Activity Diagram

Learn how activity diagrams represent interactions between actors and objects.

An **activity diagram** is a communication diagram that is used to show the dynamic aspects performed by a system. This diagram is used to represent a series of actions, similar to how they may appear in a flowchart.
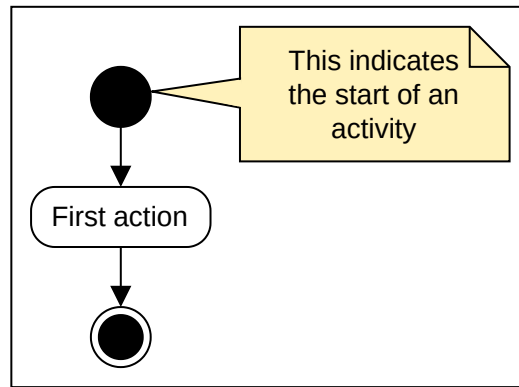
Purpose of activity diagram

Much like the previously discussed UML diagrams, activity diagrams are also used to show the sequence through which events occur in the system. Activity diagrams differ from other UML diagrams in that they specifically capture the message flow for each activity on to the next. These may look like flowcharts in appearance, but in execution, they can be much more. They can show various flows, such as single, parallel, concurrent, and branched.

Components of an activity diagram

Various components make up an activity diagram. Let's discuss some of the essential features of this diagram that appear most often.

Initial: This represents the start of the workflow of the activity diagram. They can be visualized as the node in a tree structure.
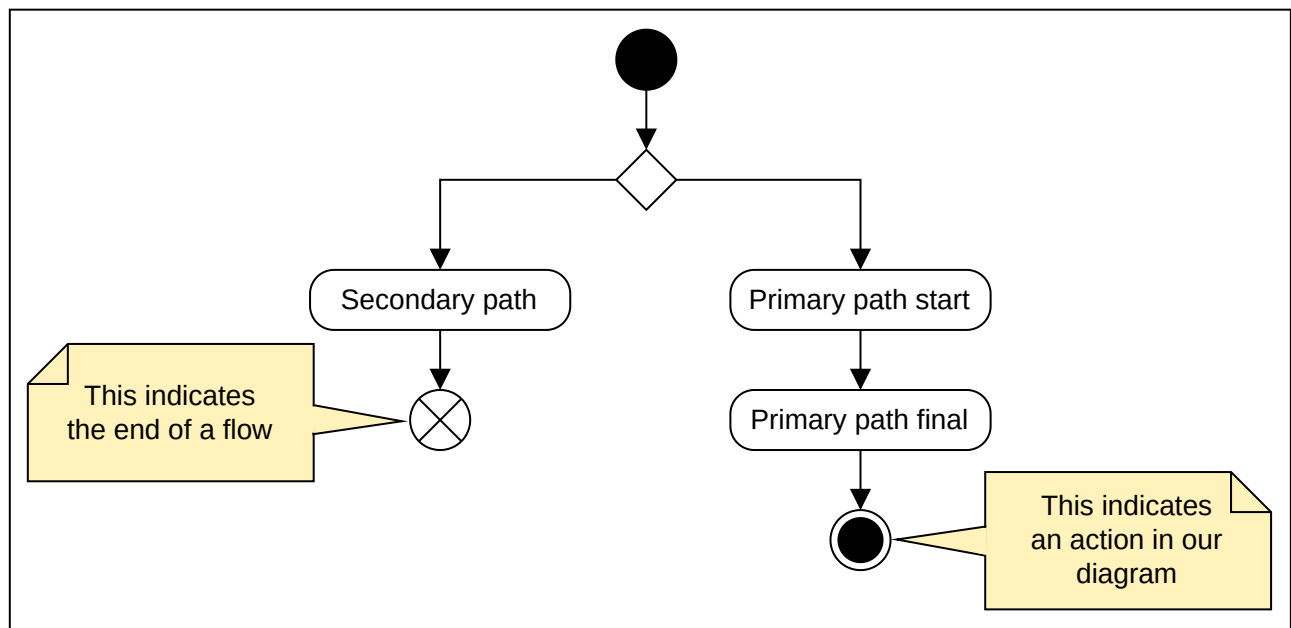

