Fox School of Business
TEMPLE UNIVERSITY®

Information Technology Audit & Cyber Security

Use Cases

Systems & Infrastructure Lifecycle Management

OBJECTIVES

- Understand the process used to identify business processes and use cases.
- Understand the process used to create use-case diagrams
- Understand the process used to model business processes with activity diagrams.
- Understand the rules and style guidelines for activity diagrams.
- Understand the process used to create use case descriptions.
- Understand the rules and style guidelines for use case descriptions.
- Be able to create functional models of business processes using use-case diagrams, activity diagrams, and use case descriptions.

INTRODUCTION

Now begin the process of turning the requirements into functional models

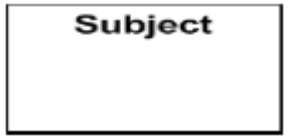
- Models are logical; i.e., independent of how they are implemented (manual or computerized)
- Develop use-cases from the requirements
  - Use-case: how a business system interacts with its environment
  - Includes a diagram and a description to depict the discrete activities that the users perform
- Develop activity diagrams from the use-cases
  - These model the business processes or how a business operates
  - Used to illustrate the movement of objects (data) between activities

ELEMENTS OF USE-CASES

Elements of Use-Case Diagrams

- Actors: users or other interacting systems
- Associations: lines to connect actors and use-cases
  - Interactions, inclusions, extensions or generalizations
- Use-case: a major process in the system that gives a benefit to the users
- Subject boundary: a named box that depicts the scope of the system
- An association relationship: links an actor with the use case(s) with which it interacts

Actor/Role



IDENTIFYING MAJOR USE-CASES

Review the requirements definition

Identify the subject's boundaries

Identify the primary actors and their goals

Identify the business processes and major use-cases

Carefully review the current set of use-cases

- Split or combine some to create the right size
- Identify additional use-cases

IDENTIFYING USE CASES

Use case— an activity that the system performs, usually in response to a request by a user

- Use cases define **functional** requirements
- Analysts decompose the system into a set of use cases (functional decomposition)
- Two techniques for Identifying use cases
  - User goal technique
  - Event decomposition technique

Name each use case using *Verb-Noun*

USER GOAL TECHNIQUE

Step One: *Identify* all possible users for the new "system"

Step Two: *Classify* all *users* into functional roles
- Ex, shipping, marketing, sales, etc.

Step Three: *Classify users* into organizational level
- Ex, operations, executive, professional, etc.

Step Four: *Interview* to find a list of specific goals
- Start with current goals, then imagine new functionality that would add value
- Use verb-noun phrasing
  - Add customer
  - Update order
  - Produce report

Step Five: *Create* use case list organized by user type

Step Six: *Look* for duplicate use cases and resolve inconsistencies

Step Seven: *Identify* where different types of users need the same use cases

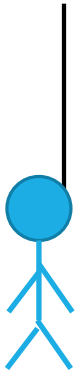
Step Eight: *Review* all use cases with each type of user and the stakeholders

EVENT DECOMPOSITION TECHNIQUE

More Comprehensive and Complete Technique

- Identify the events that occur to which the system must respond.
- For each event, name a use case (verb-noun) that describes what the system does when the event occurs

Event– something that occurs at a specific time and place, can be described, and should be remembered by the system



“Customer pays their bill…”
Record Payment

“Customer changes their address…”
Update Customer

“Time to send late notices…”
Send Late Notices

“Time to send statements…”
Produce Statements

“Time to produce end of week reports…”
Produce Reports

“Customer makes a charge…”
Process a Charge

TYPES OF EVENTS

External Event

- an event that occurs outside the system, usually initiated by an external agent or actor

Temporal Event

- an event that occurs as a result of reaching a point in time

State Event

- an event that occurs when something happens inside the system that triggers some process
- reorder point is reached for inventory item

EXTERNAL EVENT CHECKLIST

External agent or actor wants something resulting in a transaction

- Customer buys a product

External agent or actor wants some information

- Customer wants to know product details

External data changed and needs to be updated

- Customer has new address and phone

Management wants some information

- Sales manager wants update on production plans

TEMPORAL EVENT CHECKLIST

Internal outputs needed at points in time

- Management reports (summary or exception)
- Operational reports (detailed transactions)
- Internal statements and documents (including payroll)

External outputs needed at points of time

- Statements, status reports, bills, reminders

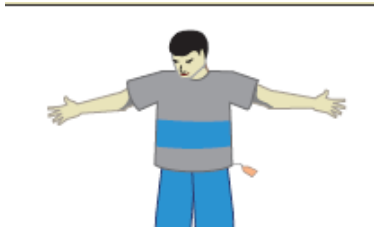
FINDING THE EVENT

Customer thinks about getting a new shirt

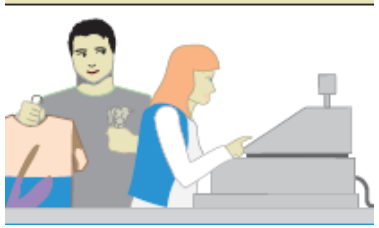
Customer drives to the mall

Customer tries on a shirt at Sears

Customer goes to Sprawl Mart



Customer tries on a shirt at Sprawl Mart



Customer buys a shirt

TRACING SEQUENCE OF TRANSACTIONS (TO FIND EVENTS)

Customer requests a catalog

Customer wants to check item availability

Customer places an order

Customer changes or cancels order

Customer checks order status

Customer updates their account

Customer returns the item

“PERFECT TECHNOLOGY ASSUMPTION”

Don’t worry about functions built into system because of limits in technology and people. Concentrate on events salient to the business process… these come later.

