

#### **DA-IICT**

## **ASE: Use Case Modeling**

Dr Saurabh Tiwari

1

## What is a Use Case?

- Created by Ivar Jacobson (1994)
- "A use case is a sequence of transactions in a system whose task is to yield a measurable value to an individual actor of the system"
- Describes WHAT the system (as a "Black Box") does from a user's (actor) perspective
- The Use Case Model is NOT an inherently object oriented modeling technique

#### **Benefits of Use Case**

- Captures operational requirements from user's perspective
- Gives a clear and consistent description of what the system should do
- A basis for performing system tests
- Provides the ability to trace functional requirements into actual classes and operations in the system

# UML Use Case Diagram

- A Use Case model is described in UML (Unified Modeling Language) as one or more Use Case Diagrams (UCDs)
- A UCD has 4 major elements:
  - The **system** described
  - The actors that the system interacts with
  - The use-cases, or services, that the system knows how to perform
  - The **relationships** between the above elements

# System

- As part of use-case modeling, the boundaries of the system developed must be defined
- · Defining the boundaries of the system is not trivial
  - Which tasks are automated and which are manual?
  - Which tasks are performed by other systems?
    - The entire solution that we supply should be included in the system boundaries
    - · Incremental releases

# System

- A system in a UCD is represented as a box
- The name of the system appears above or inside the box

Traffic Violations Report System

#### Actor

- Someone or something that interacts with the system (exchanges information with the system)
- An actor represents a role played with respect to the system, not an individual user of the system
- Example:
  - Policeman Enters data
  - Supervisor Allowed to modify/erase data
  - Manager Allowed to view statistics.
- A single user may play more than one role

#### Actor

- · Actors have goals:
  - Add a Traffic Violation
  - Lookup a Traffic Violation
- Actors don't need to be human
  - May be an external system that interfaces with the developed system
- · An actor has a name that reflects its role

# **Relationship between Actors**

- When several actors as part of their roles, also play a more generalized role, it is described as **generalization**
- The behavior of the general role is described in an actor superclass
- The specialized actors inherit the behavior of the super-class and extend it in some way
- · Relationships between actors are not always necessary



## **Use Case**

- Represent a complete behavior as perceived by an actor
  - A use case satisfies an actor's goal
- · Always initiated by an actor
- A use case is complete
  - Don't divide a use case into smaller use cases that implement each other (functional decomposition)

# **Use Case Description**

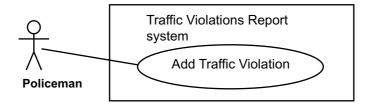
- The scenarios of a use case are normally described textually
  - A simple and consistent specification about how the actors and the system interact
  - Use case description template
- Describe at the level of user intentions and system responses
  - Free of technology and mechanism details, especially those related to user interface

# **Use Case Description Template**

- Name
  - Name of use case, usually close to the user's goal
  - Forward traceability (unique)
- Actors
- Goal description
- Reference to requirements
  - Backward traceability
- Pre-conditions
  - The necessary conditions before the use case can be performed
  - Could be other Use Cases as well
- Description
  - A description of the basic or normal course that should be taken by the system if the system should perform as intended

## **Use Case Icon**

- Use Case Icon
  - An ellipsis containing the name of the Use Case
  - Placed inside the boundaries of the modeled system
  - Connected to at least one actor with a communication association
    - Except for specialized / extending use cases.



# **Use Case Relationships**

- **Generalization**: A generalized Use Case describes the common of other specialized Use Cases.
- Inclusion: A Use Case is a part of another Use Case.
- Extension: A Use Case may extend another Use Case.

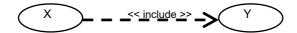
## **Generalization Relationships**

- Used when a number of Use Cases all have some subtasks in common, but each one has something different about it
- The generalized and specialized use cases share the same goal
- A specialized Use Case may capture an alternative scenario of the generalized Use Case
- The Specialized use case may interact with new actors.
- The Specialized use case may add pre-conditions and post-conditions (AND semantics).



