

Subject:

# Object-oriented Software design

## Chapter 2: User requirements




**Lê Văn Vinh, PhD**

Department of Software Engineering

Faculty of Information Technology

HCMC University of Technology and Education




# Contents

-  **I. Requirements**
-  **II. Use case diagrams**
-  **III. Scenarios**

# Requirement

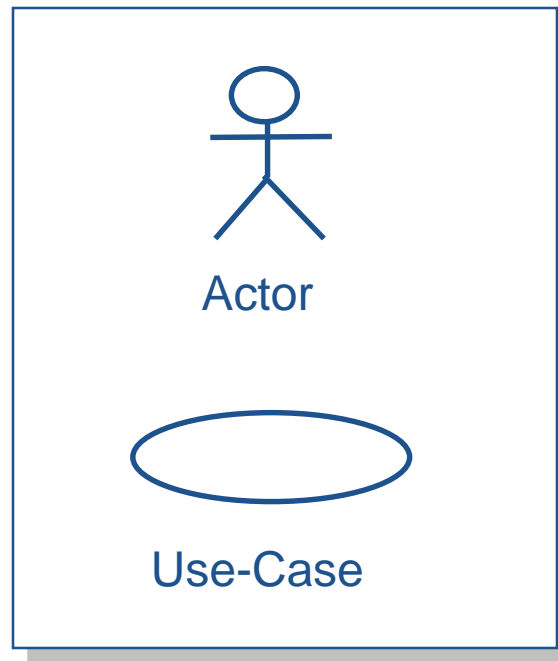
- A requirement is a **feature** that the system must have or a **constraint** that it must satisfy to be accepted by the client

# Contents

-  **I. Requirements**
-  **II. Use case diagrams**
-  **III. Scenarios**

# Use Case Diagrams

- Use case diagrams describe relationships between users and use cases
- Use case diagrams consist of two components:



# Use case Diagrams

## ❖ The benefits of Use case diagrams

- Is used to **communicate with** the end users and domain experts
- Is used to identify
  - Who will **interact** with the system and what the system should do
  - What **interfaces** the system should have
- Is used to verify
  - All requirements are captured
  - That the developers have **understood** the requirements

# Use case Diagrams

## ❖ **Activities**

- Identifying actors
- Identifying scenarios
- Identifying and refining use cases
- Identifying relationships among actors and use cases

# Actors

- Actors **are not part** of the system, they can represent **roles a user** of the system can play
- An actor may actively **interchange** information with the system
- An actor may be a **passive** recipient of information (e.g. Database)
- An actor can represent a human, a **machine** or another **system**



