Object Oriented Analysis and Design with Java

UE19CS353

Prof. Samatha R S & Prof. Vinay J

Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

PES UNIVERSITY ONLINE

UE19CS353: Object Oriented Analysis and Design with Java

Activity Modelling: UML Activity Diagrams and Modelling Guidelines

Prof. Samatha R S & Prof. Vinay J
Department of Computer Science and Engineering

ACTIVITY DIAGRAM

Introduction: Interaction Model

Interaction model describes how objects interact to produce useful results. Interactions can be modeled at different levels of abstraction:

1. Use Case Models
2. Sequence Models
3. Activity Models

Use Case Model -> Shows how a system interacts with outside actors. Each use case represents a piece of functionality that a system provides to its user.

Sequence Model -> Shows more detailed messages exchanged among set of objects of the system over a specific period of time.

Activity Model -> Shows how activities are coordinated. Represents the workflow of the process.

ACTIVITY DIAGRAM

Introduction: Activity Model

Activity Model and Use Case Model are logical models which describe the business domain's activities without suggesting how they are conducted.

- It emphasizes the flow of control from activity to activity in an object. Similar to the traditional program flowchart.
- It is particularly useful when you know that an operation has to achieve a number of different things, and we want to model what the essential dependencies between them are, before we decide in what order to do them.
- Records the dependencies between activities, such as which things can happen in parallel and what must be finished before something else can start.

ACTIVITY DIAGRAM

Activity Model: Purpose

- The purpose of the activity diagram is to model the procedural flow of actions that are part of a larger activity.
- In projects in which use cases are present, activity diagrams can model a specific use case at a more detailed level.
- However, activity diagrams can be used independently of use cases for modelling a business-level function, such as buying a concert ticket or registering for a college class.
- Activity diagrams can also be used to model system-level functions, such as how a ticket reservation data mart populates a corporate sales system's data warehouse.

ACTIVITY DIAGRAM

Activity Model

Activity diagrams are like flow chart showing flow of control & focuses on operations rather than objects.

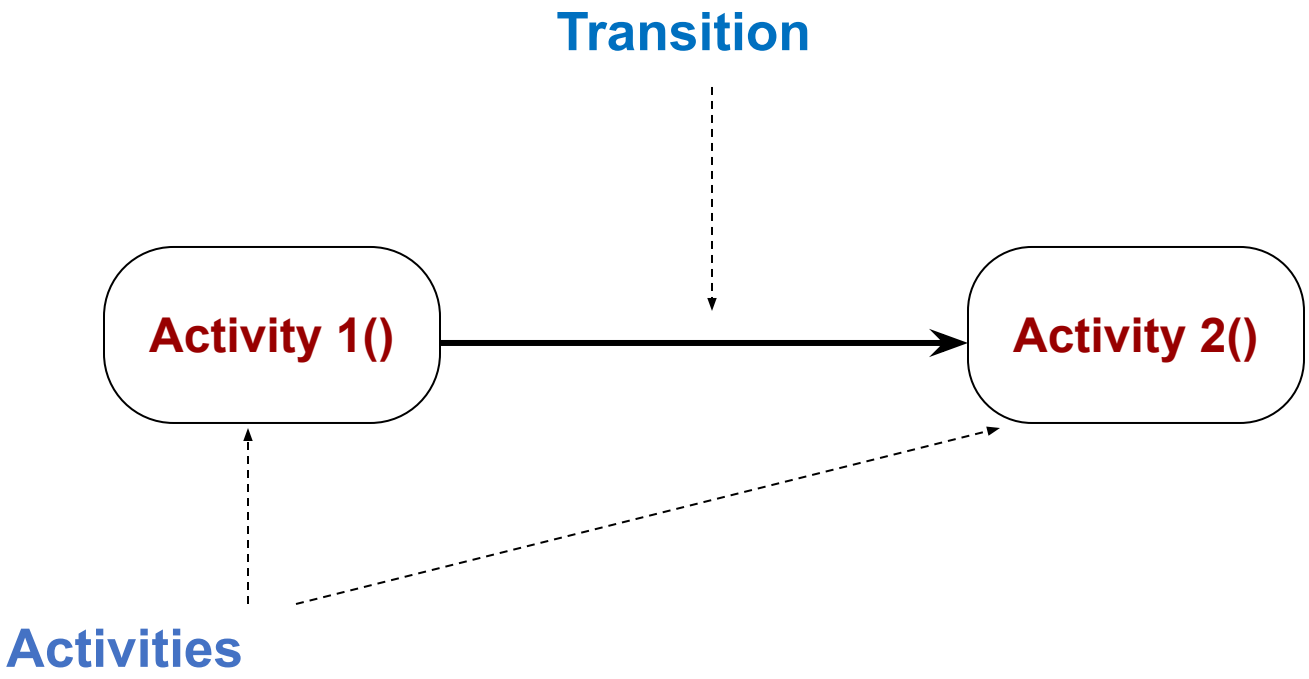
It shows

- **Sequential flow of control**
- **Concurrent flow of control**

ACTIVITY DIAGRAM

Activity Model: UML Notation

- They are graphically rendered as rounded rectangle.



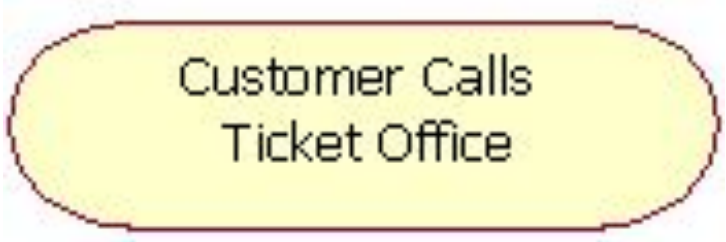
ACTIVITY DIAGRAM

Activity Model:

- An action is indicated on the activity diagram by a "capsule" shape – a rectangular object with semi circular left and right ends.

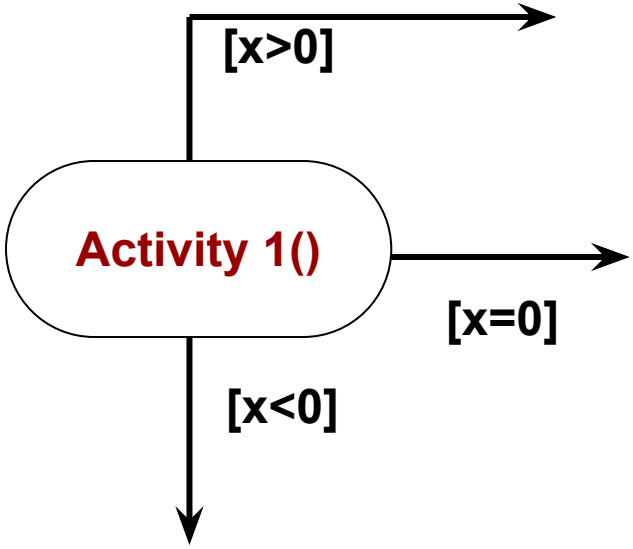
- The text inside it indicates the action (e.g., Customer Calls Ticket Office or Registration Office Opens).

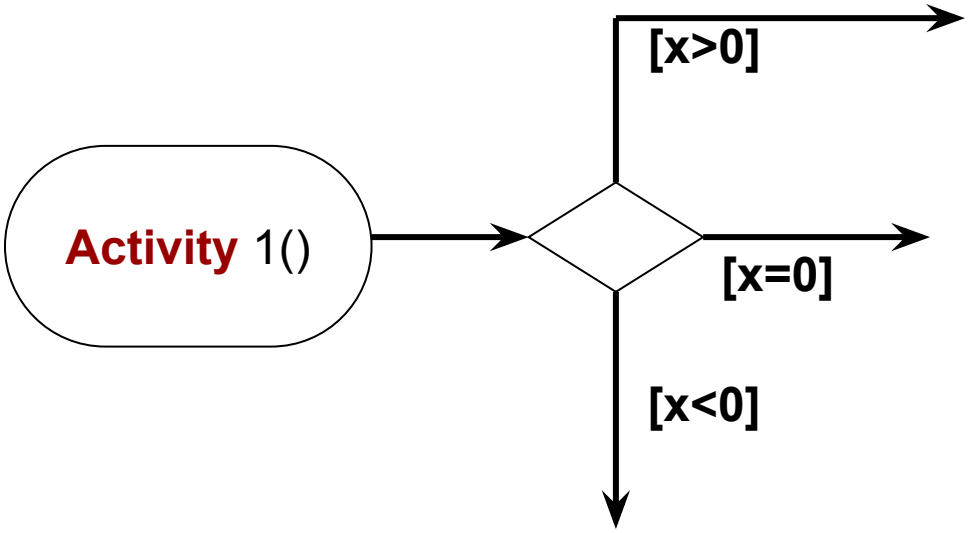
- A sample action that is part of an activity diagram



ACTIVITY DIAGRAM

Activity Model: UML Notation

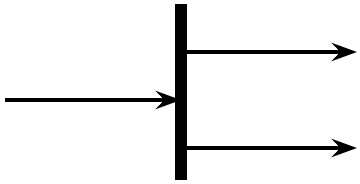




Decision Diamond

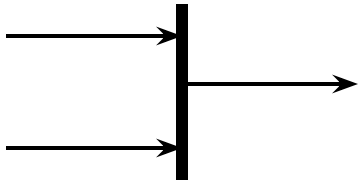
ACTIVITY DIAGRAM

Activity Model: UML Notation



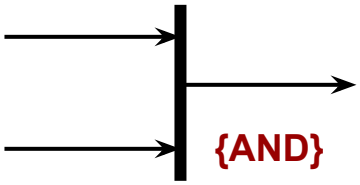
Splitting Bar (Fork)

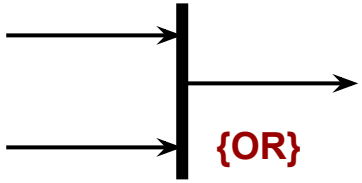
At the Fork, the activities of each of outgoing transitions are concurrent

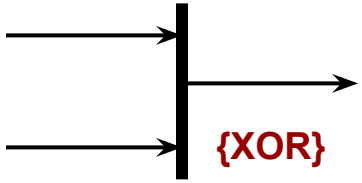


Synch. Bar (Join)

At the join, the concurrent flows synchronize. Each waits until all incoming flows have reached the join.







Join Variants

ACTIVITY DIAGRAM

Activity Model: UML Notation

A solid circle with outgoing arrow shows starting point of activity diagram so control starts at solid circle.

A bull's eye shows termination point which has incoming arrow.

At bull's eye activity is completed & execution is completed.

