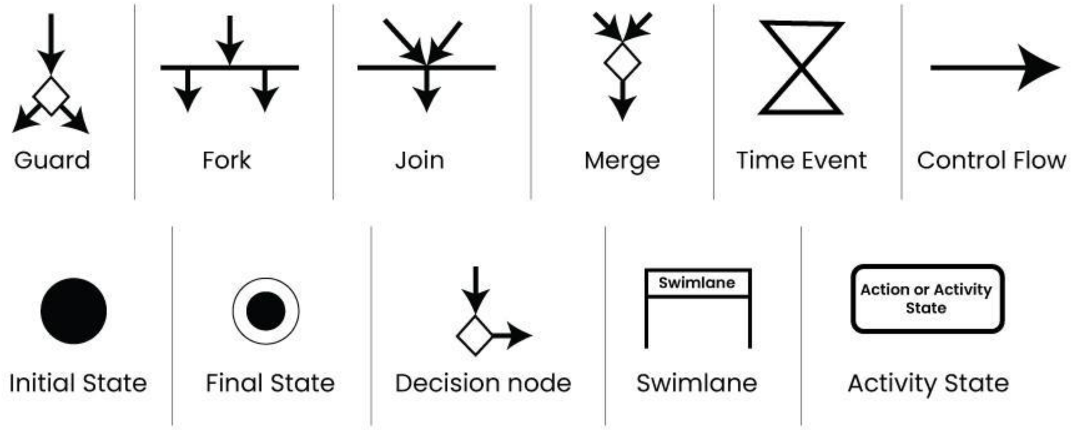
Activity and State Transition Diagram

Activity diagrams are an essential part of the Unified Modeling Language (UML) that help visualize workflows, processes, or activities within a system. They depict how different actions are connected and how a system moves from one state to another. By offering a clear picture of both simple and complex workflows, activity diagrams make it easier for developers and stakeholders to understand how various elements interact in a system.

When to use Activity Diagram?

Activity diagrams are useful in several scenarios, especially when you need to visually represent the flow of processes or behaviors in a system. Here are key situations when you should use an activity diagram:

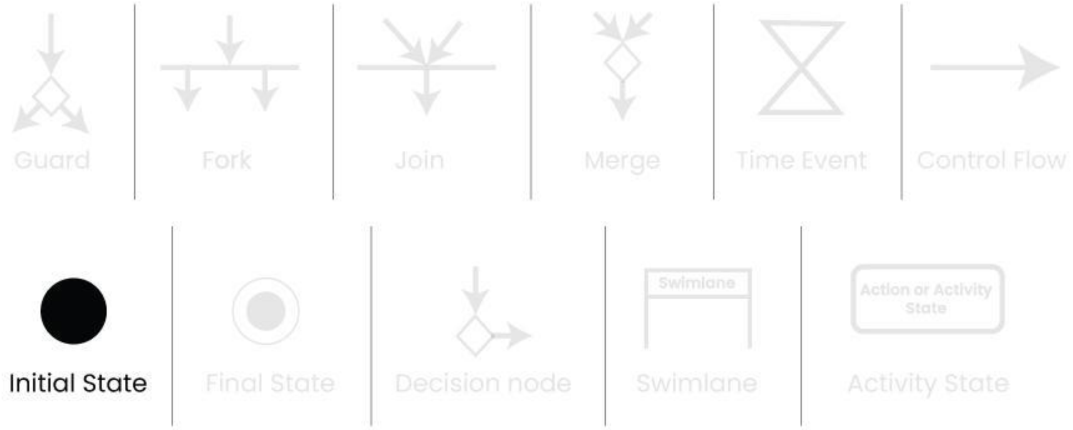
1. **Modeling Workflows or Processes**: When you need to map out a business process, workflow, or the steps involved in a use case, activity diagrams help visualize the flow of activities.
2. **Concurrent or Parallel Processing**: If your system or process involves activities happening simultaneously, an activity diagram can clearly show the parallel flow of tasks.
3. **Understanding the Dynamic Behavior**: When it's essential to depict how a system changes over time and moves between different states based on events or conditions, activity diagrams are effective.
4. **Clarifying Complex Logic**: Use an activity diagram to simplify complex decision-making processes with branching paths and different outcomes.
5. **System Design and Analysis**: During the design phase of a software system, activity diagrams help developers and stakeholders understand how different parts of the system interact dynamically.
6. **Describing Use Cases**: They are useful for illustrating the flow of control within a use case, showing how various components of the system interact during its execution.