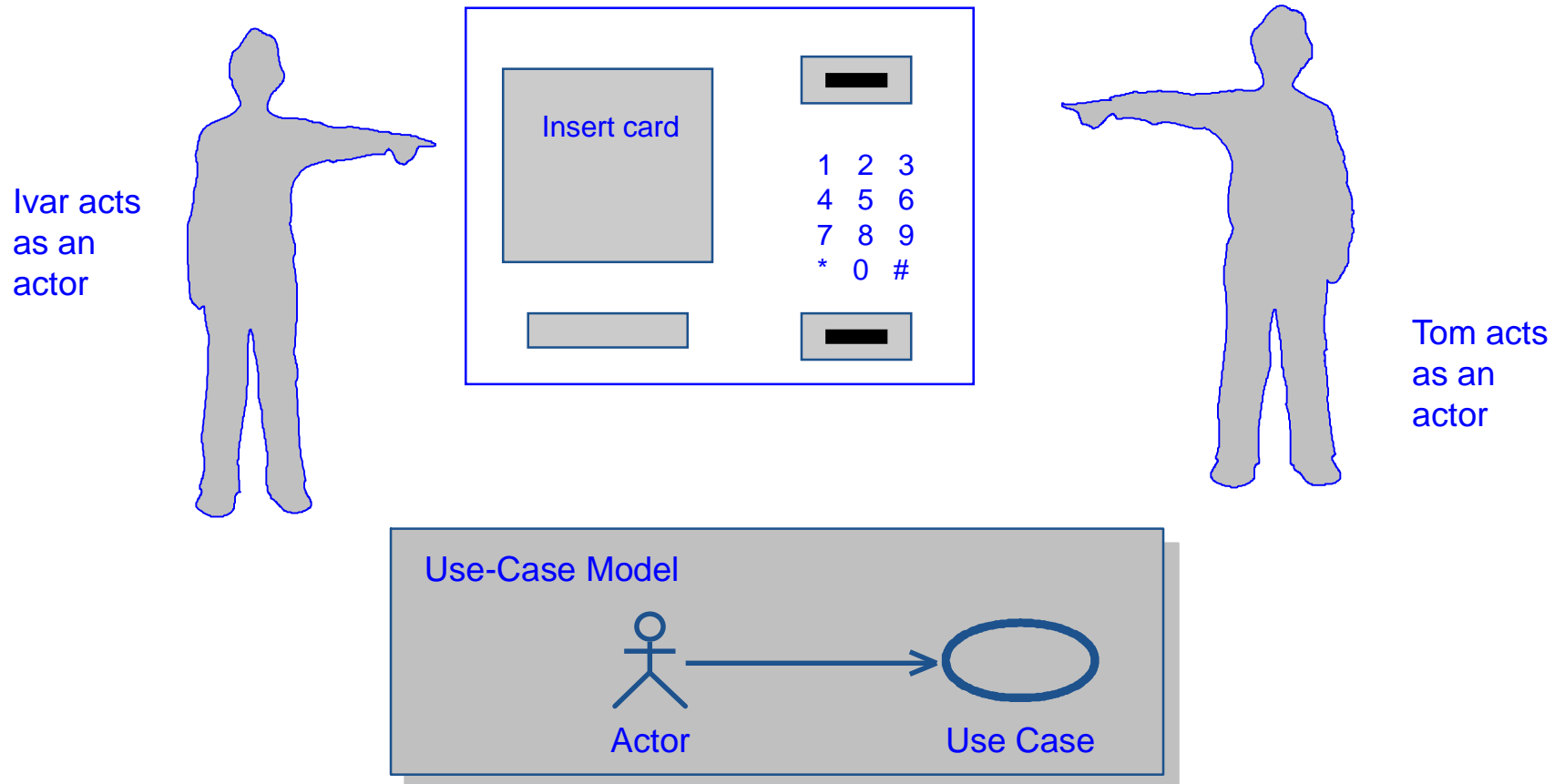
# Actors

## ❖ Finding actors: Useful questions

- Which user groups are supported by the system to perform their work?
- Which user groups execute the system's main functions?
- Which user groups perform secondary functions, such as maintenance and administration?
- With what external hardware or software system will the system interact?