- Log in
- Change password
- Change preferences
- System error handling
- Backups
- Security

EVENT DECOMPOSITION STEPS

1. Consider the external events in the system environment that require a response from the system
2. For each external event, identify and name the use case that the system requires
3. Consider the temporal events that require a response from the system
4. For each temporal event, identify and name the use case that the system requires and then establish the point of time that will trigger the use case
5. Consider the state events that the system might respond to, particularly if it is a real-time system in which devices or internal state changes trigger use cases.
6. For each state event, identify and name the use case that the system requires and then define the state change.
7. When events and use cases are defined, check to see if they are required by using the perfect technology assumption. *Do not include events that involve such system controls as login, logout, change password, and backup or restore the database, as these are put in later.*

WHICH IS BETTER? (USER GOAL VS EVENT DECOMPOSITION)

Events are broader than user goal: Capture temporal and state events

Help decompose at the right level of analysis: an elementary business process (EBP)

- *EBP is a fundamental business process performed by one person, in one place, in response to a business event*

Uses perfect technology assumption to make sure functions that support the users work are identified and not additional functions for security and system controls

CRUD TECHNIQUE

Often introduced in database context

Technique to validate, refine or cross-check use cases

NOT for primarily identifying use cases

For each domain class, verify there are use cases for CRUD

- "Domain Class" – Fancy way of saying a type of user that will interact with the system.

- Example:

Domain Class	CRUD	Use Case
Customer	Create	Delete Customer
	Read/Report	Look up customer Produce customer report
	Update	Process account adjustment Update customer account
	Delete	Update customer account (to archive)

CRUD STEPS

1. Identify all the data entities or domain classes involved in the new system. (more on this later…)
2. For each type of data (data entity or domain class), verify that a use case has been identified that creates a new instance, updates existing instances, reads or reports values of instances, and deletes (archives) an instance.
3. If a needed use case has been overlooked, add a new use case and then identify the stakeholders.
4. With integrated applications, make sure it is clear which application is responsible for adding and maintaining the data and which system merely uses the data

USE CASE DESCRIPTION

1. Once you have your use cases, provide a list with a brief description describing the basic steps to complete the use case.
2. Example:
    1. Create Customer Account
        1. User enter new customer account data, and the system assigns account number, creates a customer record, and creates an account record
    2. Look up customer
        1. User enters customer account number, and the system retrieves and displays customer and account data
    3. Process account adjustment
        1. User enters order number, and the system retrieves customer and order data, actor enters adjustment amount, and the system creates a transaction record for the adjustment

USE CASE DIAGRAMS

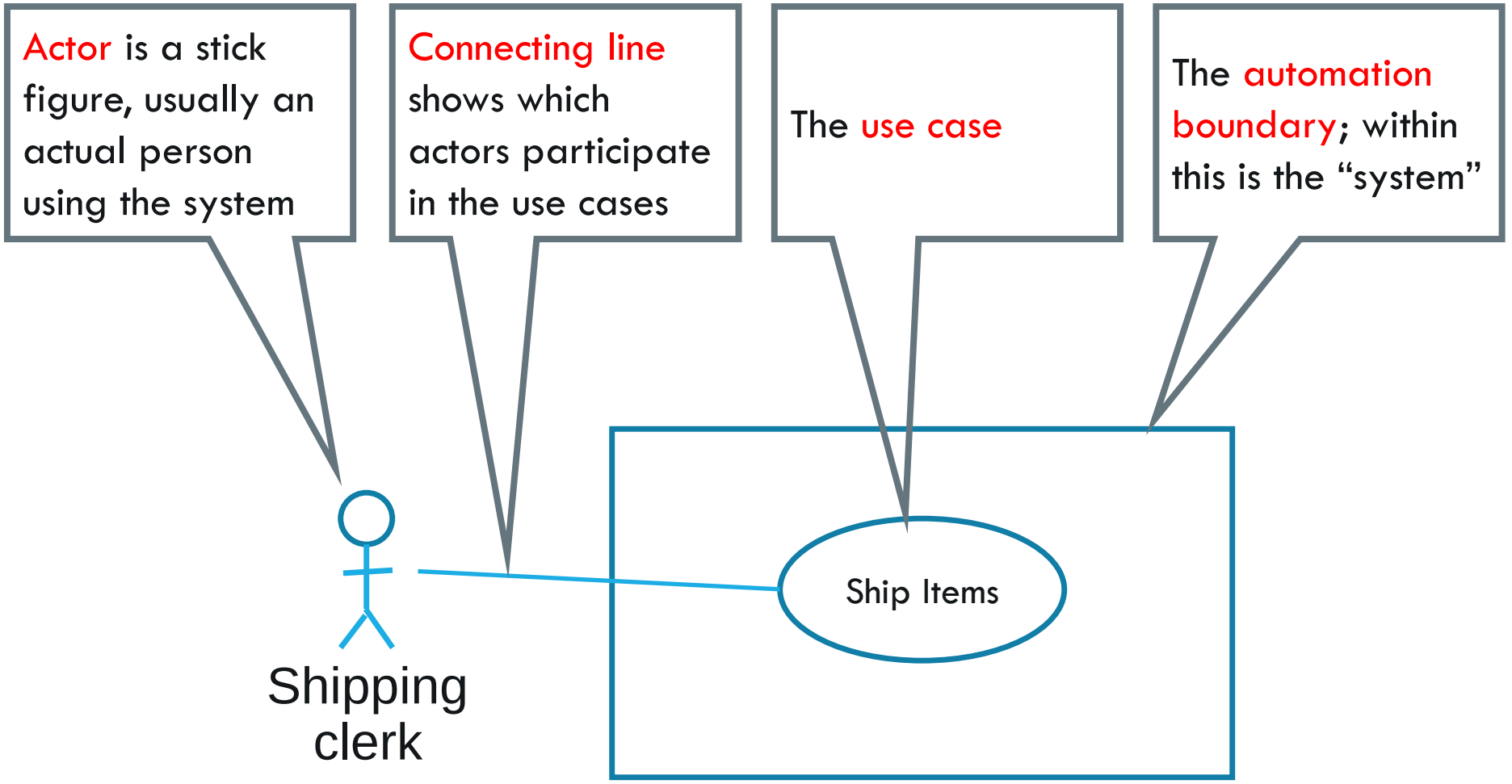
Use case diagram— a UML model used to graphically show uses cases and their relationships to actors

