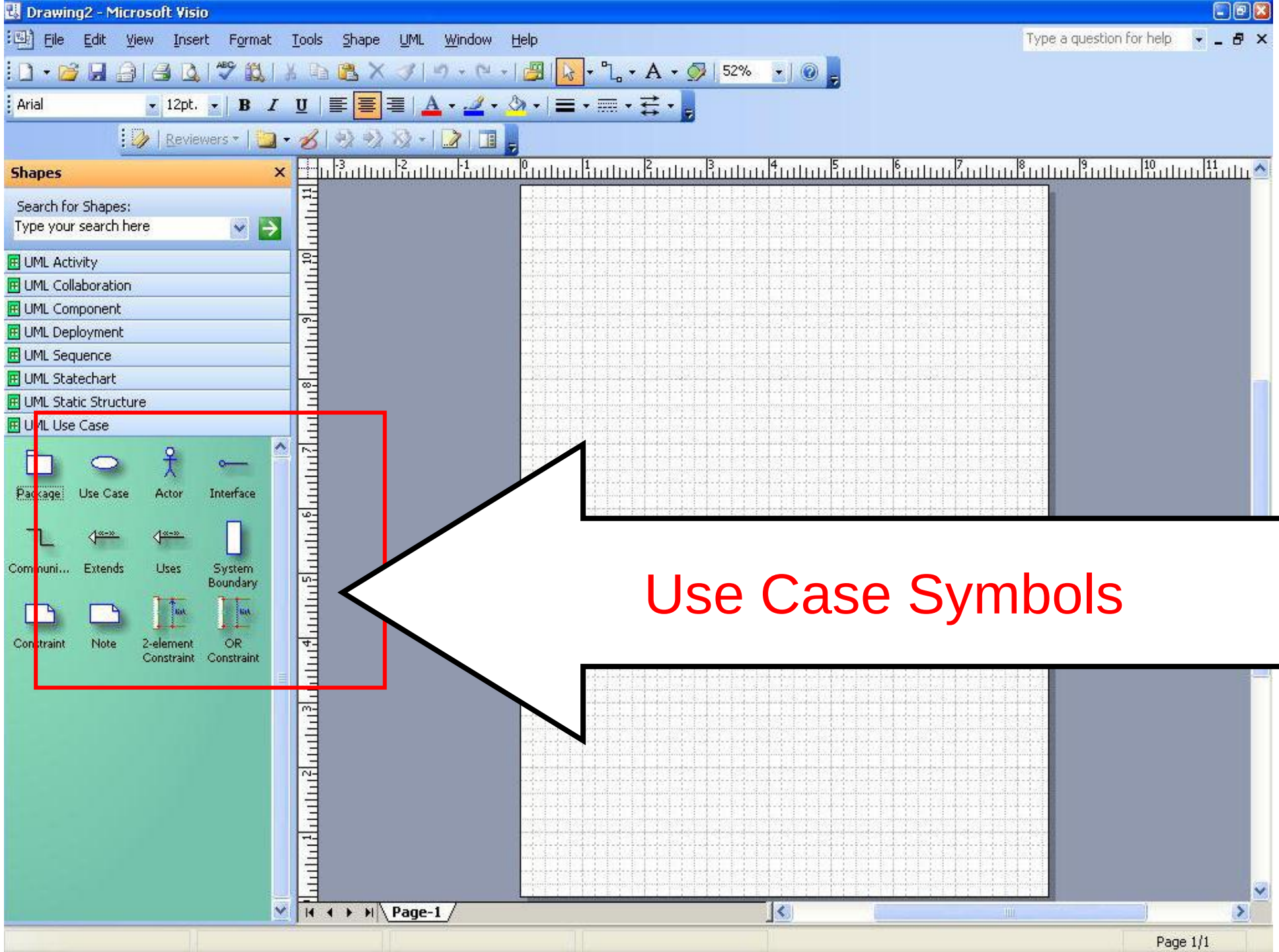
Use Case Diagrams

Describes the **functional behavior** of the system from the user's point of view.

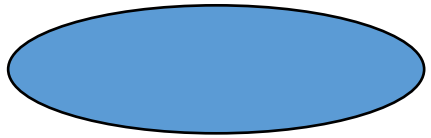
Actor: User, external system, or a physical environment.

Use case: a class of functionality provided by the system as an event flow.

Has entry and exit conditions, flow of events, and participating actors.

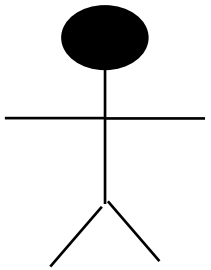

Drawing2 - Microsoft Visio
File Edit View Insert Format Tools Shape UML Window Help
Type a question for help
Arial 12pt.
Reviewers
Shapes
Search for Shapes:
Type your search here
UML Activity
UML Collaboration
UML Component
UML Deployment
UML Sequence
UML Statechart
UML Static Structure
UML Use Case
Package
Use Case
Actor
Interface
Communi...
Extends
Uses
System Boundary
Constraint
Note
2-element Constraint
OR Constraint
Page-1
Page 1/1
Use Case Symbols

Use case diagram symbols



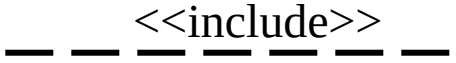
Use Case

Boundary



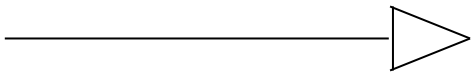
Actor

Connection



Include relationship

Extend relationship



Generalization

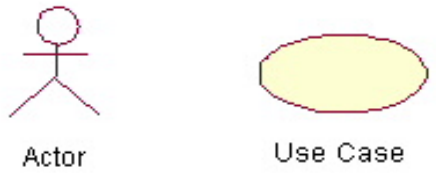
Use Case Diagram

- Used for describing a set of user **scenarios**
- Mainly used for capturing user requirements
- Work like a **contract** between end user and software developers

Use Case Diagram(core relationship)

Actors: A role that a user plays with respect to the system, including human users and other systems. an external system that needs some information from the current system.