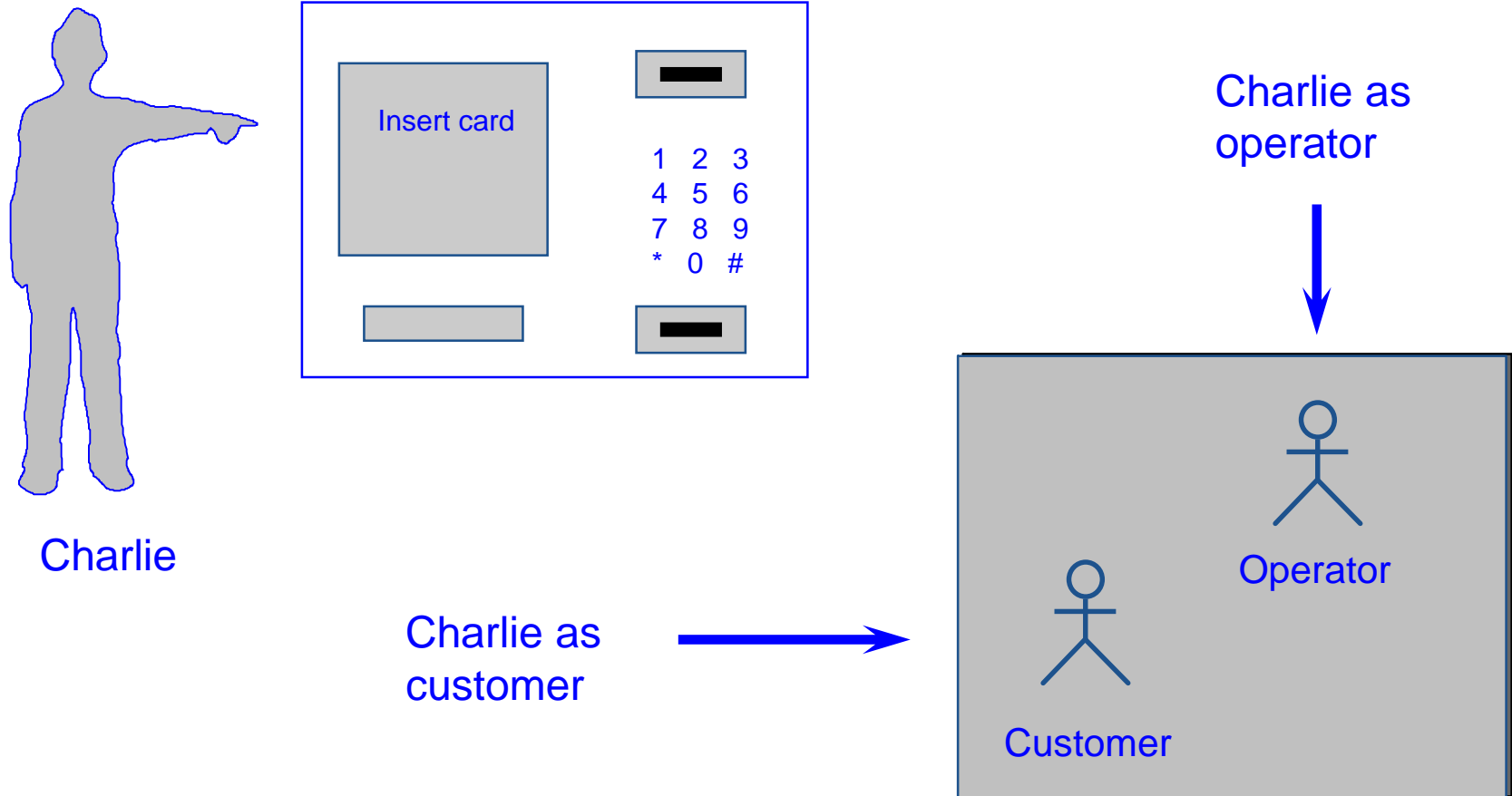
# Actors

## ❖ Instances of actors

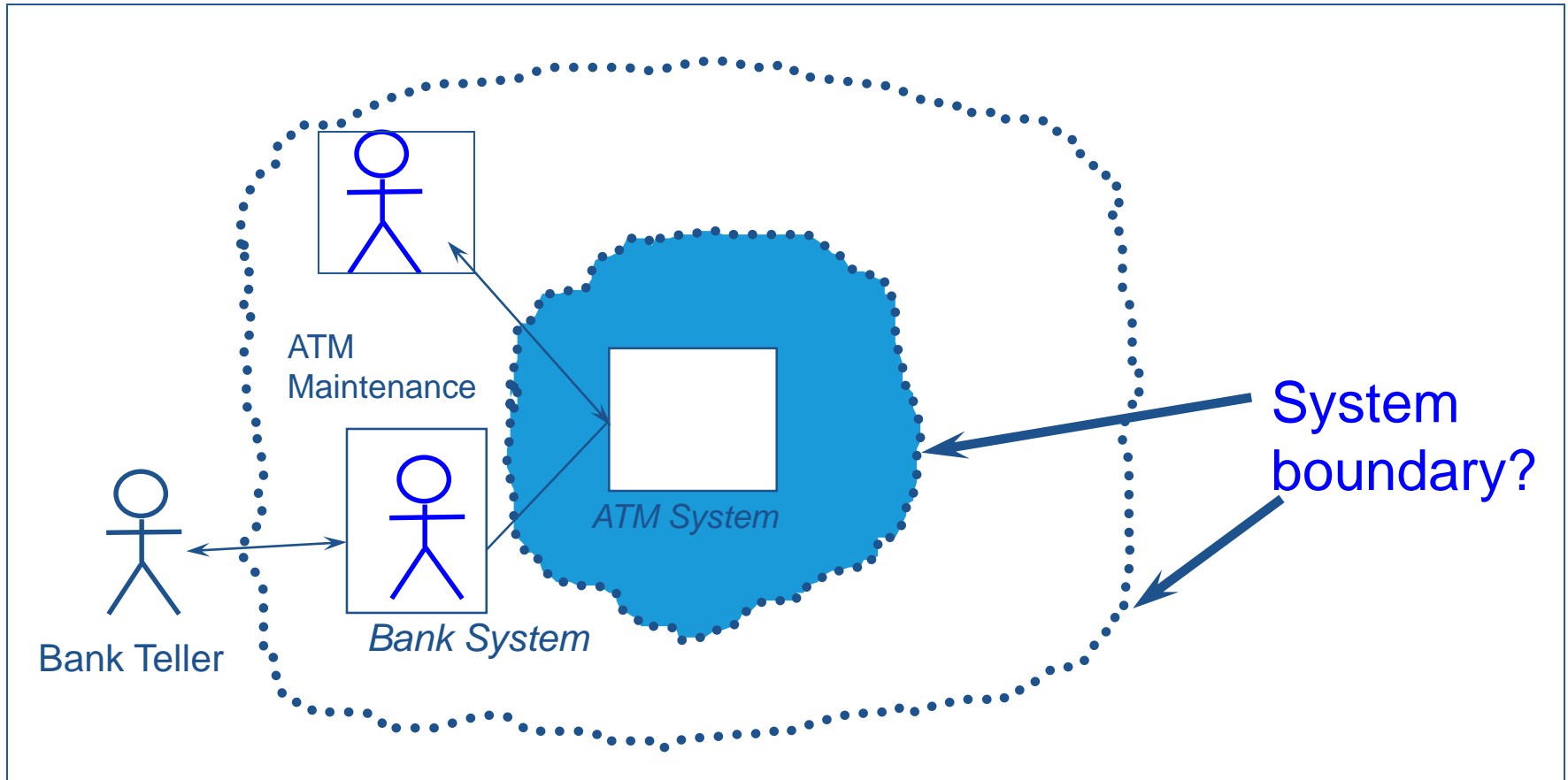


# Actors

## ❖ A user may act as several actors



# Actors and System Boundaries



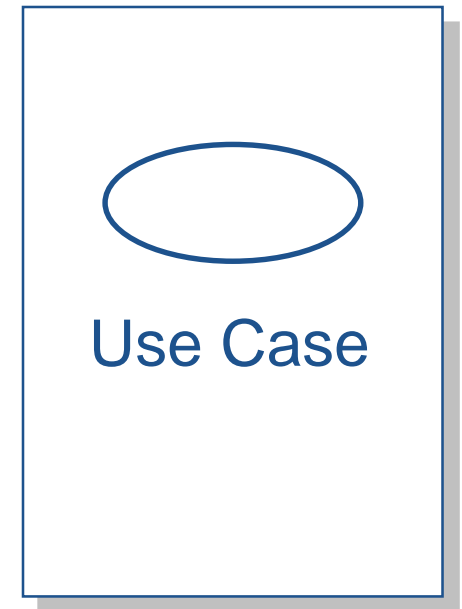
# Questions

## ❖ **List of actor of the following systems?**

- Library management system
- Academic affair system
- Hotel management system
- An e-commerce website

# Use Cases

- A use case is initiated by an actor to invoke a certain functionality in the system
- A use case is a **complete** and **meaningful** flow of events
- Usecase name should be a **verb** or **verb phrase**



# Use Cases

## ❖ **Finding use cases: Useful questions**

- What are the tasks of this actor?
- Will the actor create, store, change, remove or read information in the system?
- What use case will create, store, change, remove, or read, this information?
- Will the actor need to inform the system about sudden, external changes?
- Does the actor need to be informed about certain occurrences in the system?
- Does the system supply the business with the correct behavior?
- What use cases will support and maintain the system?
- Can all functional requirements be performed by the use cases?

# Use Cases

## ❖ Sources of information for use cases

- System specifications/problem statement
- Domain relevant literature
- Interviews with domain experts
- Personal knowledge of the domain
- Legacy systems (Hệ thống trước đây)



# Example

## ❖ Which ones are usecases?

- Sign in a system (facebook, ...)
- Create an account
- Select an item in a list (of products,...)
- Buy a product
- Print a bill