Recall UML is Unified Modeling Language, the **standard** for diagrams and terminology for developing information systems

Actor is the UML name for a end user

Automation boundary— the boundary between the computerized portion of the application and the users who operate the application

USE CASE DIAGRAM STANDARD (UML)



USE CASE DIAGRAMMING STEPS

1. Identify all the stakeholders and users who would benefit by seeing a use case diagram
2. Determine what each stakeholder or user needs to review in a use case diagram: each subsystem, for each type of user, for use cases that are of interest
3. For each potential communication need, select the use cases and actors to show and draw the use case diagram. There are many software packages that can be used to draw use case diagrams
4. Carefully name each use case diagram and then note how and when the diagram should be used to review use cases with stakeholders and users

TYPES OF USE CASES

Purpose		Amount of information: Overview	Amount of information: Detail
	Essential	High-level **overview** of issues **essential** to understanding required functionality	**Detailed** description of issues **essential** to understanding required functionality
	Real	High-level **overview** of a specific set of steps performed on the **real** system once implemented	**Detailed** description of a specific set of steps performed on the **real** system once implemented

ELEMENTS OF A USE CASE DESCRIPTION

- Overview:
  - Name, ID Number, Type, Primary Actor, Brief Description, Importance Level, Stakeholder(s), Trigger(s)
- Relationships:
  - Association: Communication between the use case and the actors
  - Extend: Extends the functionality of a use case
  - Include: Includes another use case
  - Generalization: Allows use cases to support inheritance
- Flow of events
  - Normal flow: the usual set of activities
  - Sub-flows: decomposed normal flows to simplify the use-case
  - Alternate or exceptional flows: those not considered the norm
- Optional characteristics (complexity, time, etc.)

USE CASE WRITING GUIDELINES

1. Write in the form of subject-verb-direct object
2. Make sure it is clear who the initiator of the step is
3. Write from independent observer’s perspective
4. Write at about the same level of abstraction
5. Ensure the use case has a sensible set of steps
6. Apply the KISS principle liberally.
7. Write repeating instructions after the set of steps to be repeated

CREATING USE-CASE DESCRIPTIONS

1. Pick a high priority use-case and create an overview:
   - List the primary actor
   - Determine its type (overview or detail; essential or real)
   - List all stakeholders and their interests
   - Determine the level of importance of the use-case
   - Briefly describe the use-case
   - List what triggers the use-case
   - List its relationship to other use-cases
2. Fill in the steps of the normal flow of events required to complete the use-case
3. Ensure that the steps listed are not too complicated or long and are consistent in size with other steps
4. Identify and write the alternate or exceptional flows
5. Carefully review the use-case description and confirm that it is correct
6. Iterate over the entire set of steps again

EXAMPLE USE-CASE DESCRIPTION

Use Case Name: Borrow Books	ID: 2	Importance Level: High
Primary Actor: Borrower	Use Case Type: Detail, Essential	

Stakeholders and Interests:
Borrower - wants to check out books
Librarian - wants to ensure borrower only gets books deserved

Brief Description: This use case describes how books are checked out of the library.

Trigger: Borrower brings books to check out desk.
Type: External

Relationships:
Association: Borrower, Personnel Office, Registrar's Office
Include:
Extend:
Generalization:

Normal Flow of Events:
1. The Borrower brings books to the Librarian at the check out desk.
2. The Borrower provides Librarian their ID card.
3. The Librarian checks the validity of the ID Card.
   If the Borrower is a Student Borrower, Validate ID Card against Registrar's Database.
   If the Borrower is a Faculty/Staff Borrower, Validate ID Card against Personnel Database.
   If the Borrower is a Guest Borrower, Validate ID Card against Library's Guest Database.
4. The Librarian checks whether the Borrower has any overdue books and/or fines.
5. The Borrower checks out the books.

