



SOFTWARE DESIGN (SWD392)

CH06 – USE CASE MODELING

- Requirement Modeling
- Use Cases
- Identifying Use Cases
- Documenting Use Cases
- Activity Diagrams

- The requirements of a system describe
 - What the user expects from the system
 - What the system will do for the user
- When defining the requirements of a system
 - The system should be viewed as a black box. Only the external characteristics of the system are considered
 - Both functional and nonfunctional requirements need to be considered
- Requirements modeling consists of requirements analysis and requirements specification

- The software requirements describe the functionality that the system must provide for the users.
- Requirements analysis involves analyzing the requirements by
 - Interviewing user
 - Analyzing the existing system(s)
 - Understanding and documenting the current system
 - Determining which features of the current system should be automated and which should remain manual
 - Discussing with users what functions could be done differently when the system is automated.

Requirement Modelling

Requirements Specification 1/2

- The document that needs to be agreed on by the requirements analysts and the users.
 - Starting point for the subsequent design & development
 - Both functional requirements & nonfunctional requirements need to be specified
- A functional req. describes the functionality the system must be capable of providing in order to fulfill the purpose of the system
 - Functionality the system needs to provide
 - Input to the system from the external environment
 - Output to the external environment
 - What stored information the system reads or updates
- A nonfunctional requirement: quality attribute, or quality-of-service goal that the system must fulfill

- Quality Attributes for a well-written Software Requirement Specification (**SRS**)
 - Correct
 - Complete
 - Unambiguous
 - Consistent
 - Verifiable
 - Understandable by non-computer specialists
 - Modifiable
 - Traceable

The ***use case model*** describes the functional requirements of the system in terms of the actors & use cases

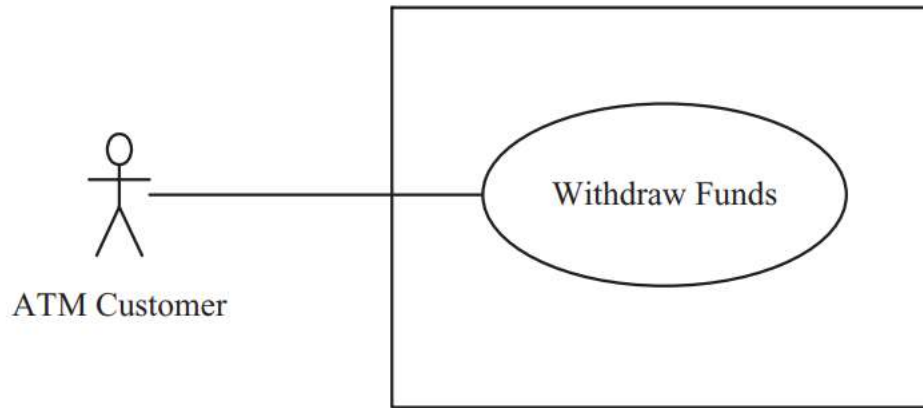
- The system is treated as a black box (dealing with ***what*** the system does in response to the actor's inputs)
- Functional requirements are described in terms of actors, which are users of the system, and use cases

A ***use case (UC)*** defines a sequence of interactions between one or more actors and the system

- A use case always starts with input from an actor
- Typically consists of a sequence of interactions between the actor and the system
- Each interaction consists of an input from the actor followed by a response from the system

An ***actor*** provides inputs to the system and the system provides responses to the actor

An automated teller machine (ATM) allows customers to withdraw cash from their bank accounts



The Withdraw Funds use case describes the sequence of interactions between the customer and the system

- The use case starts when the customer inserts an ATM card into the card reader,
- Then responds to the system's prompt for the PIN,
- And eventually receives the cash dispensed by the ATM machine

Emergency Monitoring System

As an example of a very simple use case, consider View Alarms from the Emergency Monitoring System. There is one actor, the Monitoring Operator, who can request to view the status of all alarms

Use case name: View Alarms

Summary: The monitoring operator views outstanding alarms.

Actor: Monitoring Operator

Main sequence:

1. The monitoring operator requests to view the outstanding alarms.
2. The system displays the outstanding alarms. For each alarm, the system displays the name of the alarm, alarm description, location of alarm, and severity of alarm (high, medium, low).

Alternative sequences:

Step 2: Emergency situation. System displays emergency warning message to operator.

