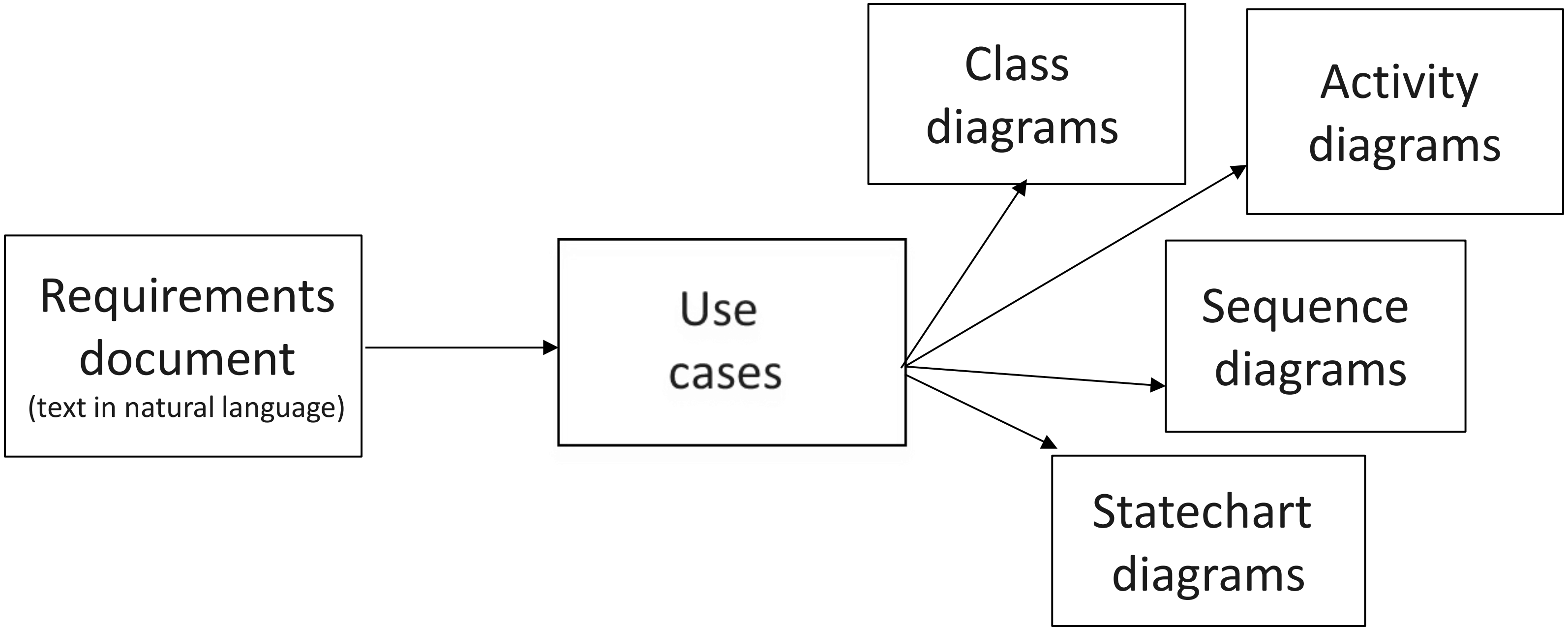
Use Case Diagram

Introduction

- *Use Case*: "... a typical interaction between a user and a computer system", Booch
  - Here, "user" is anything that needs or invokes the functionality of the system
  - "Computer system" is the system being modeled.
- Use cases capture and document the user-visible functionality of a system (functional requirements)
- Use cases capture how the system will benefit the user
- Each use case represents a discrete goal for the user
- Illustrate the developer's understanding of the user's requirements.

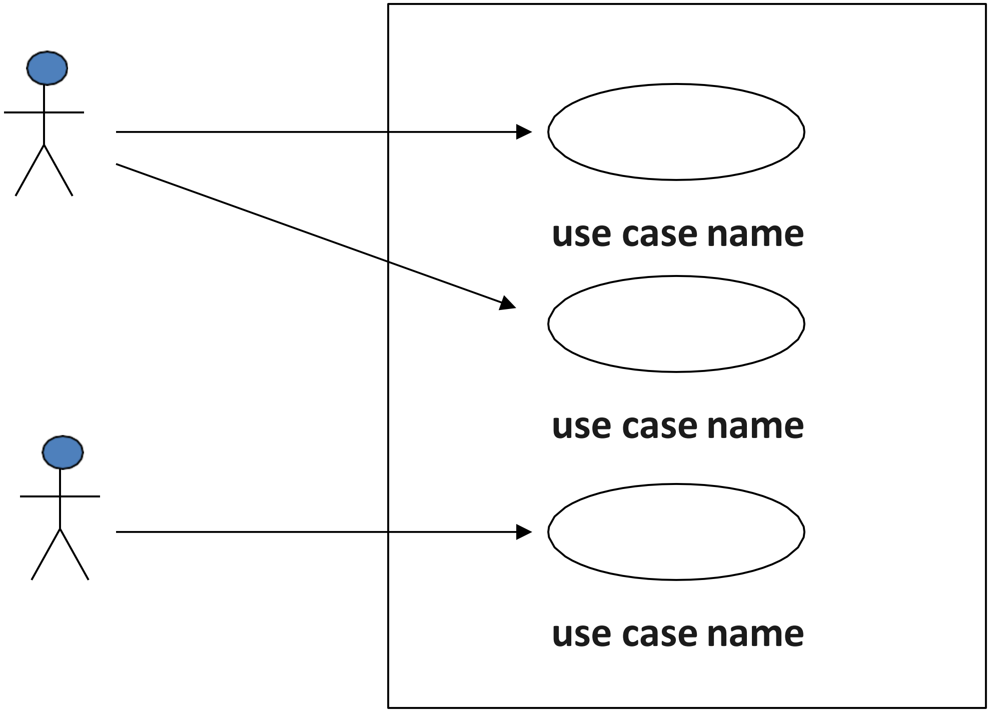
Use Case Diagram, purpose



- Use case models are developed at different levels of abstraction
  - system, system component, or a class.
- Use case modelling is an iterative and incremental process.
  - If user requirements change, the changes should be made in all the affected documents.

Use Case diagrams, basic UML notation

- Use Case Diagrams provide a visual way to document user goals and explore possible functionality.
- Components of use case diagram:
  - Actor
  - Use case
  - System boundary
  - Relationship

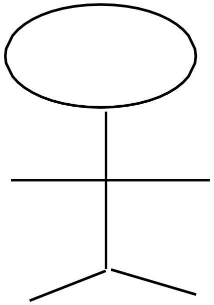


Use cases: Information captured

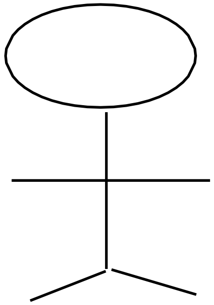
- Actors
- Relationships with other use cases
- Pre-conditions
- Details
- Post-conditions
- Exceptions
- Constraints
- Alternatives

ACTOR

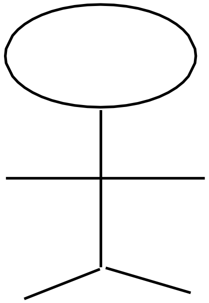
- An actor is some one or something that must interact with the system under development
- Actors can be human or automated systems, hardware device, timer.
- Actors are not part of the system.
- UML notation for actor is stickman, shown below.



Student



Faculty



Employee

ACTOR cont.

- It is a role, an user plays with respect to system.
- Actors carry out use cases and a single actor may perform more than one use case.
- Actors are determined by observing the direct uses of the system

Primary and Secondary Actors

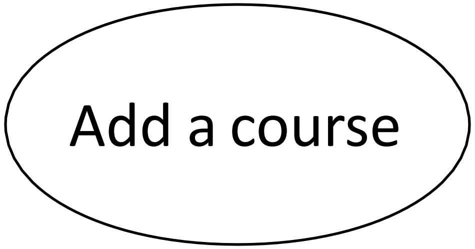
- **Primary Actor**
  - Acts on the system
  - Initiates an interaction with the system
  - Targeted end user of the system
    - leave management system - an employee or executive.
- **Secondary Actor**
  - Helps the system to fulfills its goal
  - required for the correct functionality of the system.
    - leave management system - system administrator
    - make sure that the proper leaves are credited, records are reconciled

Hints for Finding ACTOR

- Who or what will use the main functionality of the system?
- Who or what will provide input to this system?
- Who or what will use output from this system?
- Who will need support from the system to do their work?
- Are there any other software systems with which this one needs to interact
- Are there any hardware devices used or controlled by this system?

USE CASE

- An use case is a pattern of behavior, the system exhibits
- The use cases are sequence of actions that the user takes on a system to get particular target
- USE CASE is a dialogue between an actor and the system.
- Examples:



Add a course

USE CASE cont.

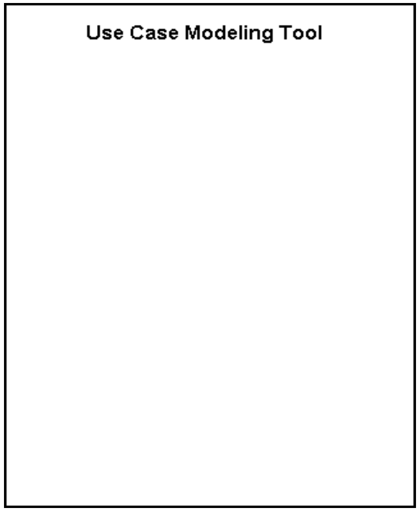
- A use case typically represents a major piece of functionality that is complete from beginning to end.
- Most of the use cases are generated in initial phase, but may add some more after proceeding.
- A use case may be small or large. It captures a broad view of a primary functionality of the system in a manner that can be easily grasped by non technical user.

Finding Use-cases

- For each actor ask these questions:
  - Which functions does the actor require from the system?
  - What does the actor need to do?
  - Could the actor's work be simplified or made efficient by new functions in the system?
  - What events are needed in the system?
  - What are the problems with the existing systems?
  - What are the inputs and outputs of the system?