SubFlows:

Alternate/Exceptional Flows:
4a. The ID Card is invalid, the book request is rejected.
5a. The Borrower either has overdue books, fines, or both, the book request is rejected.

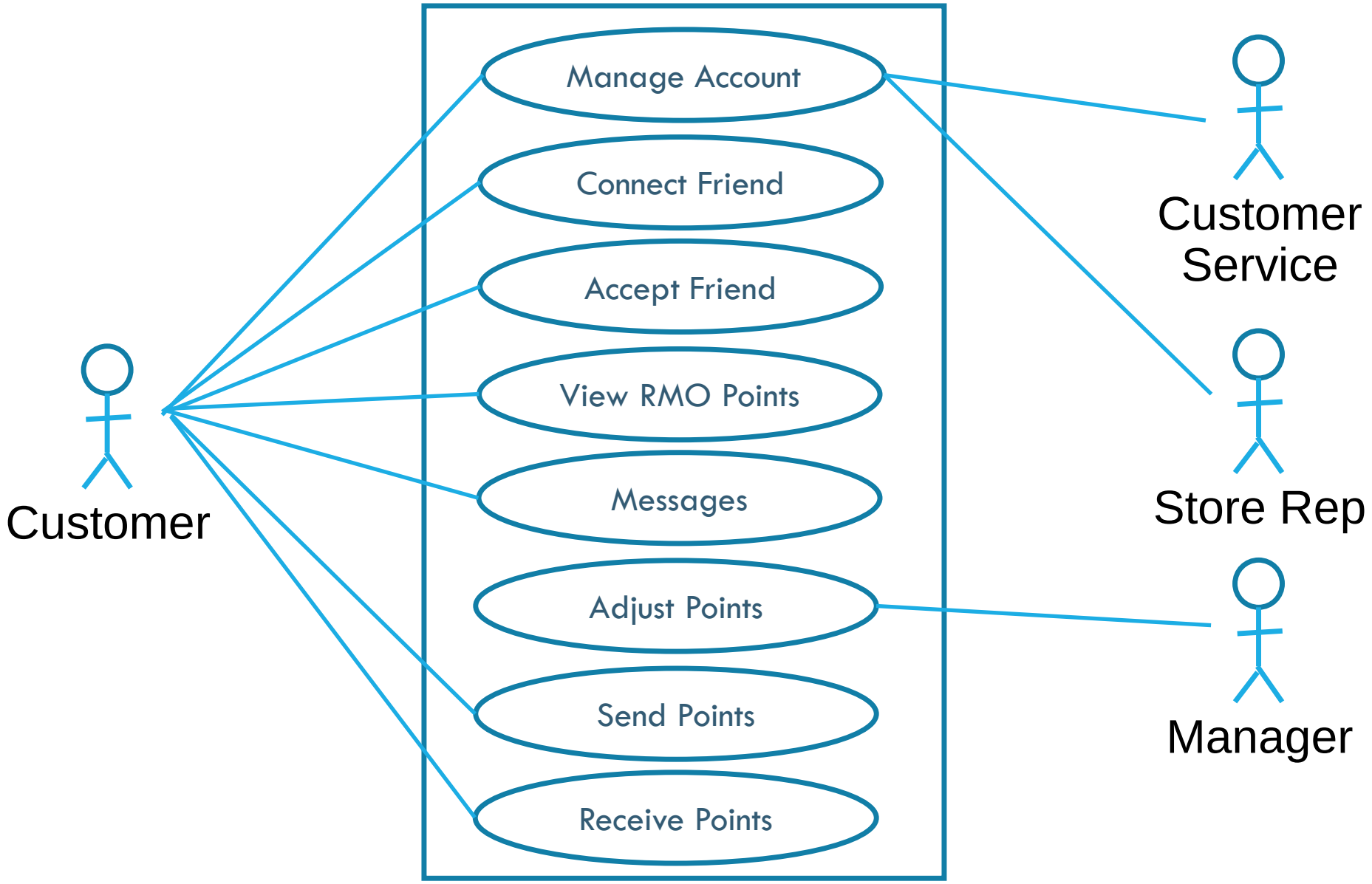
VERIFYING & VALIDATING A USE-CASE

Use-cases must be verified and validated before beginning structural and behavioral modeling

