SLIIT UNI
THE KNOWLEDGE UNIVERSITY

Lecture 5: Activity Diagrams

IT2030 – Software Engineering

Content

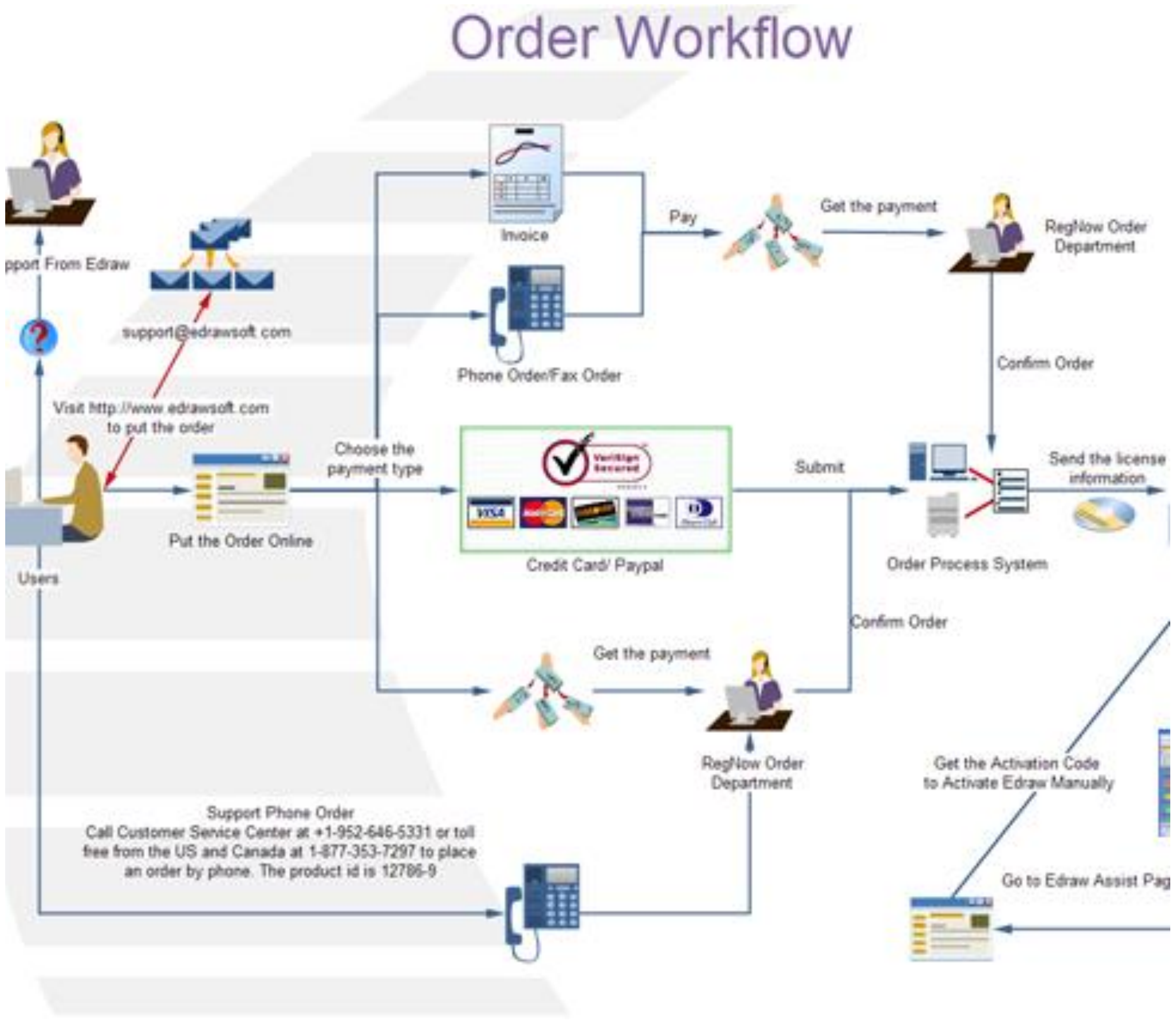
- Introduction.
- Elements of Activity diagram.
- Partitioning an activity diagram.
- Applying activity diagrams in real world applications.

Lesson Learning Outcomes

- Define activity diagrams and explain their purpose in modeling workflows and system behavior.
- Identify the main elements of an activity diagram, including actions, transitions, decision/merge nodes, forks/joins, and swimlanes.
- Differentiate between branching, merging, forking, and joining in activity diagrams.
- Interpret activity diagrams to describe the flow of activities for a given business or system process.
- Create accurate activity diagrams from textual scenarios, applying correct UML notation.

What is an Activity Diagram?

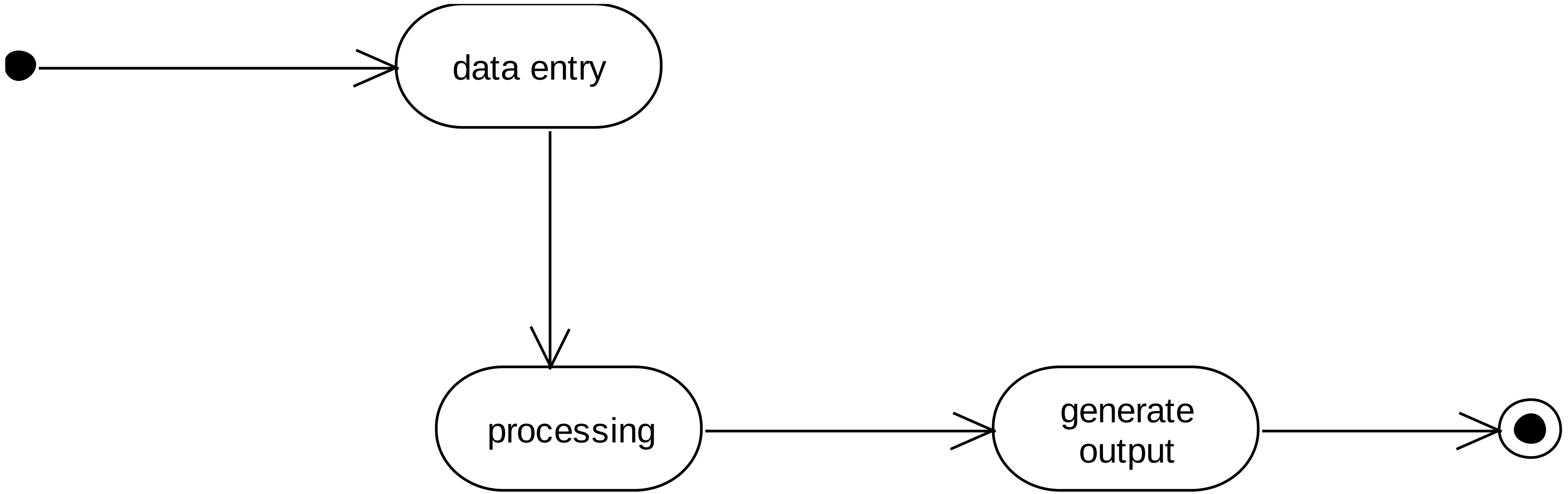
- An Activity Diagram models
  - Activities of a system.
  - Dependencies between activities.
  - Represents Workflows.



When to Use Activity Diagrams ?

- Modeling business processes.
- Analyzing system functionality to identify the use cases.
- Analyzing individual use cases in detail.
- Clarifying concurrency issues.

A Simple Activity Diagram

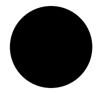
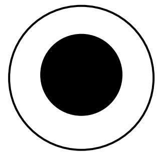


Elements of an Activity Diagram

- Action
  - Simple Action
  - Call Action
- Transition
- Controls
  - Initial/Start
  - Final/End
  - Nodes
    - Decision/Branch
    - Merge
    - Fork
    - Join

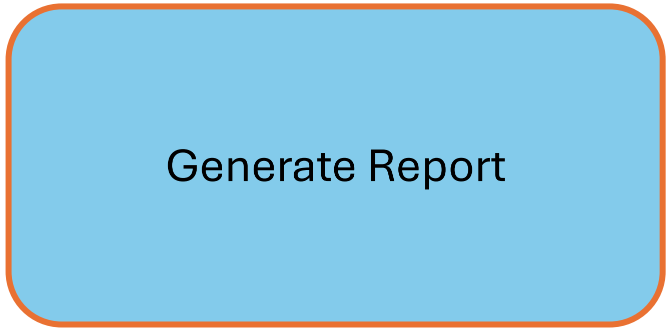
Initial & Final Nodes

Represent the beginning and end of a diagram respectively.