System Boundary

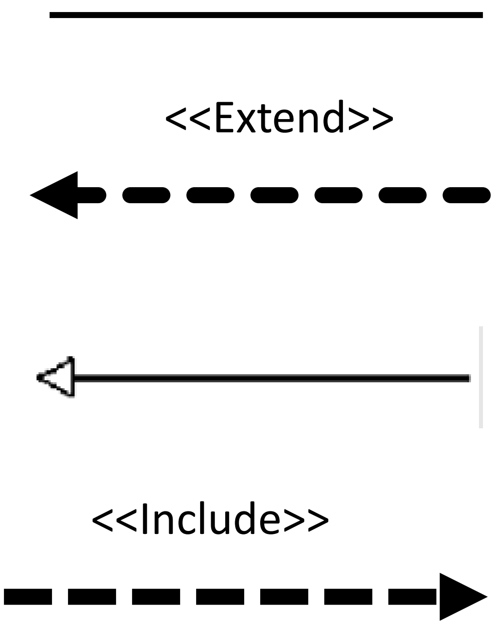
- It is shown as a rectangle.
- It helps to identify what is external versus internal, and what the responsibilities of the system are.
- The external environment is represented only by actors.



Relationship

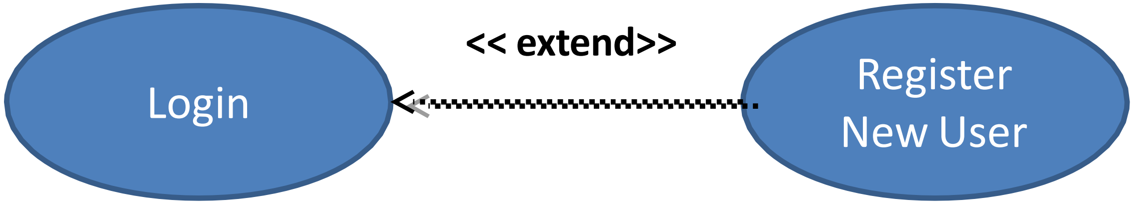
- Relationship is an association between use case and actor.
- There are several Use Case relationships:

- Association
- Extend
- Generalization
- Include



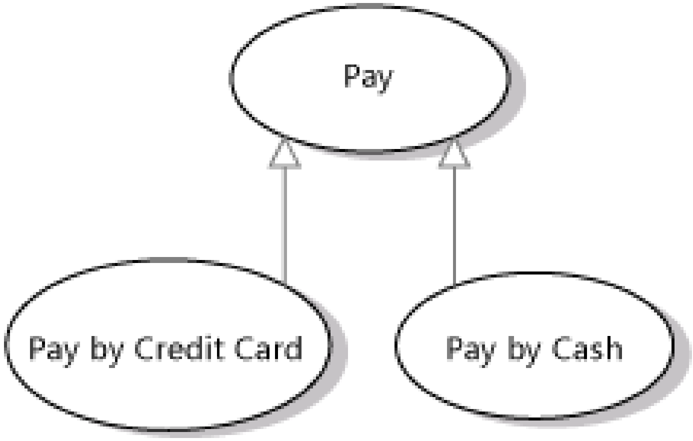
Extend Relationship

- The extended relationship is used to indicate that use case completely consists of the behavior of another use case at one or specific point
- use cases that extend the behavior of other core use cases. Enable to factor variants
- The base use case implicitly incorporates the behavior of another use case at certain points called extension points
- It is shown as a dotted line with an arrow point and labeled <<extend>> (guillemets)



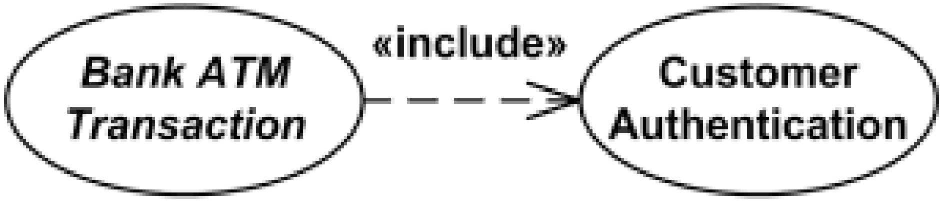
Generalization

- Generalization is a relationship between a general use case and a more specific use case that inherits and extends features to it
- use cases that are specialized versions of other use cases
- It is shown as a solid line with a hollow arrow point

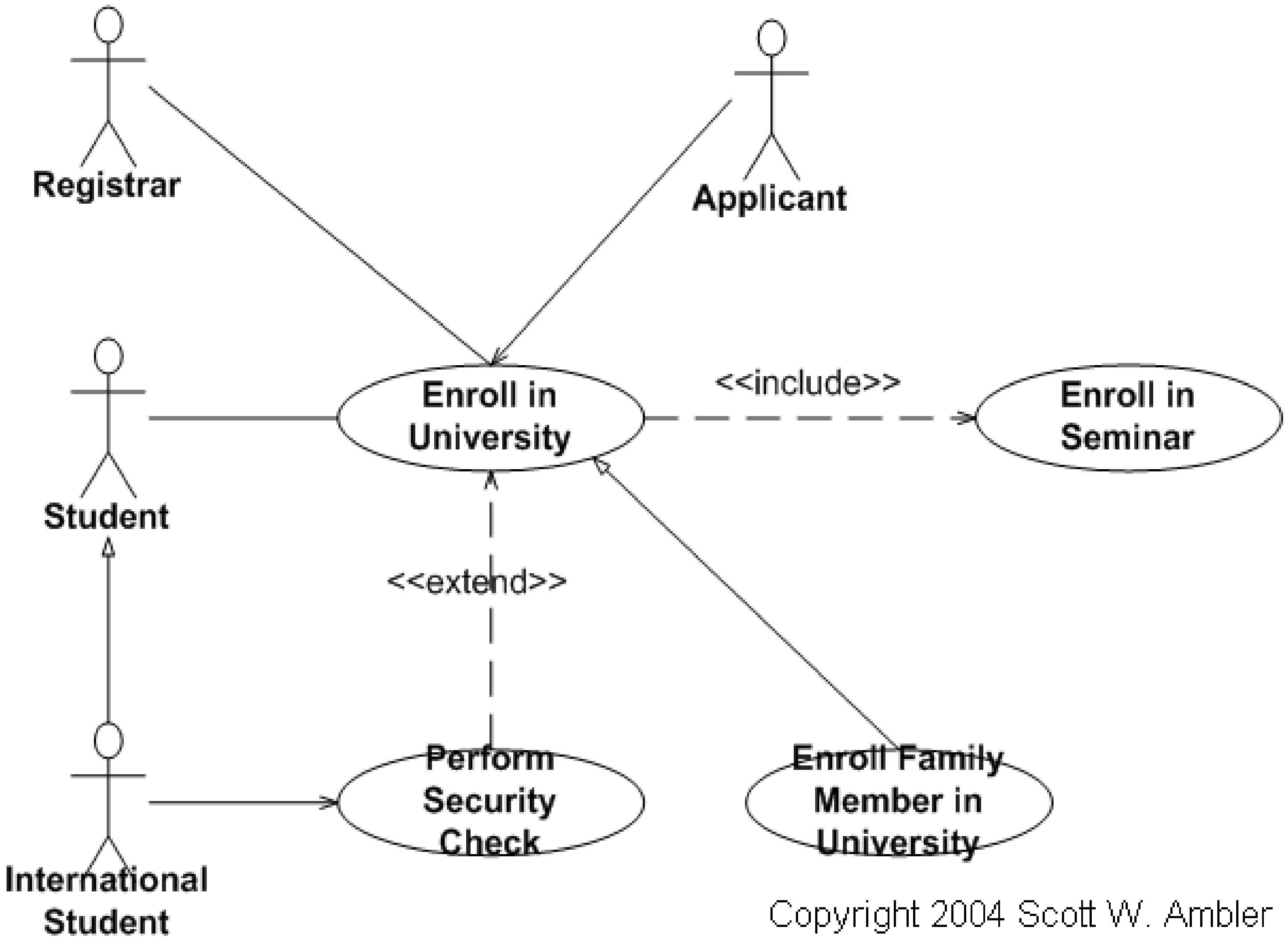


Include Relationship

- Include relationships insert additional behavior into a base use case
- use cases that are included as parts of other use cases. Enable to factor common behavior.
- The base use case explicitly incorporates the behavior of another use case at a location specified in the base.
- They are shown as a dotted line with an open arrow and the key word <<include>>