# Include Relationships

- In older versions: "uses"
- When a number of Use Cases have common behavior, which can be modeled in a single use case
- X << includes >> Y indicates that the process of doing X always involves doing Y at least once
- The included Use Case must be complete
- X must satisfy the pre-conditions of Y before including it
- Not necessarily preserves the pre or post conditions.



# **Extend Relationships**

- Serves as extension point to another Use Case
- The extended Use Case must explicitly declare its extension points
- The extension conditions of the extended Use Case are part of the pre-conditions (AND semantics)



#### **Recommended Workflow**

- 1. Identify actors (and their relationships if necessary)
- For each actor identified and until no new UC is discovered do
  - a. Find all the goals of the actor
  - b. Decide on the main course of success for each goal
  - c. Create a Use Case for each of the goals
    - · New actors/goals may be discovered
  - d. Validate/correct existing Use Cases
- 3. Draw the Use Case diagram
  - Simplify model by repeating the process incase the produced diagram is too complex

# Question? - Design a use case diagram

A POS (Point-Of-Sale) system is a computer system typically used to manage the sales in retail stores. It includes hardware components such as a computer, a bar code scanner, a printer and also software to manage the operation of the store.

The most basic function of a POS system is to handle sales. When a customer arrives at a POS counter with goods to purchase, the cashier will start a new sale transaction. When the barcode of a good is read by the POS system, it will retrieve the name and price of this good from the backend catalog system and interact with inventory system to deduce the stock amount of this good.

When the sale transaction is over, the customer can pay in cash, credit card or even check. After the payment is successful, a receipt will be printed. Note that for promotion, the store frequently issue gift

coupons. The customer can use the coupons for a better price when purchasing goods.

Another function of a POS system is to handle returns.... [The details of which are not given here]

# Use Case Name Description Precondition Postcondition Postcondition Process gift coupon sale cincludes Alternate Flow J. Jacobson, M. Christerson, P. Jonsson and G. Övergraad. Object-Oriented Software Engineering: A Use-Case Driven Approach, Addison-Washer edition, 1997

Use Case – Fui		unctional Requirements	
uo	Use Case Name	Process Sale	
	Description	The Cashier records the purchased items and collects payment. The Catalog system displays the name and price of the good entered by the Cashier in his terminal. The Catalog system then interact with Inventory system to deduce the stock amount of this good.	

		sale session.
	Postcondition	Sale is saved. Receipt is printed. Stock data updated. Payment authorization approvals are recorded.
	Basic Flow (system happiness)	A Customer arrives at a POS checkout with goods to purchase. The Cashier uses the POS system to starts a new sale. The Cashier enters item identifier [A1][A2]. The Catalog system determines the item information, records sale line item and presents item description, price, and running total. The Cashier repeats the above steps until item entry is complete in POS. The Inventory system calculates and presents total price. The Cashier tells Customer the total, and asks for payment. The Customer pays and the Catalog system handle payment [A3][A4]. The Catalog system records completed sale and sends sale information to the external Inventory system for stock update [A5]. The Catalog system presents receipt and the Customer leaves with receipt and

The Customer arrives at a POST checkout with the items to purchase. The Cashier starts a new

Alternate Flow [A1] If there is more than one of the same item, the cashier can enter the quantity as well. The subtotal is shown
[A2] Invalid identifier entered. Indicate error.
[A3] The Catalog system process gift coupon , if applicable for the customer

[A4] The Customer does not have enough money. Cancel transaction
[A5] If the items stock gets below a predefined minimum place a reposition order

# **Question?**

**Use Case Textual Description** 

Precondition

A library maintains multiple reading materials which include books, journals, and magazines. The books are issued to the registered students and faculties of the institute, for a specified period of time. The issued books are to be returned back to the library. Delayed returns are subjected to stipulated fines. The issuereturn process is administered by one of the librarians who work in library."