- Initial/Start Node – Filled Circle 
- Final node is within circle 

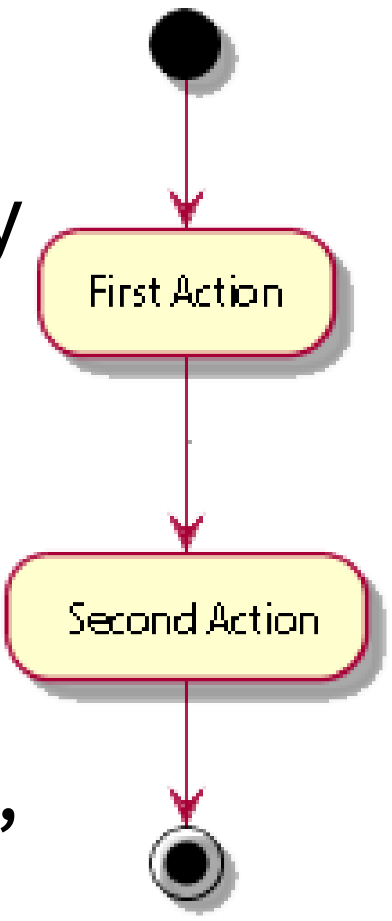
Action Node

- An action is some task which needs to be done.
- Each action can be followed by another action .
- Represented with a rounded rectangle.
- Text in the action box should represent an activity (verb phrase in present tense).
- An activity is a sequence of actions.

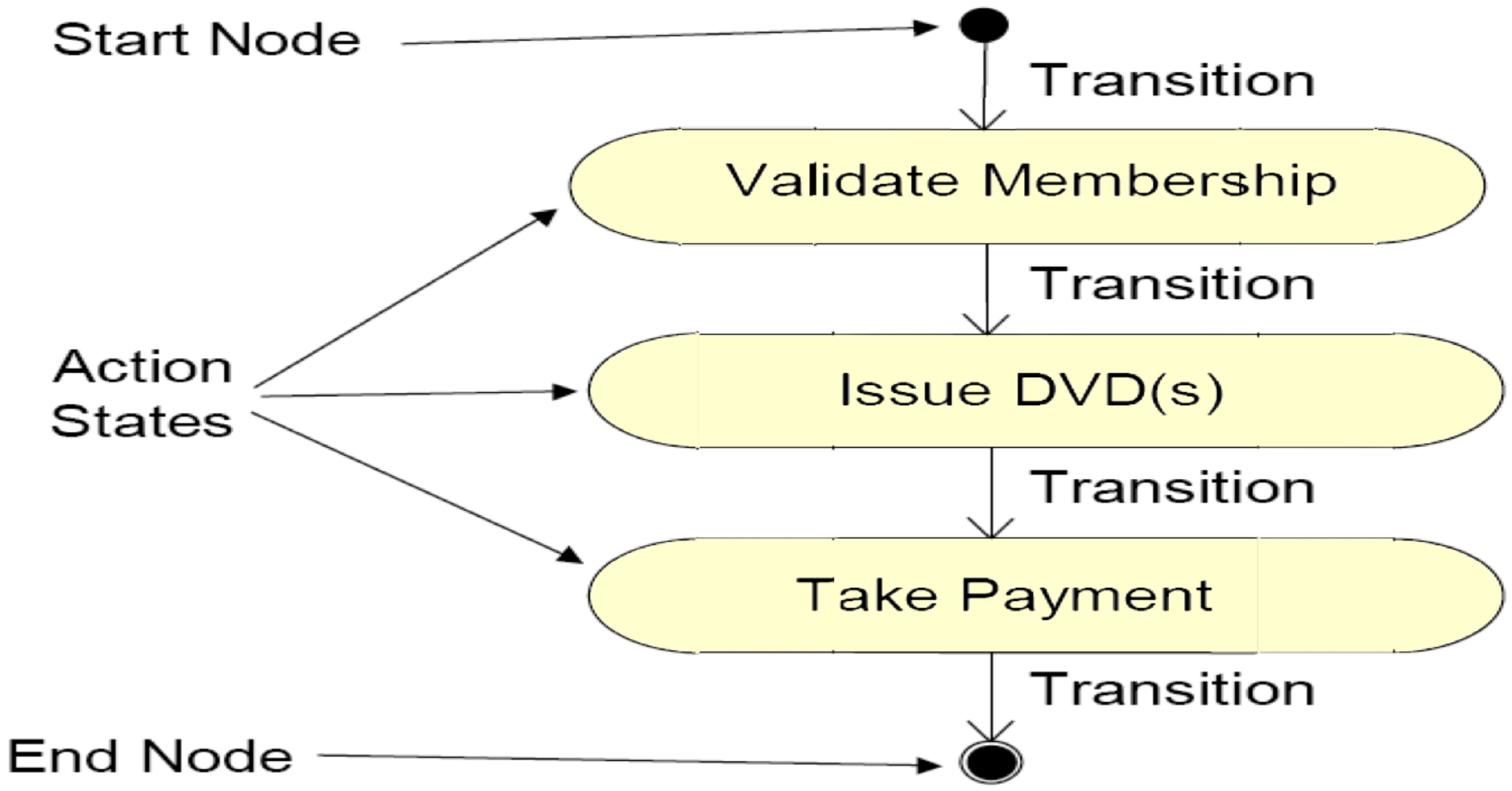


Transition $\longrightarrow$

- Activity/Action nodes are connected by Transitions.
- Also known as a control flow/directed flow /edge
- When the execution of the node completes, execution proceeds to the node found on the output flow.



Activity Diagrams: Example



Activity - 1

Draw an activity diagram for becoming a registered member of the CA-Sri Lanka Learning Management System.

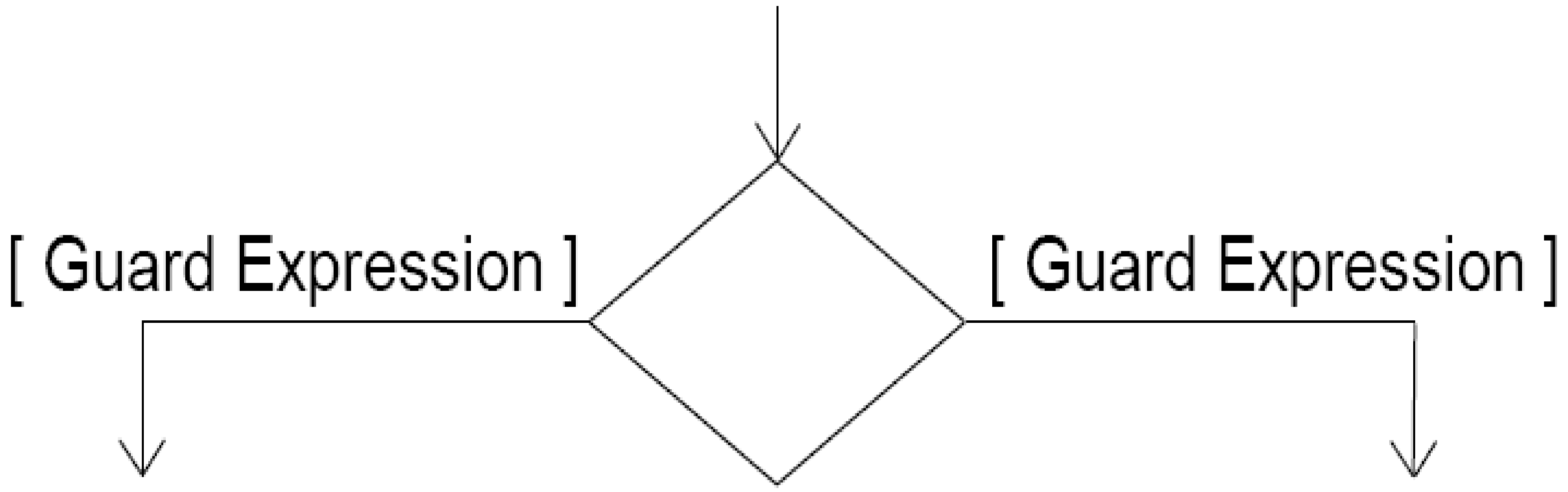
To become a registered member, a student must first upload a copy of their National Identity Card. Then, a few registration forms need to be filled out to proceed. A deposit of 3,000 must be paid to the bank, and the payment slip should be uploaded to the system. The system will then generate an ID and display it on the screen.

Decision Node

- A Decision represents a conditional flow of control:
  - The alternatives are mutually exclusive.
- Similar to IF/ELSE statements in programming languages like C/C++/Java.