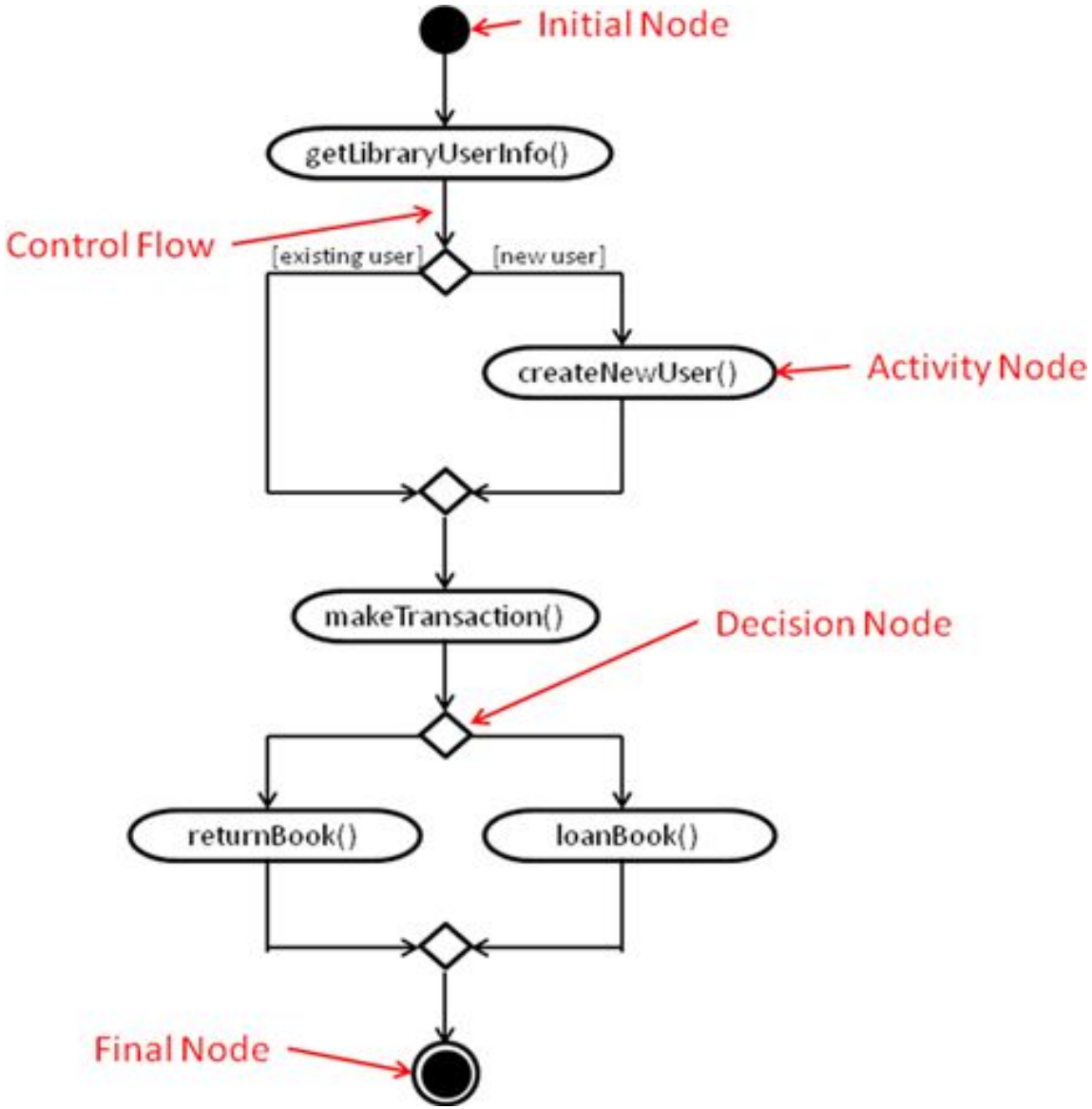
Start Marker

Stop Marker

5. Start & Stop Activity

ACTIVITY DIAGRAM

Activity Model: Diagram

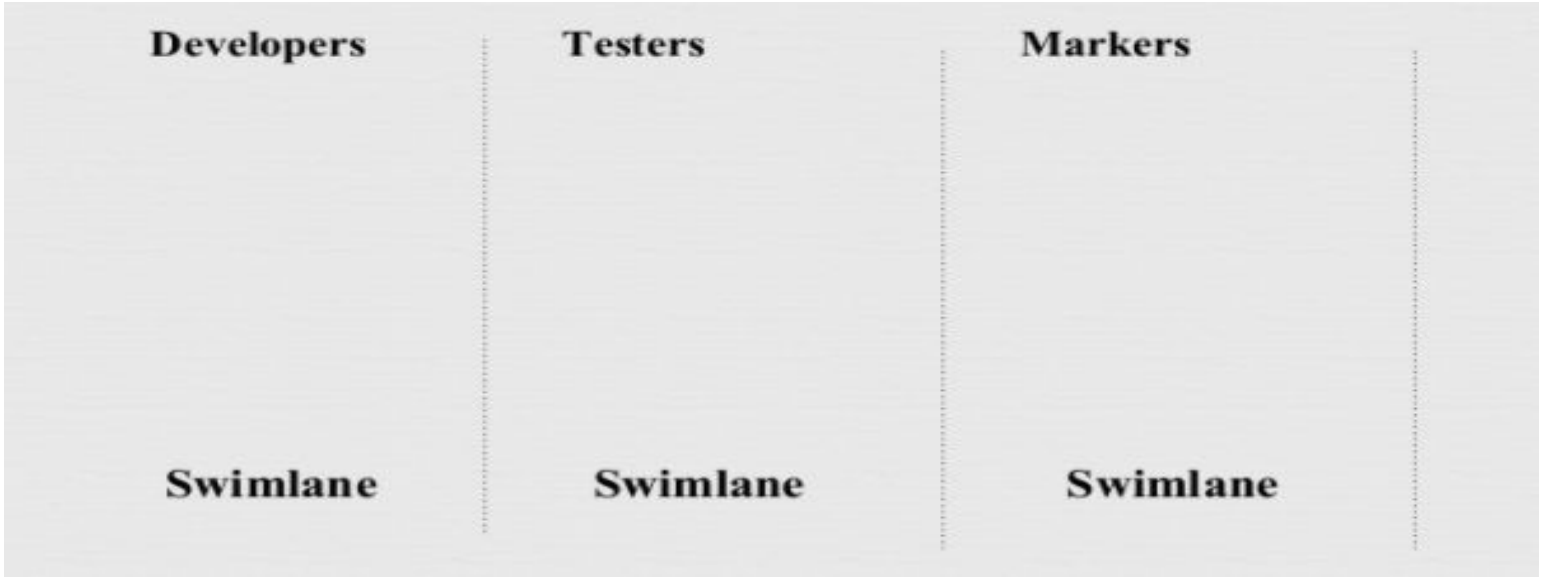


ACTIVITY DIAGRAM

Activity Model: UML Notation

A swimlane specifies a locus of activities. To partition the activity states on an activity diagram into groups each group representing the business organization responsible for those activities each group is called a swimlane.

Each swimlane is divided from its neighbor by a vertical/Horizontal solid line.

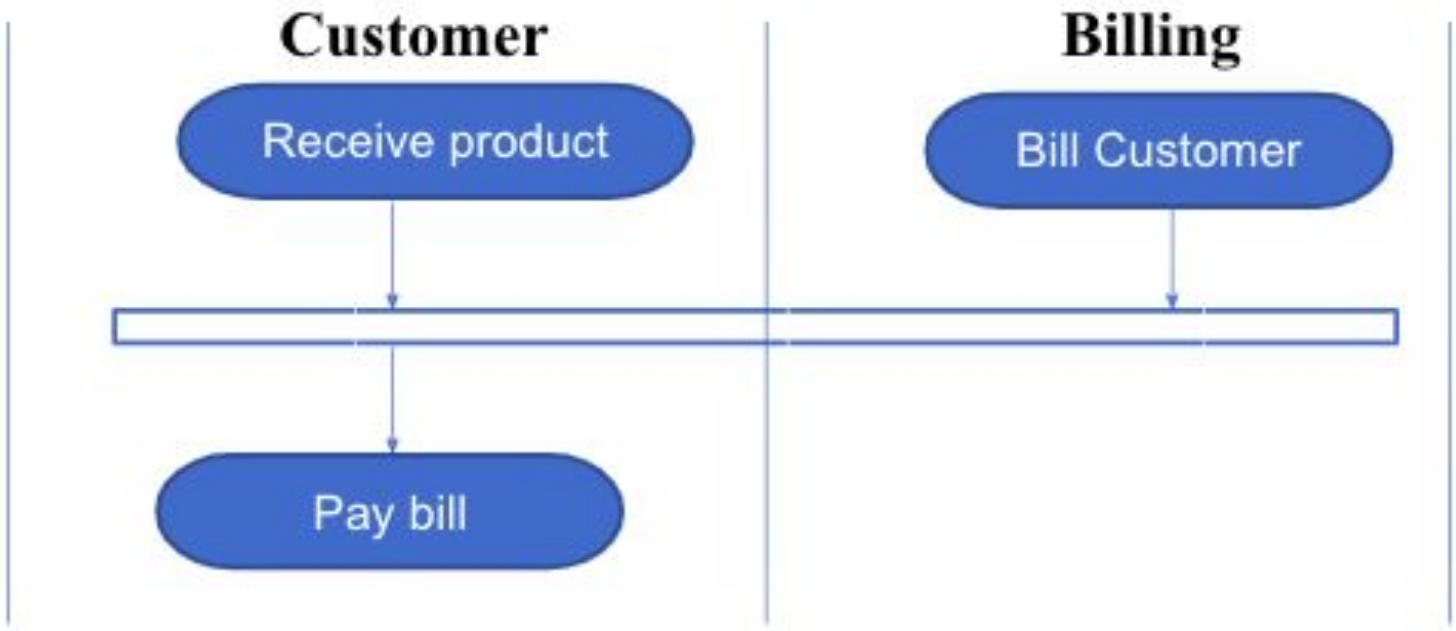


Application/Department/Group/Role Boundaries

ACTIVITY DIAGRAM

Activity Model: Swimlanes

Swimlanes partition groups of activities based on, for instance, business organizations



ACTIVITY DIAGRAM

Activity Model: Swimlanes

- In activity diagrams, it is often useful to model the activity's procedural flow of control between the objects (persons, organizations, or other responsible entities) that actually execute the action.

- To do this, we can add *swimlanes* to the activity diagram (swimlanes are named for their resemblance to the straight-line boundaries between two or more competitors at a swim meet).

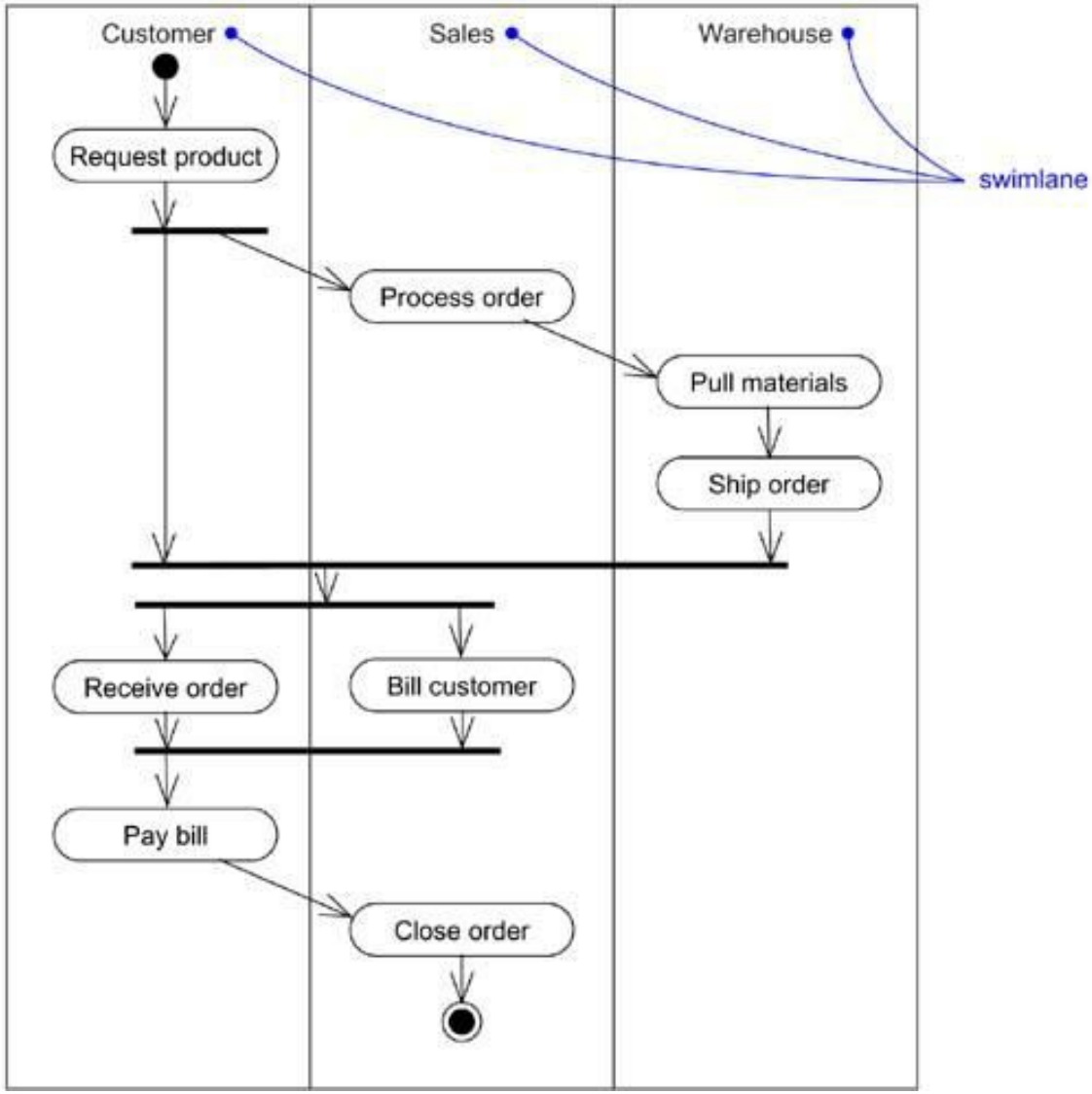
- To put swimlanes on an activity diagram, use vertical columns. For each object that executes one or more actions, assign a column its name, placed at the top of the column. Then place each action associated with an object in that object's swimlane.

- Each swimlane has a name unique within its diagram. Each swimlane may represent some real-world entity.

- Each swimlane may be implemented by one or more classes. Every activity belongs to exactly one swimlane, but transitions may cross lanes