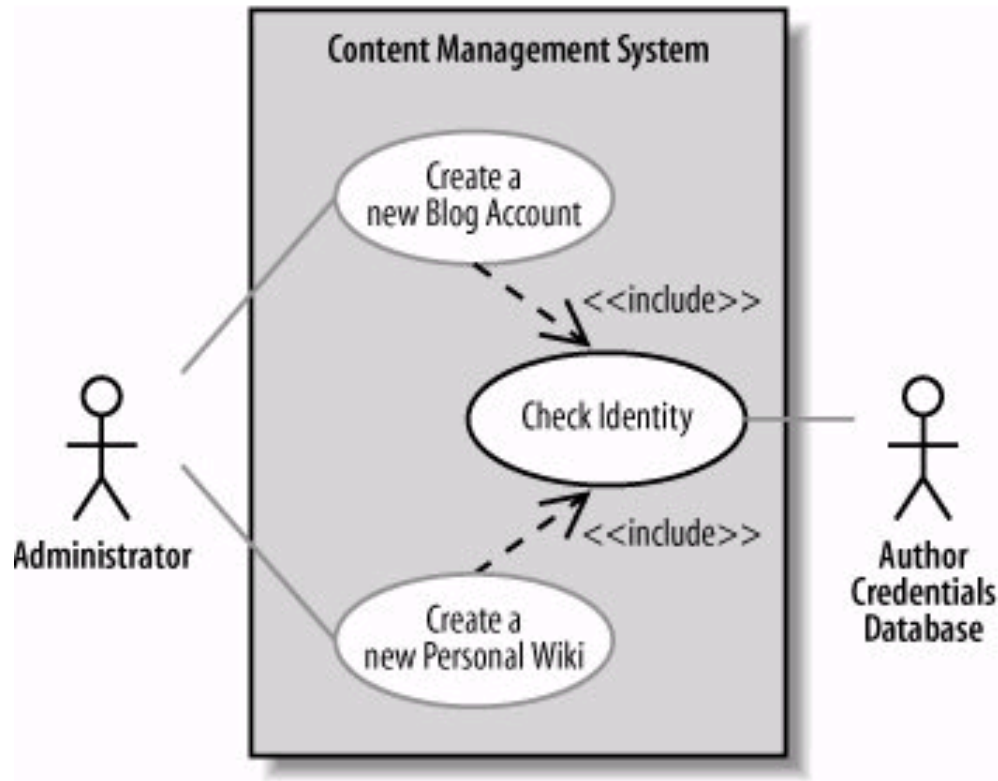
# Example

## ❖ Which ones are usecases?

- Manage users (Add, Update, Remove)
- Add a user
- Update a user
- Remove a user

# Use case relationships

## ❖ <<include>> relationship supports reuse between use cases

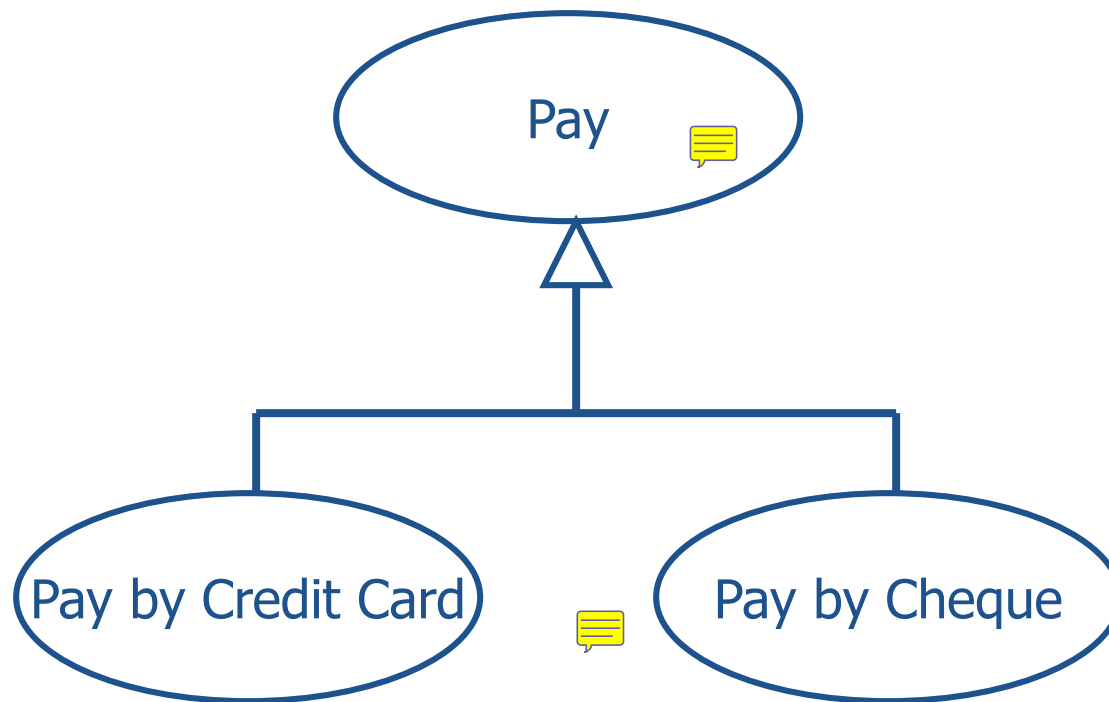


## Use case relationships (2)

❖ **<<extend>> relationship is somehow similar to <<include>>, but the reuse is optional**