Activity Diagram Notations

1. Initial State

The starting state before an activity takes place is depicted using the initial state.



Initial State

A process can have only one initial state unless we are depicting nested activities. We use a black filled circle to depict the initial state of a system. For objects, this is the state when they are instantiated. The Initial State from the UML Activity Diagram marks the entry point and the initial Activity State.

Action or Activity State

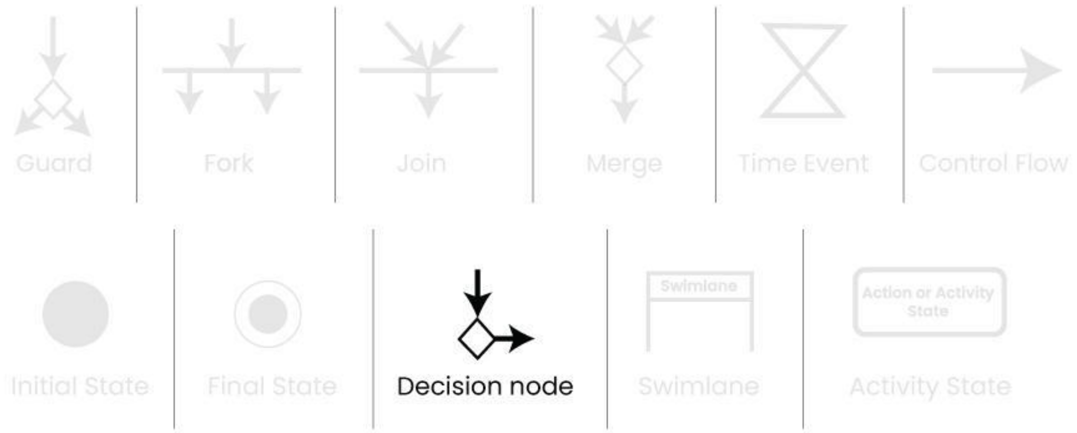
An activity represents execution of an action on objects or by objects. We represent an activity using a rectangle with rounded corners. Basically any action or event that takes place is represented using an activity.

Action Flow or Control flows

Action flows or Control flows are also referred to as paths and edges. They are used to show the transition from one activity state to another activity state.

Decision node and Branching

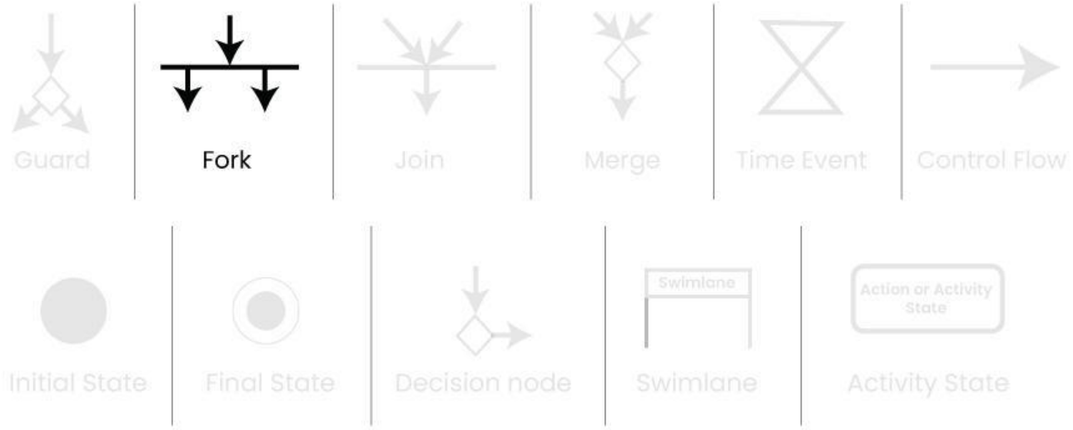
When we need to make a decision before deciding the flow of control, we use the decision node. The outgoing arrows from the decision node can be labelled with conditions or guard expressions. It always includes two or more output arrows.