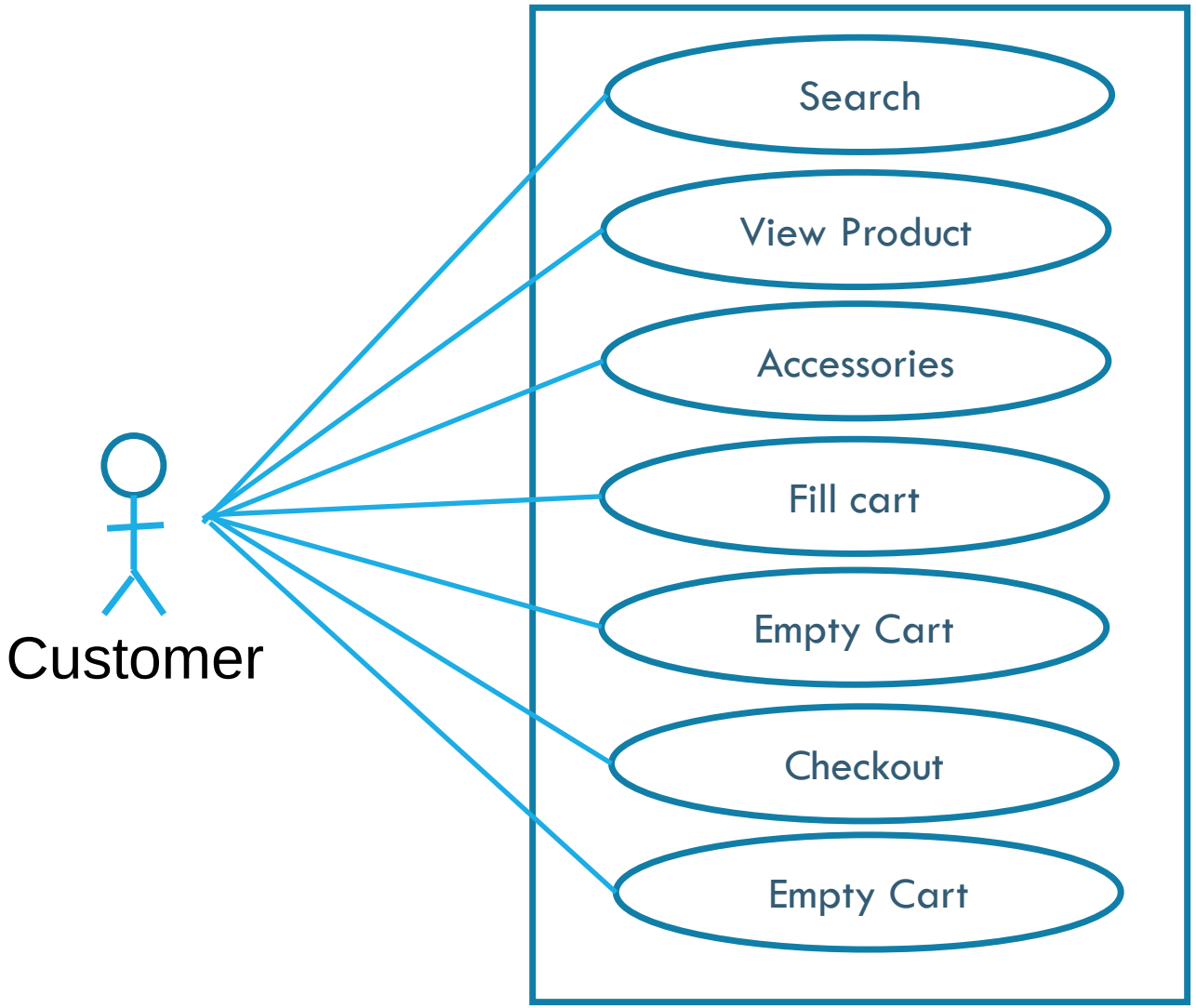
Utilize a walkthrough:

- Perform a review of the models and diagrams created so far
- Performed by individuals from the development team and the client (very interactive)
  - Facilitator: schedule and set up the meeting
  - Presenter: the one who is responsible for the specific representation being reviewed
  - Recorder (scribe) to take notes and especially to document errors