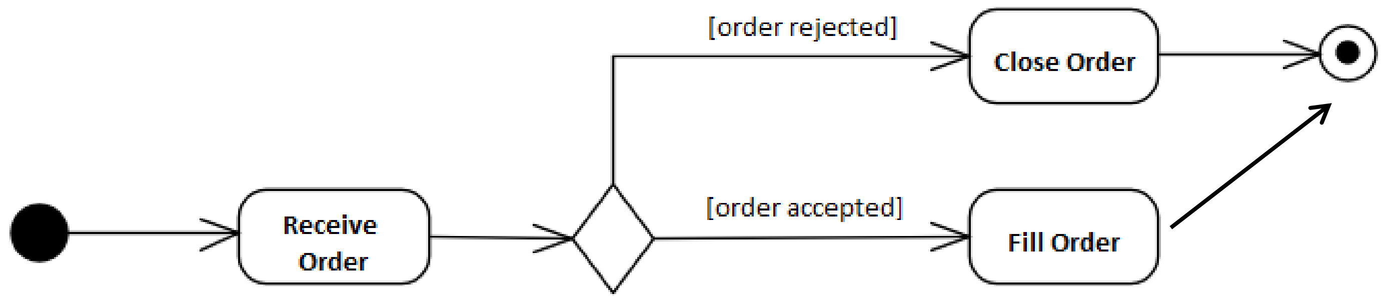
Decision Node

- A Decision/Branch is represented by:
  - a diamond at the branch point.
  - a guard expression for each possibility.

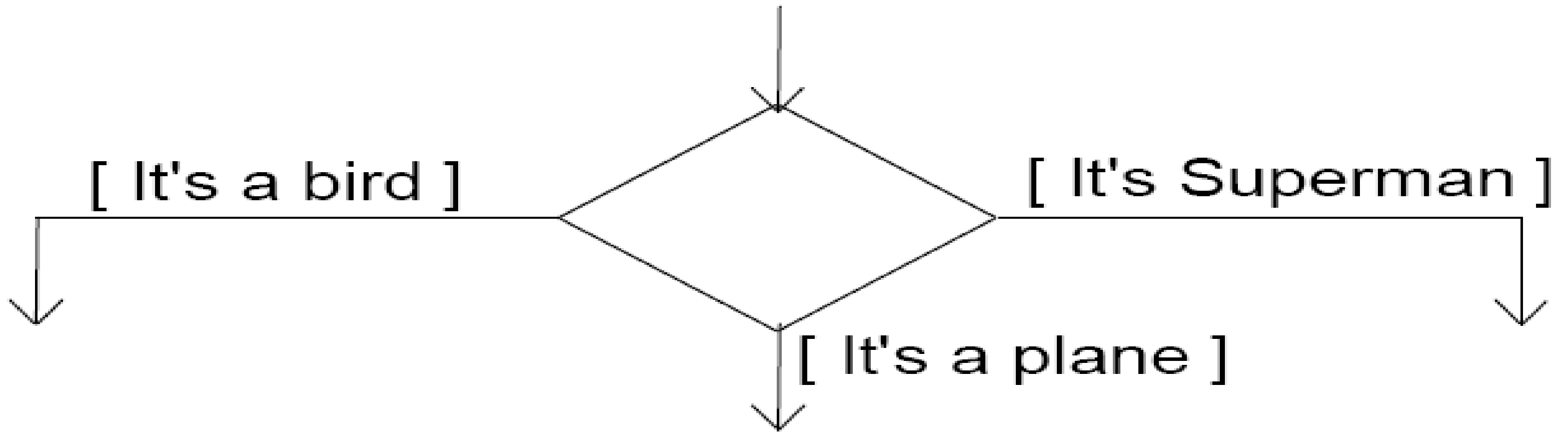


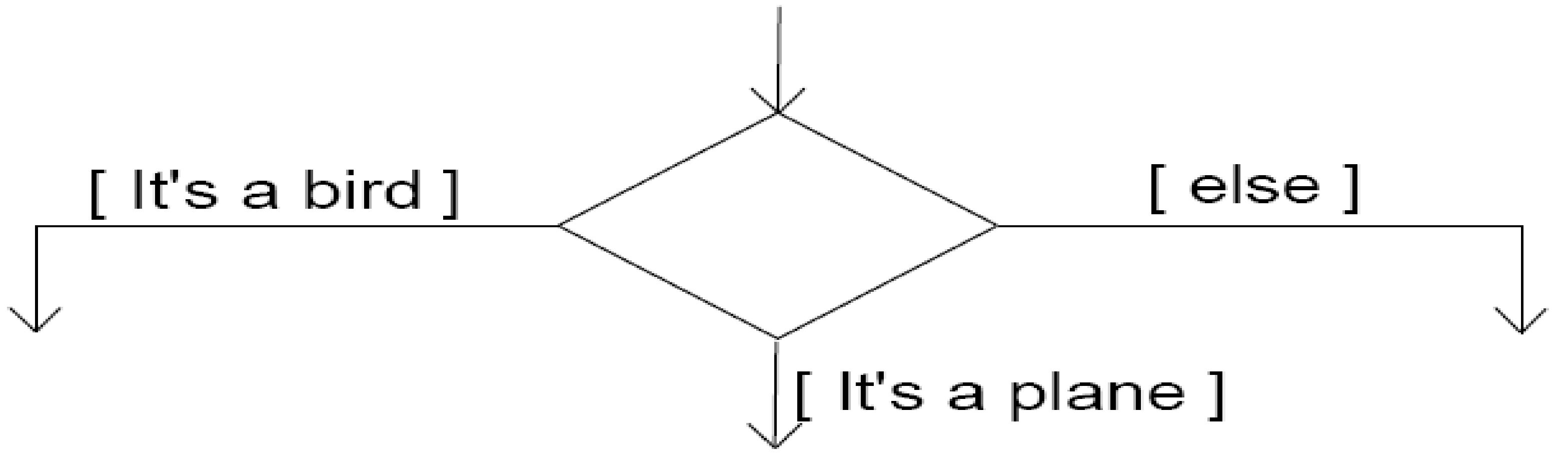
Decision Node

- Decision/Branch points.
  - Each branch must have a guard condition.
  - The flow of control flows down the single path where the branch condition is true.
  - There is no limit on the number of branches each branch point may have.