An actor characterizes an external user (i.e., outside the system) that interacts with the system

- The only external entities that interact with the system
- Actors are outside the system and not part of it
- A user is an individual, whereas an actor represents the role played by all users of the same type
- There are other types of actors in addition to or in place of human actors: external systems, I/O devices, or timers

- Users
 - Homeowners are actors on HOLIS system
 - Authors are actors on a word processing system
- Other systems or applications
 - HOLIS Control Switch is an actor on the HOLIS Central Control Unit
- A device
 - Lights
 - Printer

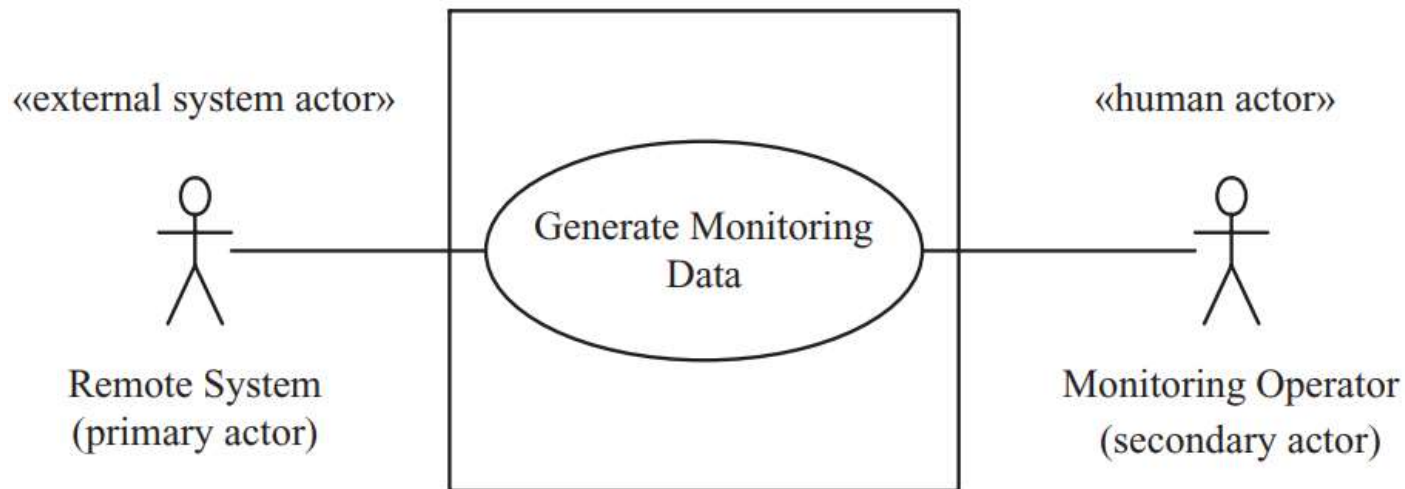
Following are some questions you might ask to help user representatives identify actors

- Who (or what) is notified when something occurs within the system?
- Who (or what) provides information or services to the system?
- Who (or what) helps the system respond to and complete a task?

Primary vs secondary actors

A **primary actor** initiates a use case. Thus, the use case starts with an input from the primary actor to which the system has to respond (gain value from the UC)

Other actors, referred to as **secondary actors**, can participate in the use case



Identifying Use Cases

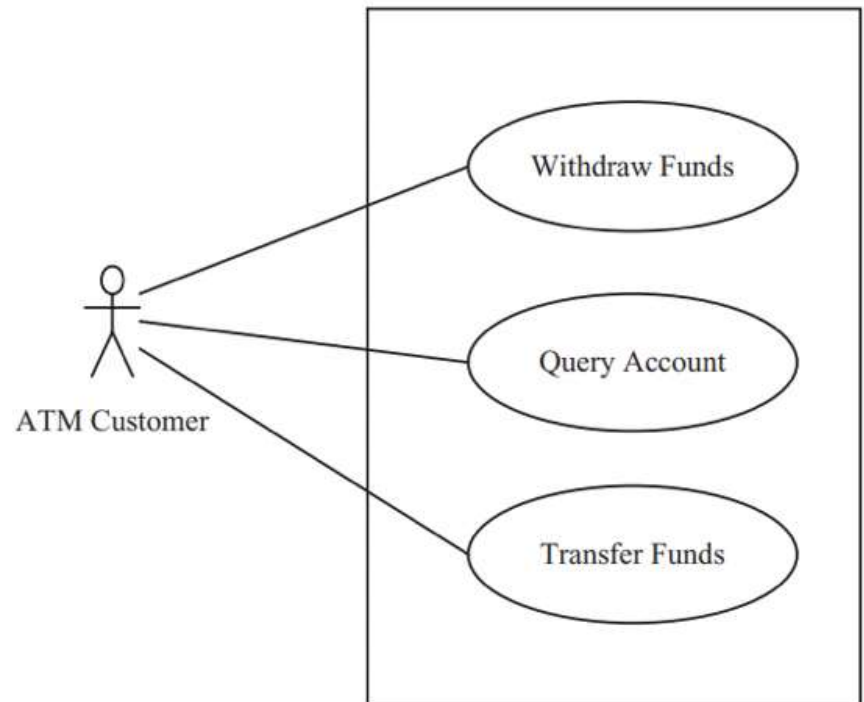
- A use case (UC) describes a sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value
- In this way, the functional requirements of the system are described in terms of the UCs, which constitute functional specification of a system.
- To determine the UCs in the system, it is useful to start by considering the actors and the interactions they have with the system.
- When developing UCs, it is important to avoid a functional decomposition in which several small UCs describe small individual functions of the system rather than describe a sequence of events that provides a useful result to the actor

