



Activity Diagram

Written Date : September 02, 2016

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operation, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to create a workflow for an organization.

When to use Activity Diagram?

1. Represent business workflows
2. Identify candidate use cases, through the examination of business workflows
3. Identify pre- and post-conditions (the context) for use cases
4. Model workflows between/within use cases
5. Model complex workflows in operations on objects
6. Model in detail complex activities in a high level activity Diagram

Basic Model Elements

The activity diagram's notation is very similar to that of a statechart diagram. In fact, according to the UML specification, an activity diagram is a variation of a statechart diagram.

Actions

Action is a named element which represents a single atomic step within activity i.e. that is not further decomposed within the activity.

A blue rounded rectangular node with the text "Process Order" inside, representing an action in an activity diagram.

Activity

Activity represents a behavior that is composed of individual elements that are activity nodes which could be:

1. action
2. object
3. control flow

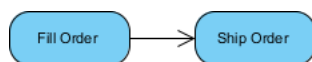
Control Flow

A Control Flow represents a transfer of execution control from one action to another action.

It is drawn as a solid line with a line-arrow at one end pointing to the next action.

Example

When the *Fill Order* action is ended, the execution control is transferred to the *Ship Order* action.

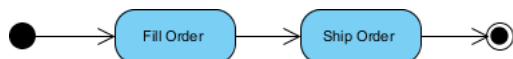


Start and Final

They represent the starting point and the ending point of an action execution sequence within an activity. Start Notation and Final Notation are also called Initial State notation and Final State.

1. A Start element is drawn as a small solid circle.
2. A Final element is drawn as a small solid circle enclosed in an outer circle.

Example



Decision

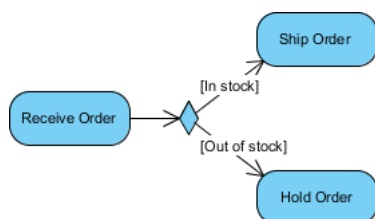
It represents a decision action that evaluates certain conditions and decides which action path to continue the execution.

A Decision Notation is drawn as a small diamond shape with one incoming control flow and multiple outgoing control flows.

Each outgoing control flow should be labeled with the condition that leads to this flow.

Example

The decision action performed after the *Receive Order* action to check the stock can be drawn in the diagram as shown below:



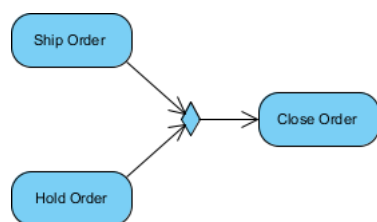
Merge

It represents a merge point where multiple alternate execution paths will meet and continue.

It is drawn as a small diamond shape with multiple incoming control flows and one outgoing control flow.

Example

The merge point where *Ship Order* and *Hold Order* meet can be drawn in the diagram as a Merge Notation as shown below:

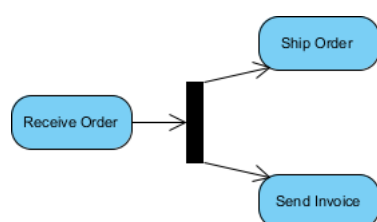


Fork

It represents a fork action that splits a single execution flow into multiple concurrent execution flows. It is drawn as a short solid line with one incoming control flow on one side and multiple outgoing control flows on the other side.

Example

The fork action performed after the *Receive Order* action to start *Ship Order* action and *Send Invoice* action concurrently can be drawn in the diagram as shown below:

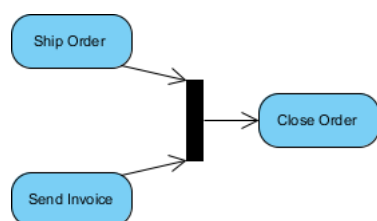


Join

It represents a join action that waits for multiple concurrent execution flows to finish. It is drawn as a short solid line with multiple incoming control flows on one side and one outgoing control flow on the other side.

Example

Join action performed before the *Close Order* action to wait for both *Ship Order* action and *Send Invoice* action to finish can be drawn as in the diagram as shown below:



Object

It represents an object which could be an input and/or an output of an action. An object in this case is considered as an instance of a class in a given state.

It is represented by a rectangle with its name placed inside.