Initial node in an activity diagram

Action: These are the main building blocks of an activity diagram and are used to show the activities that a modeled process is made of.
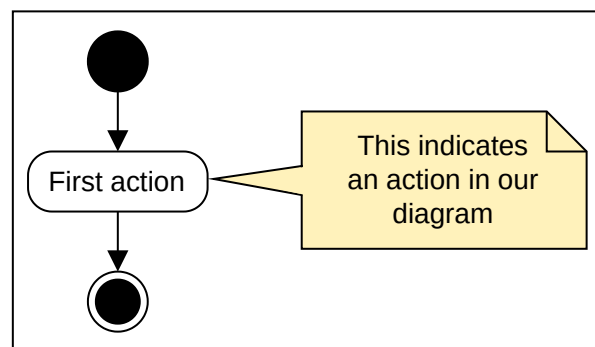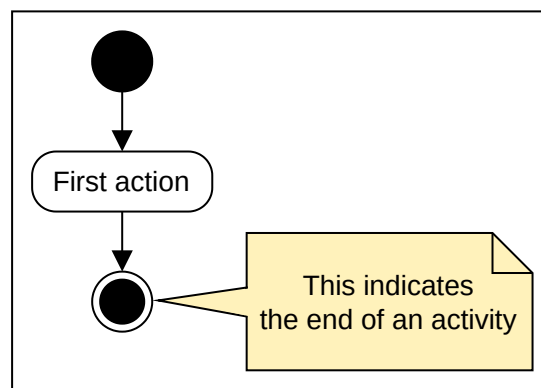

Action in an activity diagram

Flow final: This represents the end of a single path in the activity diagram. They can be visualized as a leaf in a tree structure.
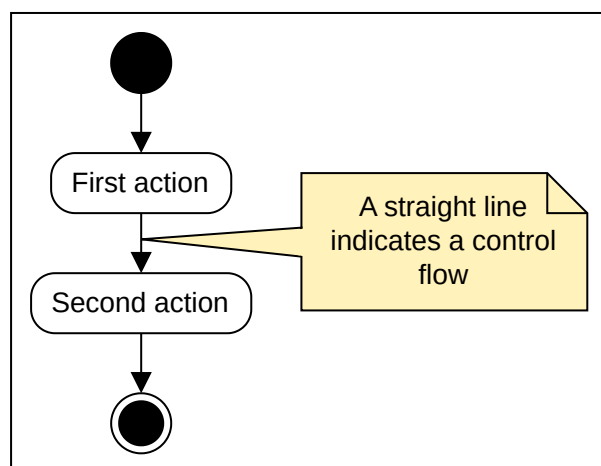
Flow final in an activity diagram

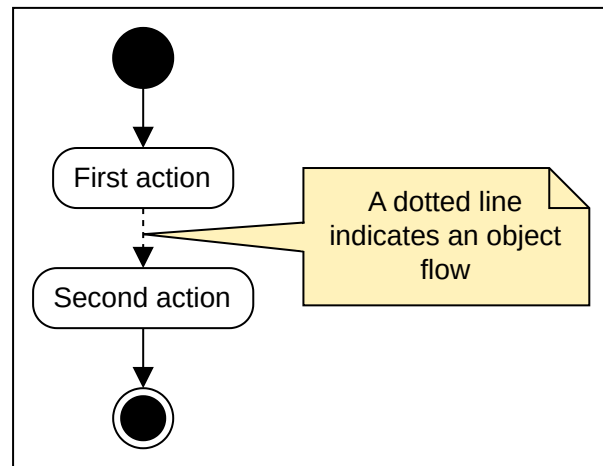**Activity final:** This represents the end of all the activities in the activity diagram.