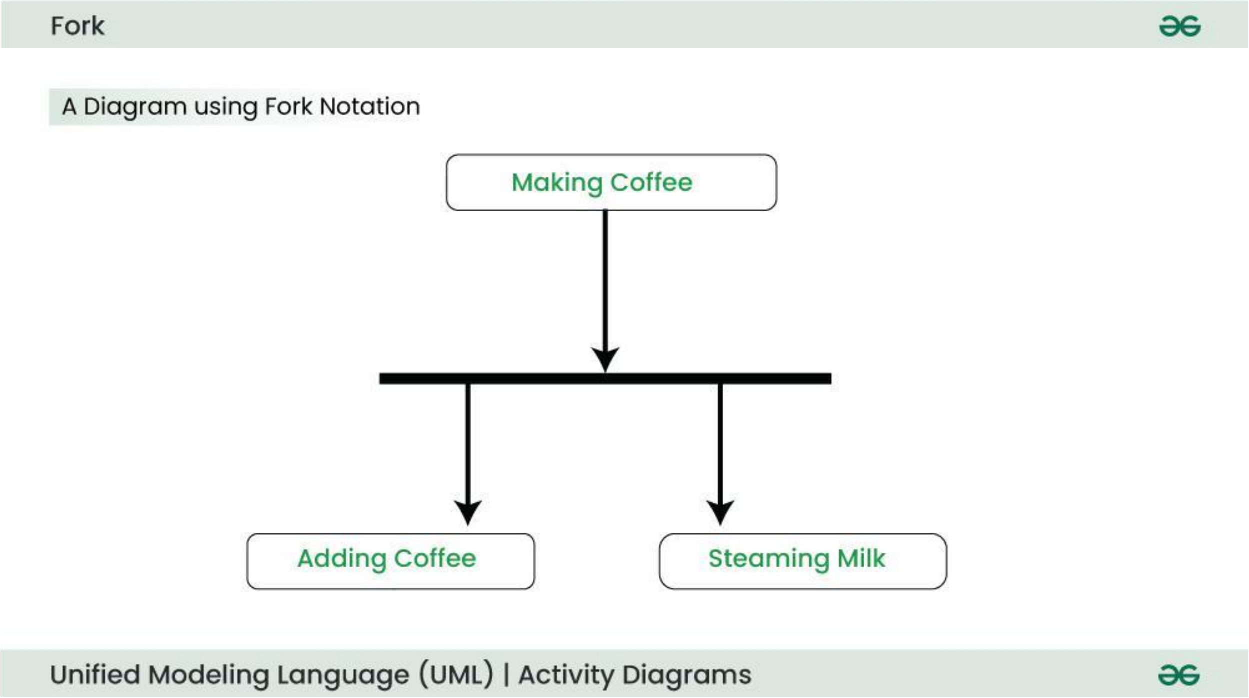


Decision Node

Fork

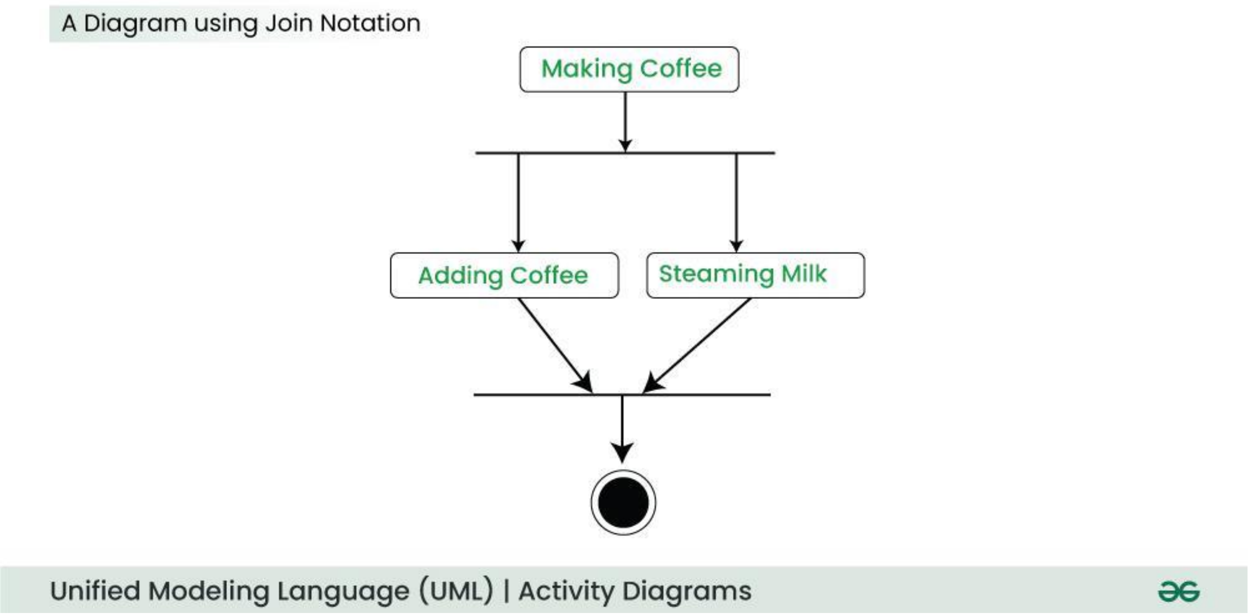
Fork nodes are used to support concurrent activities. When we use a fork node when both the activities get executed concurrently i.e. no decision is made before splitting the activity into two parts. Both parts need to be executed in case of a fork statement. We use a rounded solid rectangular bar to represent a Fork notation with