Use case: A set of scenarios that describing an interaction between a user and a system, including alternatives.

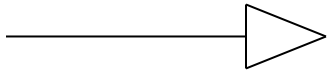


System boundary: Rectangle diagram representing the boundary between the actors and the system.

Use Case Diagram (core relationship)

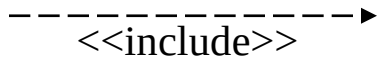
Association: Communication between an actor and a use case; Represented by a solid line.

Generalization: Relationship between one general use case and a special use case (used for defining special alternatives)
Represented by a line with a triangular arrow head toward the parent use case.



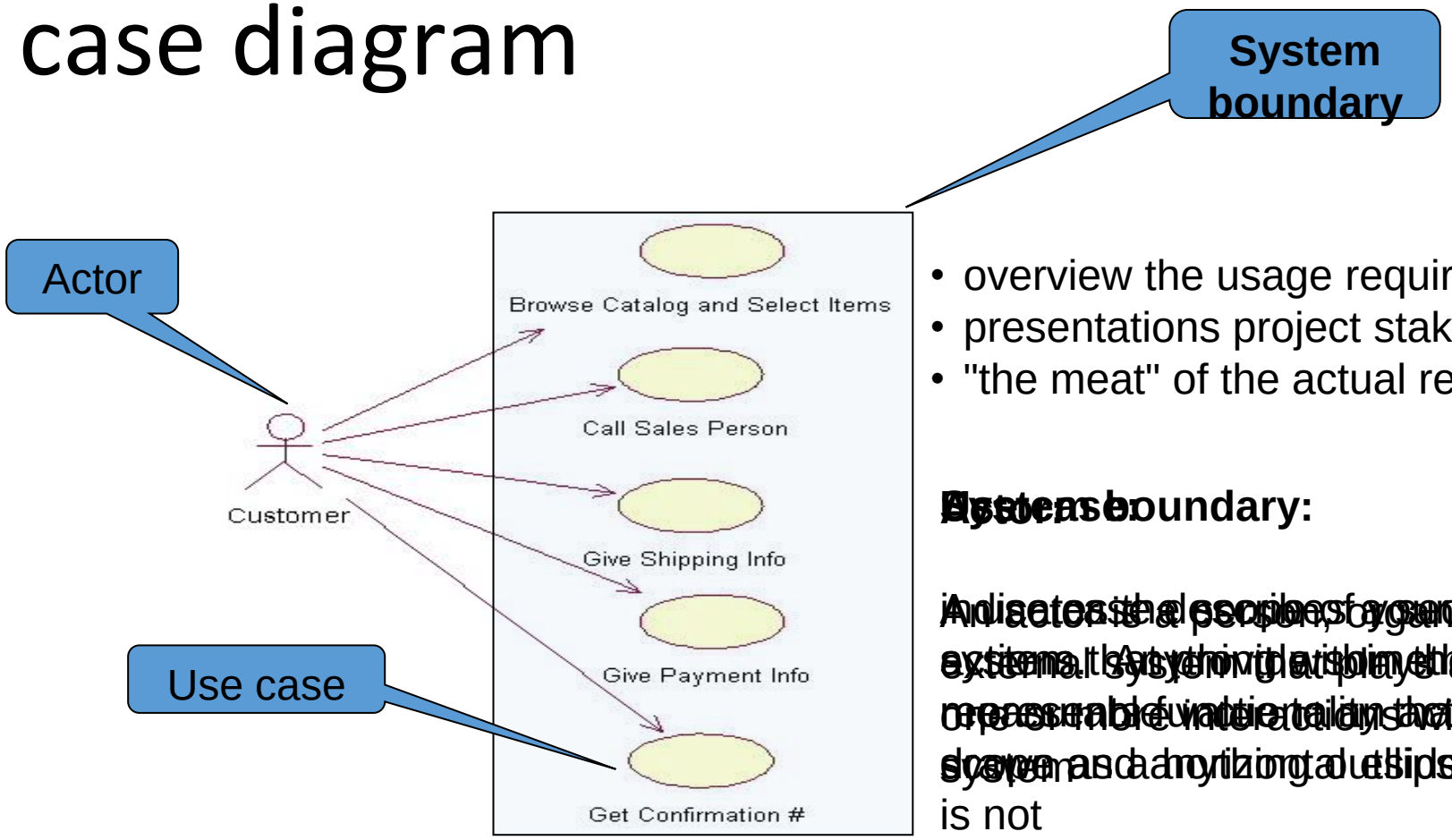
Use Case Diagram (core relationship)

Include: a dotted line labeled <<include>> beginning at base use case and ending with an arrows pointing to the include use case. The include relationship occurs when a chunk of behavior is similar across more than one use case. Use "include" in stead of copying the description of that behavior.



Extend: a dotted line labeled <<extend>> with an arrow toward the base case. The extending use case may add behavior to the base use case. The base class declares "extension points".