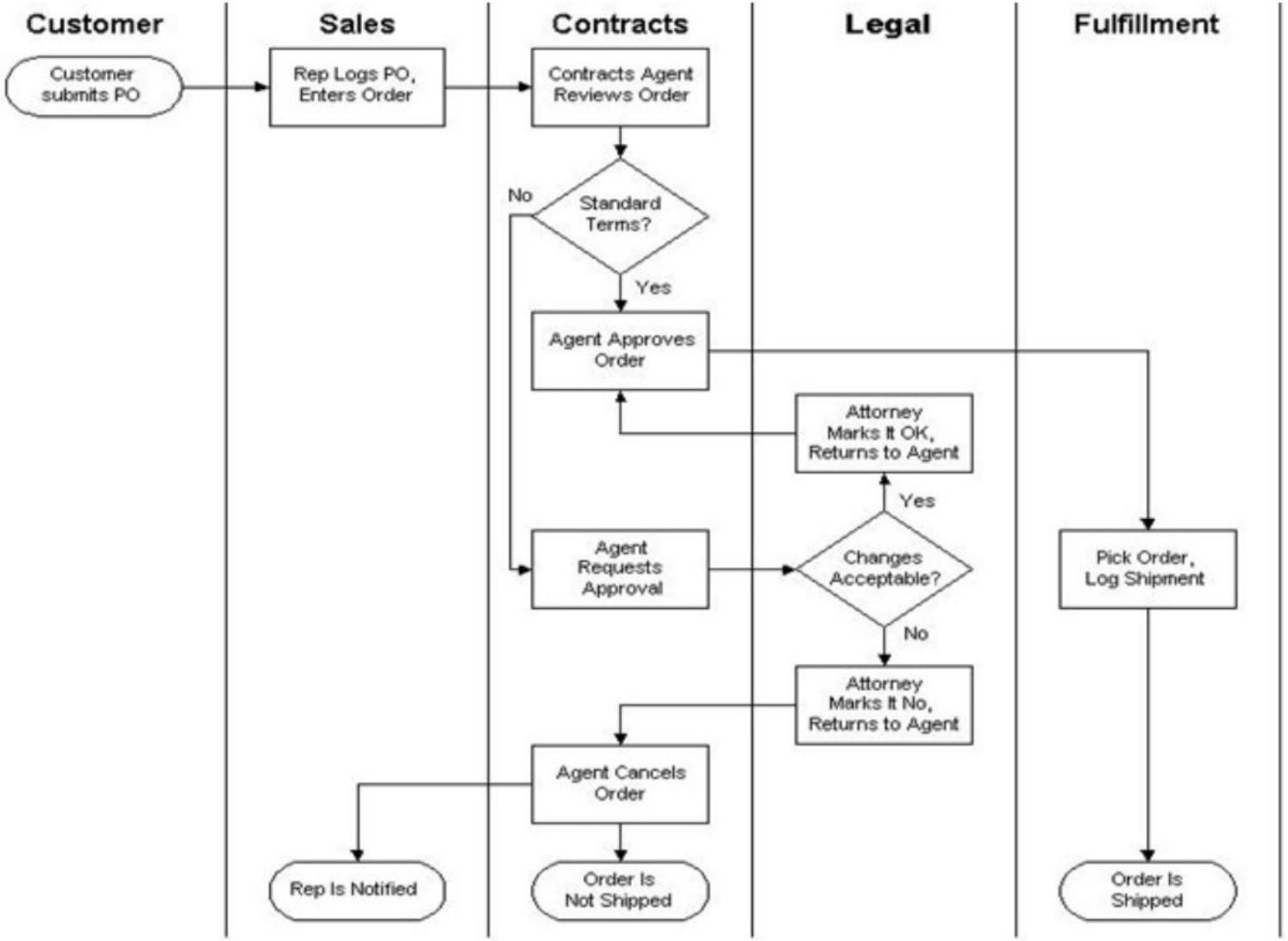
ACTIVITY DIAGRAM

Activity Model: Swimlanes



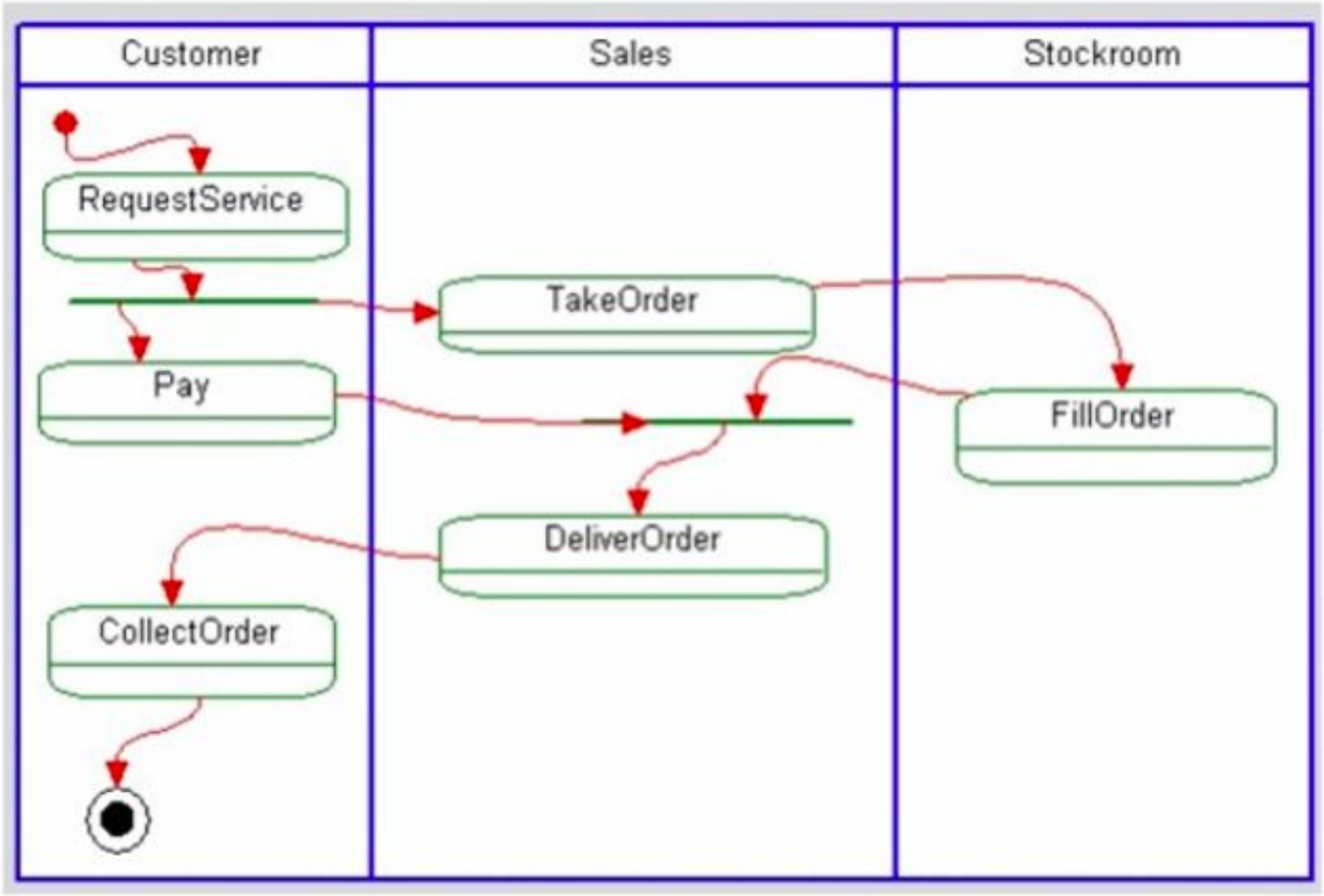
ACTIVITY DIAGRAM

Activity Model: Swimlanes



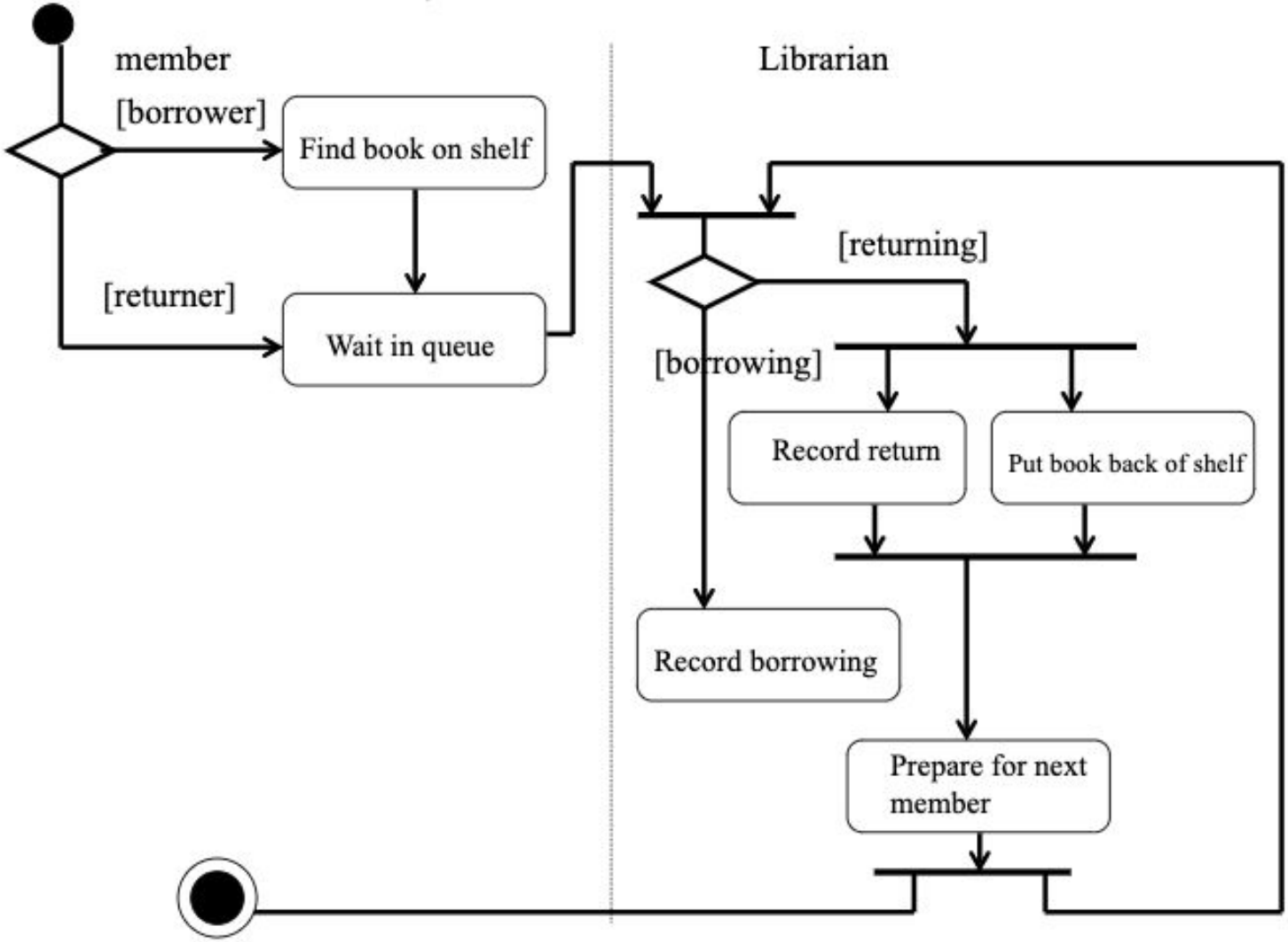
ACTIVITY DIAGRAM

Activity Model: Swimlanes



ACTIVITY DIAGRAM

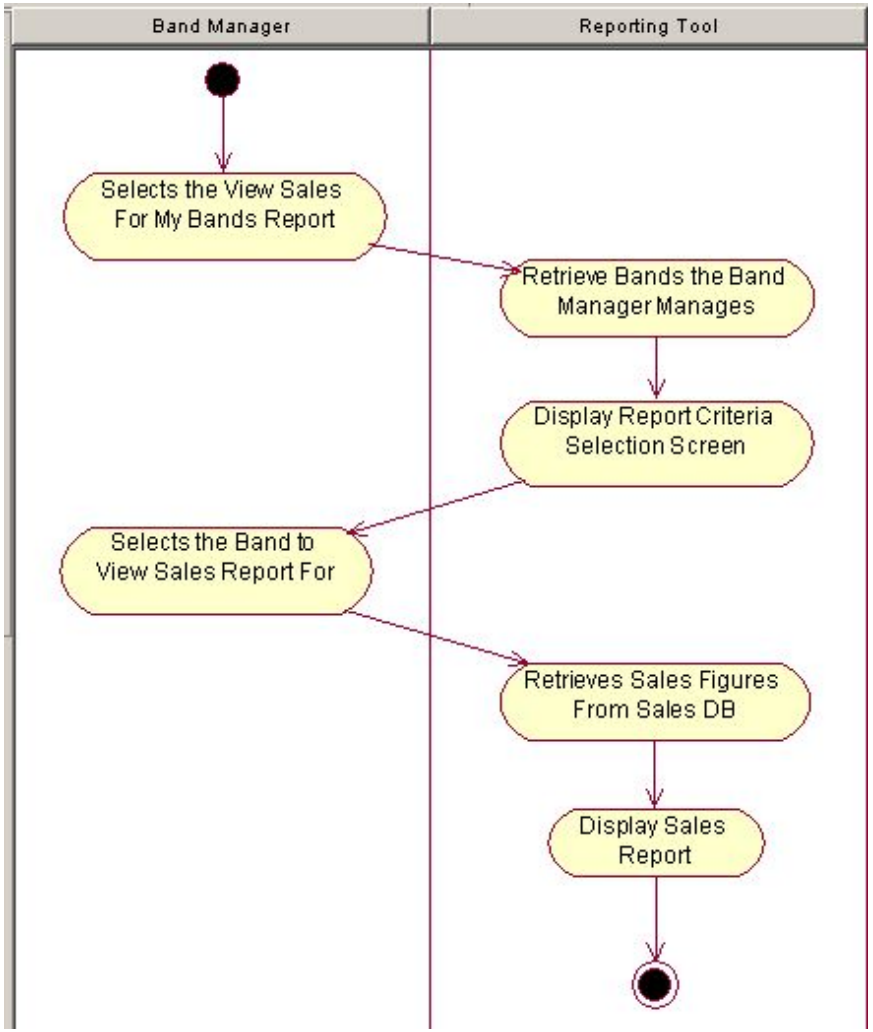
Activity Model: Swimlanes



ACTIVITY DIAGRAM

Swimlane: Band Manager and Reporting Tool

The Band Manager object executes the "Selects the View Sales For My Band Report" action and the Reporting Tool executes the "Retrieve Bands the Band Manages" action.



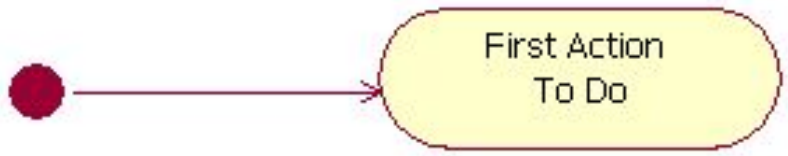
ACTIVITY DIAGRAM

Activity Model: The Initial State

The initial state clearly shows the starting point for the action sequence within an activity diagram. Because activity diagrams show a sequence of actions, they must indicate the starting point of the sequence.

The official UML name for the starting point on the activity diagram is *initial state*, and it is the point at which you begin reading the action sequence.

It is important to note that there can *be only one* initial state on an activity diagram and *only one* transition line connecting the initial state to an action.