Extend vs Include



Copyright 2004 Scott W. Ambler

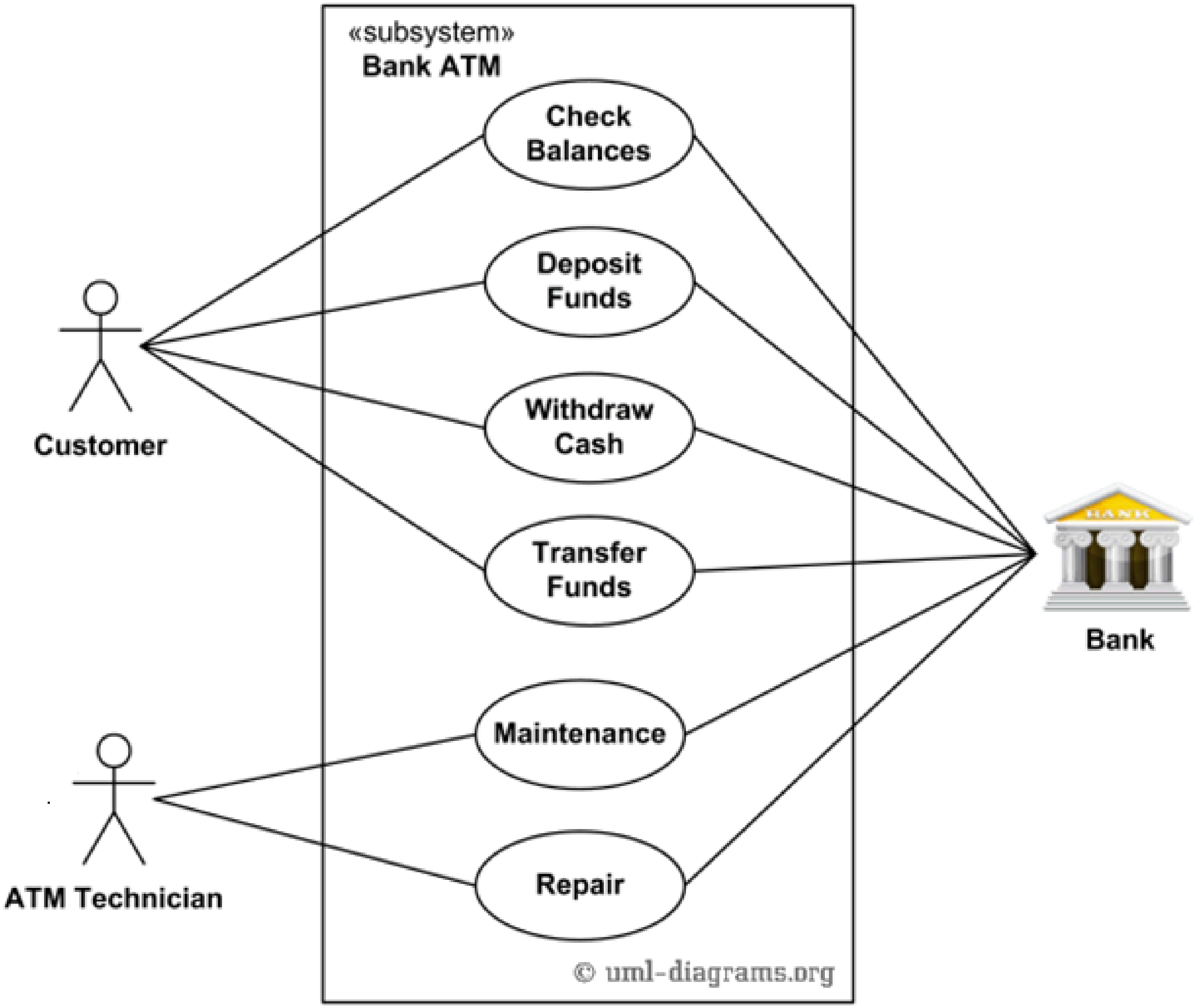
Bank ATM

- An automated teller machine (**ATM**) is a banking subsystem that provides bank customers with access to financial transactions in a public space without the need for a ~~cashier~~, ~~clerk~~, or ~~bank teller~~.
- *Customer* uses bank ATM to *Check Balances* of his/her bank accounts, *Deposit Funds*, *Withdraw Cash* and/or *Transfer Funds*.
- On most bank ATMs, the customer is authenticated by inserting a plastic ATM card and entering a personal identification number (PIN).
- *Customer Authentication* is required for every ATM transaction.

Bank ATM cont.

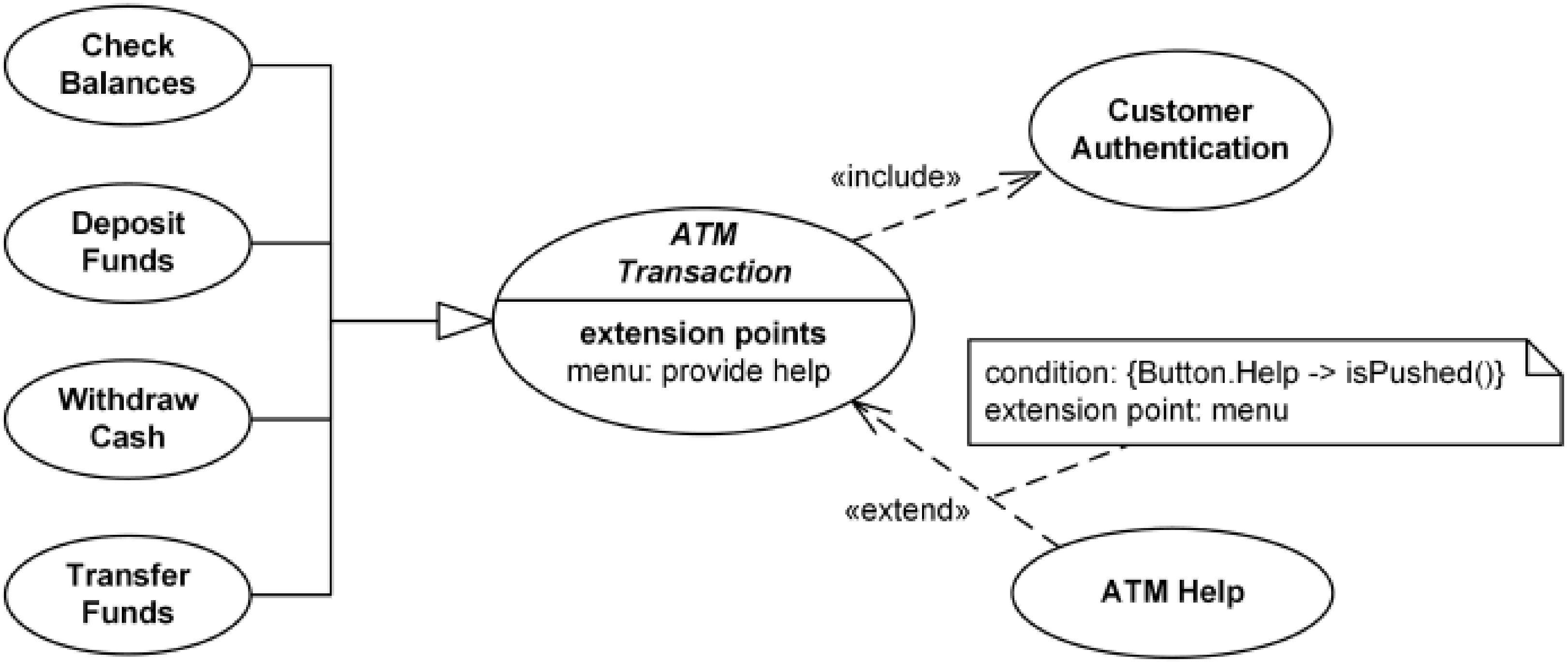
- Customer may need some help from the ATM.
- *ATM Technician* provides *Maintenance* and *Repairs*.
- *ATM Technician* maintains or repairs Bank ATM.
- *Maintenance* includes *Replenishing* ATM with cash, ink or printer paper, *Upgrades* of hardware, firmware or software, and remote or on-site *Diagnostics*.
- All these use cases also involve *Bank* whether it is related to customer transactions or to the ATM servicing.

Bank ATM subsystem - top level



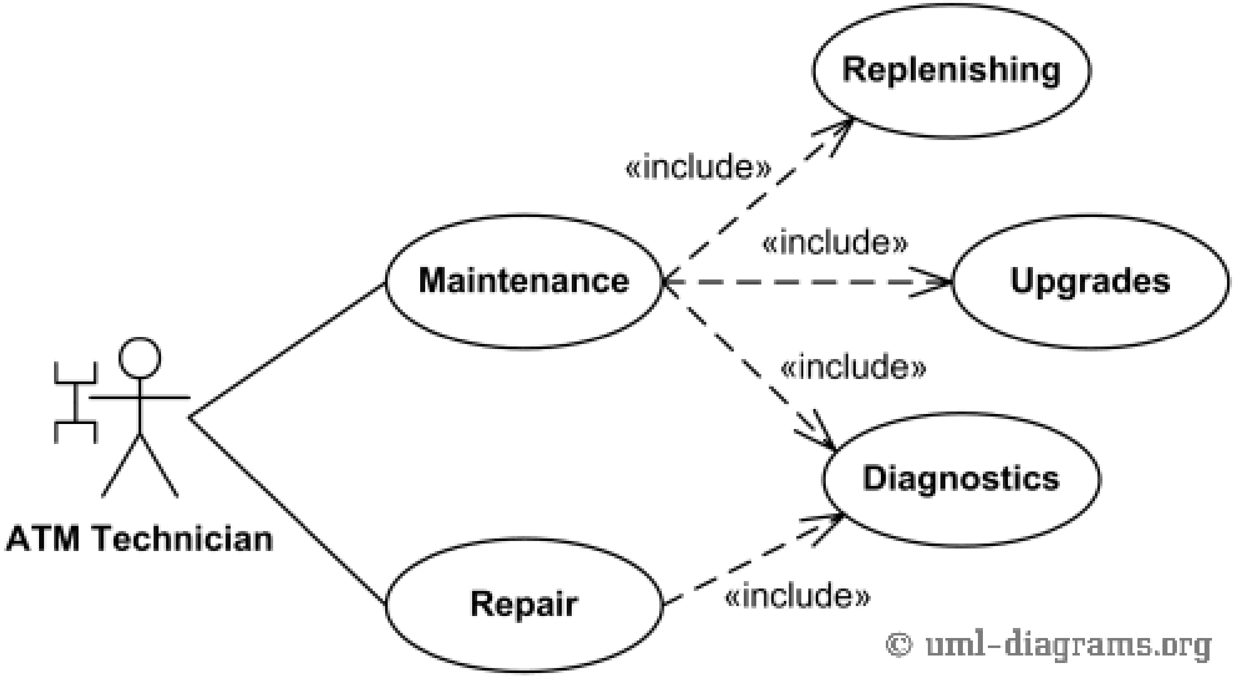
Bank ATM Transactions and Customer Authentication Use Cases

- *Customer Authentication* use case - show it as include relationship.
- Transaction generalizations make the *ATM Transaction* an abstract use case.
- *ATM Transaction* use case is extended by the *ATM Help* use case