Use case diagram



- overview the usage requirements
- presentations project stakeholders
- "the meat" of the actual requirements

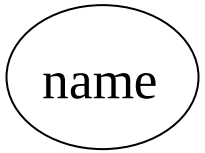
System boundary:

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

Indicates the scope of your use case and what is system and what is outside the box

Online C2C shopping

Use Case



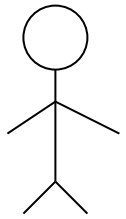
- A use-case is a set of sequences of actions a system performs that yield an observable result of value to a particular actor.
- A use-case describes a requirement for the system, that is, what it should do, but not how it should do it.
- A use-case is a set of scenarios tied together by a common user goal.

Actors

❖ An actor is someone or something that interacts with the system. It is who or what uses the system.

❖ An actor communicates with the system by sending and receiving messages.

❖ An actor is a role that a user plays with respect to the system.

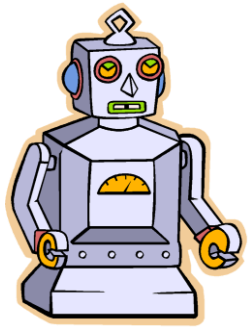


name

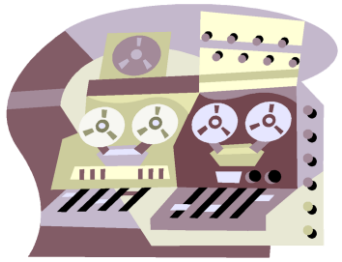
Finding Actors

- External objects that produce/consume data:
  - Must serve as sources and destinations for data
  - Must be external to the system