ACTIVITY DIAGRAM

Activity Model: The Initial State

The sample activity diagram documents the activity "Booking a Concert Ticket," with actions in the following order:

1. Customer calls ticket office.

2. Ticket incharge asks what event person wants tickets for.

3. Customer tells rep event choice.

4. Ticket incharge tells customer available seats and prices.

5. Customer tells rep seating choice.

6. Ticket incharge reserves seats.

7. Ticket incharge asks for credit card and billing address.

8. Customer gives requested information.

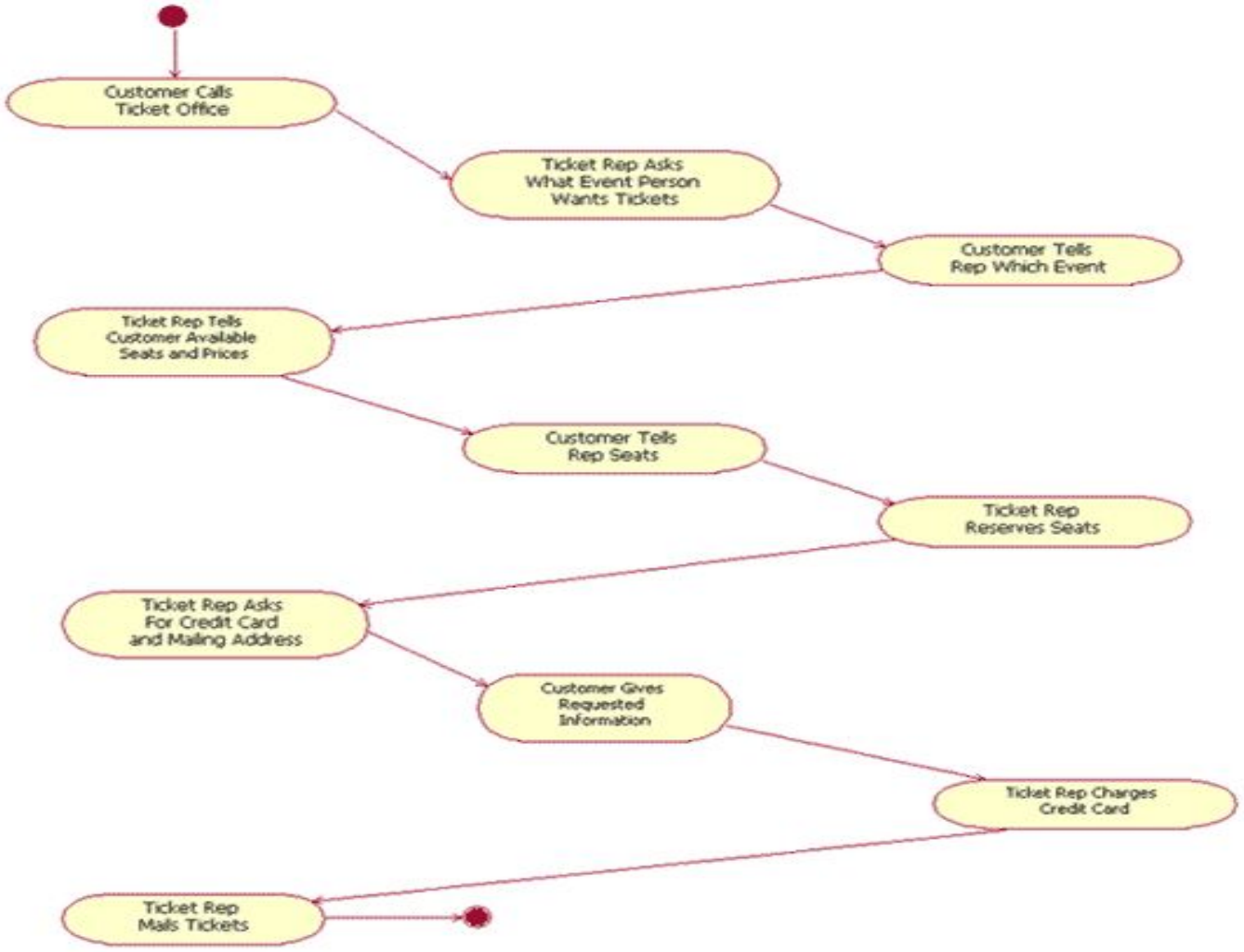
9. Ticket incharge charges credit card.

10. Ticket incharge mails tickets.

ACTIVITY DIAGRAM

Activity Model: The Initial State

A complete activity diagram makes the sequence of actions easy to understand



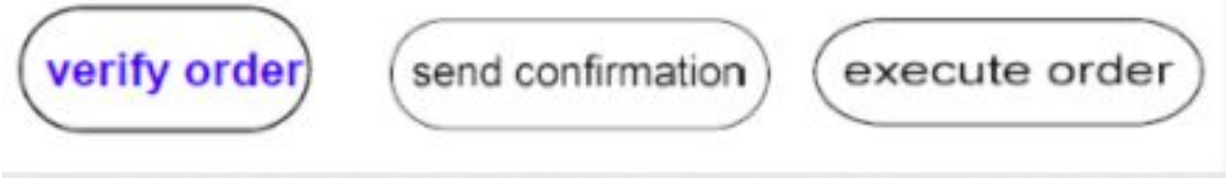
ACTIVITY DIAGRAM

Activity Diagram: Activities

The step of activity diagram are operations. Activity is a sequence of operations that takes time to complete.

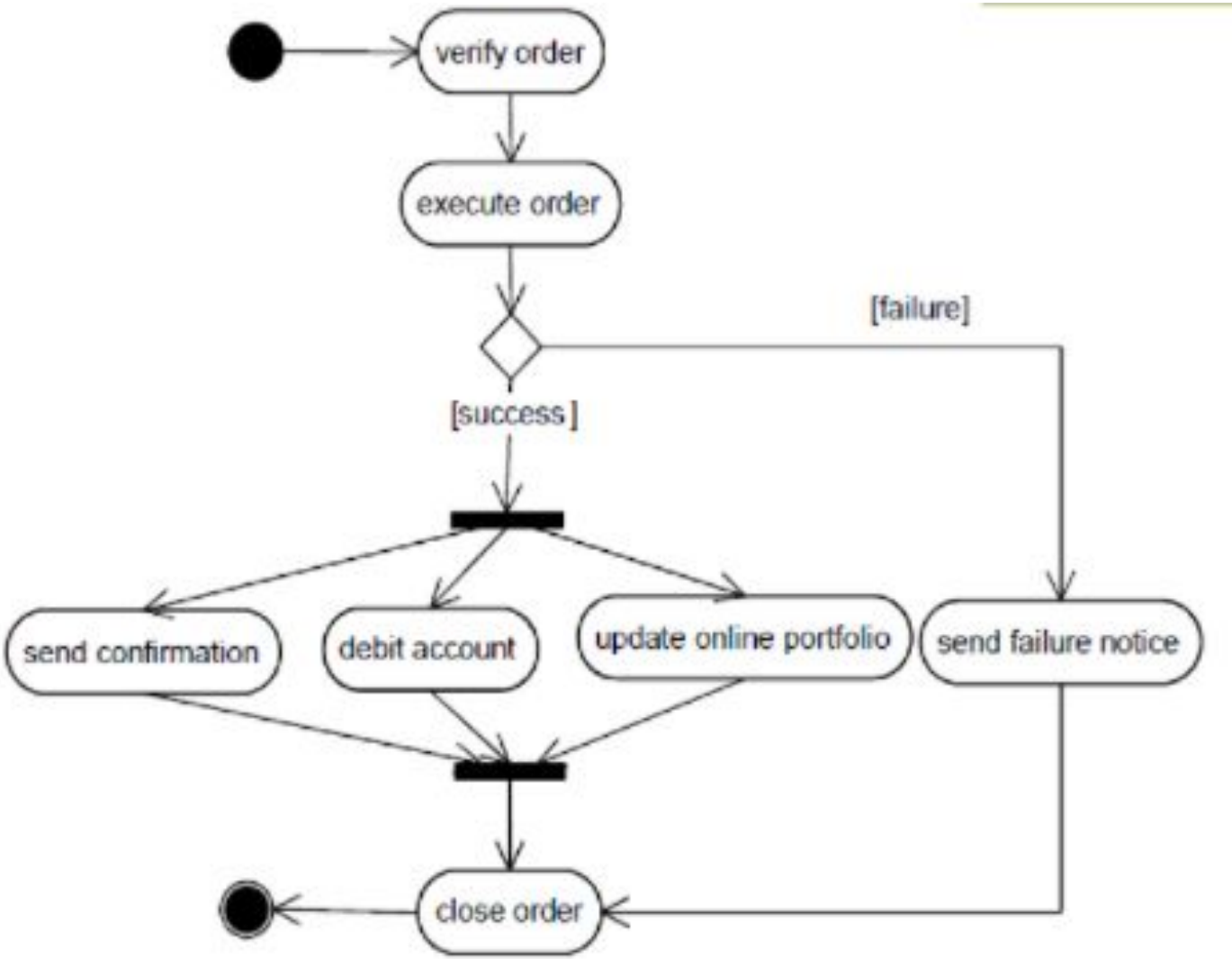
Most activities complete their work & terminate by themselves.

Termination of activity means completion of event & starting of next activity.



ACTIVITY DIAGRAM

Activity Diagram for Stock Trade Processing



ACTIVITY DIAGRAM

Activity Models:

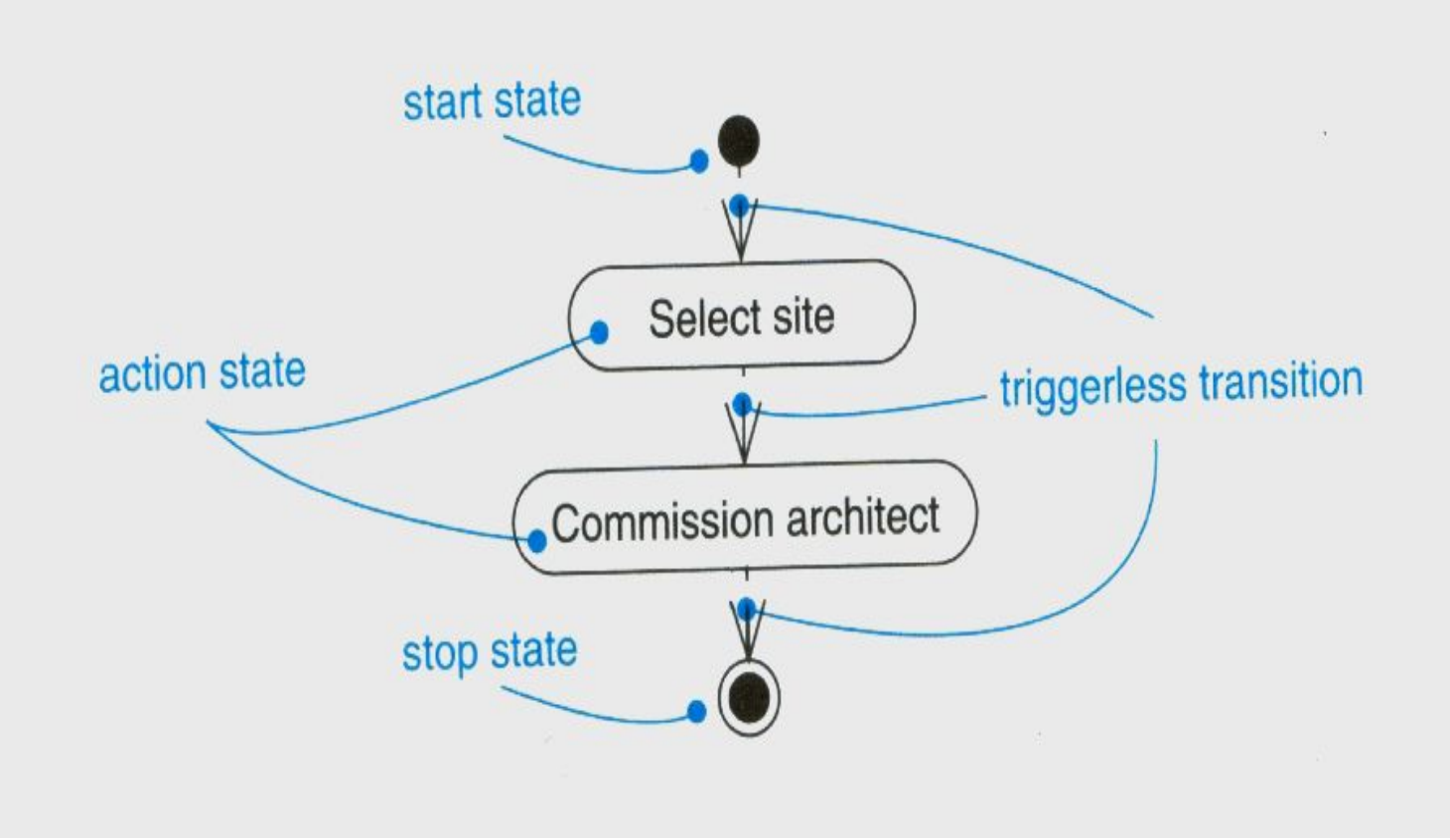
Activity diagrams commonly contain:

- **Action states** are atomic and cannot be decomposed. Work of the action state is not interrupted
- **Activity states** can be further decomposed
    - Their activity being represented by other activity diagrams. They may be interrupted.
- **Transitions**: When the action or activity of a state completes, flow of control passes immediately to the next action or activity state. A flow of control has to start and end someplace.
- **Objects:** An object node is an abstract activity node that helps to define the object flow in an activity.

ACTIVITY DIAGRAM

Activity Models:

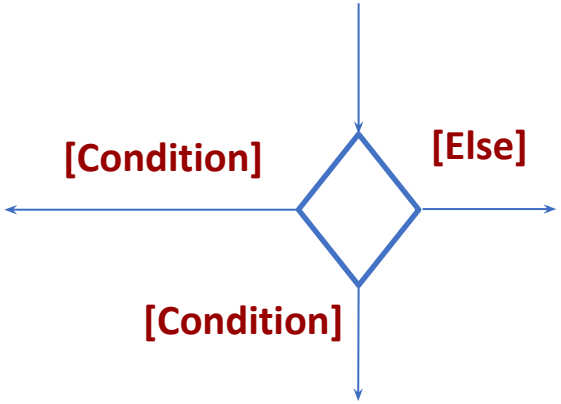
When the action or activity of a state completes, flow of control passes immediately to the next action or activity state

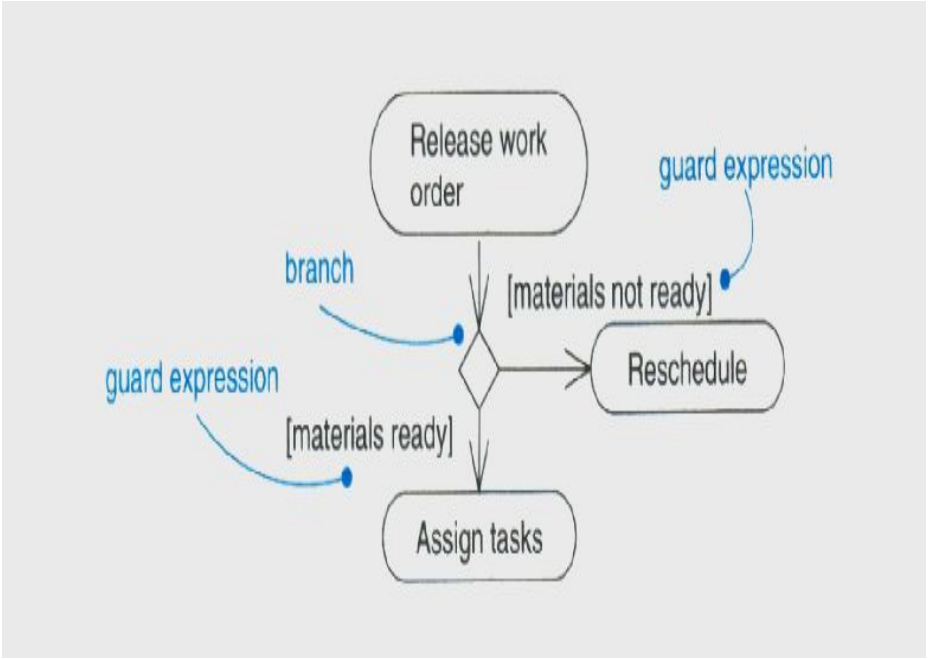


ACTIVITY DIAGRAM

Activity Models: Branching

- A branch specifies alternate paths taken based on some Boolean expression.
- A branch may have one incoming transition and two or more outgoing ones.