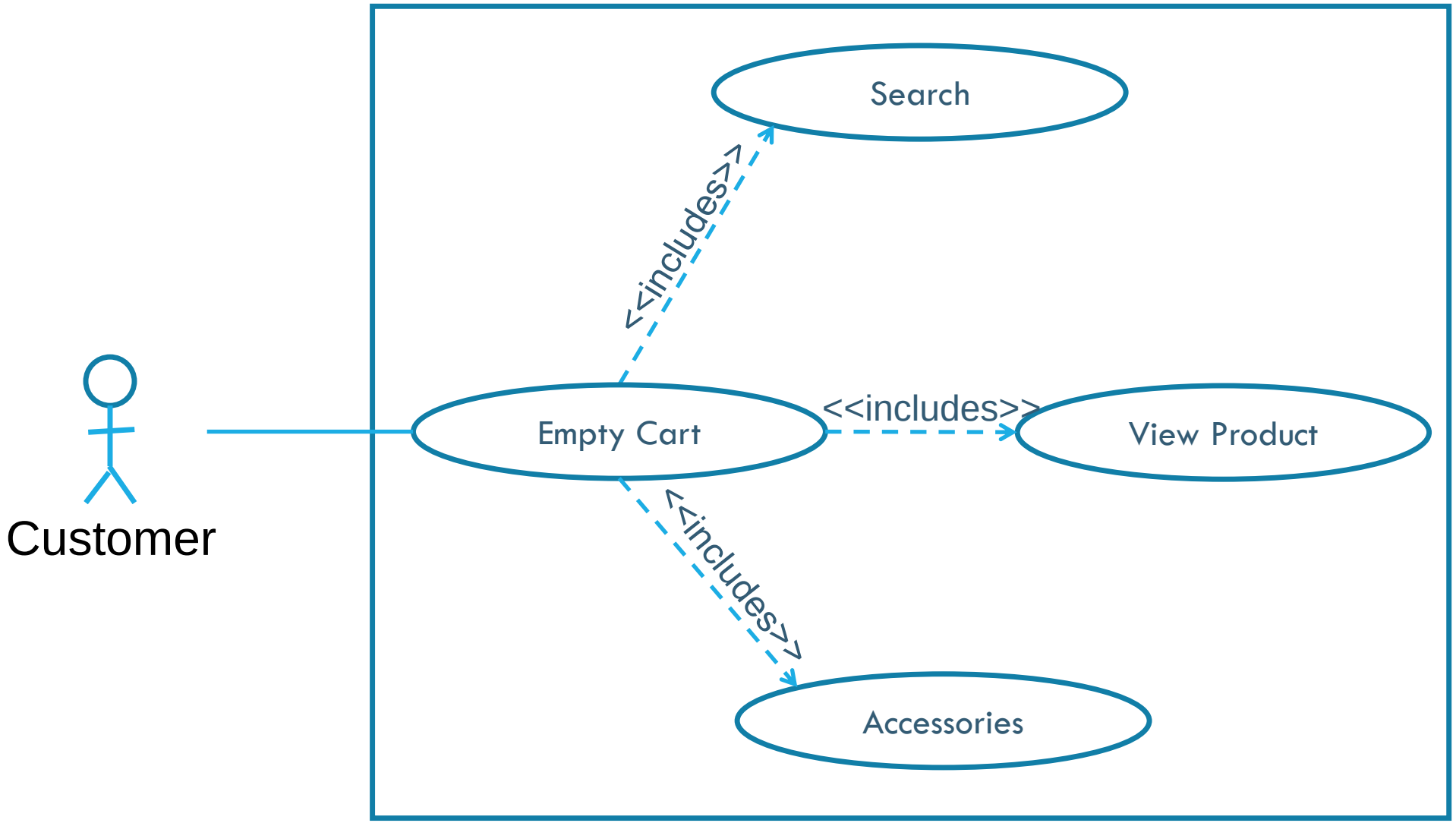
EXAMPLE


Manage Account
Connect Friend
Accept Friend
View RMO Points
Messages
Adjust Points
Send Points
Receive Points
Customer
Customer Service
Store Rep
Manager

EXAMPLE USE CASE

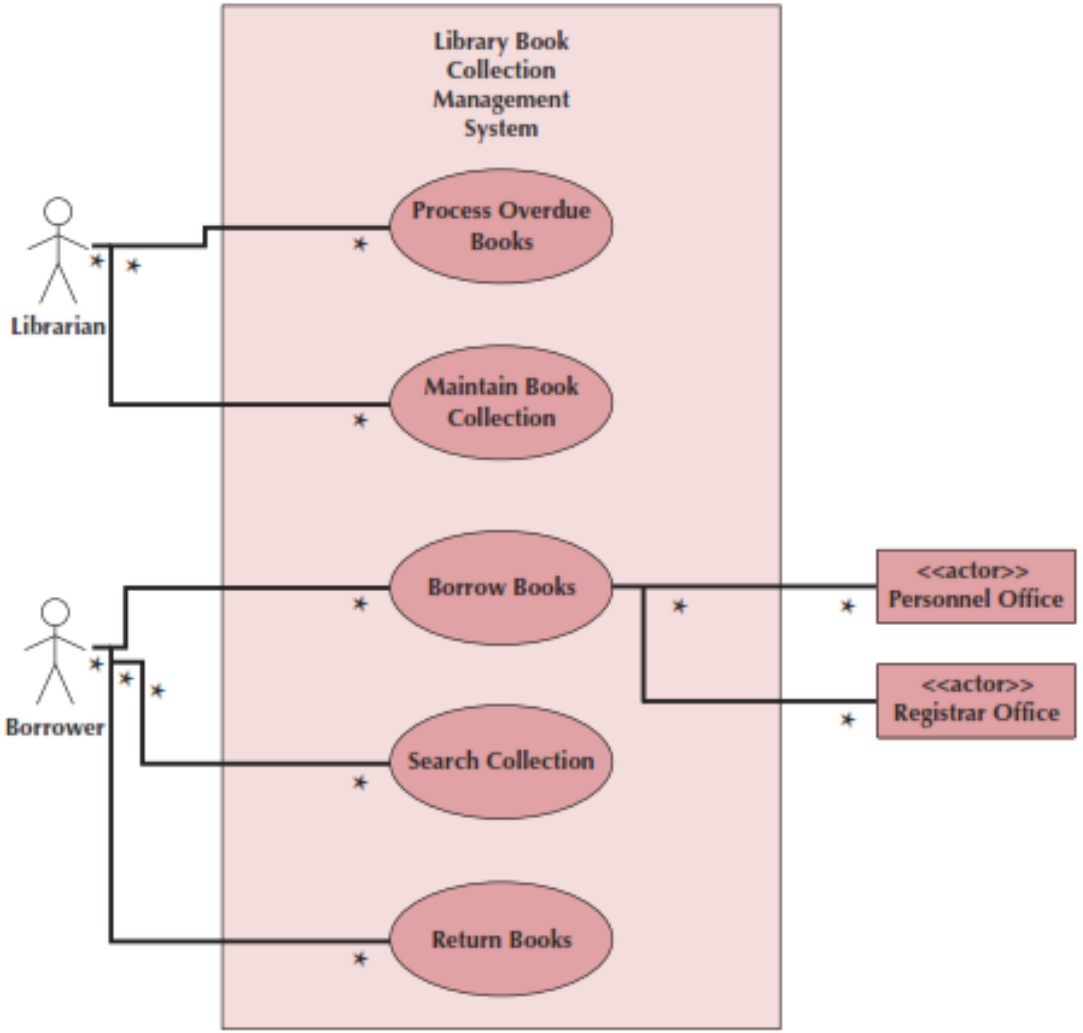


ADVANCED EXAMPLE...