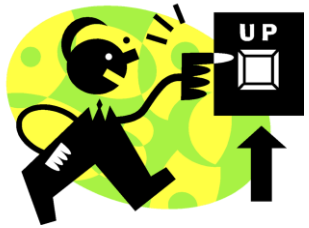
Humans



Machines

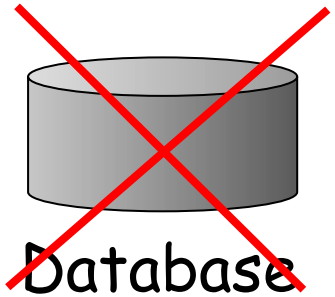


External systems



Sensors

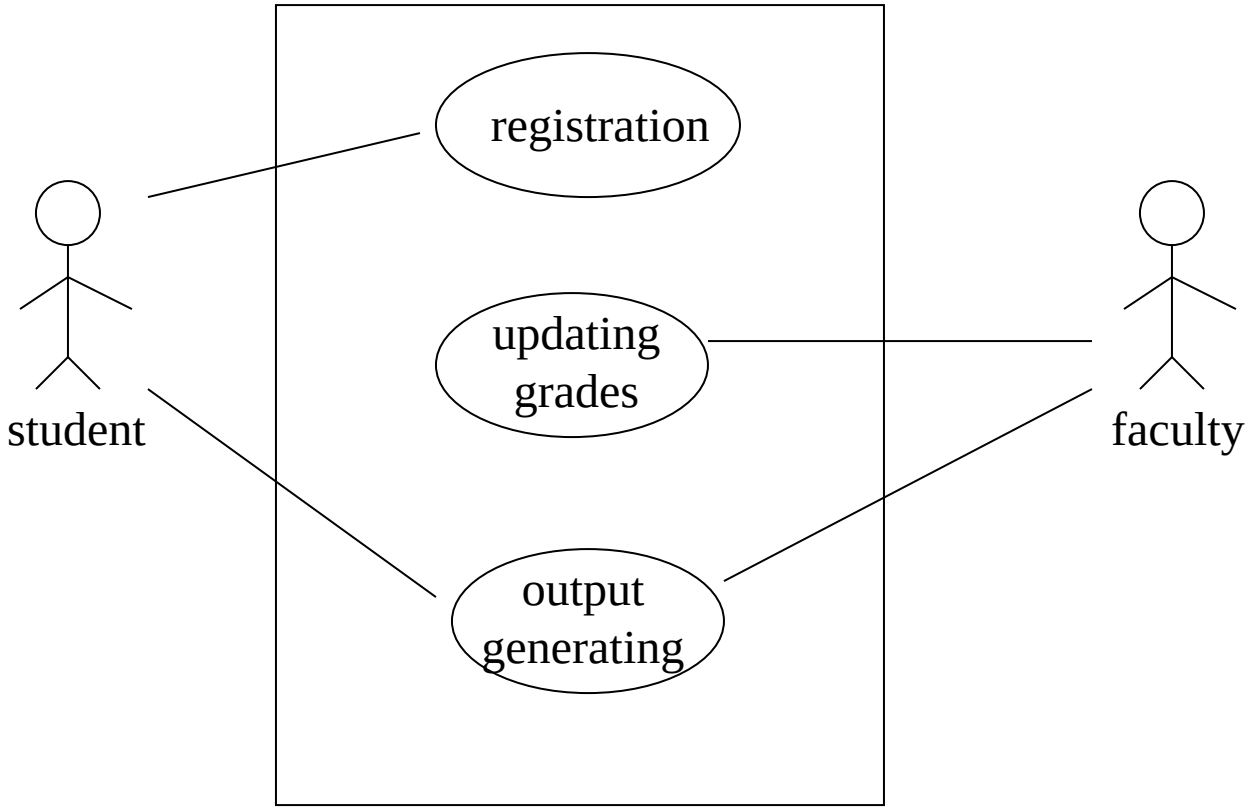
Organizational Units



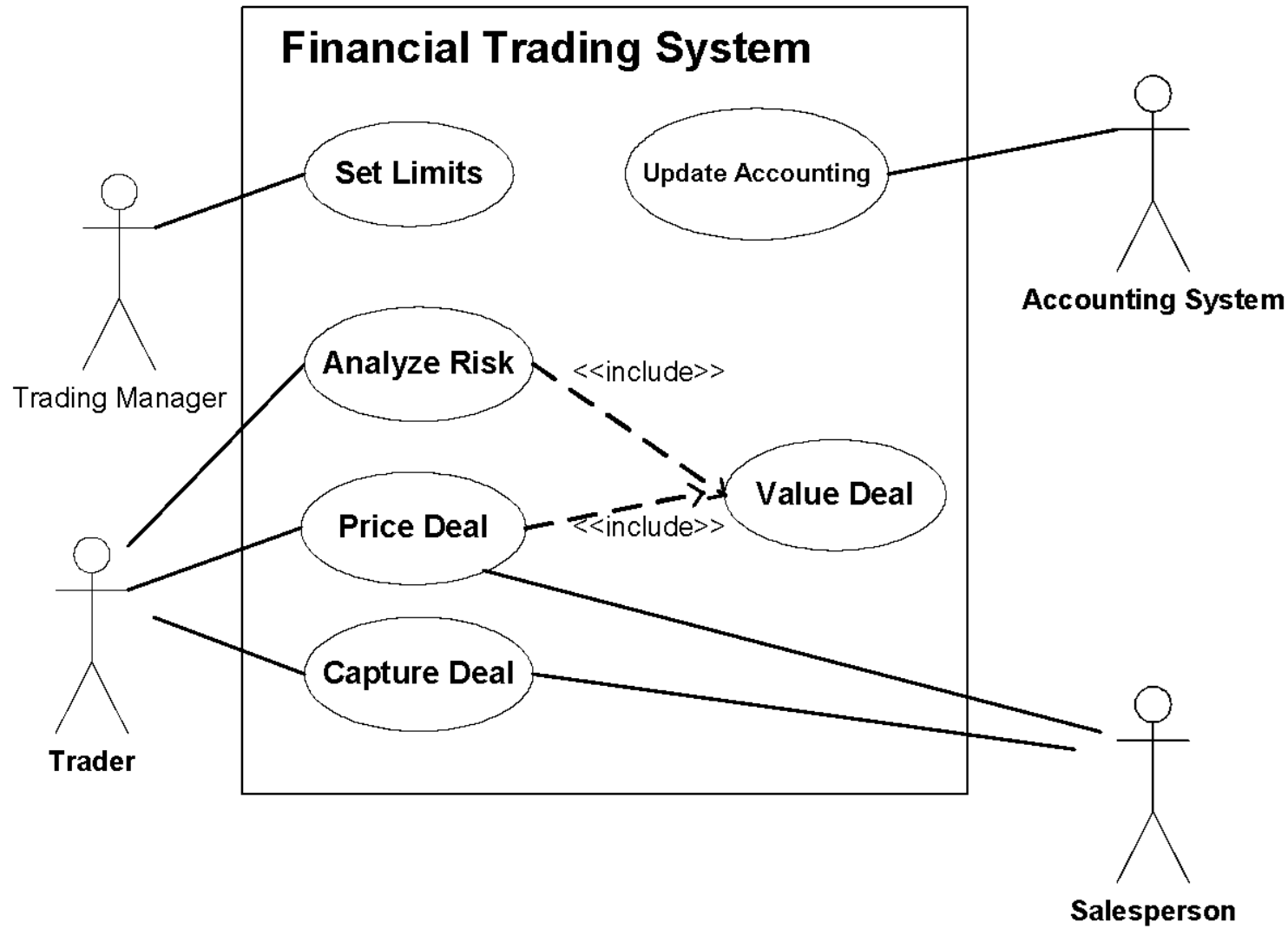
Database

Printer

Example of Use Case Diagram



Example of Use Case Diagram



Relationships between Use Cases

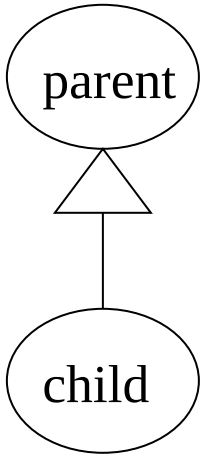
1. Generalization - use cases that are specialized versions of other use cases.

2. Include - use cases that are included as parts of other use cases. Enable to factor common behavior.

3. Extend - use cases that extend the behavior of other core use cases. Enable to factor variants.

1. Generalization

The child use case inherits the behavior and meaning of the parent use case.



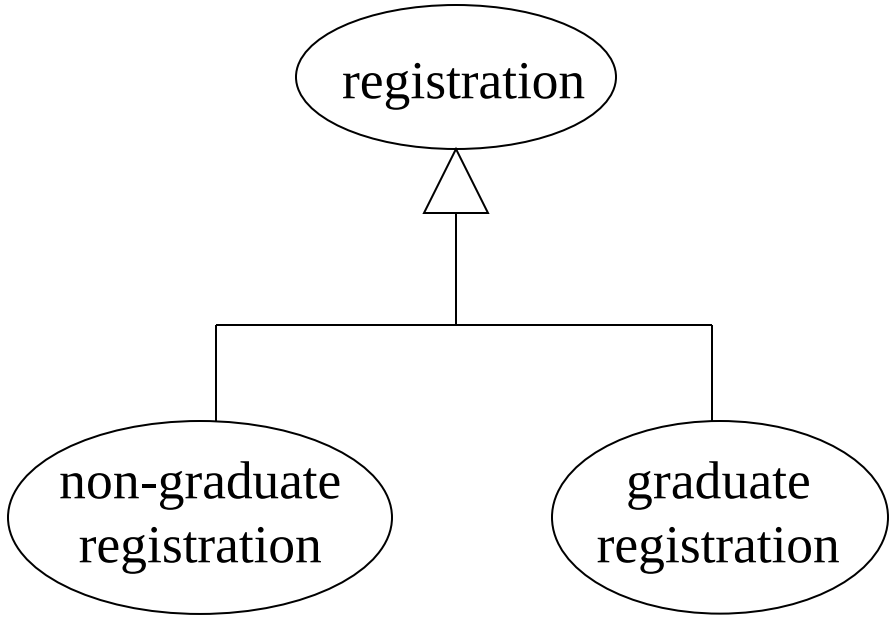
The child may add to or override the behavior of its parent.