Activity final an activity diagram

Control flow: This shows the directional flow of the diagram. This exists as a connector between one action ande another.
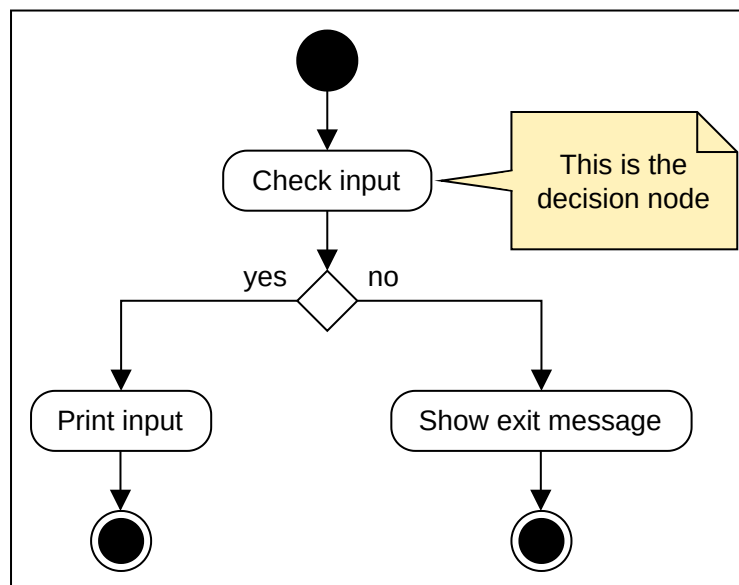
Object flow: This shows the path of the objects as they move throughout the activity.
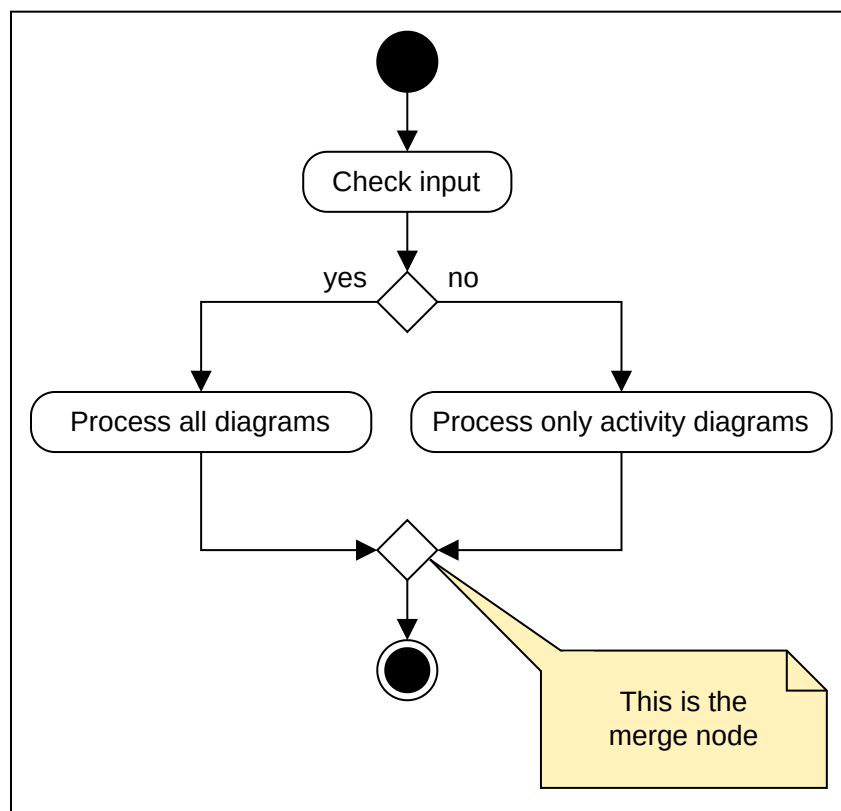