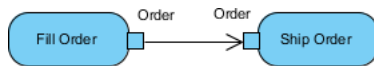
It can also be qualified by a state written within brackets below the name.

Example

Order [Filled] object generated from the *Fill Order* action will be consumed by the *Ship Order* action. This object can be drawn as an Object Notation in a UML activity diagram as shown below:



Different notations with same semantics:



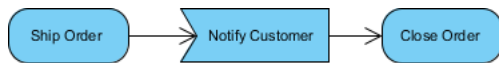
Signal and Event

It represents a signal action that sends a signal to outside of the activity. The send signal action does not wait for any responses from the receiver of the signal. It ends itself and passes the execution control to the next action.

It is drawn as a convex pentagon with its name placed inside.

Example

The *Notify Customer* send signal action in an order processing activity can be drawn as a Send Signal Notation in a UML activity diagram as shown below:



Swimlane and Partition

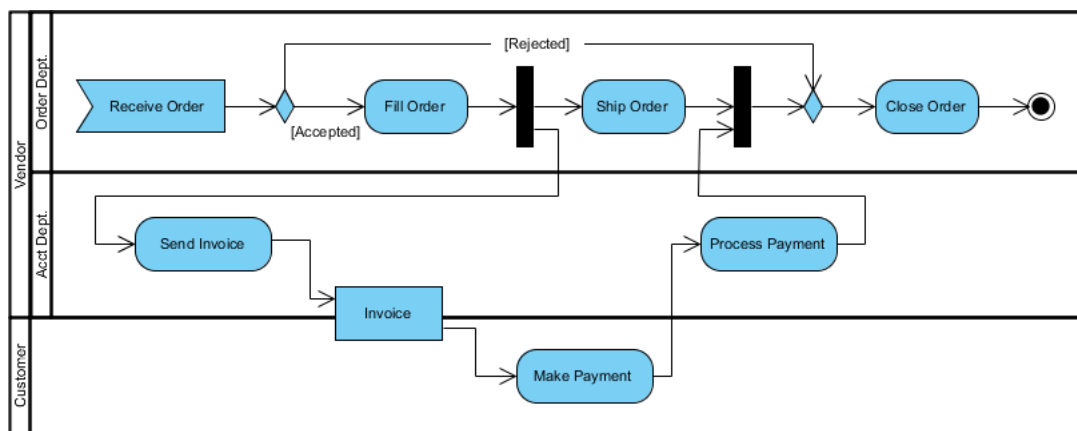
1. Partition presents a group of actions and objects that share some common properties.
2. It is drawn as two parallel lines with the partition name as one end.
3. Any actions and objects that are located within the two lines are considered to be in the group.
4. If an action or object may located on the border of multiple partitions, is considered to be in multiple groups.
5. Partitions can be nested to form a hierarchy of partitions.

Example

Actions and objects in an order processing activity can be divided into 3 groups:

1. Order Department action group
2. Accounting Department action group
3. Customer action group, based on action ownerships

These groups can be drawn as Partition Notations in a UML activity diagram as shown below:



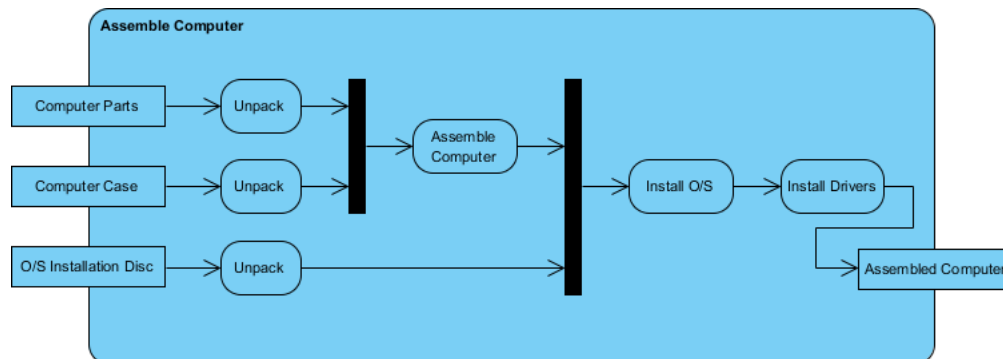
Parameterized Activity

An Activity Parameter Activity accepts input to an Activity or provides output from an Activity.