Generalization is used when there is one use case similar to another.

Inheriting parent behavior, adding and overriding with the child's behavior.

Sub use case inherits behavior and semantics from super use cases.

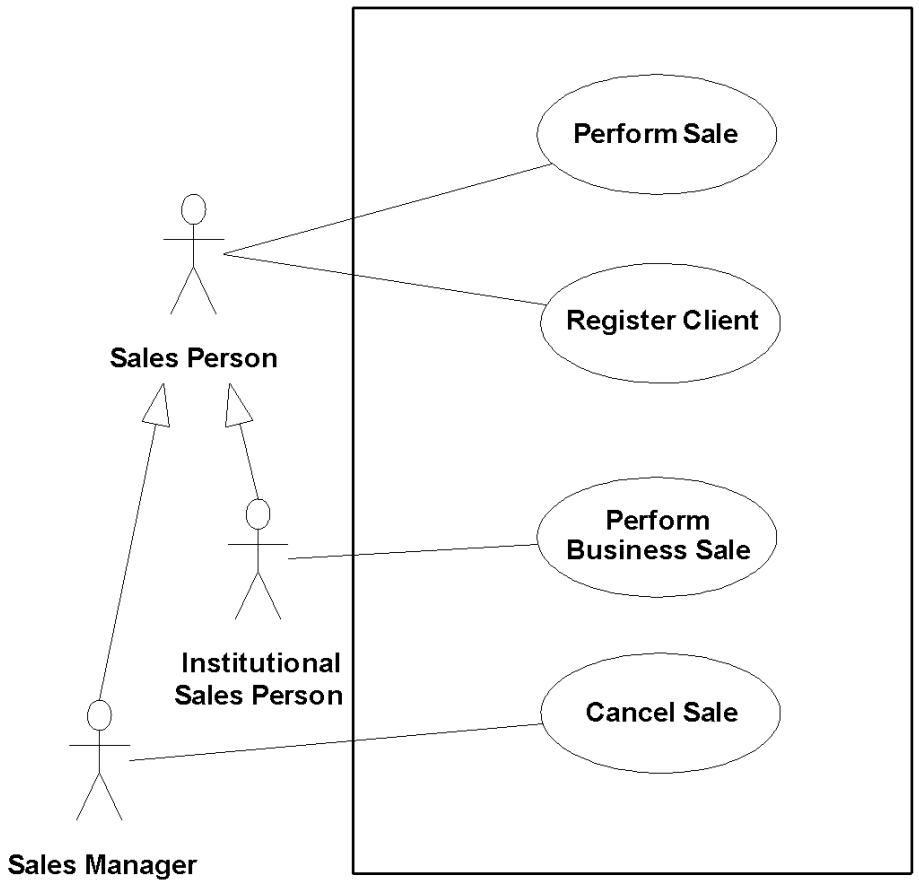
More about Generalization



Actors can be generalized

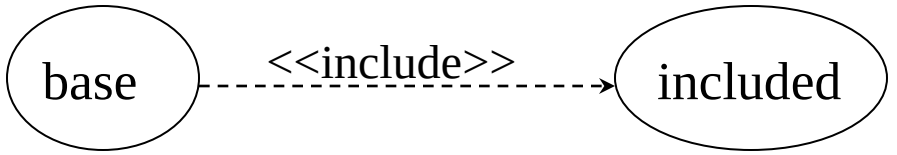
The child actor inherits all use-cases associations

Should be used if (**and only if**), the specific actor has more responsibility than the generalized one (i.e., associated with more use-cases)



Example

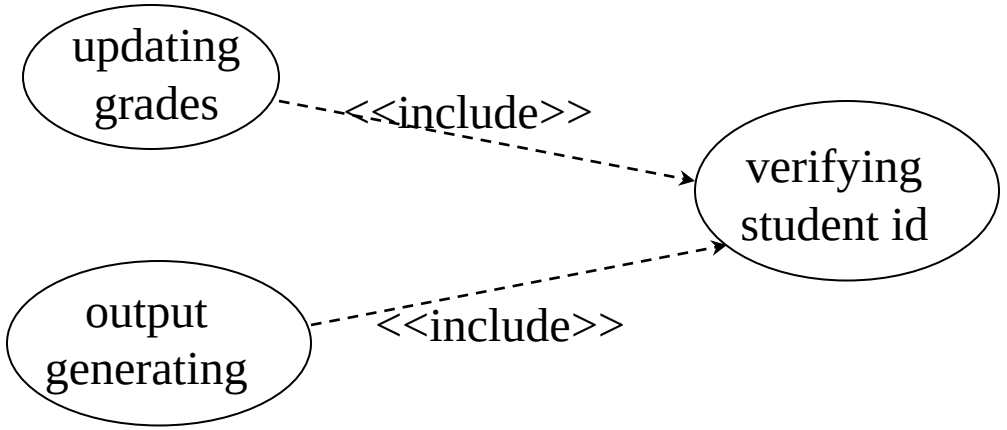
2. Include



- The base use case explicitly incorporates the behavior of another use case at a location specified in the base.
- The included use case never stands alone. It only occurs as a part of some larger base that includes it.