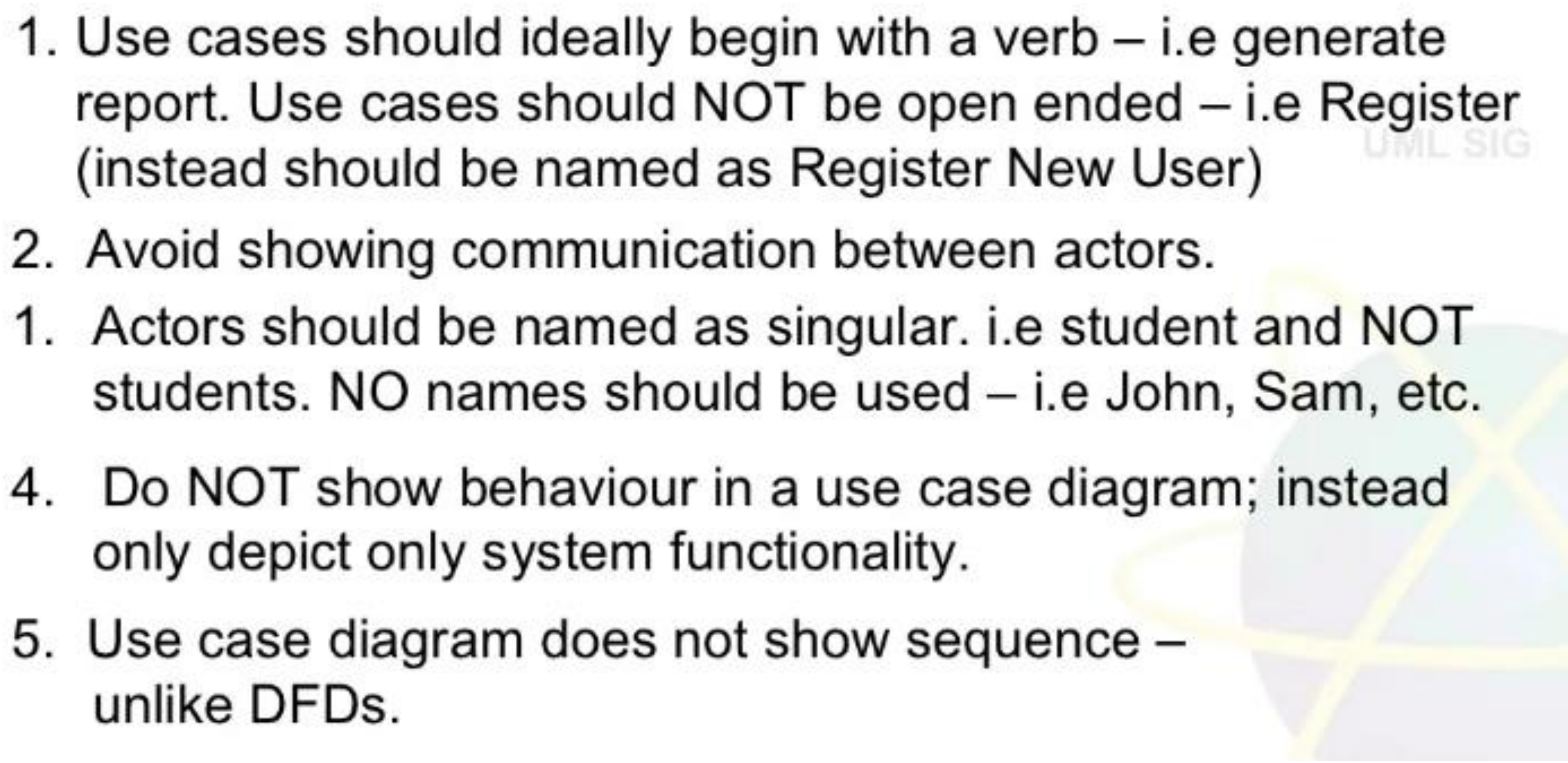


Bank ATM Maintenance, Repair, Diagnostics Use Cases

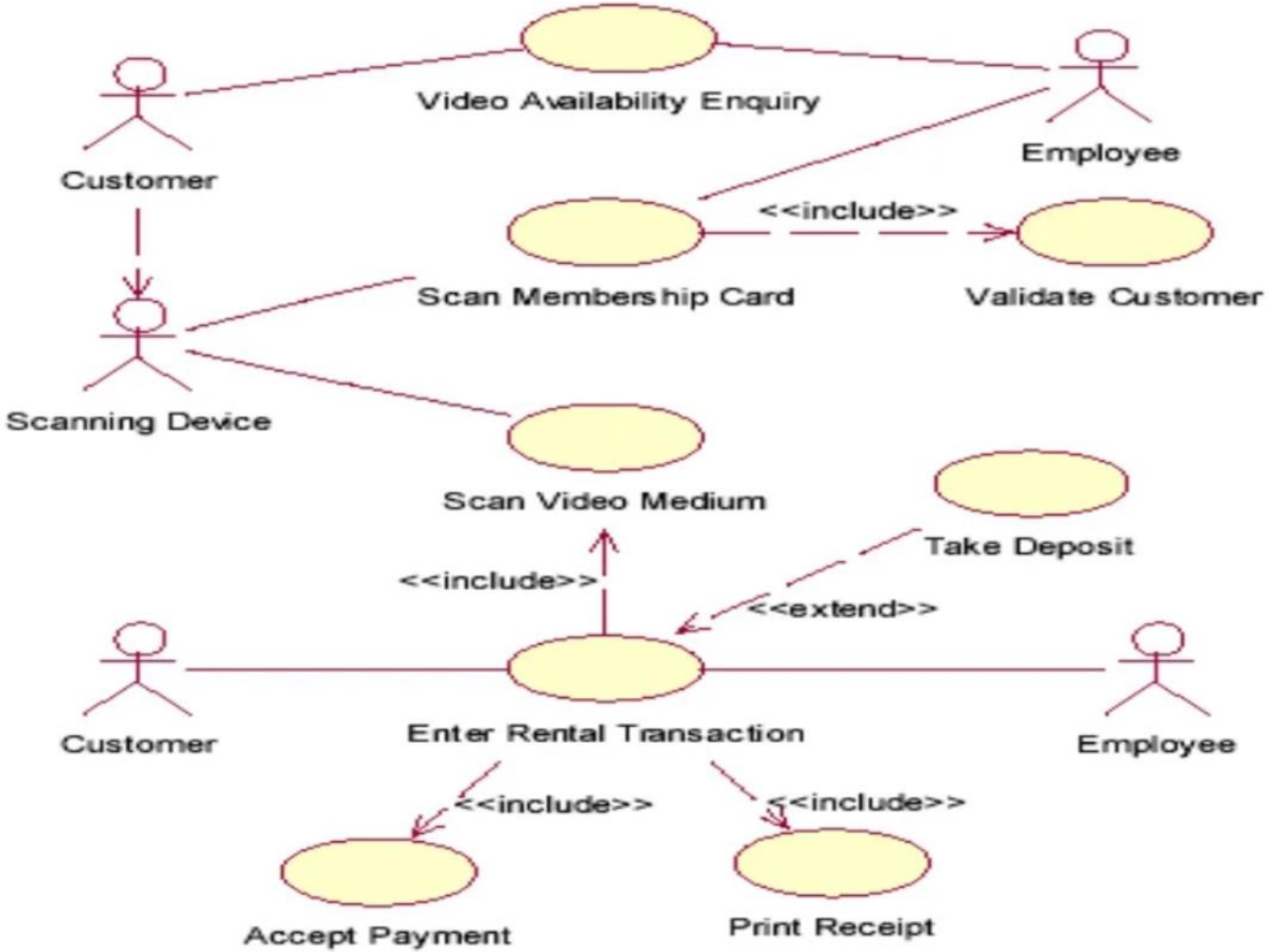
- *Maintenance* use case includes *Replenishing* ATM with cash, ink or printer paper, *Upgrades* of hardware, firmware or software, and remote or on-site *Diagnostics*.
- *Diagnostics* is also included in *Repair* use case.



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.
1. 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.


Video Availability Enquiry
Customer
Employee
<<include>>
Scan Membership Card
Validate Customer
Scanning Device
Scan Video Medium
Take Deposit
<<include>>
<<extend>>
Customer
Enter Rental Transaction
Employee
<<include>>
<<include>>
Accept Payment
Print Receipt

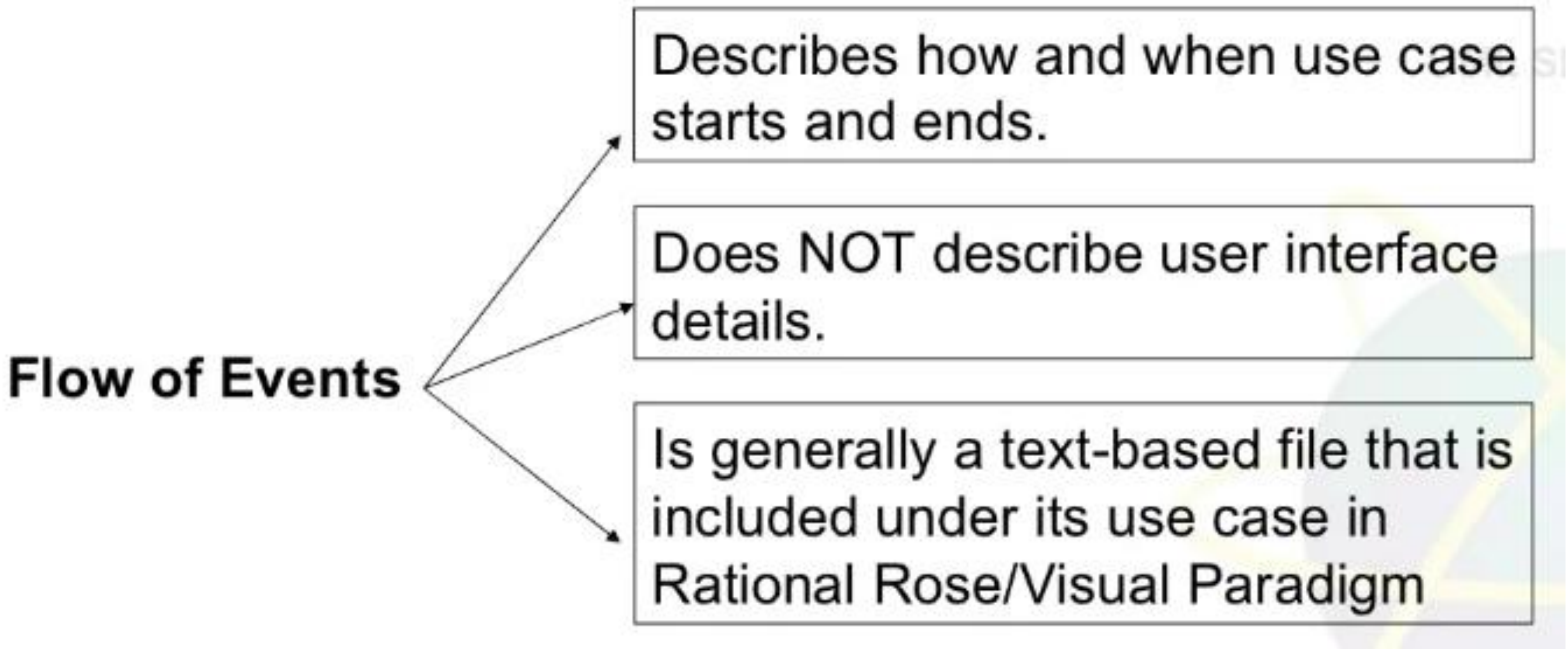
Use Case Description / Specification

- Use case specification is synonymous to use case description and use case definition and can be used interchangeably.