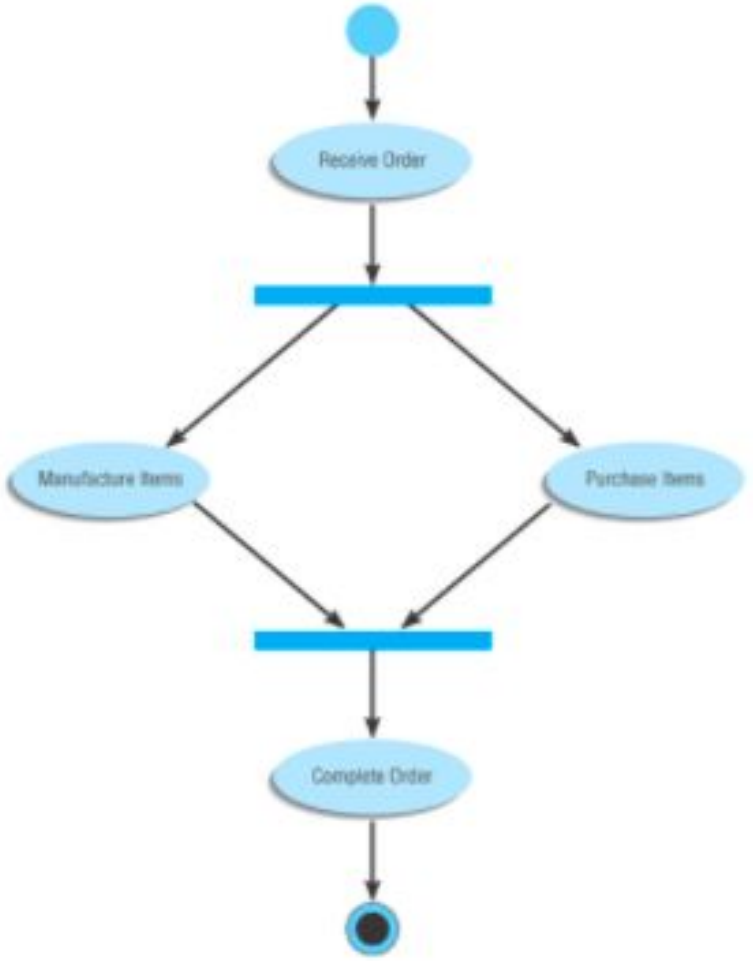




ACTIVITY DIAGRAM

Activity Models: Join and Fork

- Use a synchronization bar to specify the forking and joining of parallel flows of control.
- A synchronization bar is rendered as a thick horizontal or vertical line.
- Activity diagram with synchronization bars
- Top synchronization bar is a fork.
- Bottom synchronization bar is a join.



ACTIVITY DIAGRAM

Activity Models: Fork and Join

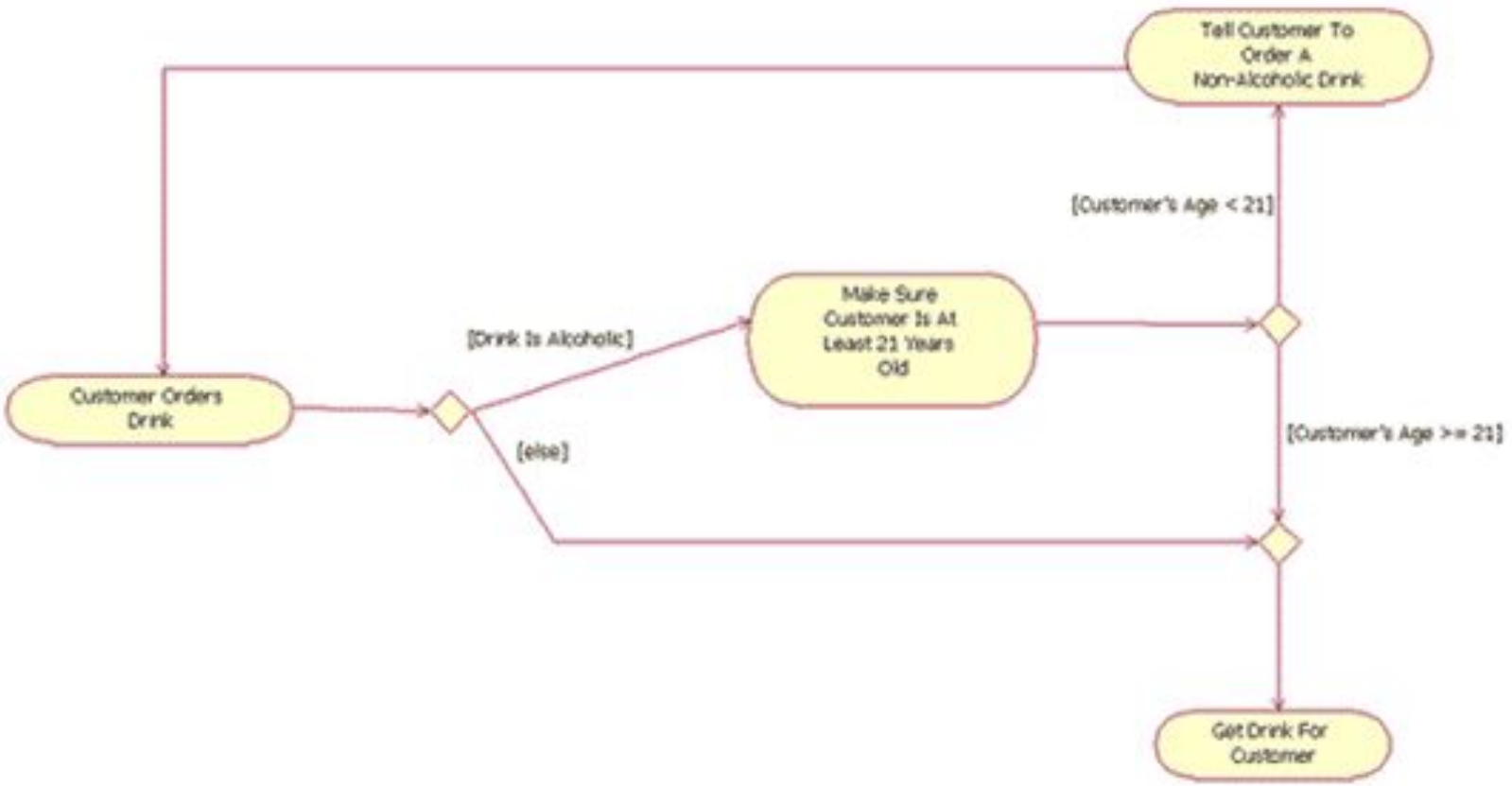
A fork may have one incoming transitions and two or more outgoing transitions. Each transition represents an independent flow of control. Conceptually, the activities of each of outgoing transitions are concurrent. Either truly concurrent (multiple nodes) or Sequential yet interleaved (one node)

A join may have two or more incoming transitions and one outgoing transition. Above the join, the activities associated with each of these paths continues in parallel. At the join, the concurrent flows synchronize. Each waits until all incoming flows have reached the join, at which point one flow of control continues on below the join.

ACTIVITY DIAGRAM

Merge Points

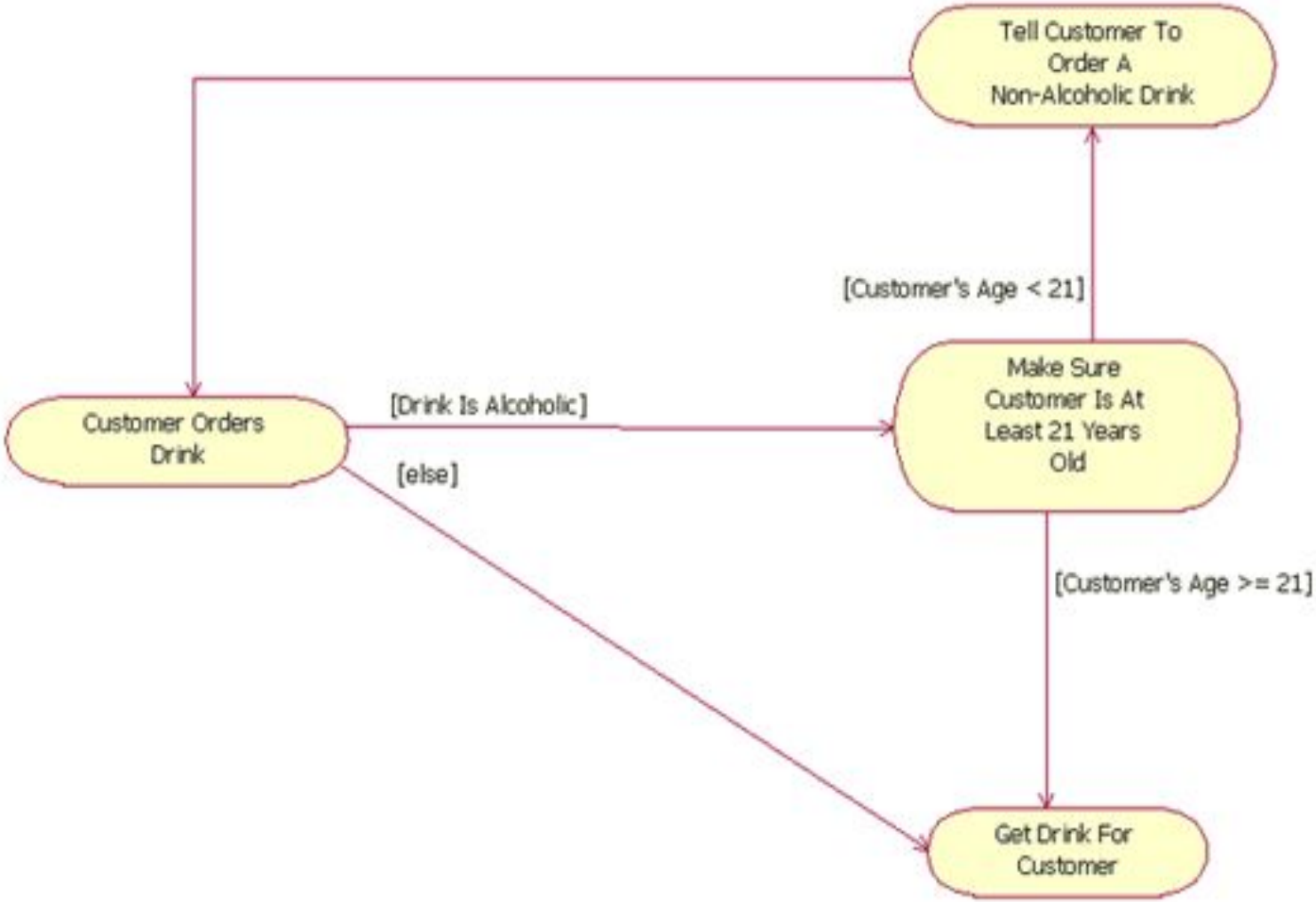
- A partial activity diagram, showing two decision points ("Drink is alcoholic" and "Customer's age < 21") and one merge ("else" and "Customer's age >= 21")



ACTIVITY DIAGRAM

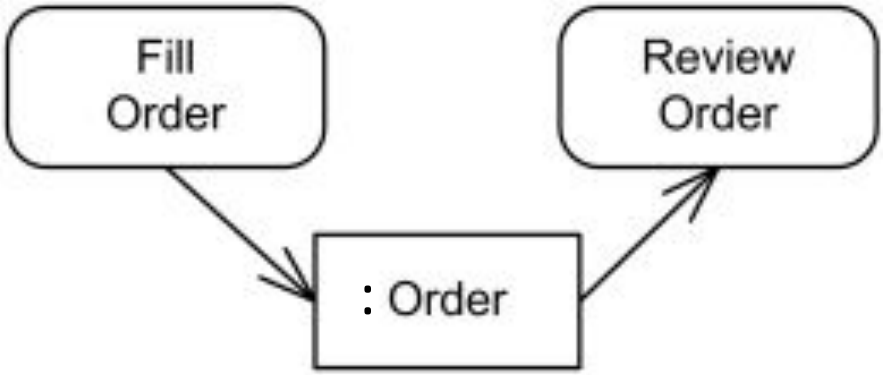
Without Merge

- The second approach to modelling decisions



Activity Model: Object & Object Flow

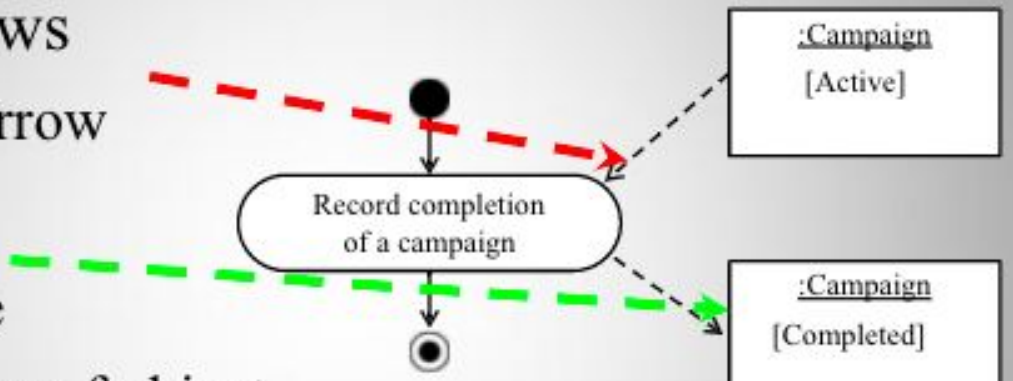
- Object flow refers to the creation and modification of objects by activities.
- An object flow arrow from an action to an object means that the action creates or influences the object.
- An object flow arrow from an object to an action indicates that the action state uses the object.