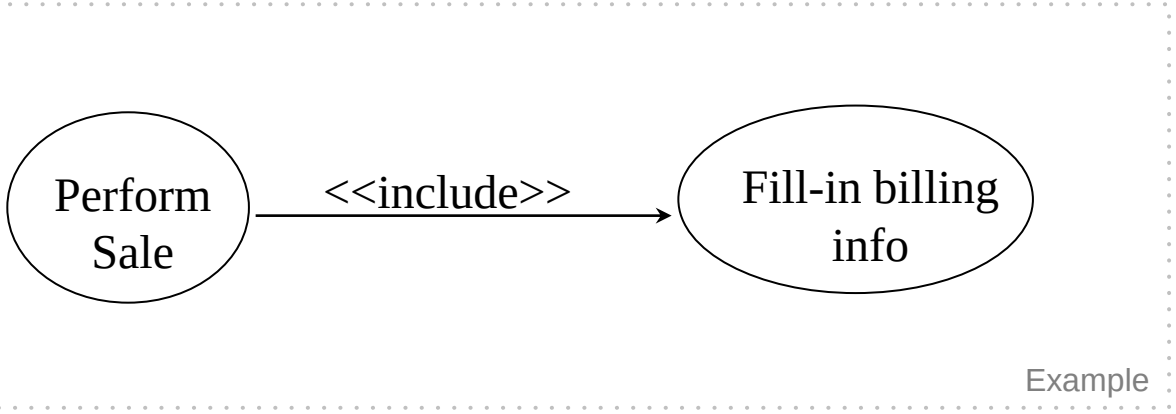
More about Include

- Enables to avoid describing the same flow of events several times by putting the common behavior in a use case of its own.