- Defines information that pertains to a particular use case which is important to understand the purpose behind the use case.

- Has one or more flow of events or pathways.
  - Is a textual description embodying sequence of events.
  - To understand the complexity in realizing the use cases.

Flow of Events / Pathways



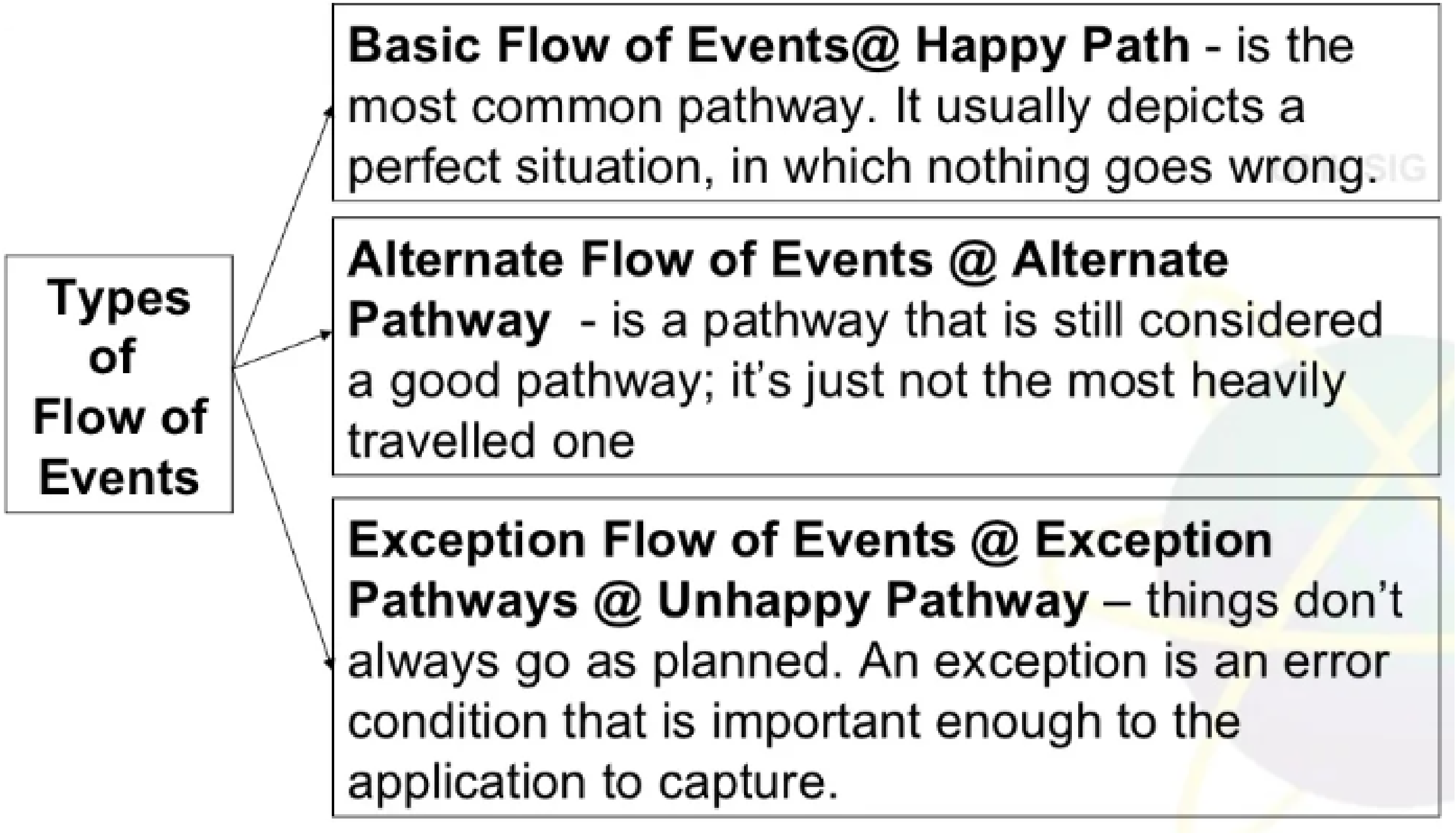
Flow of Events

Describes how and when use case starts and ends.

Does NOT describe user interface details.

Is generally a text-based file that is included under its use case in Rational Rose/Visual Paradigm

Types of Flow of Events / Pathways



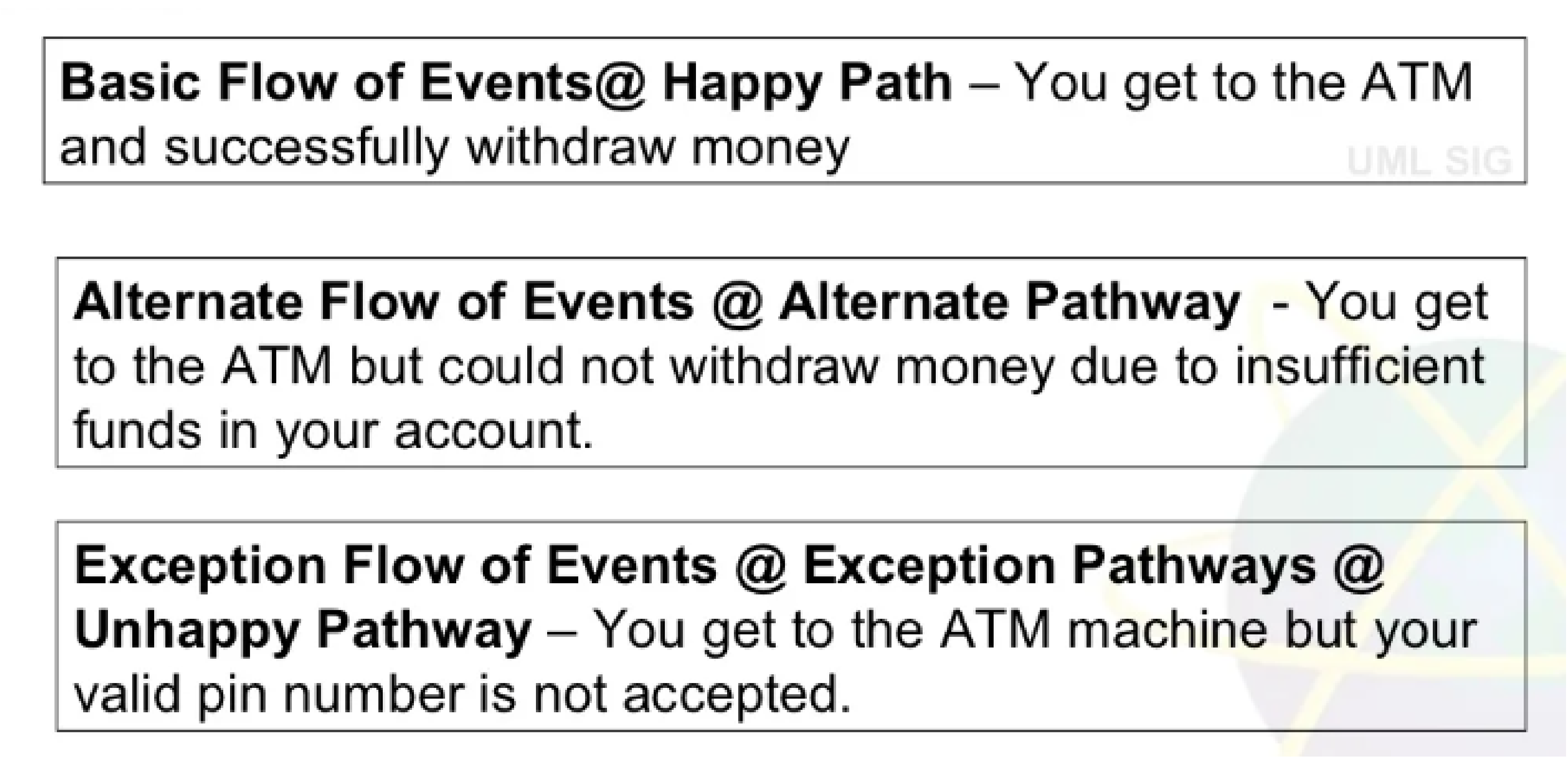
Types of Flow of Events

Basic Flow of Events@ Happy Path - is the most common pathway. It usually depicts a perfect situation, in which nothing goes wrong.

Alternate Flow of Events @ Alternate Pathway - is a pathway that is still considered a good pathway; it's just not the most heavily travelled one

Exception Flow of Events @ Exception Pathways @ Unhappy Pathway – things don't always go as planned. An exception is an error condition that is important enough to the application to capture.

Types of Flow of Events / Pathways



Basic Flow of Events@ Happy Path – You get to the ATM and successfully withdraw money

Alternate Flow of Events @ Alternate Pathway - You get to the ATM but could not withdraw money due to insufficient funds in your account.

Exception Flow of Events @ Exception Pathways @ Unhappy Pathway – You get to the ATM machine but your valid pin number is not accepted.