incoming arrow from the parent activity state and outgoing arrows towards the newly created activities.




Fork
A Diagram using Fork Notation
Making Coffee
Adding Coffee
Steaming Milk
Unified Modeling Language (UML) | Activity Diagrams

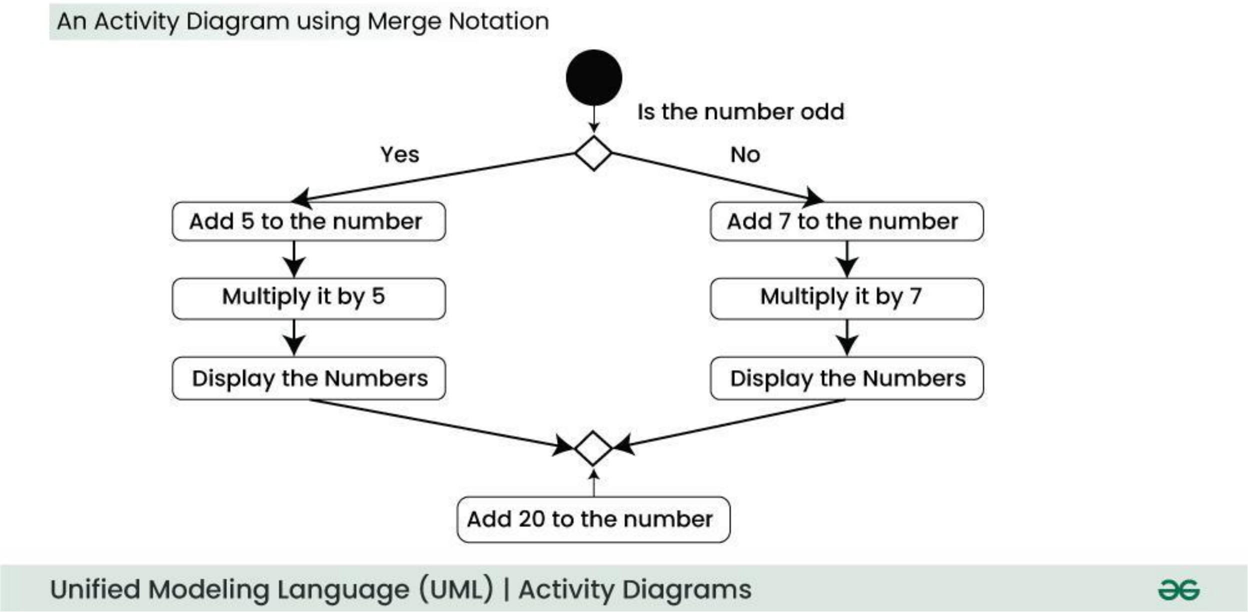
Join

Join nodes are used to support concurrent activities converging into one. For join notations we have two or more incoming edges and one outgoing edge.