- Questions to consider when identifying use cases
 - What will the actor use the system for?
 - Will the actor create, store, change, remove, or read data in the system?
 - Will the actor need to inform the system about external events or changes?
 - Will the actor need to be informed about certain occurrences in the system?
- The names of use cases are always written in the form of a verb followed by an object.
- Select strong, descriptive names to make it evident from the name that the use case will deliver something valuable for some user.

Identifying Use Cases

Simple Banking Example

- In addition to withdrawing cash from the ATM, the ATM Customer actor is also allowed to query an account or transfer funds between two accounts.
- Because these are distinct functions initiated by the customer with different useful results, the query and transfer functions should be modeled as separate UCs, rather than being part of the original UC



UC Spec

- Use case name
- Summary
- *Dependency*
- Actors
- Preconditions
- Main sequence
- Alternative sequences
- Postcondition

Use case name: Make Order Request

Summary: Customer enters an order request to purchase items from the online shopping system. The customer's credit card is checked for sufficient credit to pay for the requested catalog items.

Actor: Customer

Precondition: The customer has selected one or more catalog items.

Main sequence:

1. Customer provides order request and customer account Id to pay for purchase.
2. System retrieves customer account information, including the customer's credit card details.
3. System checks the customer's credit card for the purchase amount and, if approved, creates a credit card purchase authorization number.
4. System creates a delivery order containing order details, customer Id, and credit card authorization number.
5. System confirms approval of purchase and displays order information to customer.

Alternative sequences:

Step 2: If customer does not have account, the system creates an account.

Step 3: If the customer's credit card request is denied, the system prompts the customer to enter a different credit card number. The customer can either enter a different credit card number or cancel the order.

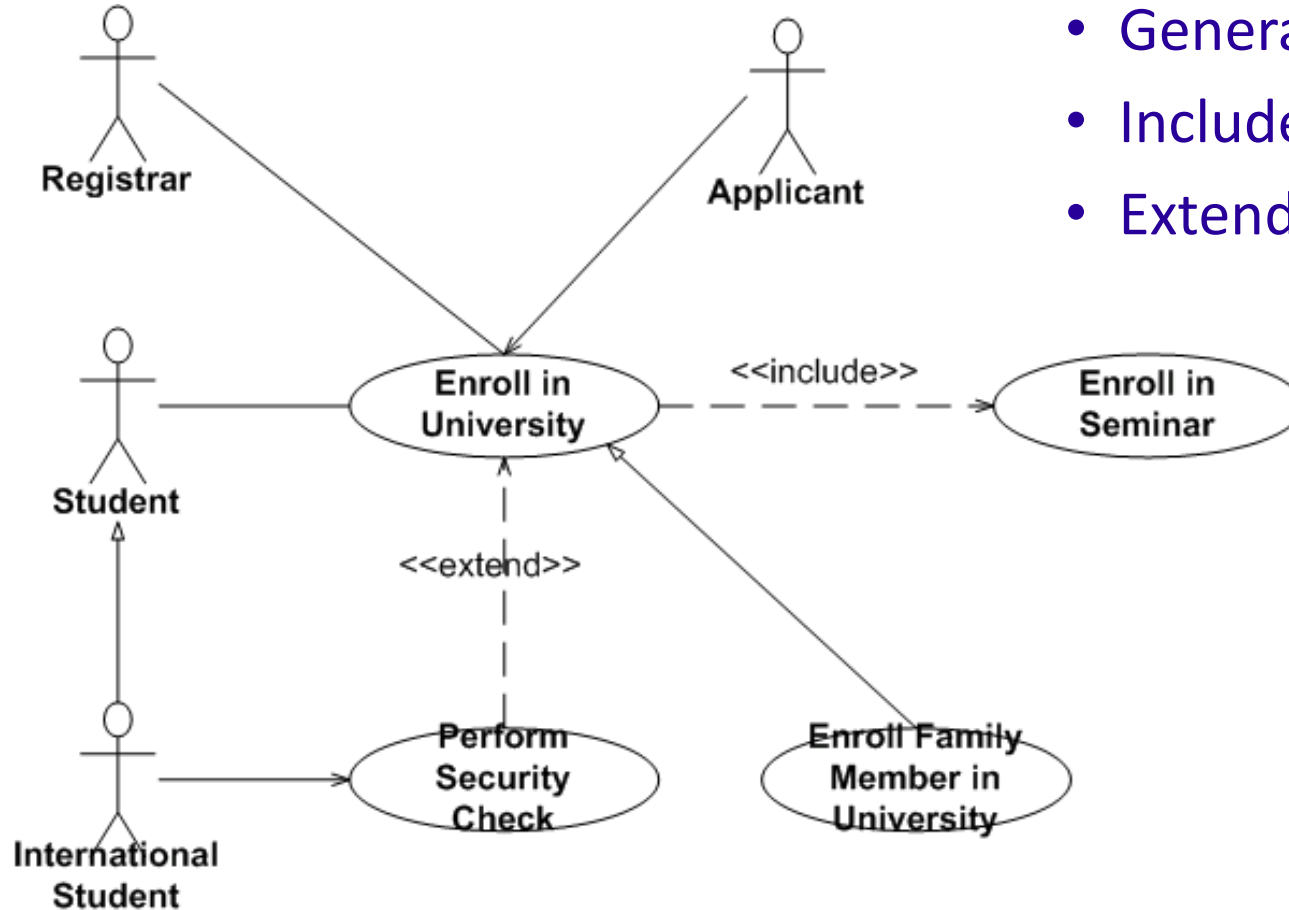
Postcondition: System has created a delivery order for the customer.