Use Case Description

Each use case may include all or part of the following

Title or Reference Name	- meaningful name of the UC
Author/Date	- the author and creation date
Modification/Date	- last modification and its date
Purpose	- specifies the goal to be achieved
Overview	- short description of the processes
Cross References	- requirements references
Actors	- agents participating
Pre Conditions	- must be true to allow execution
Post Conditions	- will be set when completes normally
Normal flow of events	- regular flow of activities
Alternative flow of events	- other flow of activities
Exceptional flow of events	- unusual situations
Implementation issues	- foreseen implementation problems

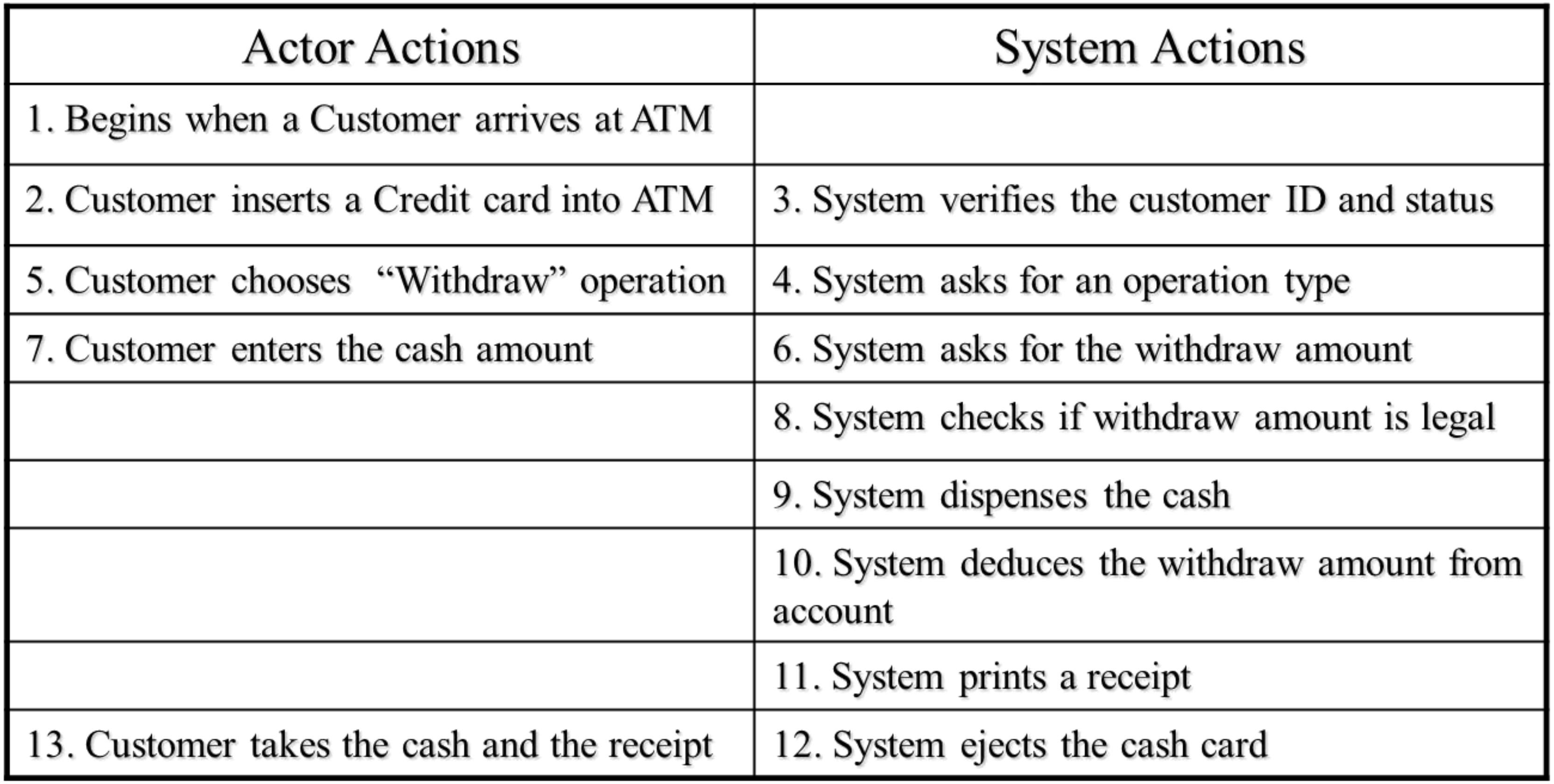
Example- Money Withdraw

- Use Case: Withdraw Money
- Author: PKD
- Date: 11-09-2013
- Purpose: To withdraw some cash from user’s bank account
- Overview: *The use case starts when the customer inserts his card into the system. The system requests the user PIN. The system validates the PIN. If the validation succeeded, the customer can choose the withdraw operation else alternative 1 – validation failure is executed. The customer enters the amount of cash to withdraw. The system checks the amount of cash in the user account, its credit limit. If the withdraw amount in the range between the current amount + credit limit the system dispense the cash and prints a withdraw receipt, else alternative 2 – amount exceeded is executed.*
- Cross References: R1.1, R1.2, R7

Example- Money Withdraw

- Actors: Customer
- Pre Condition:
  - The ATM must be in a state ready to accept transactions
  - The ATM must have at least some cash on hand that it can dispense
  - The ATM must have enough paper to print a receipt for at least one transaction
- Post Condition:
  - The current amount of cash in the user account is the amount before the withdraw minus the withdraw amount
  - A receipt was printed on the withdraw amount
  - The withdraw transaction was audit in the System log file

Normal flow of events

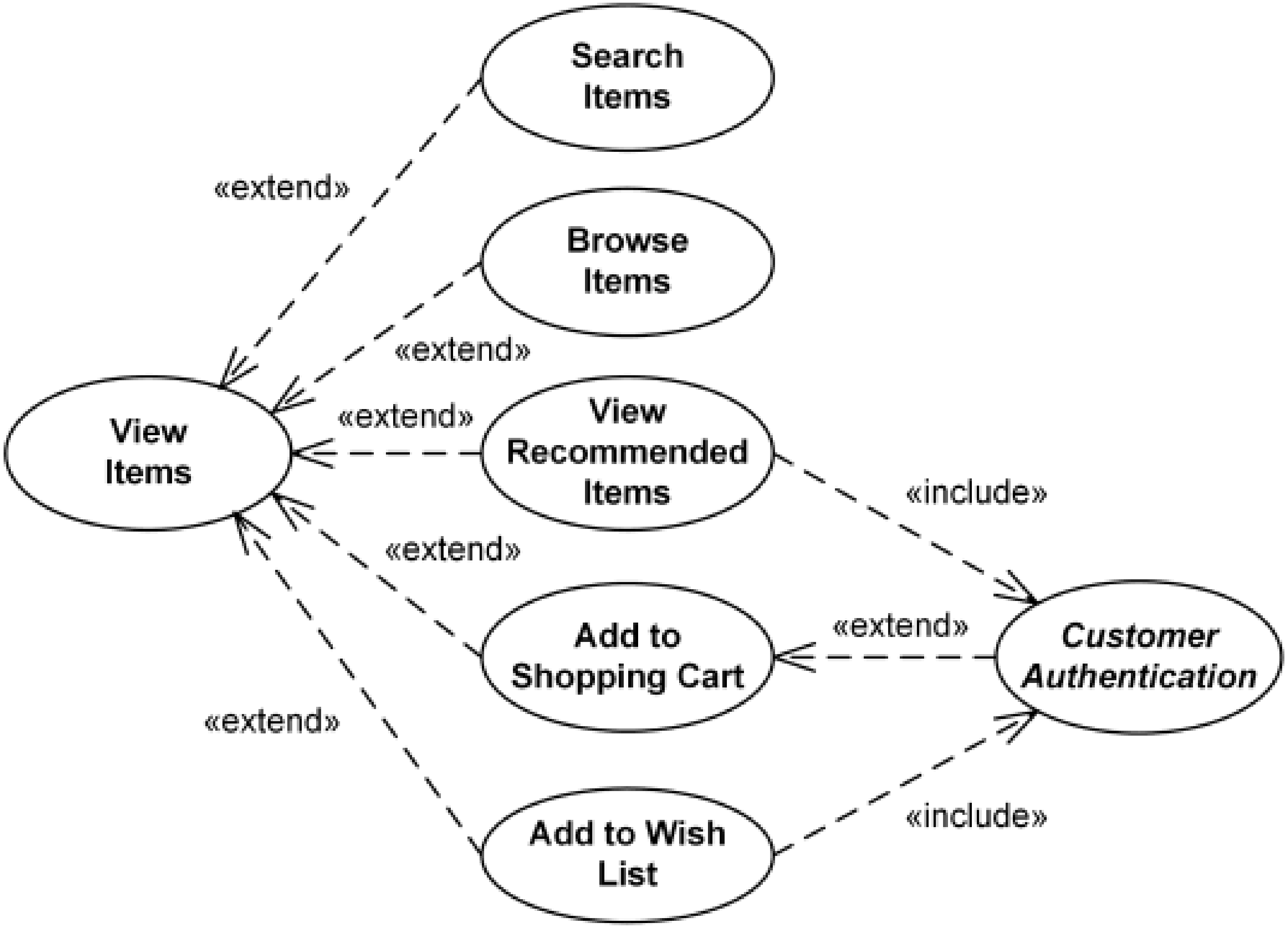


Actor Actions	System Actions
1. Begins when a Customer arrives at ATM	
2. Customer inserts a Credit card into ATM	3. System verifies the customer ID and status
5. Customer chooses "Withdraw" operation	4. System asks for an operation type
7. Customer enters the cash amount	6. System asks for the withdraw amount
	8. System checks if withdraw amount is legal
	9. System dispenses the cash
	10. System deduces the withdraw amount from account
	11. System prints a receipt
13. Customer takes the cash and the receipt	12. System ejects the cash card