Activity parameters are displayed on the border and listed below the activity name as: parameter-name: parameter-type.

Example

The following example depicts two entry parameters and one output parameter defined for the Activity.



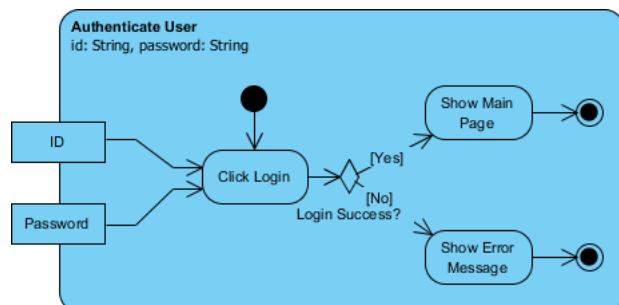
Activity Frame

It is used in a UML Activity Diagram to provide a boundary to enclose all actions and objects of the activity.

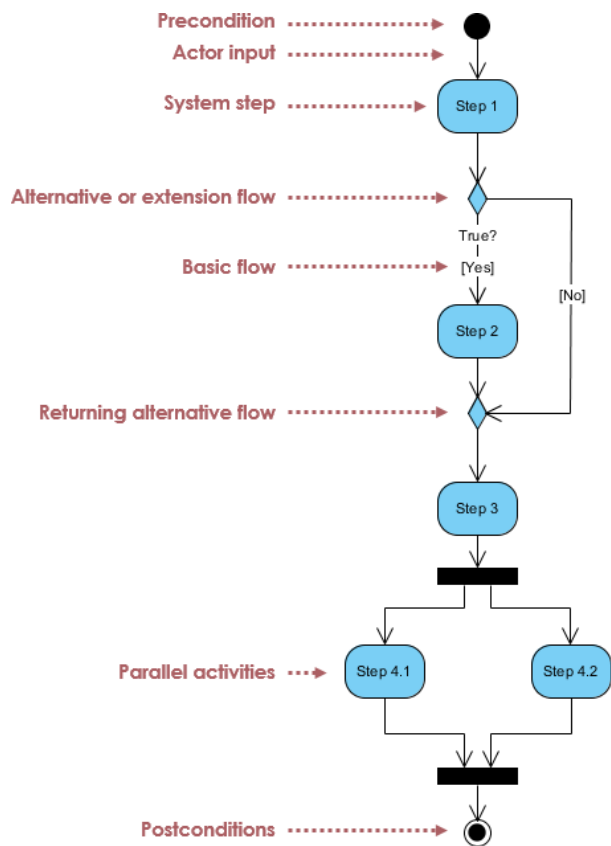
It is drawn as a large rectangle with rounded corners. The activity name, input parameters and output parameters are written near the top left corner of the rectangle.

Object Notations representing input parameters and output parameters can be placed on edges of the rectangle.