ACTIVITY DIAGRAM

Activity Model: Object & Object Flow

- Object flows
  - dashed arrow
- Objects
  - rectangle
  - with name of object underlined
  - optionally shows the state of the object in square brackets



ACTIVITY DIAGRAM

Guidelines for Activity Diagram

- **Don’t misuse activity diagram** :- Elaborate use case & sequence model to study algorithm & workflow.

- **Level diagram :-** Activities on a diagram should be consistent level of detail. place additional details for activity in separate diagram.

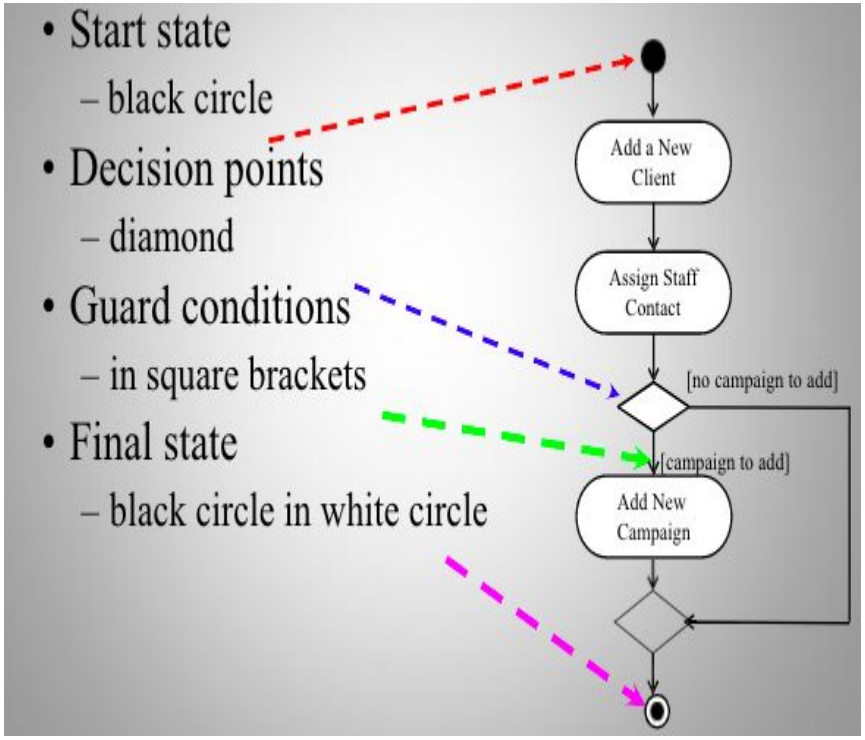
- **Be careful with branches & conditions**:- At least one condition should be satisfied.

- **Be careful with concurrent activities** :- Before a merge can happen all inputs must first complete.

ACTIVITY DIAGRAM

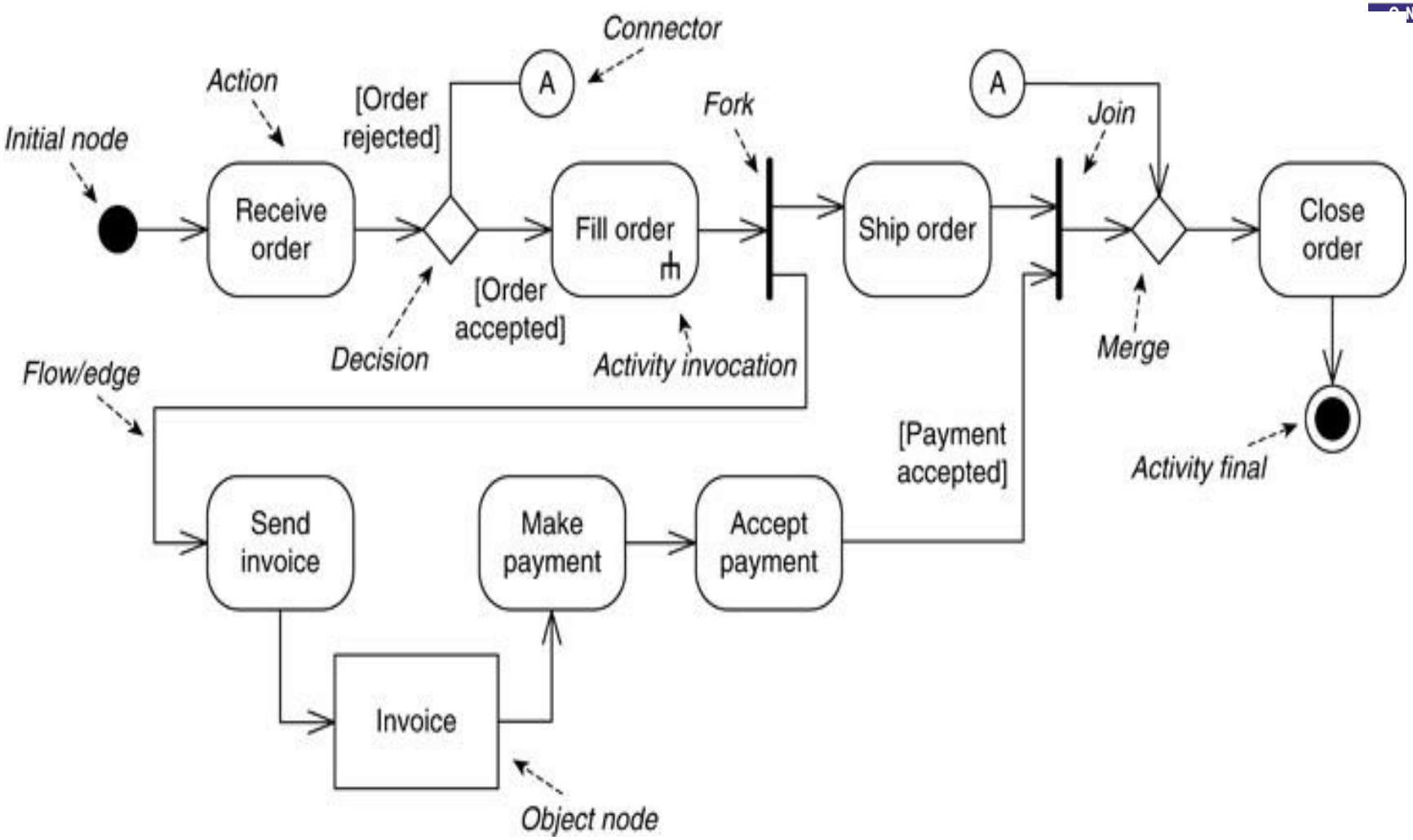
Steps to draw Activity Diagram

- Identify activities.
- Organise the activities in order with transitions.
- Identify any alternative transitions and the conditions on them.
- Add transitions and guard conditions to the diagram.
- Identify any processes that are repeated.
- Add transitions and guard conditions to the diagram.



ACTIVITY DIAGRAM

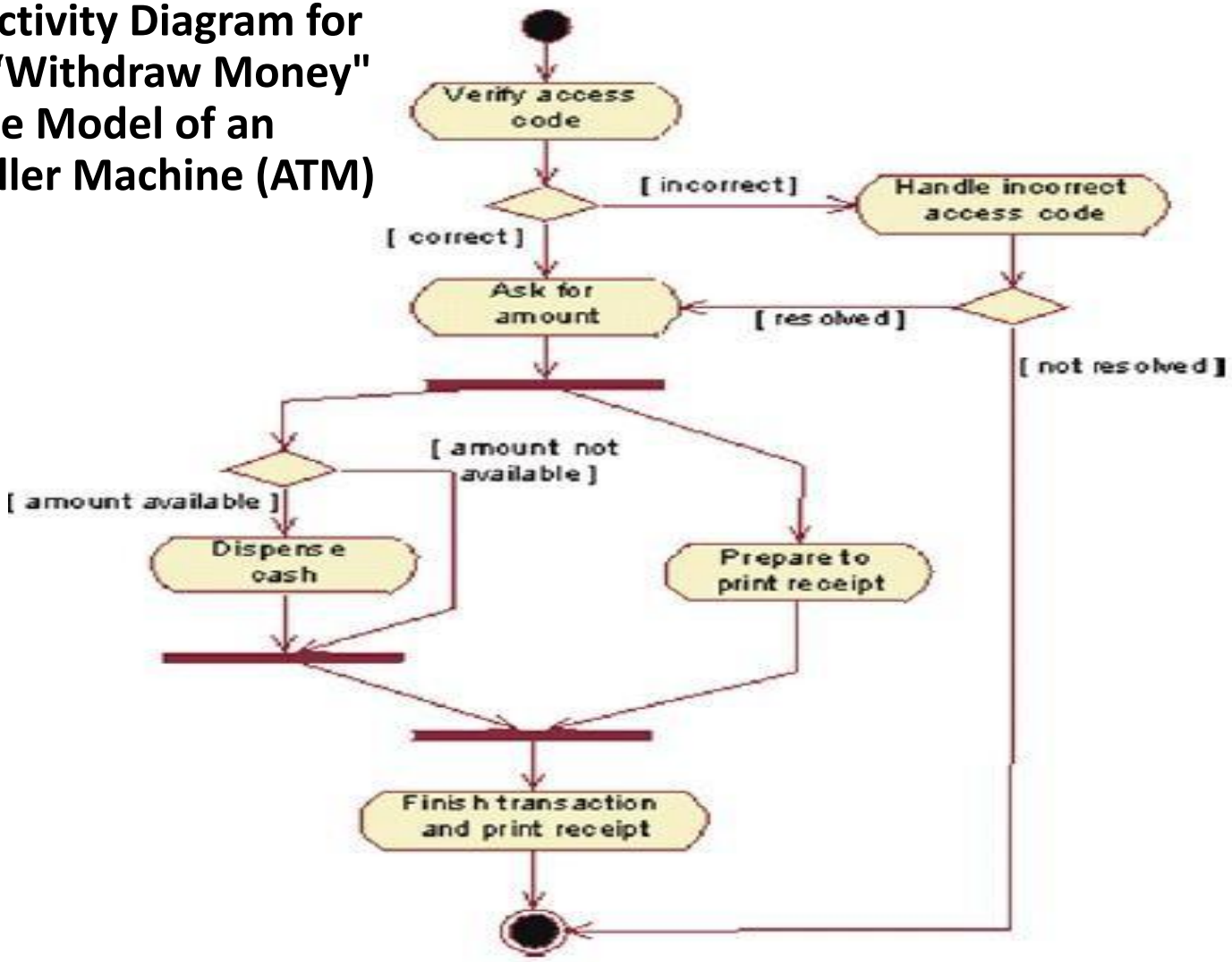
Activity Diagram Example



ACTIVITY DIAGRAM

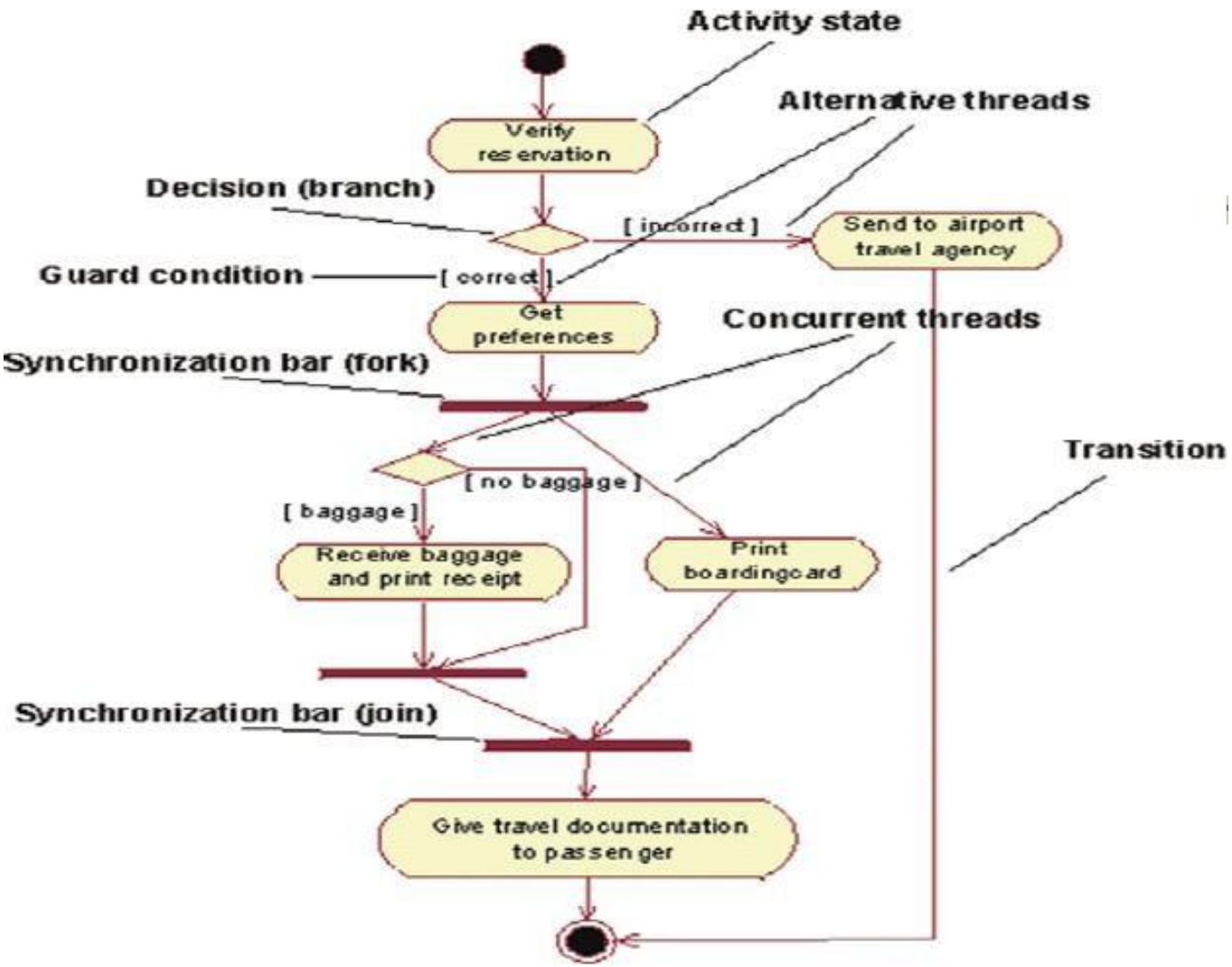
Activity Diagram of ATM

A Simplified Activity Diagram for the Use Case “Withdraw Money" in the Use-Case Model of an Automated Teller Machine (ATM)