EXAMPLE USE-CASE

Library Book Collection Management System Use Case Diagram



BPM WITH ACTIVITY DIAGRAMS

Business processes consist of a number of activities

Activity diagrams depict the sequence of these activities

- Diagrams are abstract and describe processes in general
- They model behavior independent of objects
- Can be used for any type of process

ACTIVITY DIAGRAM SYNTAX

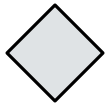
- Action or Activity
  - Represents action or set of actions

- Control Flow
  - Shows sequence of execution

- Initial Node
  - The beginning of a set of actions

- Final Node
  - Stops all flows in an activity

- Decision Node
  - Represents a test condition



ELEMENTS OF AN ACTIVITY DIAGRAM

Actions & Activities

- Something performed for some specific business reason
- Named with a verb and a noun (e.g., Get Patient Information)
- Activities can be further sub-divided; actions cannot

Object Nodes: represent the flow of information from one activity to another

Control Flows: model execution paths

Object Flows: model the flow of objects

Control Nodes: 7 types

SEVEN TYPES OF CONTROL NODES

Initial node: the beginning of the set of actions/activities

Final-activity node: stops all actions/activities

Final-flow node: stops one execution path but allows others to continue

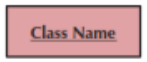
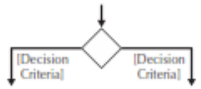
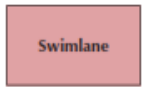
Decision node: represents a test to determine which path to use to continue (based on a guard condition)

Merge node: rejoins mutually exclusive execution paths

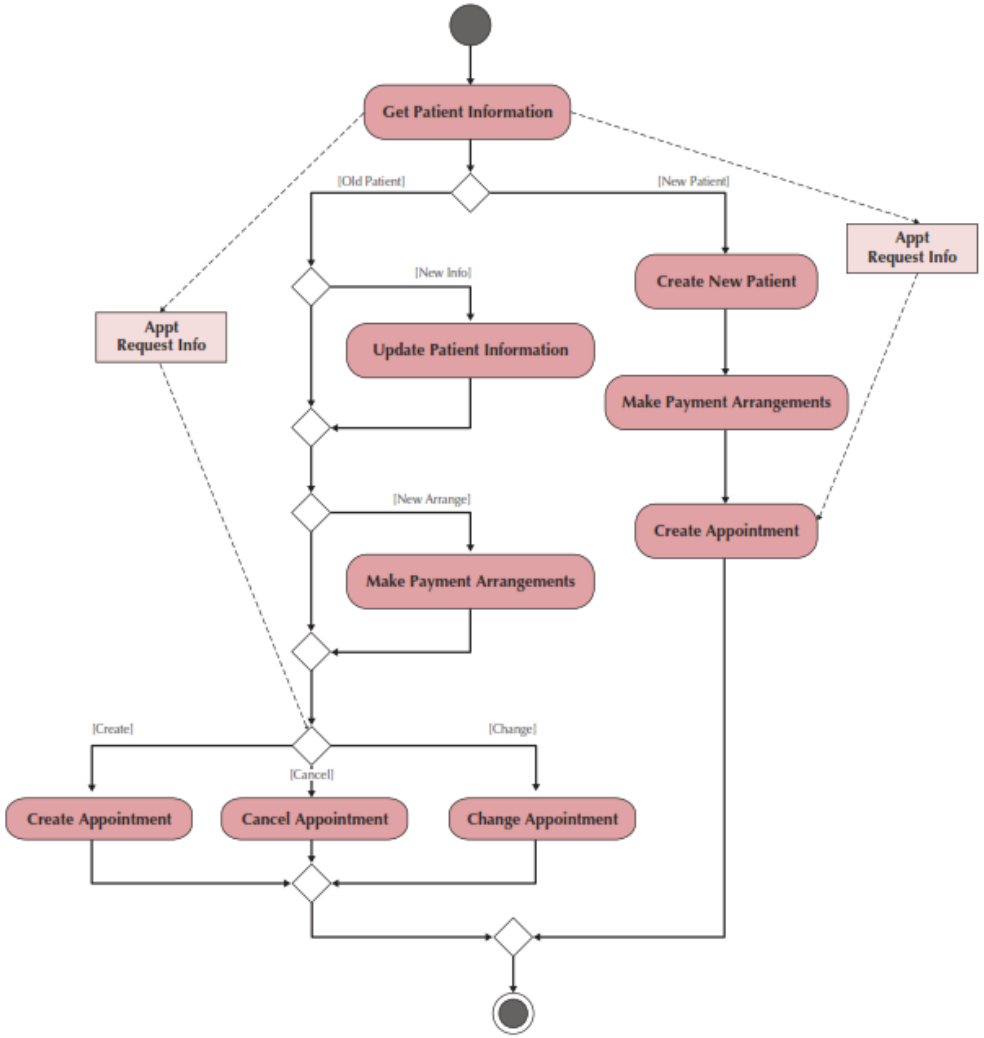
Fork node: separates a single execution path into one or more parallel paths

