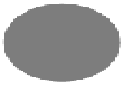
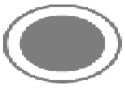
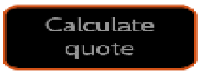
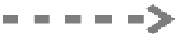
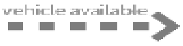
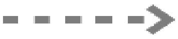
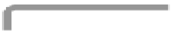
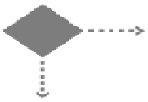
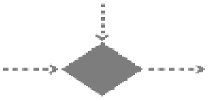
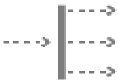
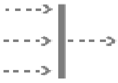
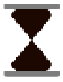
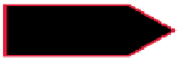
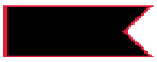
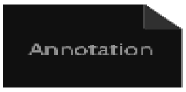
ACTIVITY DIAGRAM

basic components

- **Start node:** Symbolizes the beginning of the activity, represented by a black circle.
- **Action:** A step in the activity wherein the users or software perform a given task, represented by a rectangle usually with rounded edges
- **Decision node:** A conditional branch in the flow that is represented by a diamond shape. It includes a single input and two or more outputs
- **Control flows:** The connectors that show the flow between steps. These include connector arrows, the join symbol, and the fork symbol
- **End node:** Symbolizes the end state of an activity and represents the completion of the process.

Overview of the symbols

	Start	Used at the beginning or start of a proess. Usually just one start in an activity diagram.
	End	Used at the end of a process. There could be multiple ends within a process to show the end of each branch of the process.
	Activity	Represents an activity or task. There will be multiple tasks within an activity diagram. Named using the verb-noun naming convention. A task represents one person doing one thing at one point in time.
	Flow arrow	Represents the flow of the process. Links different shapes together flowing in the direction of the arrow.
	Annotated Arrow	Same as a flow arrow but includes annotation to add more information to a diagram,
	Flow arrow	Represents the flow of the process. Links different shapes together flowing in the direction of the arrow.
	Swim Lane	Represents the flow of the process. Links different shapes together flowing in the direction of the arrow.
	Decision	Represents where the flow follows a different path depending on the result of the decision. The process will only follow one of the outgoing paths from a deicion
	Merge	Represents where two or more paths converge back into one path. Only one of the incoming paths needs to hit the merge for the process to continue.
	Fork	Represents where the process splits in to different paths, unlike a decision the process will follow all outgoing paths.
	Join	Represents where two or morepaths converge back into the same path. All of the incoming paths must reach this point before the process continues.
	Time	Used to depict where time impacts the process such as a passage of time or a date in the calendar.
	Outgoing Event	Used where the process creates an event but the outcome of the event is outside the scope of the activity diagram.
	Incoming Event	Used were the process receives an event from another process or source that is outside the scope of the activity diagram.
	Annotation	Used to add text to annotate or elaborate on parts of the diagram.

A high-level overview of the diagram creation process

According to the UML tool provider, VISUAL PARADIGM , the following 5 steps need to be followed to create effective activity diagrams.

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

A 10-step guide to modelling processes with UML activity diagrams

The completed activity diagram of the example I am about to go through is presented at the end of the step-by-step guide.

Feel free to review it before reading through the steps, although I would recommend reading through the steps first. This will help you visualize how some of your processes would be modeled before seeing the finished product.

Preparation

Step 1: Determine the process

Firstly, we need to choose a specific process. Let’s say we want to map out the process for hiring a car.

Step 2: Identify the actors (swimlanes)

In this case, there are 3 actors – the customer, the sales desk, and the hiring office. Each of these 3 actors will represent a swimlane.

Don’t worry if you feel unsure about exactly which actors are involved in the process. It’s ok to identify actors as you map out the various activities.

Step 3: List all activities/tasks that need to be completed

Make a list of all the activities/tasks that need to be completed in order for the process to reach the desired outcome when it ends. This can also be done as you work through the model, but ideally list everything beforehand as you may identify opportunities to streamline the process.

Step 4: Name the start node and place it in the top left corner of the diagram

The start node, symbolized by a black circle, needs to be named and placed in the top left corner of the model, in the first swimlane. The swimlane at the top of the diagram belongs to the actor who begins the process, which in this case is the customer. The node is named “Customer request to hire car”.

Step 5: Insert the first activity

Once the first actor has begun the process, what is the first activity or task that needs to be completed? In the case of our example, the task is “Check availability” and is to be completed by the sales desk (second swimlane).

Step 6: Begin connecting activities with connector arrows, placing fork and join nodes where necessary

This is when the model will start to take shape. Place a connector arrow between each activity. For activities that need to be split into different paths, insert a fork node. Where two or more paths converge back into the same path, insert a join node.

Step 7: Integrate decision and merge symbols where appropriate

Following an activity such as “Check availability”, a decision needs to be made as there is a possibility that a car is available or unavailable. Depending on the result, the process will change.

Step 8: Integrate outgoing/incoming symbols where appropriate

These symbols are also referred to as send and receive symbols. Insert an outgoing symbol where a certain activity creates an event for which the outcome is outside the scope of the diagram. In the case of our example, an outgoing symbol titled “Notify stock manager” is created if a car is unavailable as this is an activity that needs to take place but how the stock manager deals with the situation is irrelevant to the car hire process.

An incoming symbol should be inserted where the process receives an event from another process our source that is outside the scope of the diagram

Step 9: Add annotations where appropriate

Annotations should be used conservatively. It can be tempting to insert a bunch of annotations in an attempt to clarify activities but this can cause confusion and create a messy diagram. Only add them where critical information needs to be represented.

You will notice that our example contains only one annotation, referring to accepted methods of payment which is obviously an important piece of information.

Step 10: Verify that the final activity completes the process and connect it to the end node

Do a final check to make sure that the final activity completes the process, and add a connector arrow to the end node. All done! Now it’s time to review the diagram and ensure that no steps have been missed.