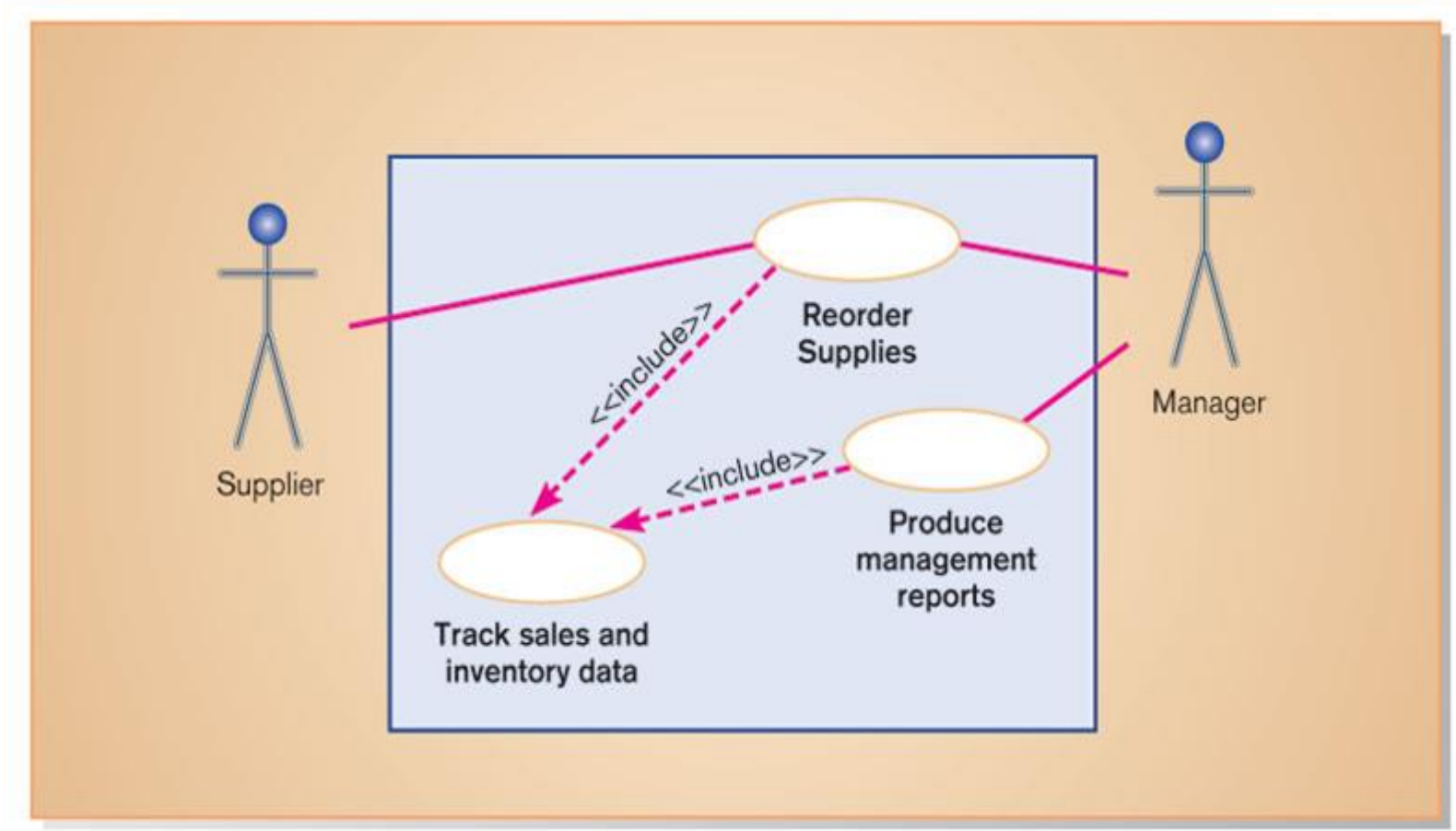


The “Include” Construct

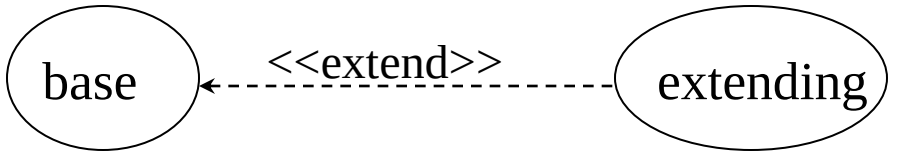
- Include is used when:
  - Decomposing complicated behavior
  - Centralizing common behavior
- The base use case explicitly incorporates the behavior of another use case at a location specified in the base.



More about Include



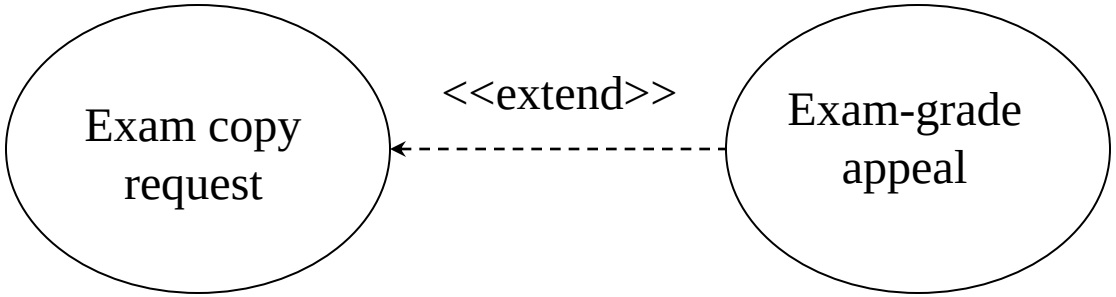
3. Extend



- The base use case implicitly incorporates the behavior of another use case at certain points called extension points.
- The base use case may stand alone, but under certain conditions its behavior may be extended by the behavior of another use case.

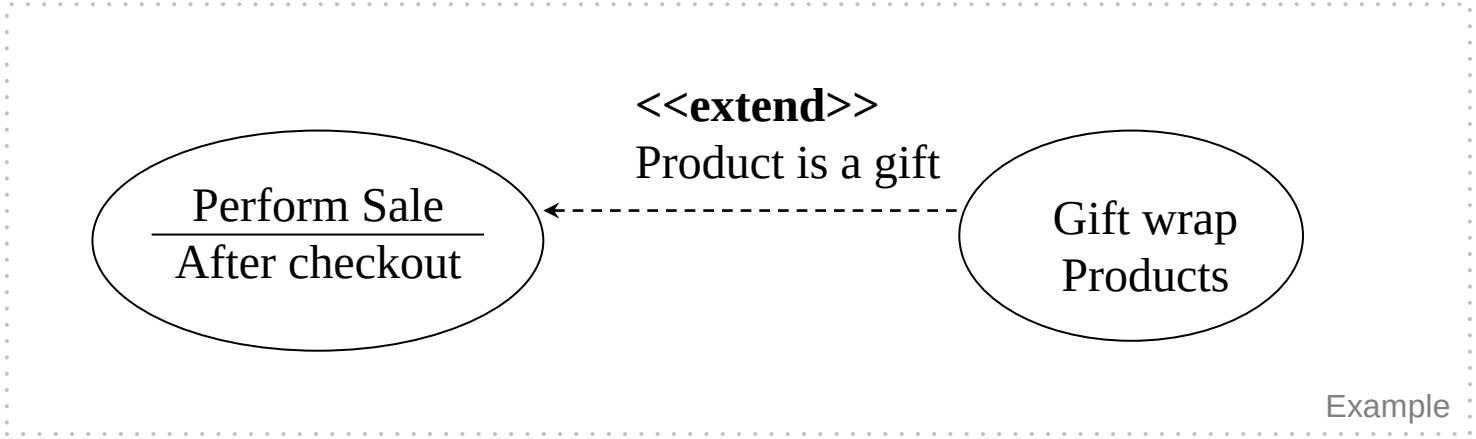
More about Extend

- Enables to model optional behavior or branching under conditions.