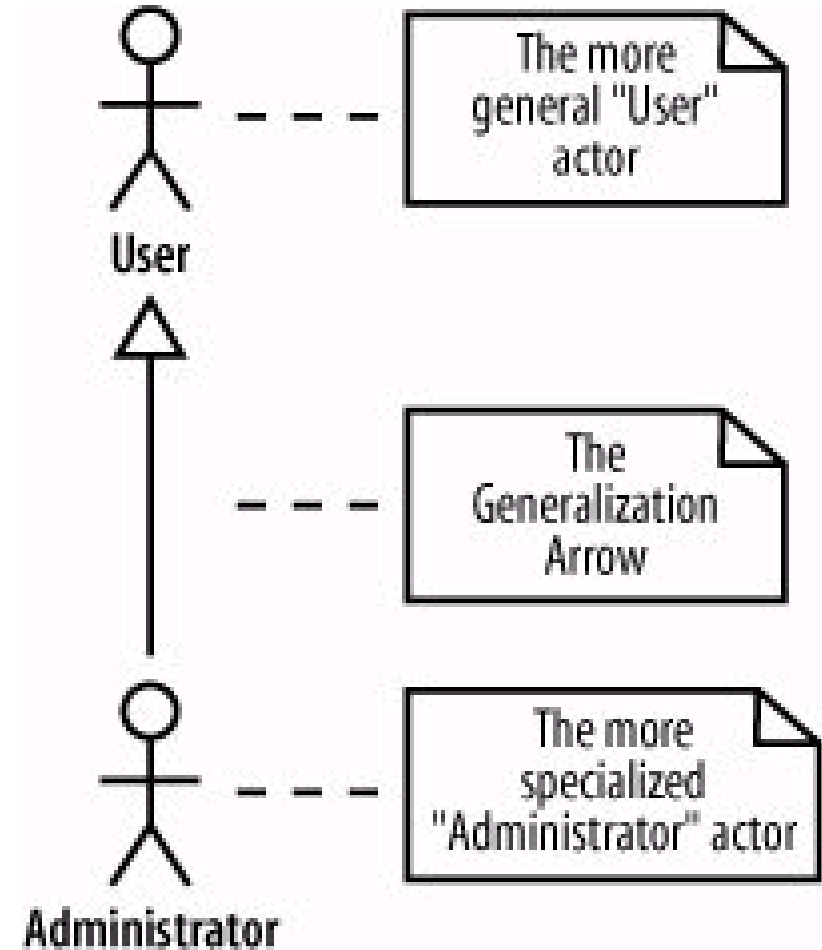


# Use Case Inheritance



# Relationships between actors

- Different actors may have some relationship to each other



# Contents

- I.** Requirements
- II.** Use case diagrams
- III.** Actors and Use cases
- IV.** Relationship
- V.** Scenarios

# Scenarios

- A scenario is an instance of a use case
  - It is one flow through a use case
  - It is “a narrative description of what people do and experience as they try to make use of computer systems and applications”



# Scenarios

## ❖ **Scenario template**

- Name
- Brief description
- Actor (s)
- Flow of events
  - Basic flow
  - Alternate flows
- Pre-conditions
- Post-conditions
- Extension points

# Scenarios

## ❖ Scenario template

- Flow of events
  - Basic flow: *Often referred to as the **primary** scenario or course of events*
  - Alternate flows: *Give any alternate flows that might occur. May include flows that involve **error** conditions. Or flows that fall outside of the basic flow*

# Scenarios

## ❖ Scenario template

- Pre-conditions
  - A list of conditions that must be true **before** the Use Case starts
- Post-conditions
  - A list of conditions that must be true when the Use Case **ends**
- Extension points
  - If the Use Case has extension points, list them here

# Example

❖ **Login**

# Questions for identifying scenarios

## Questions for identifying scenarios

- What are the tasks that the actor wants the system to perform?
- What information does the actor access? Who creates that data? Can it be modified or removed? By whom?
- Which external changes does the actor need to inform the system about? How often? When?
- Which events does the system need to inform the actor about? With what latency?