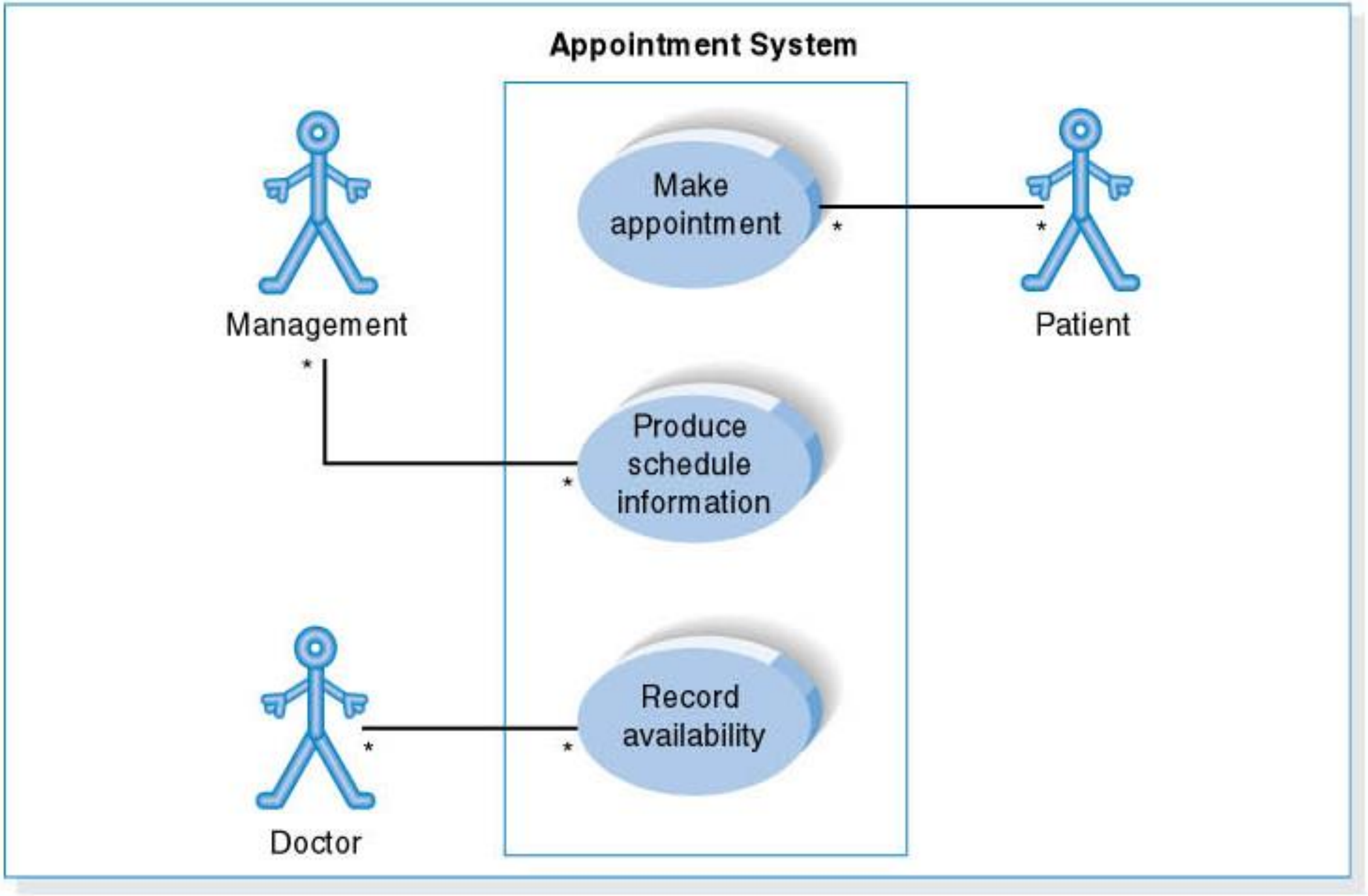
Example- Money Withdraw

- Alternative flow of events:
  - Step 3: Customer authorization failed. Display an error message, cancel the transaction and eject the card.
  - Step 8: Customer has insufficient funds in his account. Display an error message, and go to step 6.
  - Step 8: Customer exceeds the legal amount. Display an error message, and go to step 6.
- Exceptional flow of events:
  - Power failure in the process of the transaction before step 9, cancel the transaction and eject the card