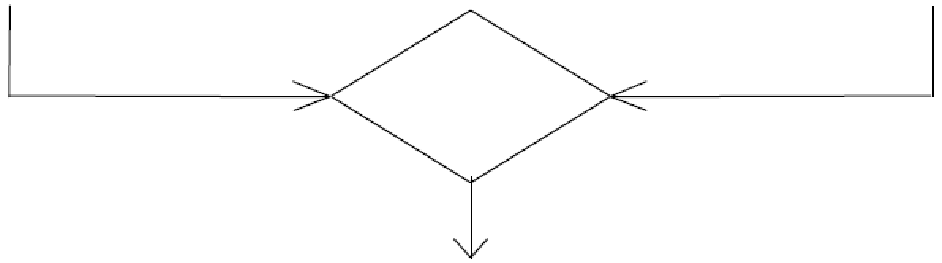
Multiple Branches: Examples



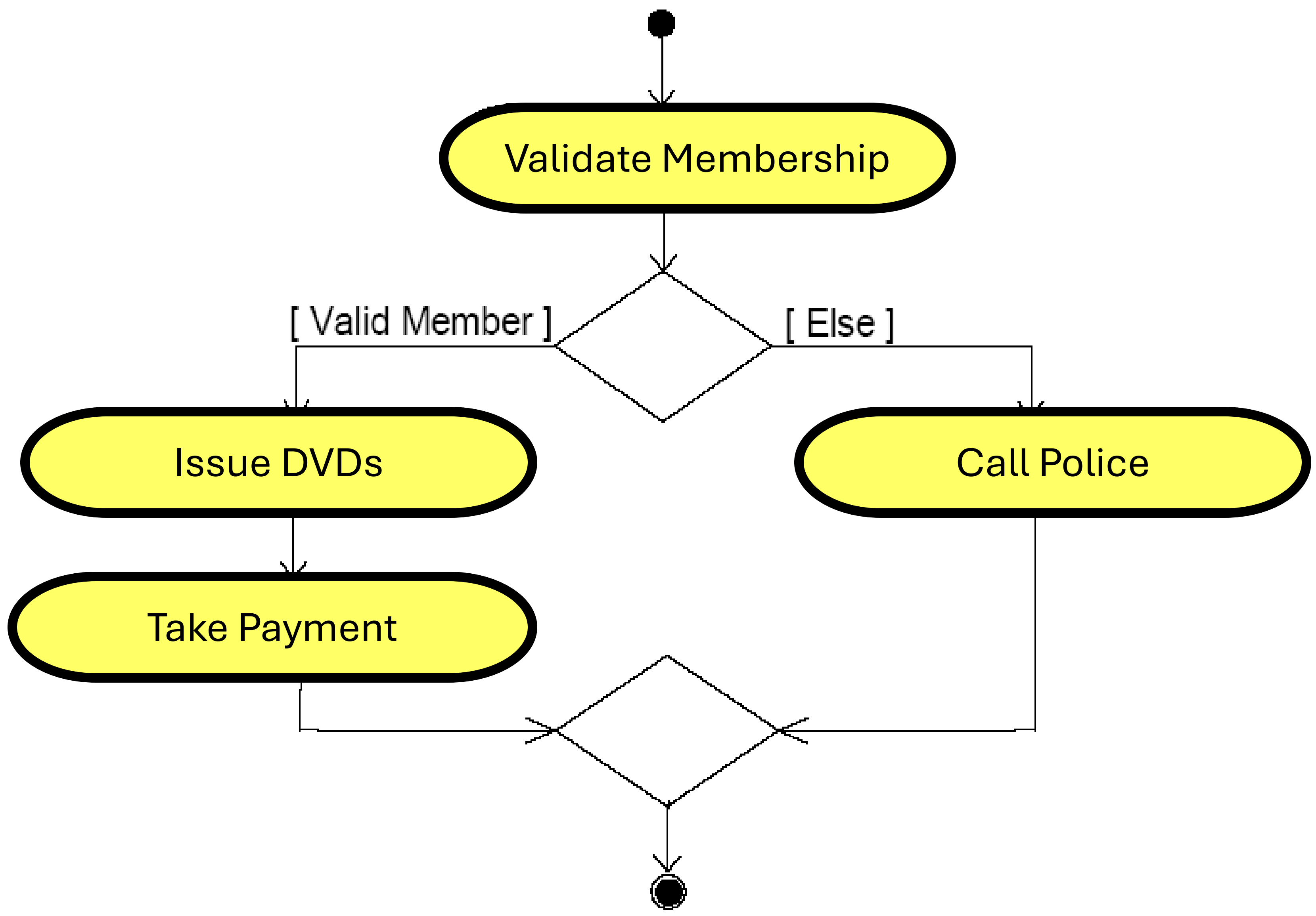


Merge Node

- A merge point is used to **merge the flow of control from two or more branch points back together.**
- The UML equivalent of ENDIF in pseudo code, or “}” in C/C++/Java.
- The diagram below shows a merge graphically – the diamond is optional:



Example


Validate Membership
[Valid Member]
[Else]
Issue DVDs
Call Police
Take Payment

Activity - 2

Flight Booking in Qatar Airways
To book a ticket, a guest must first become a member.

Members can book tickets in either **Economy Class** or **Business Class**.

- **Economy Class** includes a baggage allowance of **30 KG**.

If they need more, they can pay for **extra baggage** and get **an additional 10 KG**.

- **Business Class** includes **40 KG** of baggage by default.

All members must fill out a **mandatory information form** and select a **travel date**.

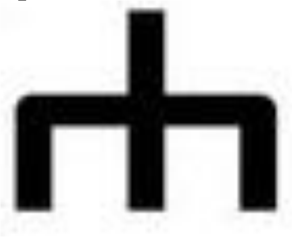
- If a flight is available on the selected date, they can book the ticket.

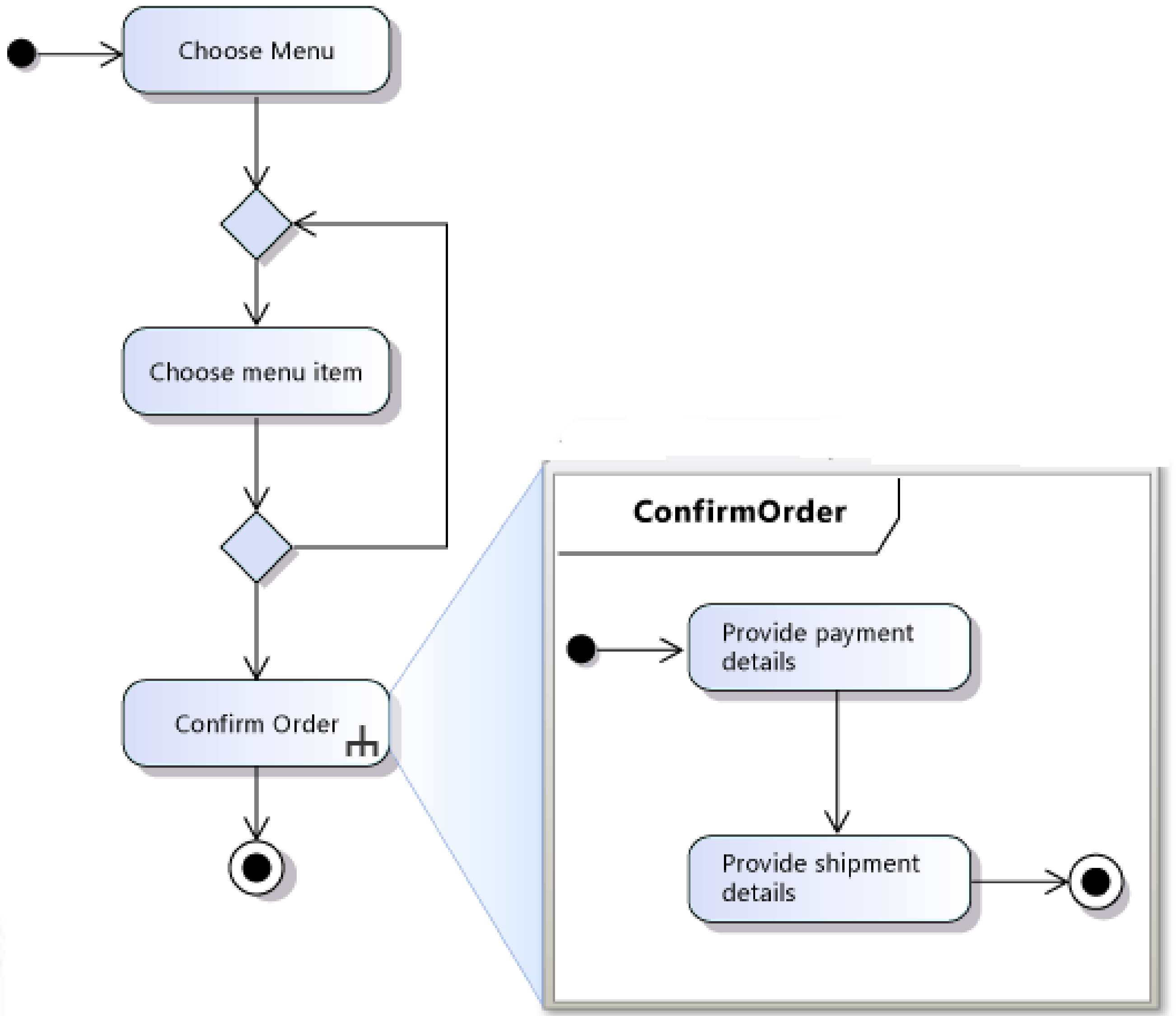
If not, they can cancel the process and exit the official website.

Call Action /Sub Activity

- **Sub activity** is an activity that is defined in more detail on another activity diagram.

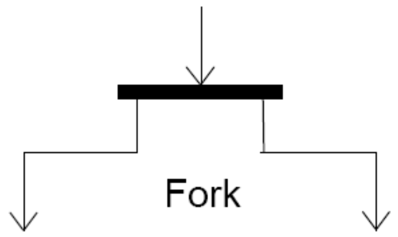
- It is indicated by a rake-style symbol within the action symbol.





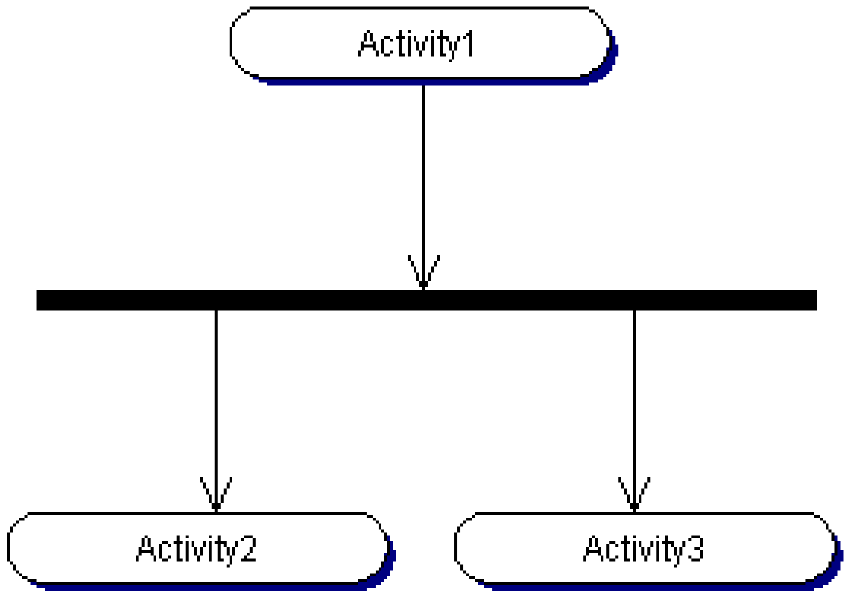
Forks & Joins

- Forks and joins are used to show activities that can occur at the **same time (in parallel).**
  - This does not mean that the activities must occur concurrently in the finished software system.
  - It means that the order of execution of the activities can be whatever is convenient for the implementation.