Documenting Use Cases

Use Case Relationships

UC Relationships

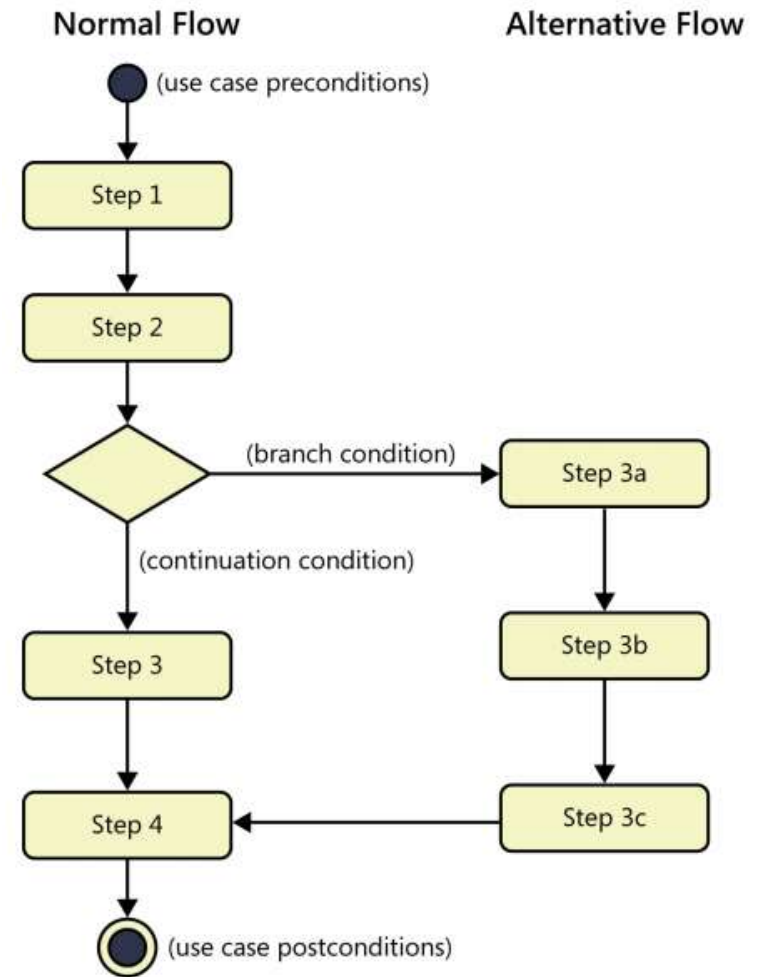
- Generalization
- Include: mandatory
- Extend: optional



Activity Diagrams 1/2

An activity diagram can be used to represent the sequential steps of a UC, including the main sequence and all the alternative sequences

- Depicting the flow of control & sequencing among activities
- Shows the sequence of activities, decision nodes, loops, and even concurrent activities



Activity Diagrams 2/2

- An activity node can be used to represent one or more sequential steps of the UC
- To depict a use case, a subset of the activity diagram capabilities is sufficient
- Activity diagrams can also be used to depict sequencing among UCs.