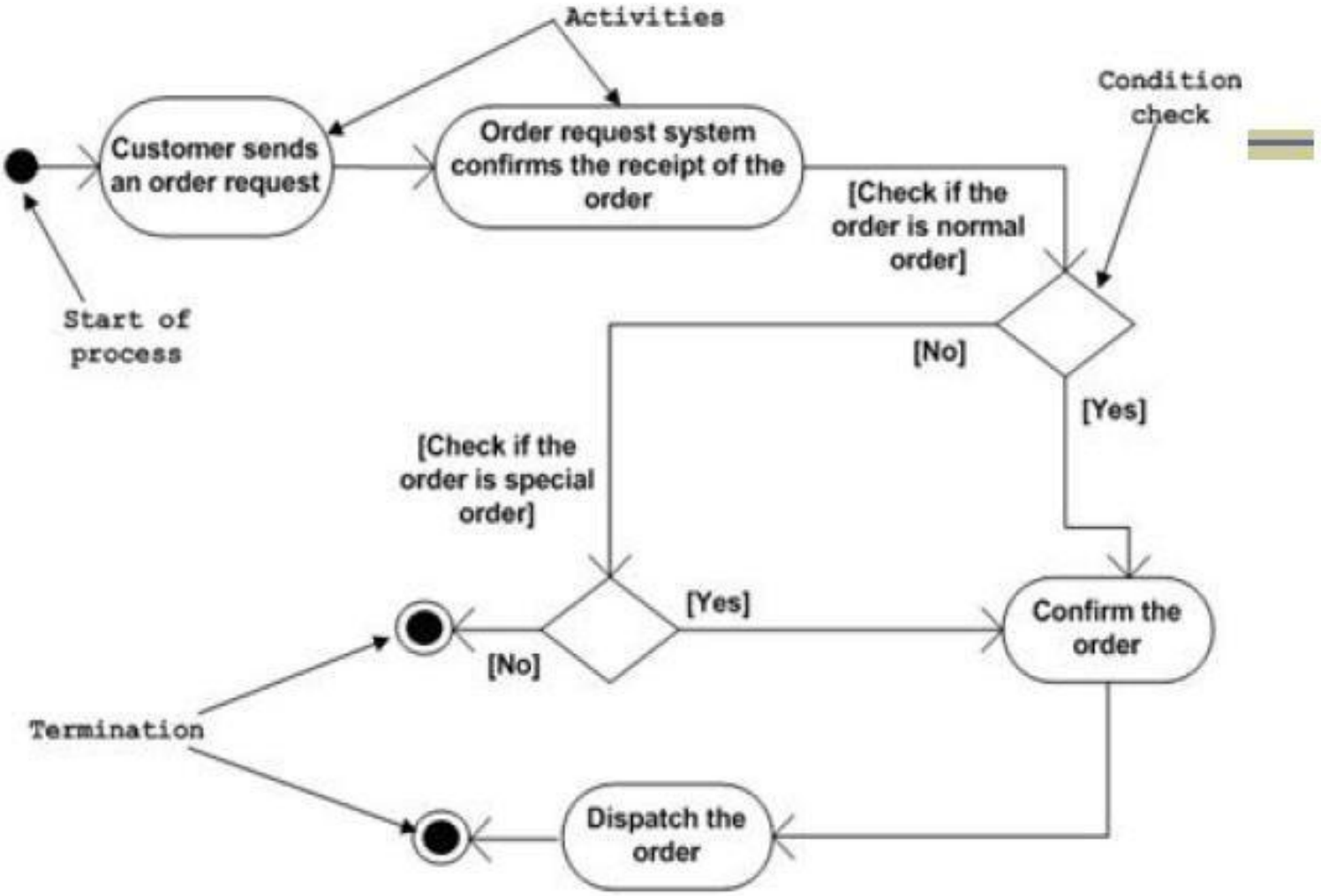
ACTIVITY DIAGRAM

Activity Diagram of Travel Reservation



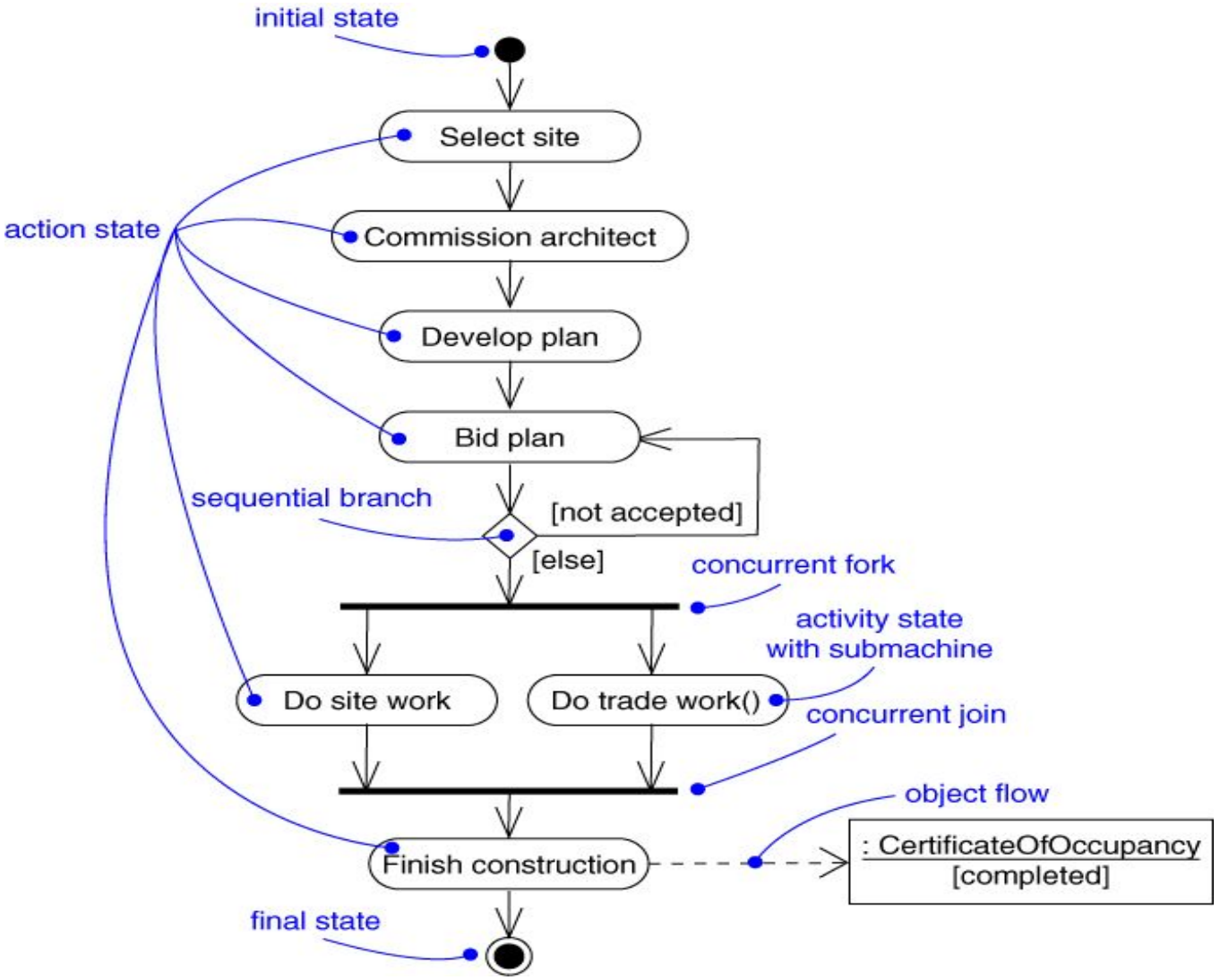
ACTIVITY DIAGRAM

Activity Diagram of Order Management System



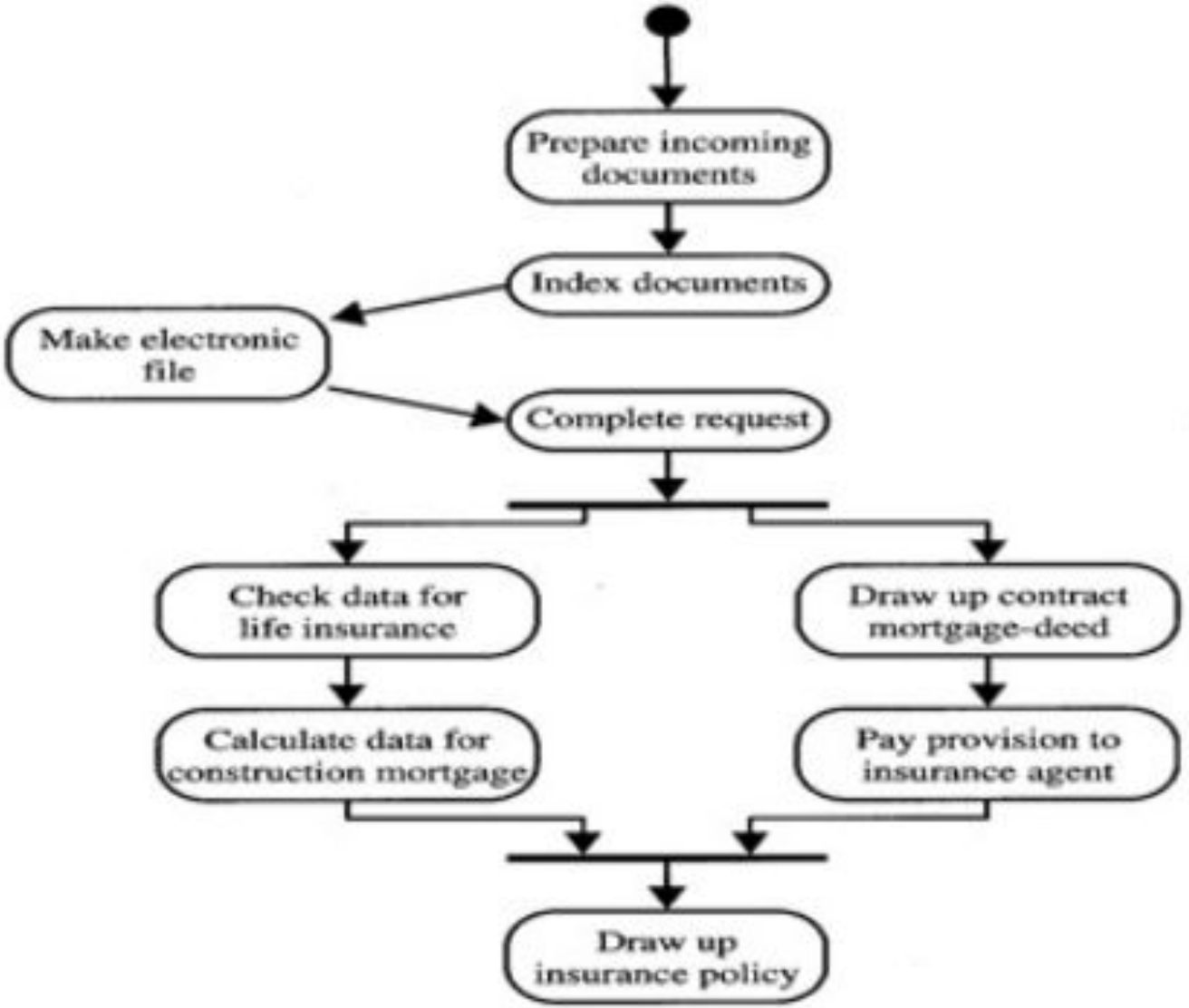
ACTIVITY DIAGRAM

Activity Diagram of Site Construction



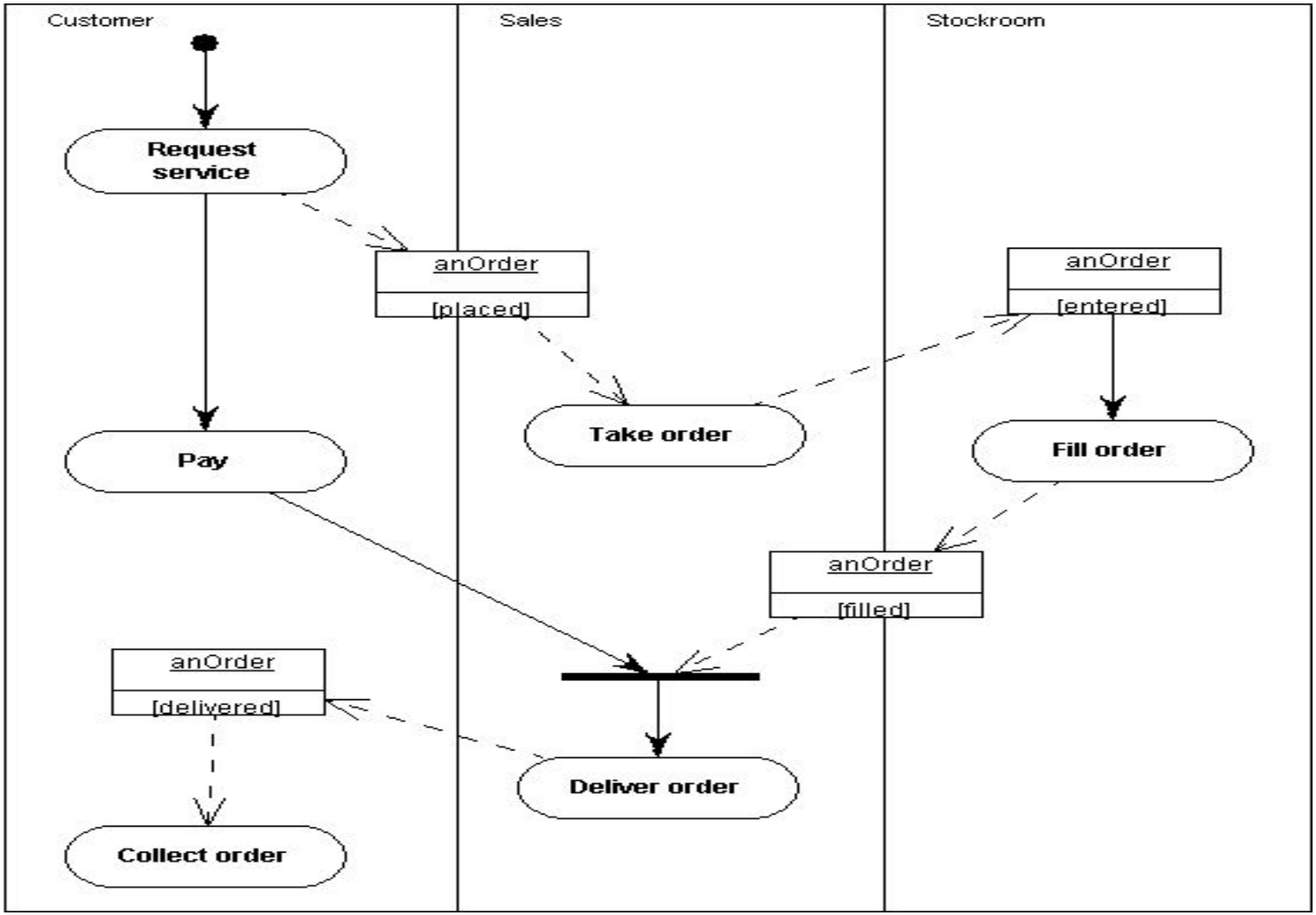
ACTIVITY DIAGRAM

Activity Diagram of Processing Mortgage Requests



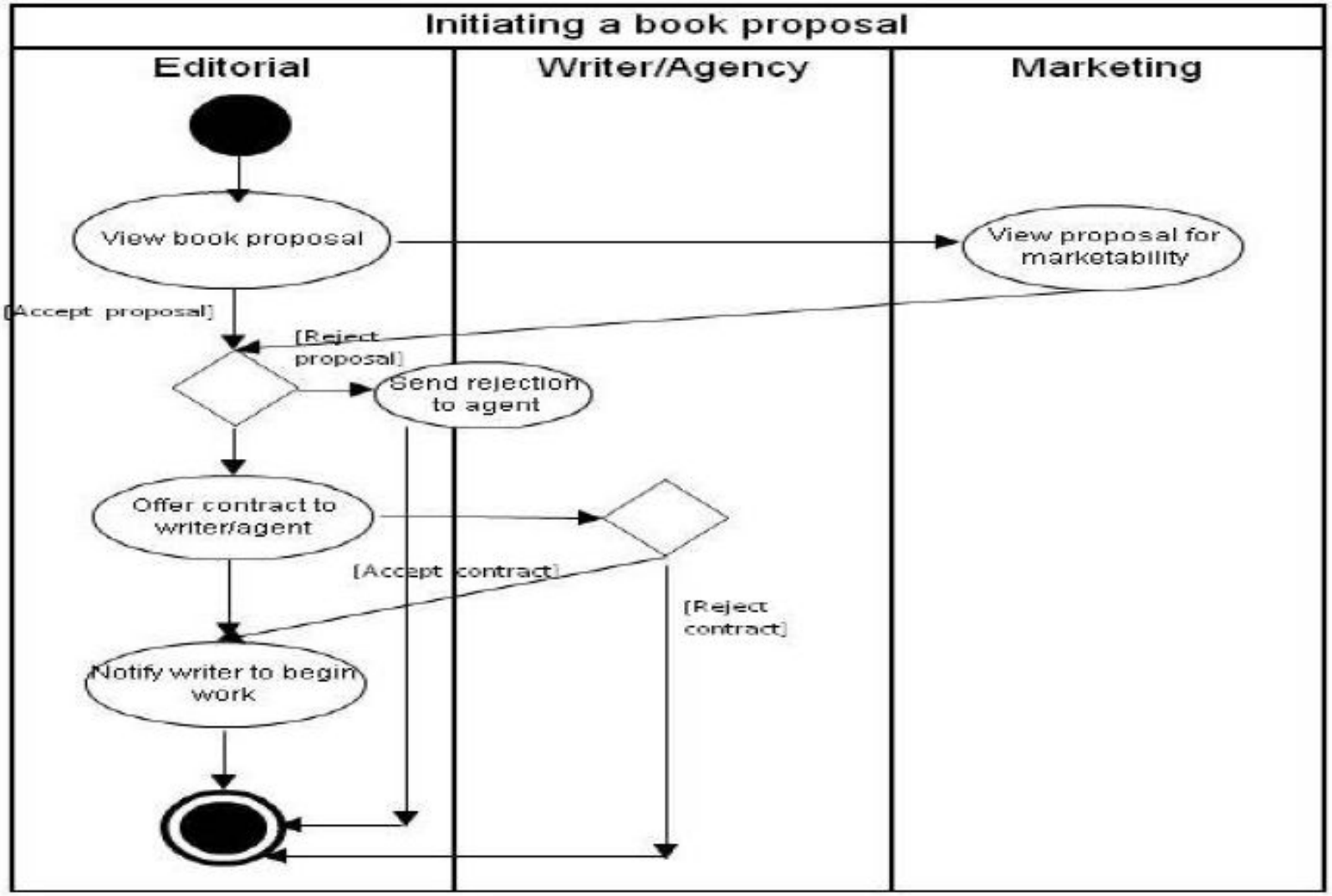
ACTIVITY DIAGRAM

Activity Diagram of Processing Stock Order