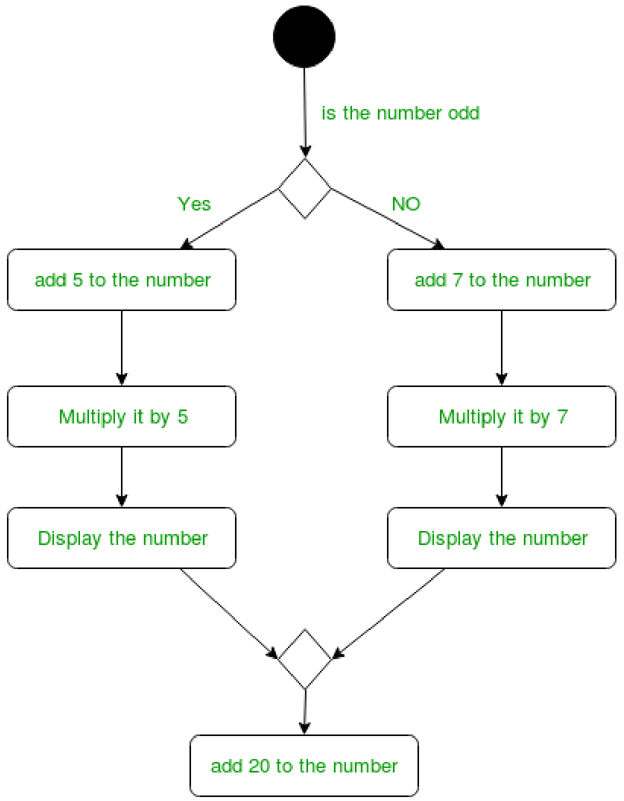
Fork

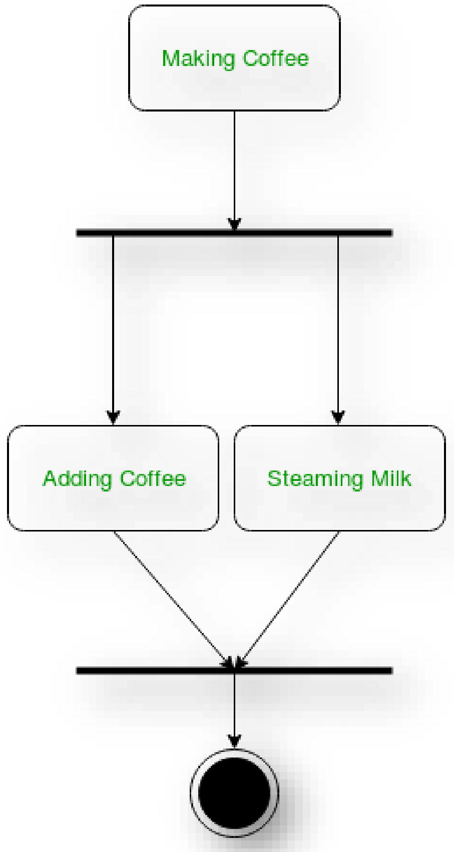
- A fork is when a single flow of control splits into two or more parallel (concurrent) flows of control.
- Represents a split in the flow of control.



Fork

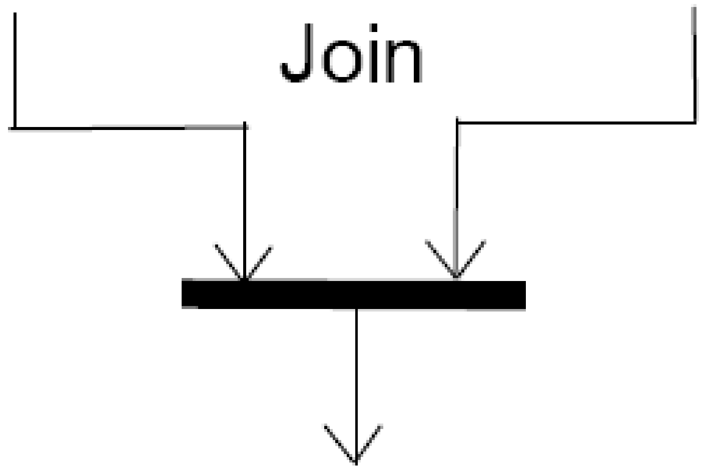
- Unlike a branch point, the control flows down **all forked paths.**
- With a branch point, the control flows down only **one path.**





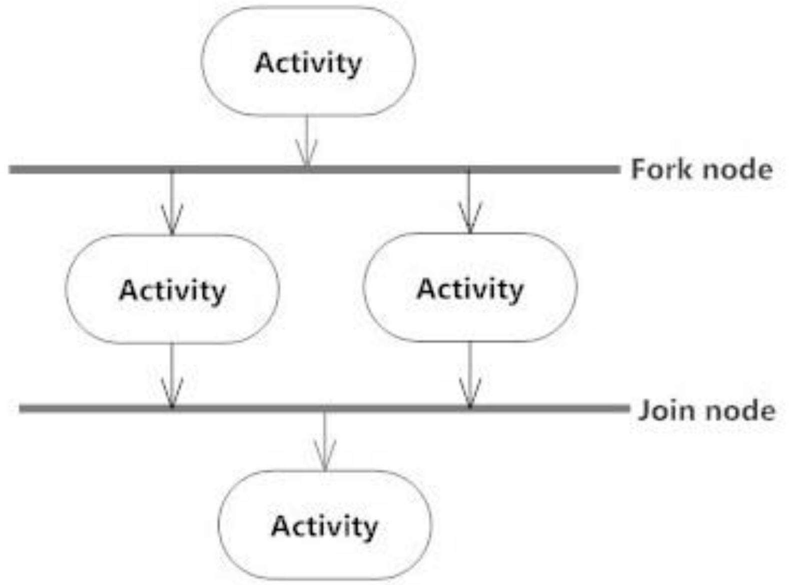
Join

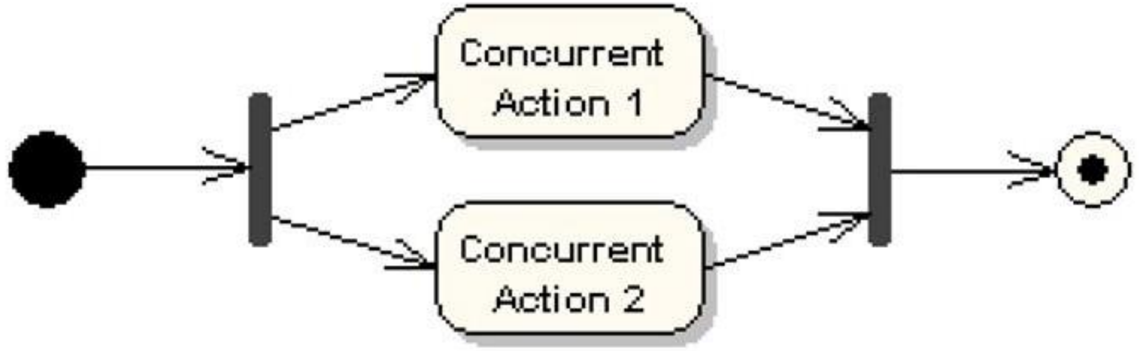
- A join is when two or more flows of control merge into a single flow of control.
- Represents the merging of multiple flows of control back into one.
- **Every fork must have a join associated with it.**