Object flow in an activity diagram

Decision: This is used to represent multiple options that are possible in a system. They appear as a branch alongside the text describing the condition for the path.
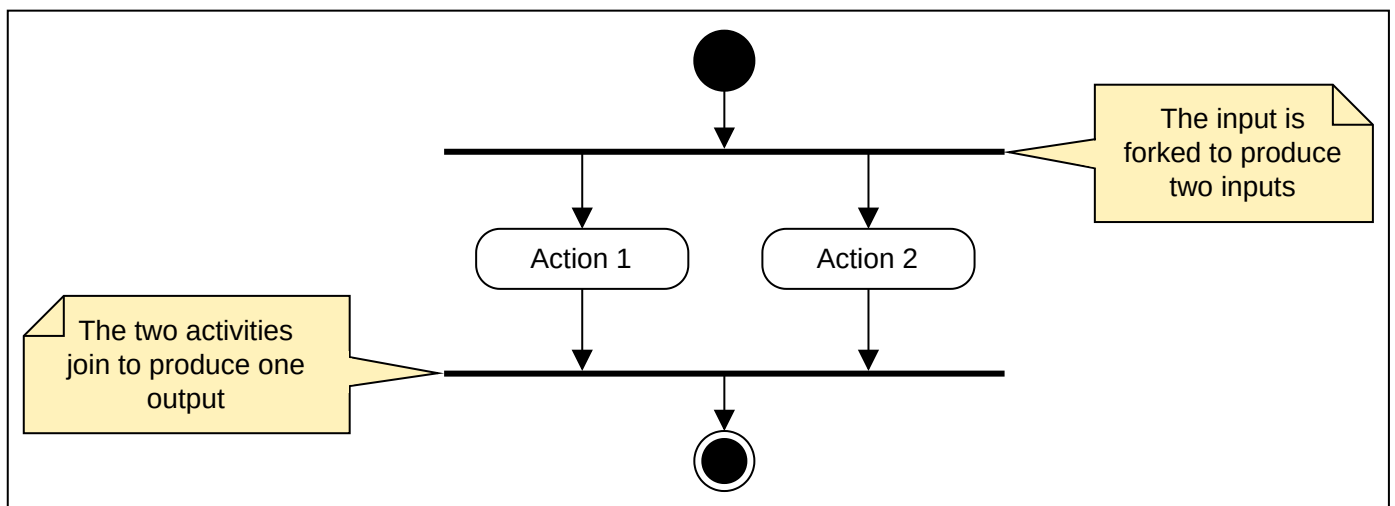

Flow final in an activity diagram

Merge: This uses the same symbol as a decision. However, this shows that multiple options join at this node, but leads to a single output.

Flow final in an activity diagram

Fork and join: The fork node represents a single activity that is split into two concurrent activities happening alongside each other. On the other hand, the join node joins two concurrent activities together to lead to a single activity.

Fork and join in an activity diagram

How to draw an activity diagram

Activity diagrams show the message flow of actions in a simple manner. Before we start, it is important to understand that there isn't one correct way of creating an

activity diagram. Some of us may have a different approach to handling these problems.

This lesson introduces a methodology that helps break down the problem into smaller, achievable tasks.

Determine the actions of the system

Here we need to recognize each action and how they interact with the others in the system. We can use the aid of use cases to make this process easier.

For now, we will use the previous example where a customer wants to withdraw cash from an ATM.
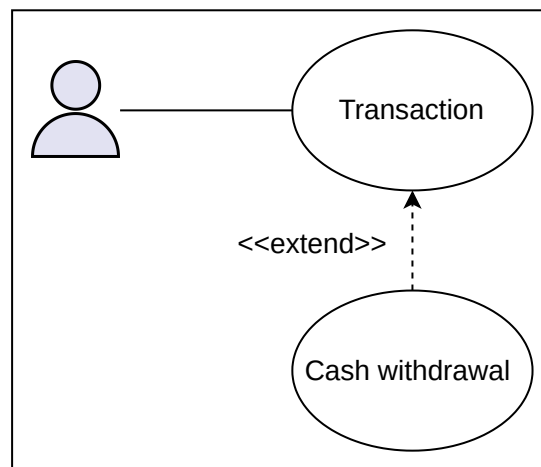
Use case for cash withdrawal

Determine the actors and their roles, and list down all the actors and objects that will be involved in the entire message flow of the activity. These would be the following:

- Customer
- ATM
- Transaction
- Account
- Cash dispenser

Finding the flow of each activity

We first work out the order in which the actions should occur, and we also note down the action that coincides with the other actions. Here, we also determine the conditions that lead to specific outcomes. Lastly, we note if there are actions that can only be executed once a previous one is completed.