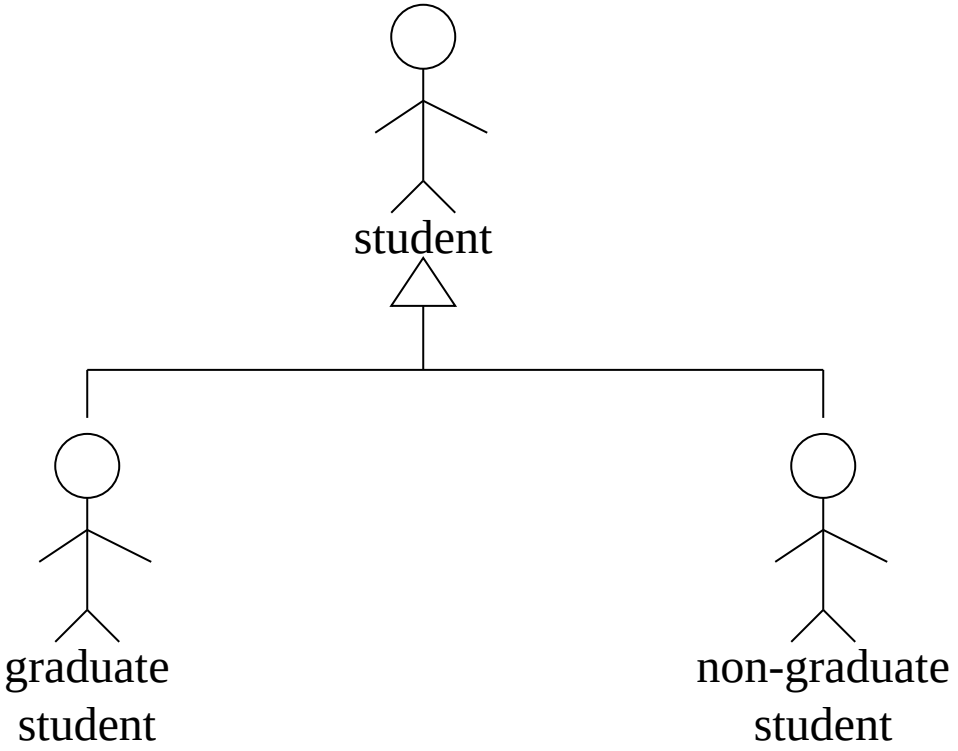
Extend – Graphical Representation

- The base use case can incorporate another use case at certain points, called extension points.
- Note the direction of the arrow
  - The base use-case does not know which use-case extends it

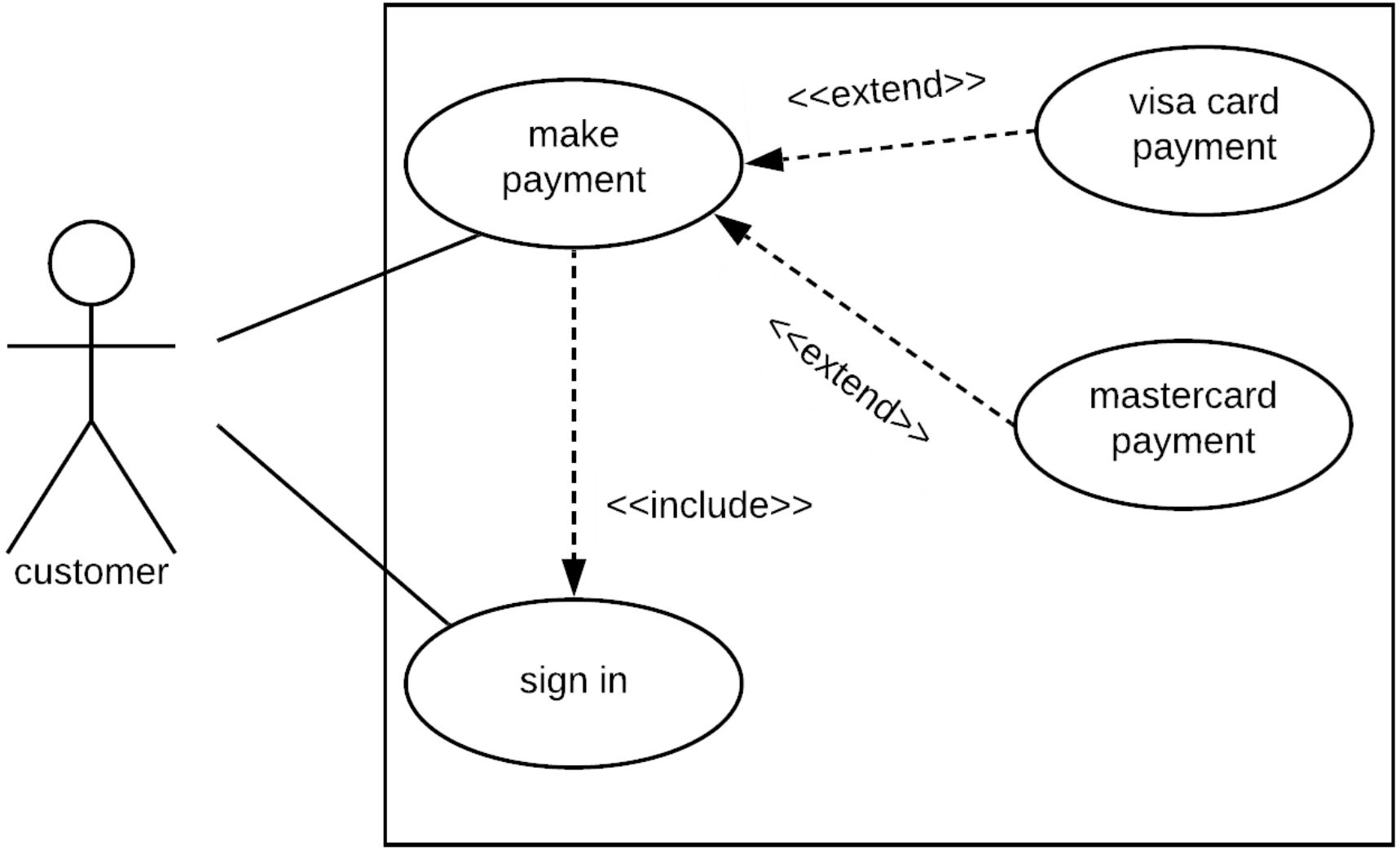


Relationships between Actors

- Generalization.



Use Case Diagram for Online Payment System



Relationships between Use Cases and Actors

- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

Use Case Diagram for Clinical Appointment