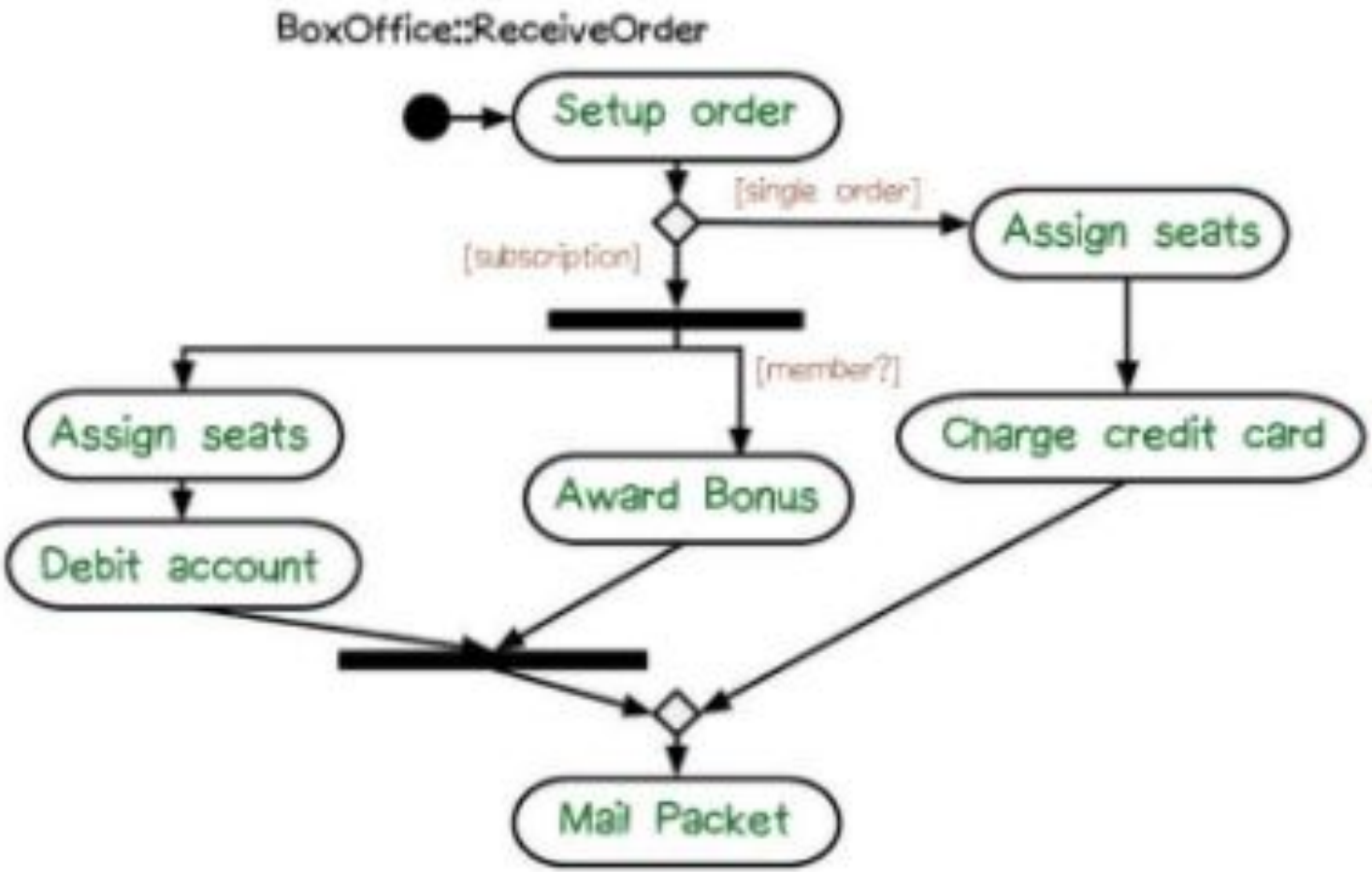
ACTIVITY DIAGRAM

Activity Diagram of Book Proposal



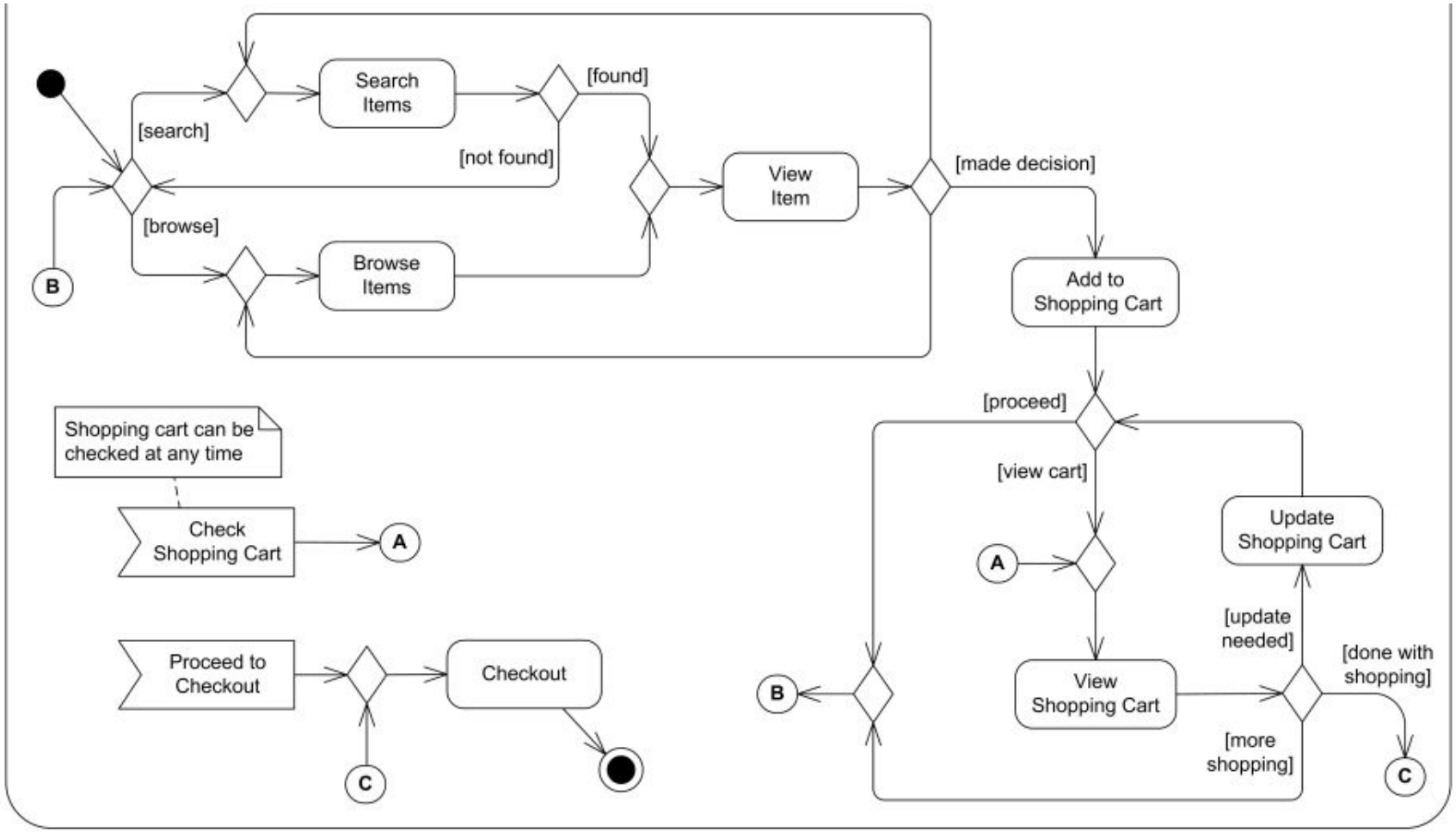
ACTIVITY DIAGRAM

Activity Diagram of Box Office Receive Order



ACTIVITY DIAGRAM

Activity Diagram of Online Shopping



ACTIVITY DIAGRAM

References:

- Object - Oriented Modeling and Design With UML by RUMBAUGH and BLAHA

- https://www.uml-diagrams.org

THANK YOU

Prof. Samatha R S and Prof. Vinay J
Department of Computer Science and Engineering
samathaswamy@pes.edu
vinayj@pes.edu