Join node: rejoins parallel execution paths

ACTIVITY DIAGRAM SYMBOLS

An action: ■ Is a simple, nondecomposable piece of behavior. ■ Is labeled by its name.	
An activity: ■ Is used to represent a set of actions. ■ Is labeled by its name.	
An object node: ■ Is used to represent an object that is connected to a set of object flows. ■ Is labeled by its class name.	
A control flow: ■ Shows the sequence of execution.	
An object flow: ■ Shows the flow of an object from one activity (or action) to another activity (or action).	
An initial node: ■ Portrays the beginning of a set of actions or activities.	
A final-activity node: ■ Is used to stop all control flows and object flows in an activity (or action).	
A final-flow node: ■ Is used to stop a specific control flow or object flow.	
A decision node: ■ Is used to represent a test condition to ensure that the control flow or object flow only goes down one path. ■ Is labeled with the decision criteria to continue down the specific path.	
A merge node: ■ Is used to bring back together different decision paths that were created using a decision node.	
A fork node: Is used to split behavior into a set of parallel or concurrent flows of activities (or actions)	
A join node: Is used to bring back together a set of parallel or concurrent flows of activities (or actions)	
A swimlane: Is used to break up an activity diagram into rows and columns to assign the individual activities (or actions) to the individuals or objects that are responsible for executing the activity (or action) Is labeled with the name of the individual or object responsible	

SAMPLE ACTIVITY DIAGRAM

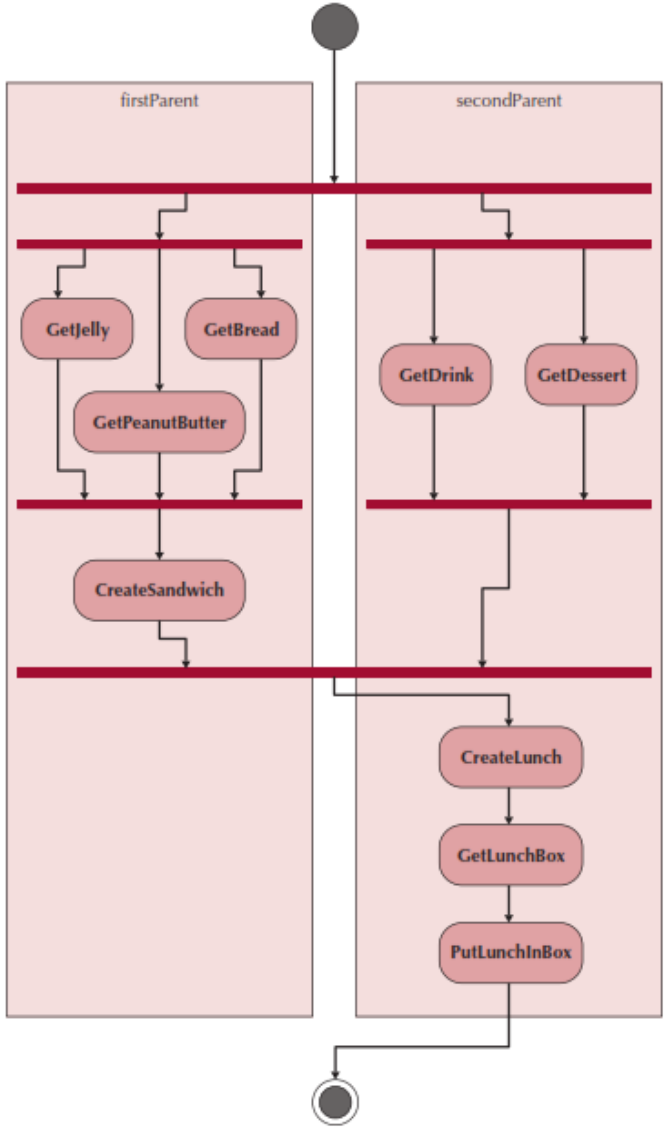


SWIM LANES

Used to assign responsibility to objects or individuals who actually perform the activity

Represents a separation of roles among objects

Can be drawn horizontally or vertically



GUIDELINES FOR ACTIVITY DIAGRAMS

1. Set the scope of the activity being modeled
2. Identify the activities; connect them with flows
3. Identify any decisions that must be made
4. Identify potential parallelism in the process
5. Draw the activity diagram

CREATING AN ACTIVITY DIAGRAM

Choose a business process identified previously

- Review the requirements definition and use-case diagram
- Review other documentation collected thus far

Identify the set of activities used in the business process

Identify control flows and nodes

Identify the object flows and nodes

Lay out & draw the diagram (minimize crossing lines)

SUMMARY

Presented in this unit:

- The identification of business processes using use-case diagrams and descriptions
- Modeling business processes with activity diagrams
- How to create the documentation of use-cases and use-case descriptions
- How to verify and validate the business processes and functional models