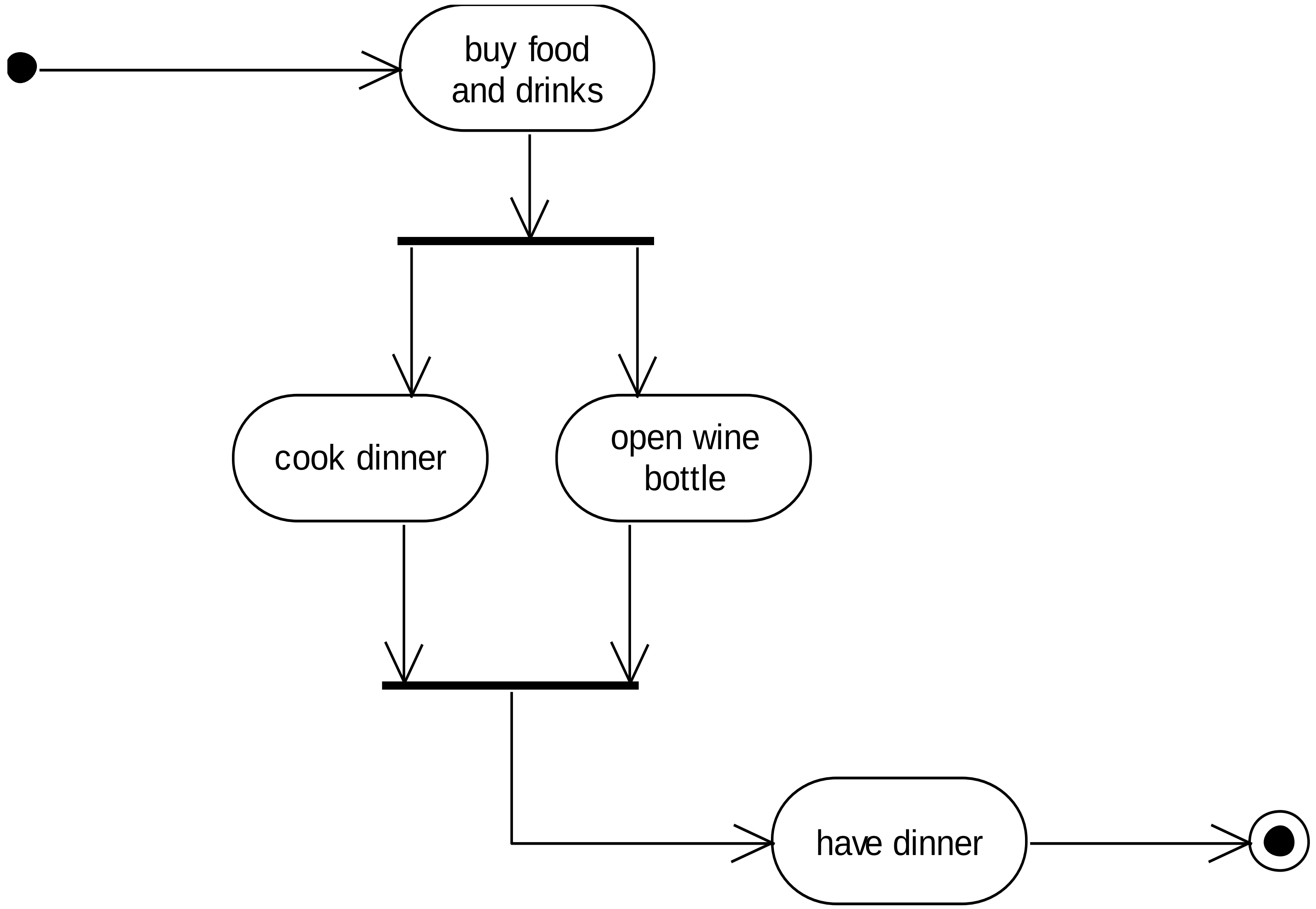
Join

- A flow of control is also known as a thread.
- A synchronization bar is used to model forking and joining and is modeled as a thick horizontal or vertical bar.

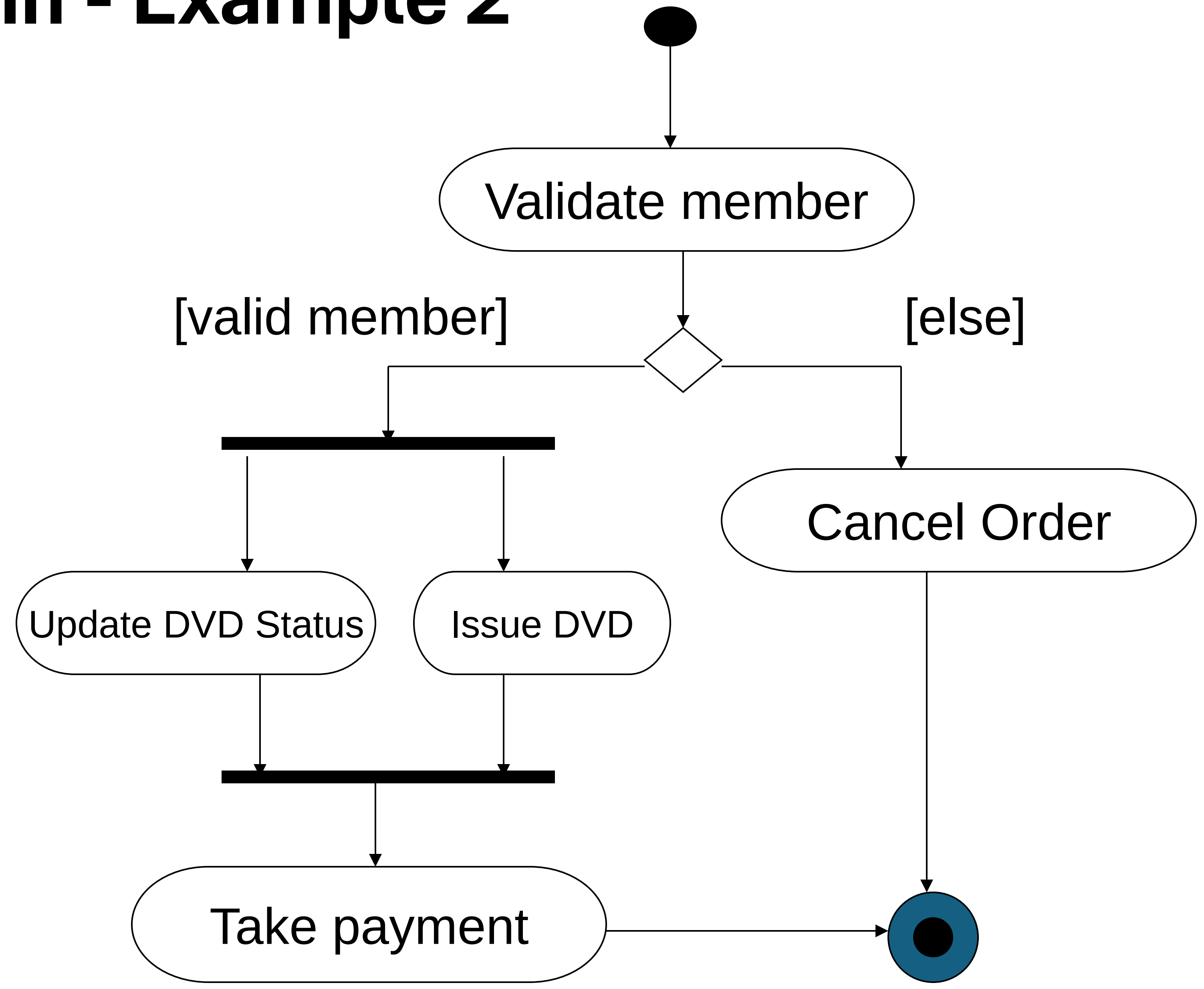




Fork and Join - Example 1



Fork and Join - Example 2



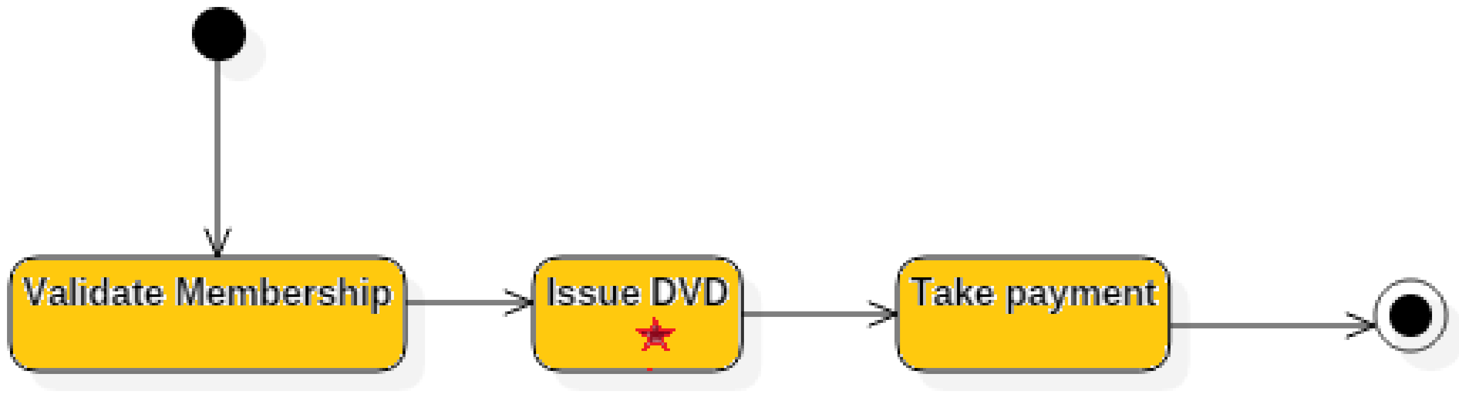
Activity - 3

<u>"Enrolling in University"</u>

1. Candidates who wish to register for university have to fill out enrolment forms and submit them promptly.
2. The forms that are being incorrectly filled will be rejected.
3. The candidates who submitted properly filled forms are being enrolled.
4. The enrolled candidates then have to register for a seminar and make payments for initial tuition.
5. Meanwhile, they have to attend a university overview presentation as a start.
6. Then, only students can start classes.