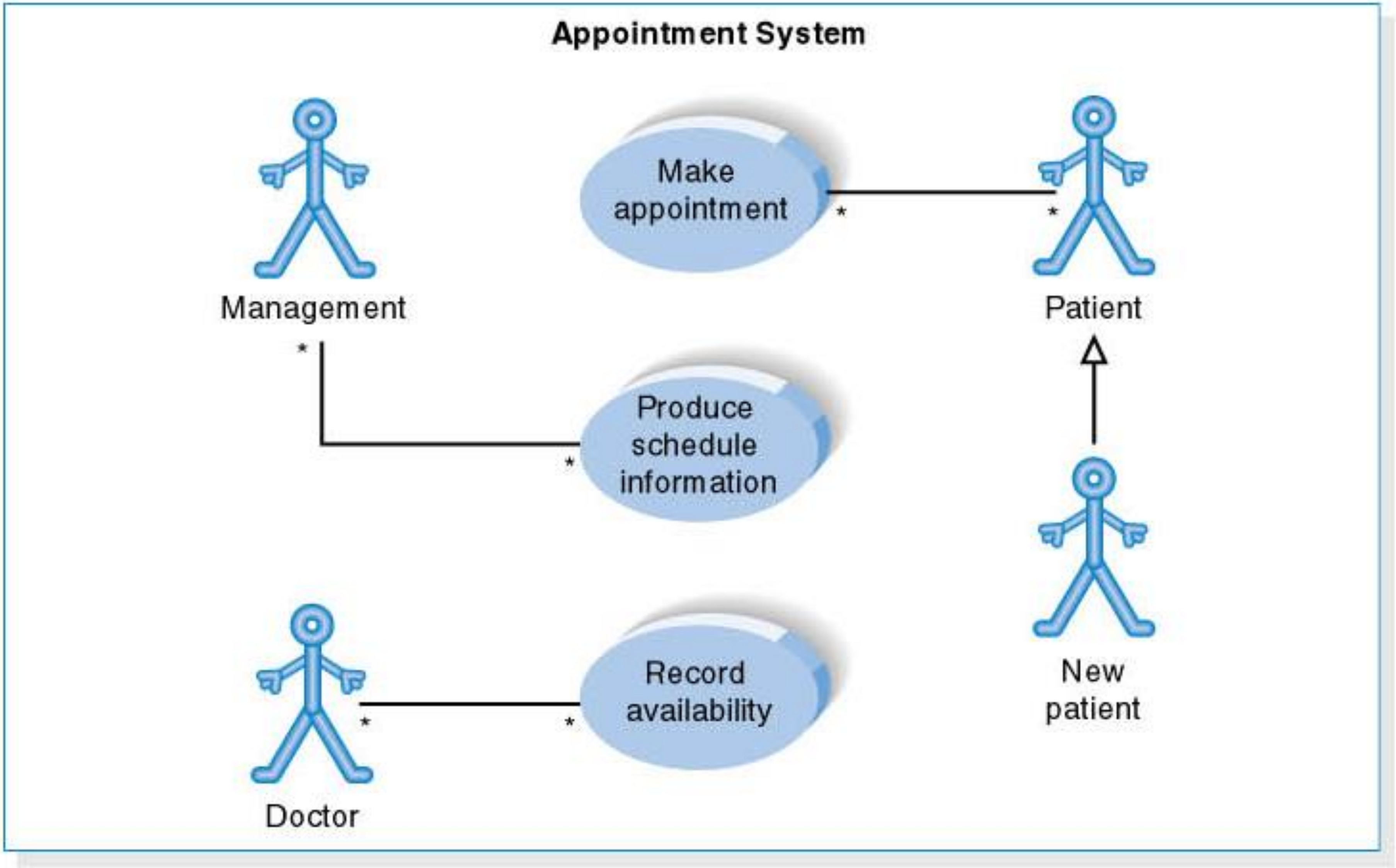
View Items Use Case



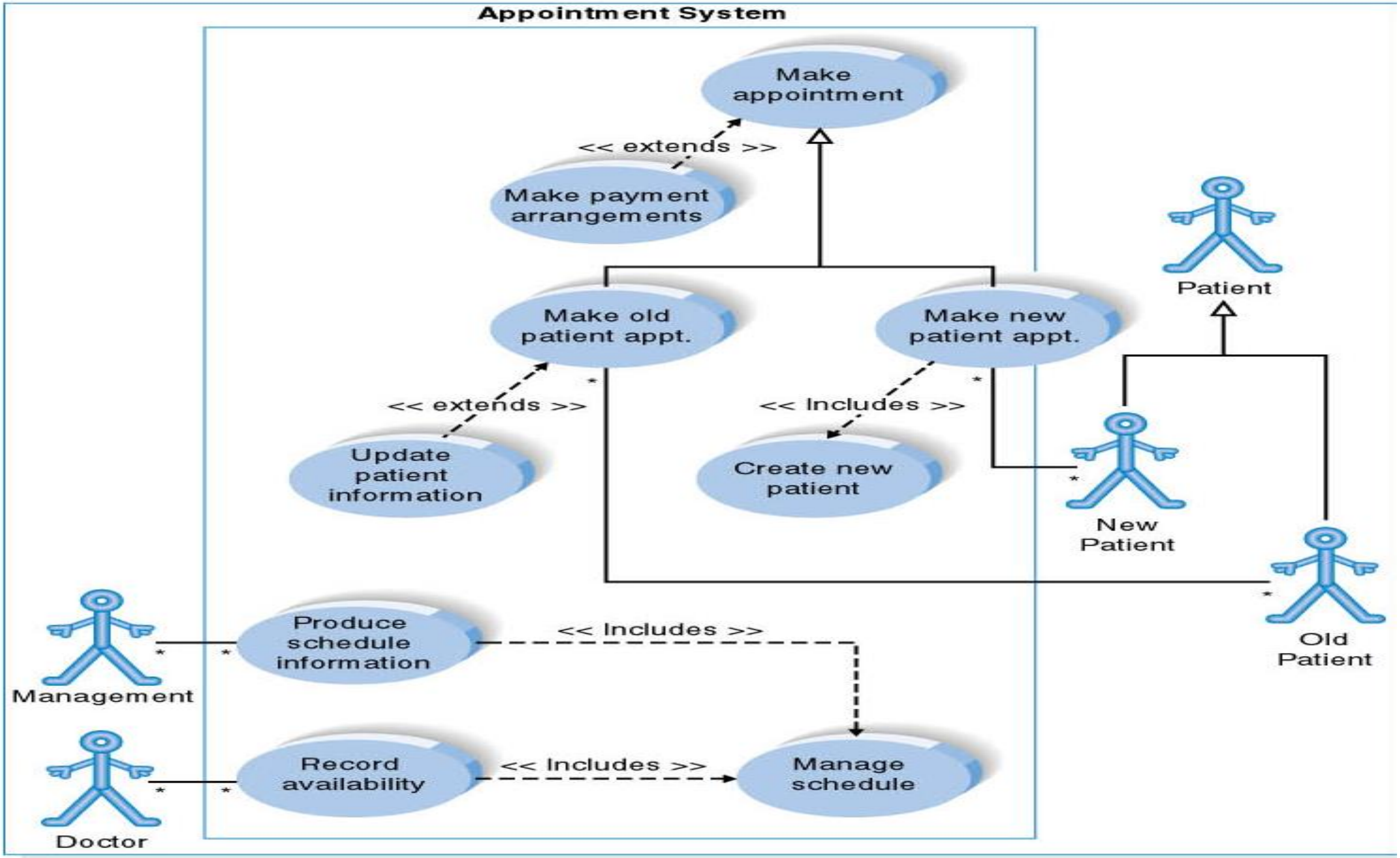
Use Case Diagram for Appointment System



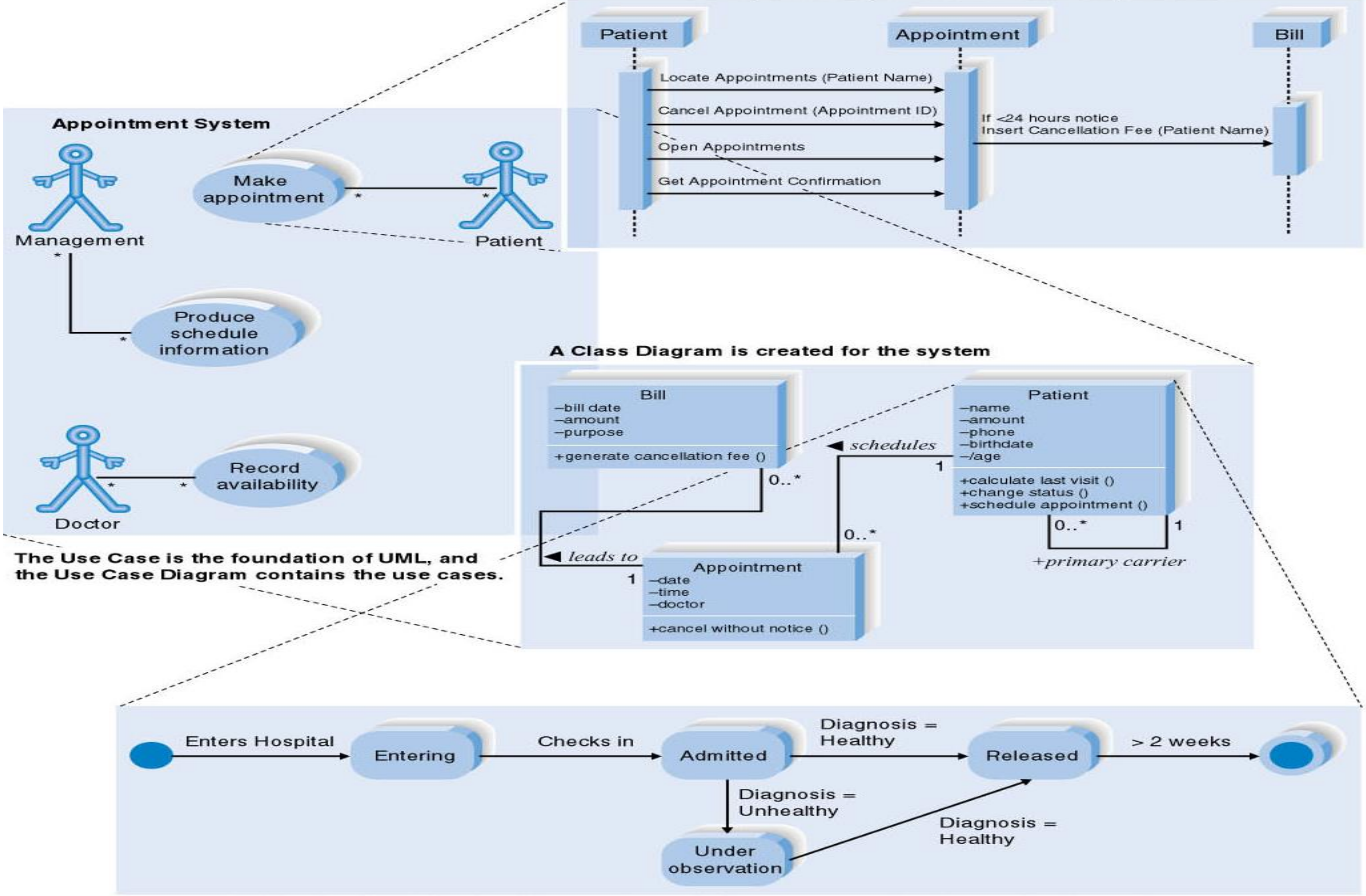
Use Case Diagram for Specialized Actor



Extends and Includes Associations



Dennis, Wixom, & Roth Systems Analysis and Design, 4th Edition Copyright 2009 © John Wiley & Sons, Inc.



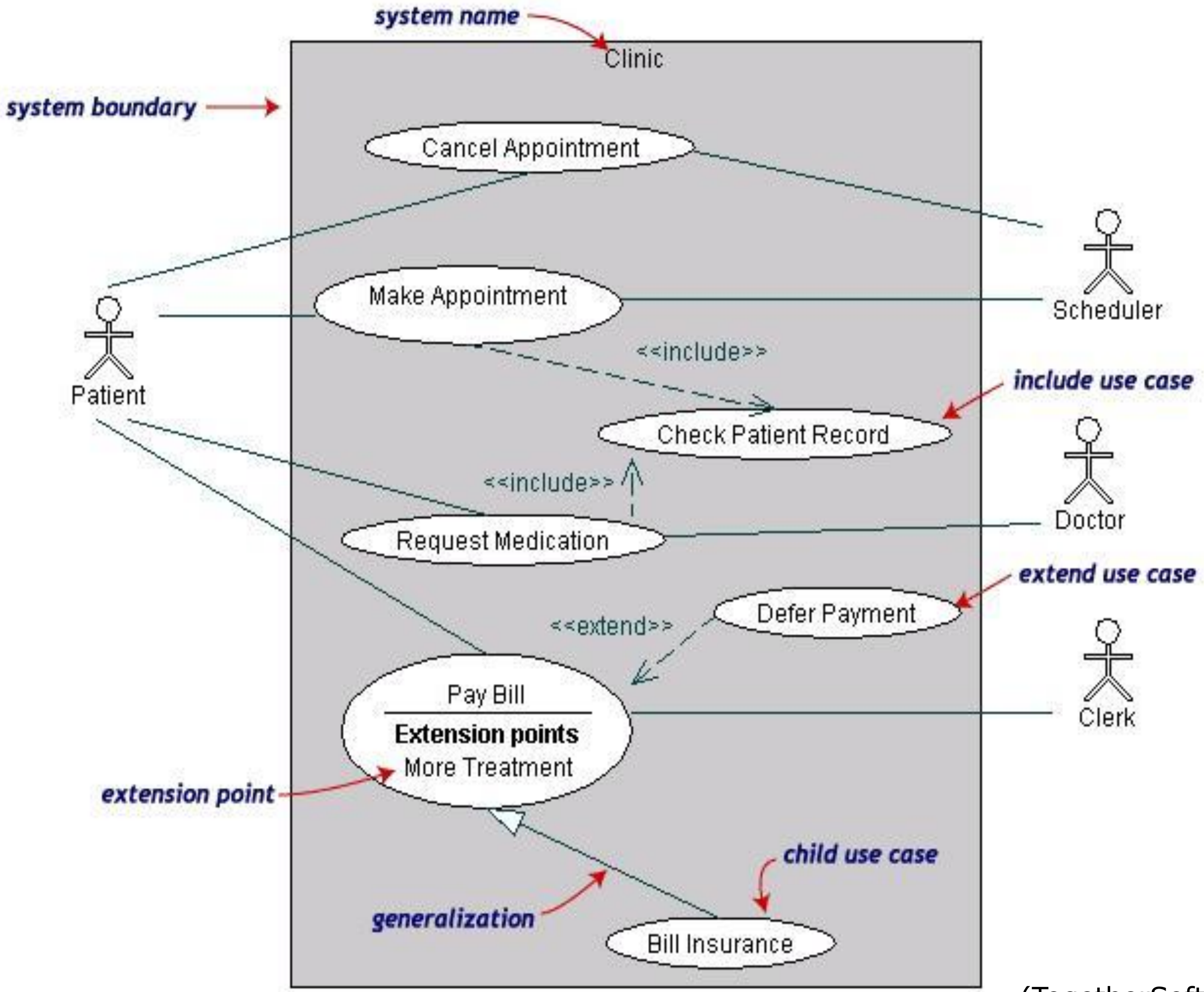
A Sequence Diagram is created for every use case.

The Use Case is the foundation of UML, and the Use Case Diagram contains the use cases.

A Class Diagram is created for the system

A Behavioral State Machine Diagram is created for every complex class on the Class Diagram.

Use Case Diagram for Hospital Management System



(TogetherSoft, Inc)

Both **Make Appointment** and **Request Medication** include **Check Patient Record** as a subtask (include)

The **extension point** is written inside the base case **Pay bill**; the extending class **Defer payment** adds the behavior of this extension point. (extend)

Pay Bill is a parent use case and **Bill Insurance** is the child use case. (generalization)

(TogetherSoft, Inc)