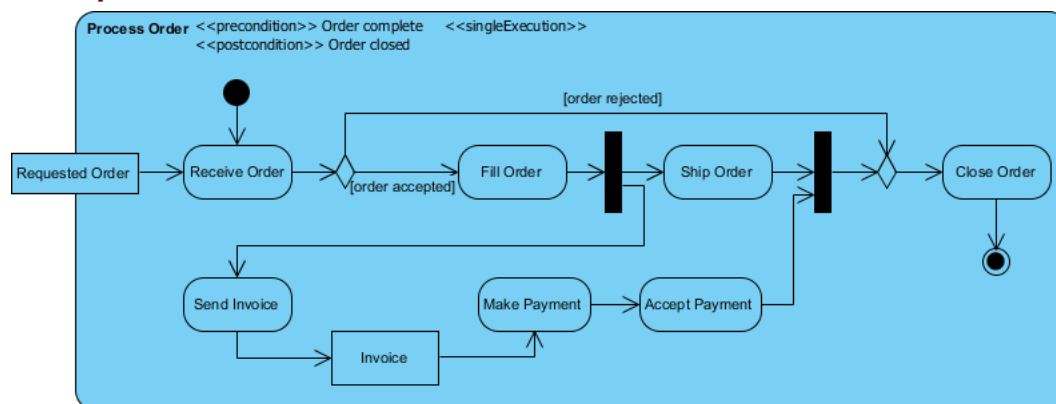
Example



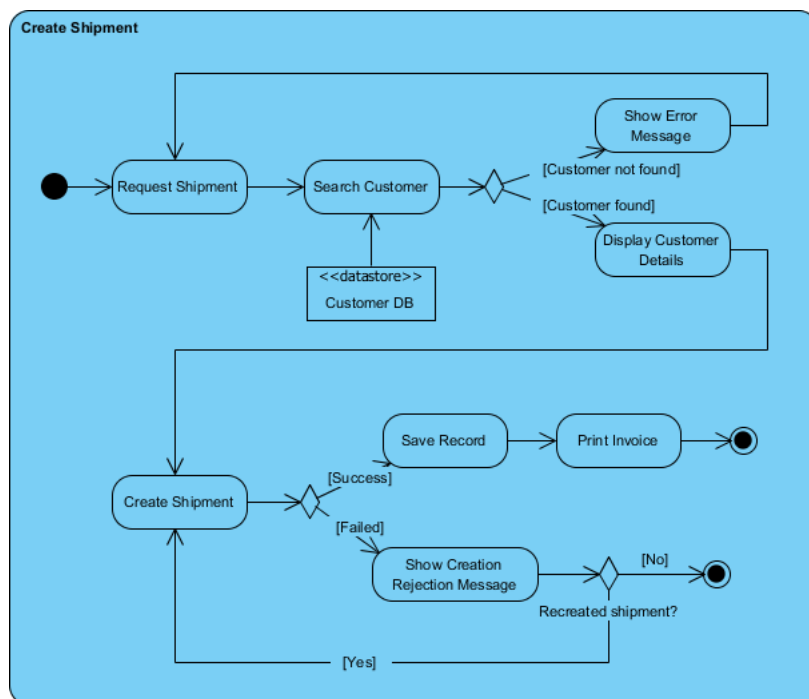
Example 1 - Activity Notations



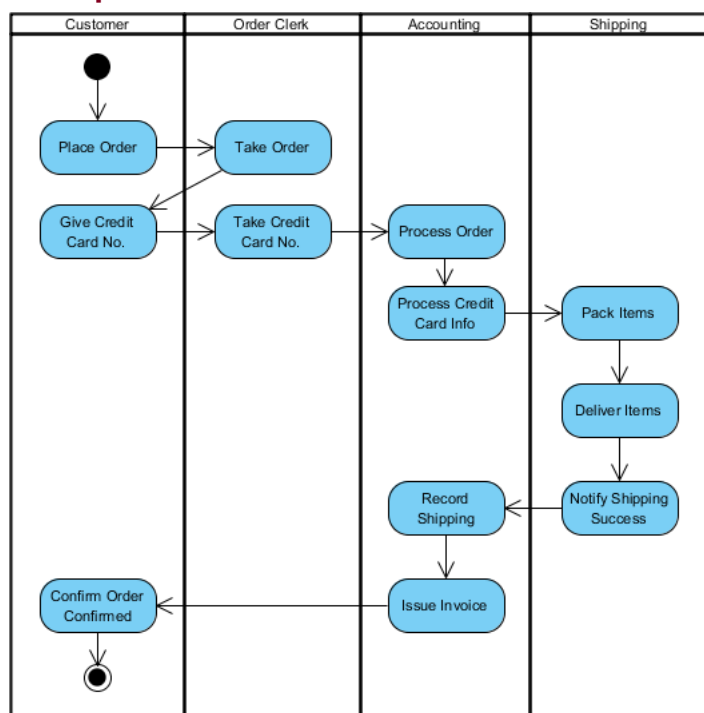
Example 2 - Process Order



Example 3 - Create Shipment



Example 4 - Swimlane



Visual Paradigm supports activity diagram and other UML diagram types. You can find all the tools you need in modeling different kinds of flows using activity diagram.

[Try now for FREE](#)

References

1. [How to draw a Activity Diagram in UML - Visual Paradigm User's Guide](#)



Visual Paradigm home page
(<https://www.visual-paradigm.com/>)

Visual Paradigm handbooks
(<https://www.visual-paradigm.com/learning/handbooks/>)