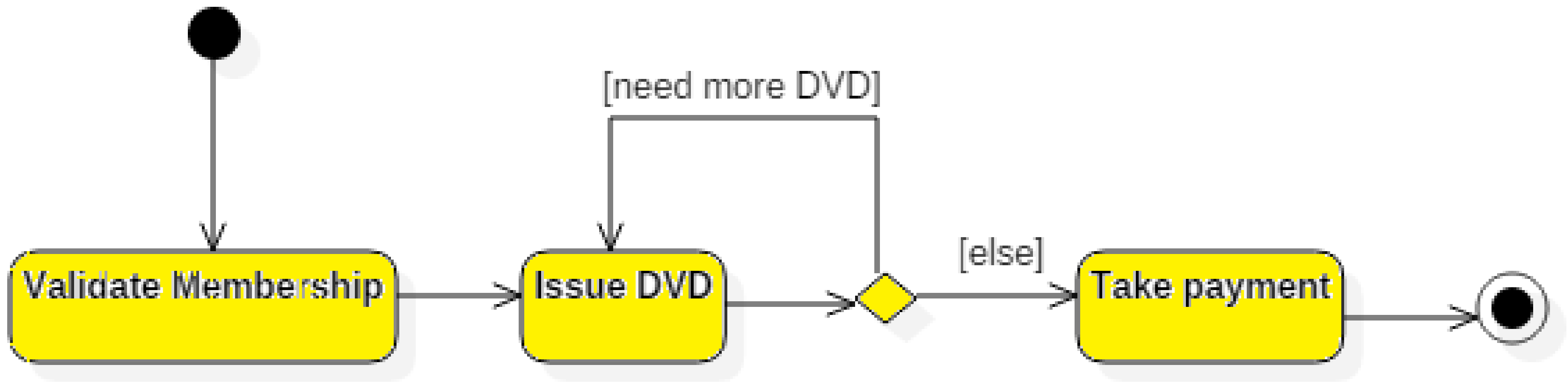
Iteration ★

- An **asterisk** inside an action state indicates that it may need to be **performed more than once**.
  - This notation is used to show iteration, without the unnecessary details of a loop construct.
- The next action state does not occur until the loop is finished.



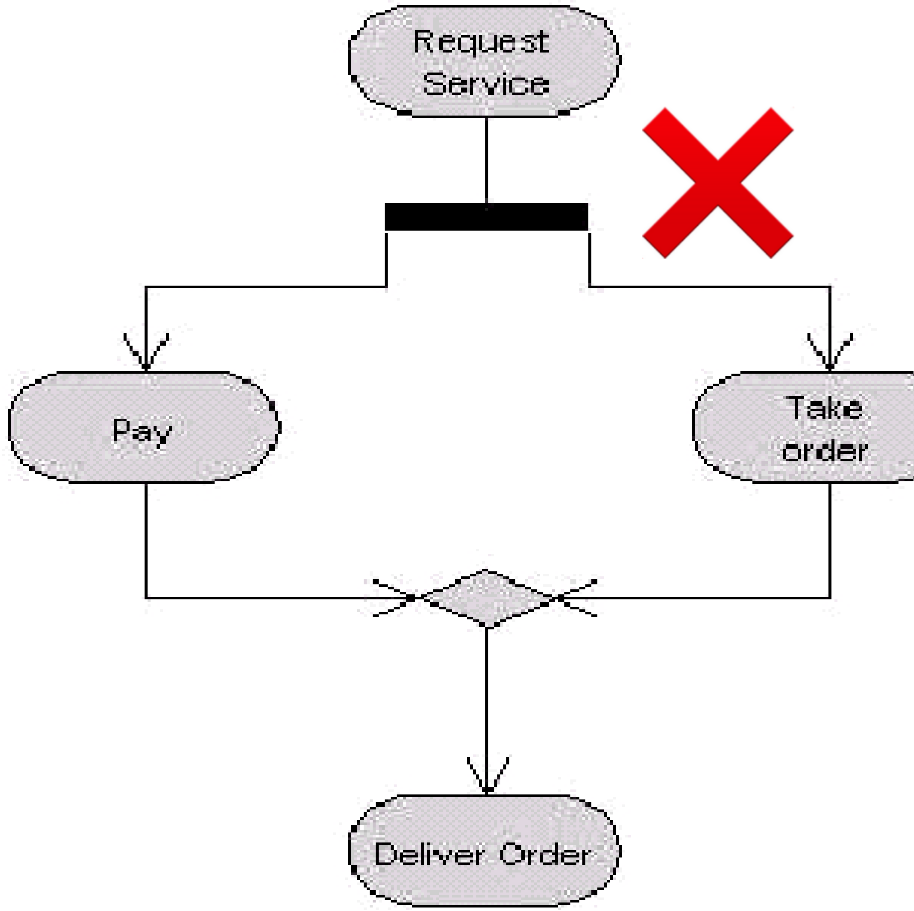
Iteration – Alternative Method

- Use of the **asterisk** to represent an iteration will not highlight the **loop termination conditions** and **number of repetitions.**
- Therefore, use of a **Decision Node** to represent an iteration would be ideal.



Common Mistakes

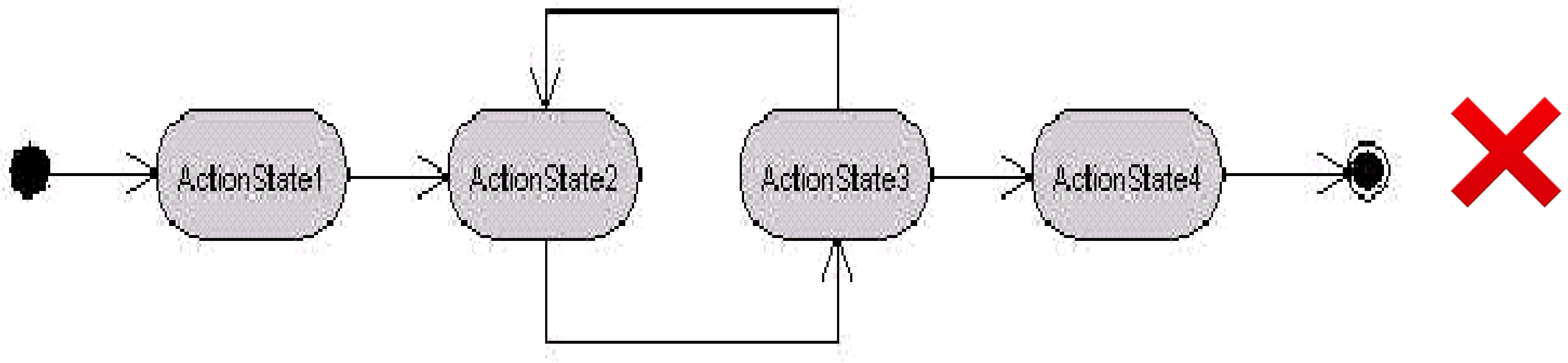
- Incorrect use of **forks, branch points, merge points and joins.**
- The fork means that the flow of control goes down both paths.
- The merge point indicates a merging of divergent **paths, not flows**.



Common Mistakes

Loops

- Guard conditions should be mentioned



Activity - 4

Develop a software system for a client

1. Project Manager (**PM**) first gathers the requirements
2. UI Engineer (**UIE**) develops the prototype
3. PM shows the prototype to **client**
4. Till the client is satisfied, UIE modifies the prototype.
5. Client signs off the prototype
6. Once the client is satisfied, UIE develop the UI screens, Software Engineer (**SE**) develops the system.
7. Once both (system and UI development) are done, SE integrates the system.
8. PM delivers the system to client
9. Client signs off the delivery.