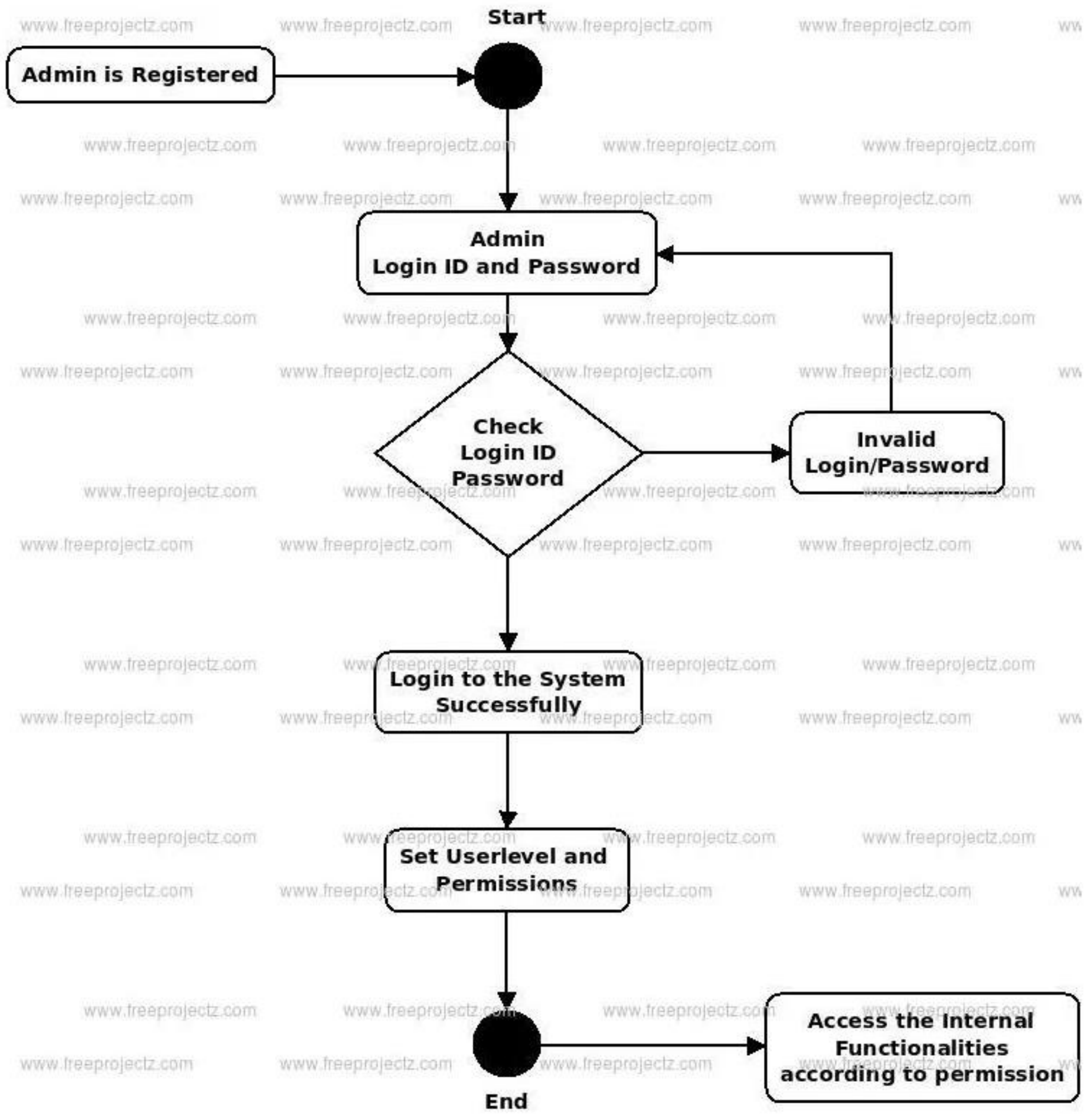

A Diagram using Join Notation
Making Coffee
Adding Coffee
Steaming Milk
Unified Modeling Language (UML) | Activity Diagrams

Merge or Merge Event

Scenarios arise when activities which are not being executed concurrently have to be merged. We use the merge notation for such scenarios. We can merge two or more activities into one if the control proceeds onto the next activity irrespective of the path chosen.

