Introduction to Activities

Activities represent one or more units of work to be performed by a participant that require time and resources to complete. The three most common activities are tasks, subprocesses, and call activities.

A task is an atomic activity within a process flow. That is, it's a fundamental unit of work that cannot be broken down to a more granular level of detail.

A subprocess, on the other hand, consists of activities, gateways, events, and sequence flows. This colored rectangle with the plus marker is a subprocess in its collapsed state. The flow objects of which it is composed are hidden. A collapsed subprocess hides complexity. This subprocess, alternatively, is in its expanded state.

The use of start and end events within a subprocess is optional. That's because they are inferred as they are embedded within their superordinate—or parent—process. If used, the start event of a subprocess should be of the "none" type. That's because a subprocess doesn't have its own trigger independent of that of its parent process.

Here's an illustration of how a token flows through a simple process containing an expanded subprocess.

Pools are containers for processes. Consequently, a subprocess, which is, by definition, embedded within a parent process, can't have its own pools or lanes. A single actor must execute a subprocess.

An embedded subprocess can only be invoked from within its parent. However, a subprocess can be made globally available for use in other processes by designating it as a call activity.

Markers indicate the execution behavior of activities. The activity markers shown here are among the most common.

These indicate loop activities. A loop task or subprocess is repeated until the condition associated with the activity is satisfied. In other words, the number of instances isn't known in advance.

For example, each of these diagrams describes the process of selecting a restaurant for lunch. Restaurant suggestions are submitted until an agreement is reached. It's unknown in advance how many suggestions will be required. Of the two representations shown here, the process incorporating the loop task is more compact and, arguably, easier to understand.

When a task or subprocess must be repeated a known number of times, a multiple-instance marker is used. Three vertical lines indicate that the instances are to be executed in parallel. Three horizontal lines indicate that the instances are to be executed sequentially.

Here is an example that incorporates loop and multiple-instance tasks: a process for hiring a new employee. First, a loop task indicates that applications will be solicited until a satisfactory number has been received. At this point, we know how many interviews to schedule. So, the "Schedule interviews" subprocess has the parallel multiple-instance marker. After the interviews have been scheduled, they are conducted in sequence, after which the process continues.

In an ad hoc subprocess, activities are executed without a pre-defined sequence. That is, it doesn’t contain a sequence flow.

Activities can be further differentiated by type.

- Abstract activities don’t specify a default behavior.
- The send task sends a message to an external participant.
- The receive task receives a message from an external participant.
- A manual task is completed physically without the assistance of information technology.
- On the other hand, a user task is completed with the assistance of information technology.
- A service task is used when an external service, such as a web service, is called to perform a task.
- A script task, on the other hand, is used when a task is executed locally using a process engine. A process engine is a software that manages the processing, storage, and distribution of data related to a business process.
- Lastly, a business rule task is used to model the evaluation of a business rule, such as a recurring decision modeled using the DMN standard.