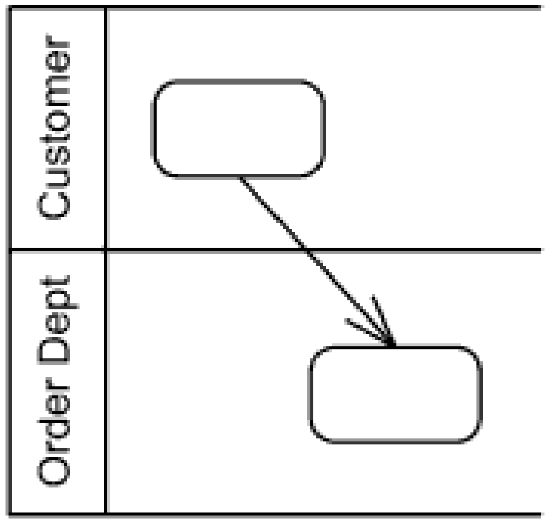
Partitioning

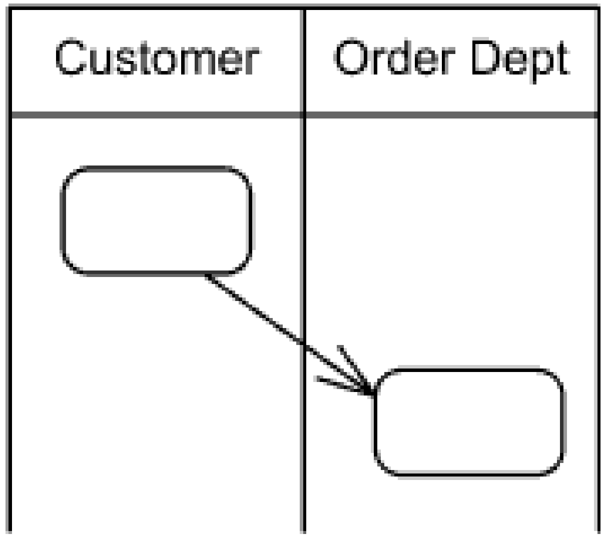
- An activity **partition** is an **activity group** for actions that have some common characteristic.
- Partitions often correspond to **organizational units** or **business actors** in a **business model**.

Partitions: SwimLanes

Activity **partition** may be shown using a **SwimLane** notation:

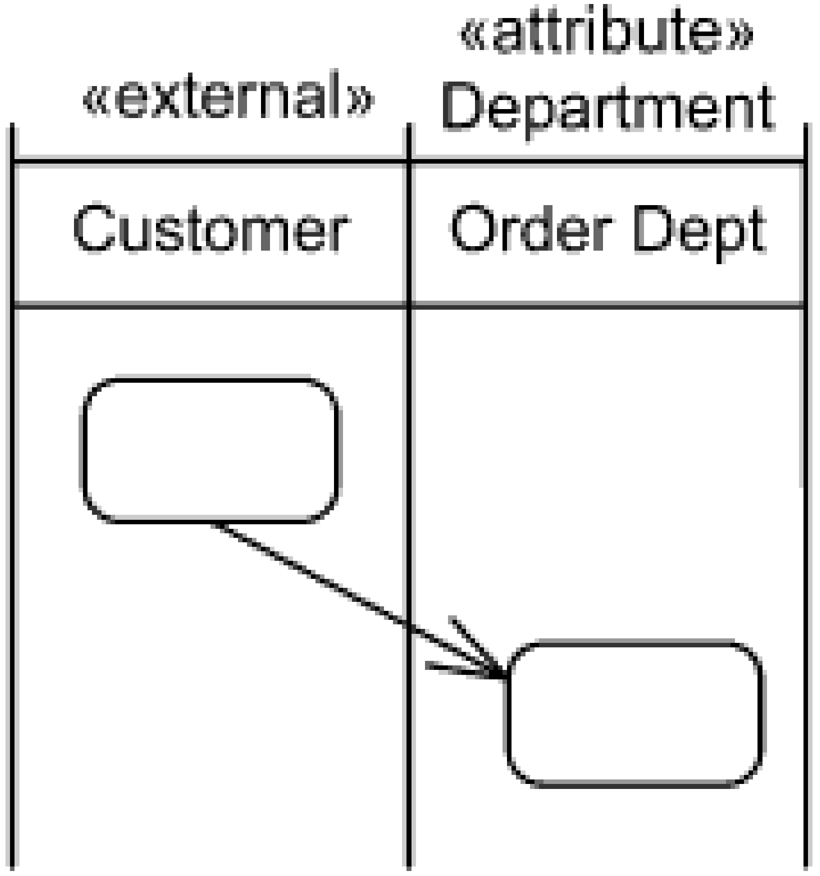
- with two, usually parallel lines, either horizontal or vertical
- a name labeling the partition in a box at one end.





Sub Partitioning in SwimLanes

- Order department is a subclass of the Department class.



Activity - 5

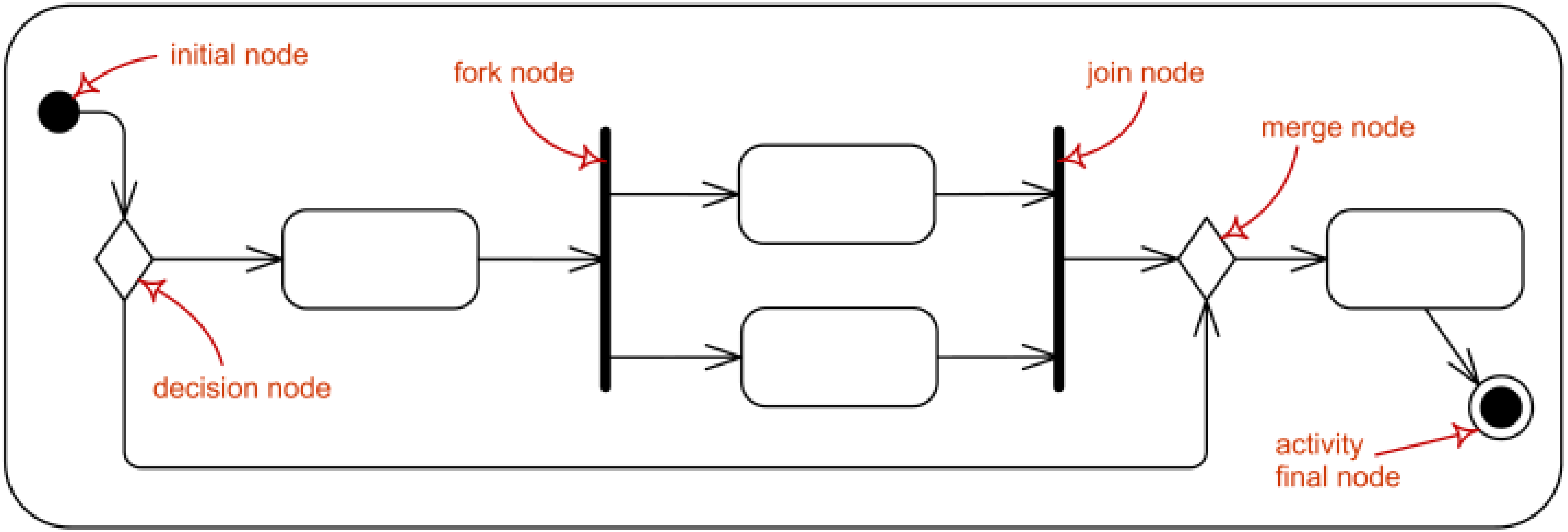
Develop a software system for a client

1. Project Manager (**PM**) first gathers the requirements.
2. UI Engineer (**UIE**) develops the prototype.
3. PM shows the prototype to **client.**
4. Till the client is satisfied, UIE modifies the prototype.
5. Once the client is satisfied, signs off the prototype
6. UIE develop the UI screens while Software Engineer (**SE**) develops the system.
7. Once both (system and UI development) are done, SE integrates the system.
8. PM delivers the system to client
9. Client signs off the delivery.

Activity – 6 : "Order Processing System"

1. Once the order is received by the customer service department, several other departments too operate on various activities to complete the order.
2. The fulfillment department should fill the order (prepare goods to deliver) and deliver the order to customer.
3. Meanwhile, as soon as the order is received, customer service department works on sending an invoice to the customer.
4. The finance department will handle the payments from the customer. The shop operates on credit basis. Therefore, receiving payments is not mandatory prior to the delivery of the goods.
5. Customer service department can close the order only after completing all the above activities.

Activity Diagram Notations



References

- The Unified Modeling Language Reference Manual – James Rumbaugh, Ivar Jacobson, et al
- UML Distilled: A Brief Guide to the Standard Object Modeling Language by Martin Fowler, Kendall Scott
- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process by Craig Larman
- The Complete UML Training Course, Student Edition by Grady Booch, et al
- UML Explained by Kendall Scott

Thank You…!!!