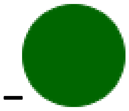
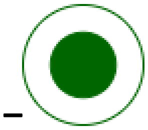
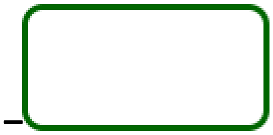
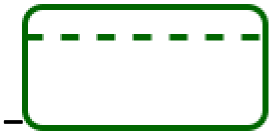


Example

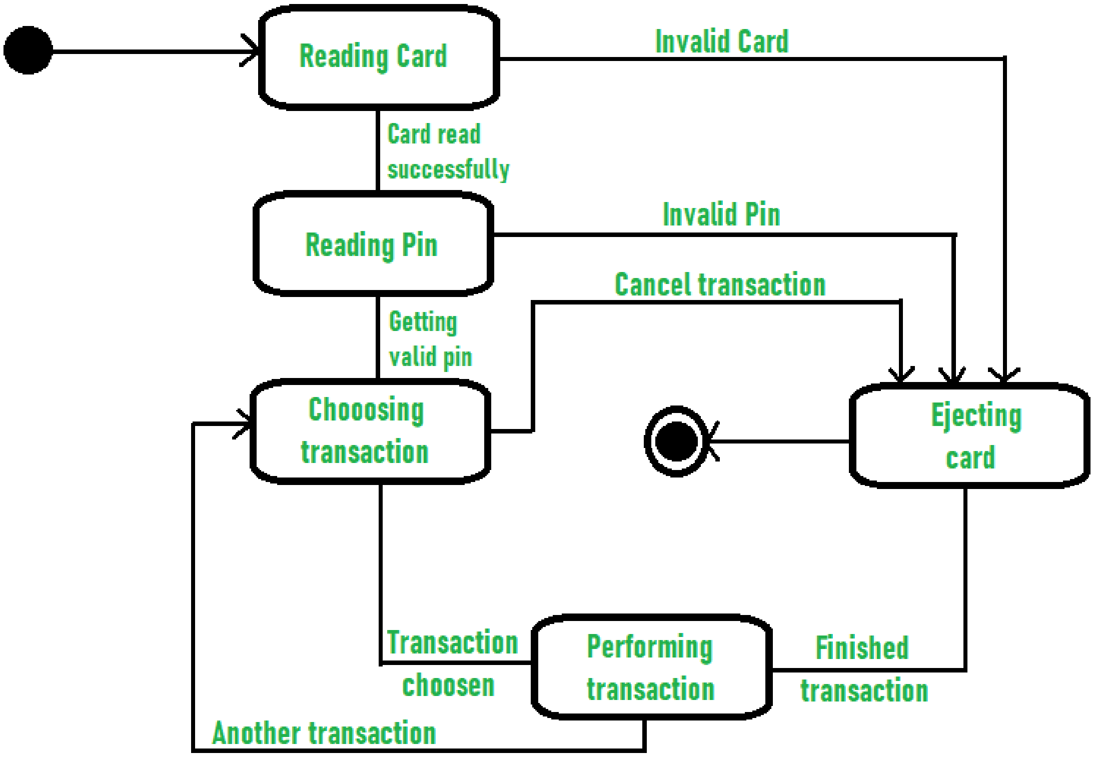


State Transition Diagram

State Transition Diagram are also known as Dynamic models. As the name suggests, it is a type of diagram that is used to represent different transition (changing) states of a System. It is generally used to graphically represent all possible transition states a system can have and model such systems. It is very essential and important and right for object-oriented modeling from the beginning. The System consists of various states that are being represented using various symbols in the state transition diagram. You can see the symbols and their description given below :

1. **Initial State –** 
2. **Final State –** 
3. **Simple State –** 
4. **Composite State –** 

Type of State	**Description**
Initial State	**In a System, it represents Starting state.**
Final State	**In a System, it represents Ending state.**
Simple State	**In a System, it represents a Simple state with no substructure.**
Composite State	**In a System, it represents a Composite state with two or more parallel or concurrent states out of which only one state will be active at a time and other states will be inactive.**



State Transition Diagram for ATM System