Create the diagram

Now, we will use these steps to build a sample activity diagram.
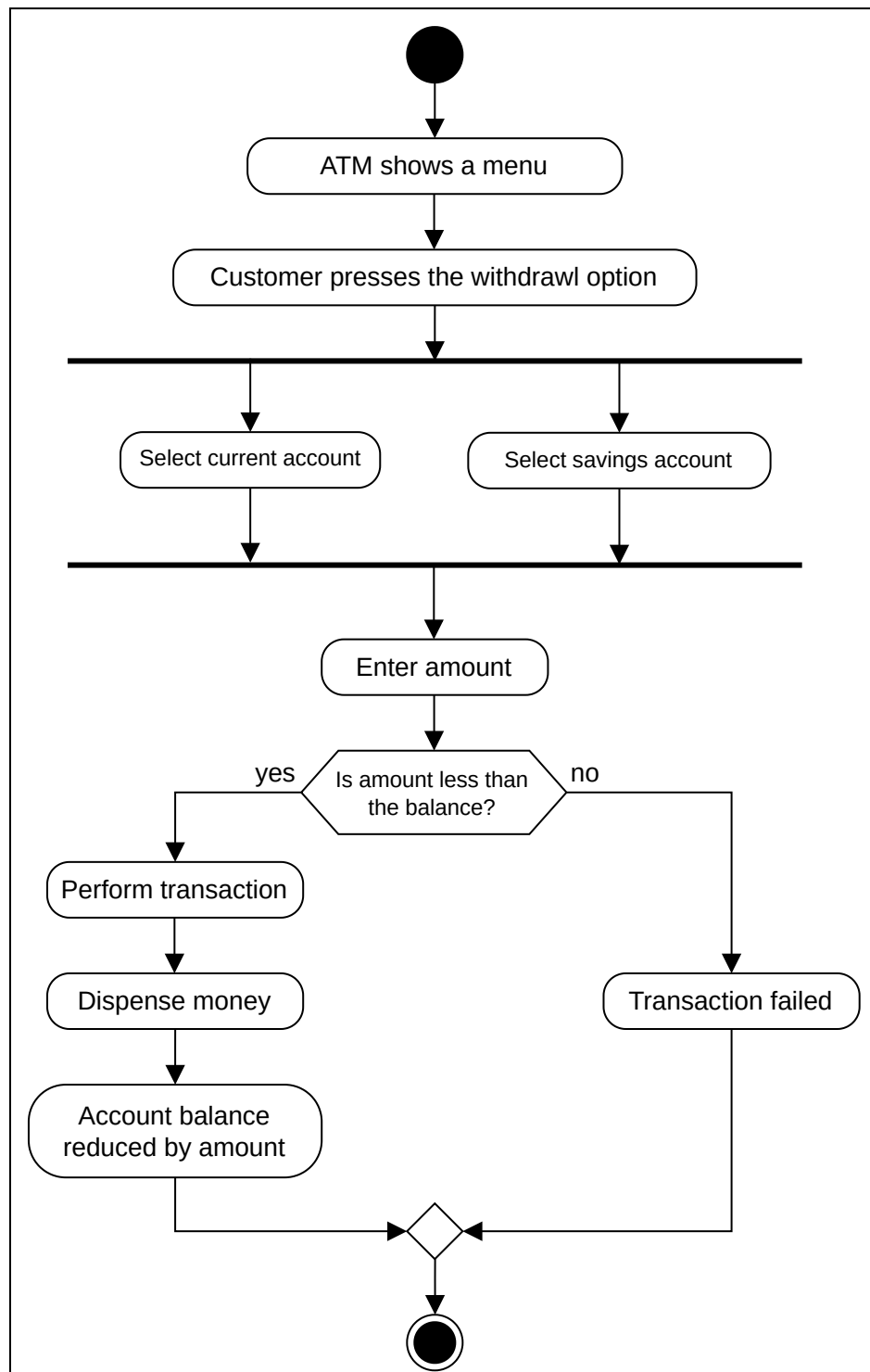
Sequence vs. activity diagram

We have now studied both the sequence and the activity diagram. As a refresher, the following table provides an overview of the differences that exist between both of them:

Sequence Diagram	Activity Diag
It is used for dynamic modeling.	It is used for functional modeling
It visualizes the flow of messages from one object to the other.	It visualizes the flow of messages other.
It helps in modeling the systems workflow.	It helps in visualizing a systems se particular functionality.
It describes the behavior of various objects just for an independent use case.	It describes the general sequence objects as well as various use case