BPMN 2.0 Activities

BPMN Activity Basics

An Activity

* is work that is performed within a Business Process
* can be atomic or non-atomic (compound)
* represent points in a Process flow where work is performed
* is executable element of a BPMN Process.

Activity can be:

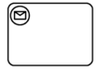
* **Task**
  + an atomic Activity within a Process flow
  + is used when the work in the Process cannot be broken down to a finer level of detail
  + an end-user or applications are used to perform the Task when it is executed
* **Sub-Process**
* **Call Activity**
  + allows the inclusion of re-usable Tasks and Processes in the diagram

Types of Tasks

**Receive Task**

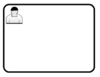
[](http://training-course-material.com/training/File:Figure10-15-receive-task-object.png)

* Waits for a Message to arrive from an external Participant
* Once the Message has been received, the Task is completed

[](http://training-course-material.com/training/File:Figure10-16-receive-task-object-that-instantiates-process.png)

* Starts the process
* MUST NOT have any incoming Sequence Flow
* Instantiate attribute MUST be set to true

**User Task**

[](http://training-course-material.com/training/File:Figure10-17-user-task-object.png)

* A typical “workflow” Task where a human performer performs the Task with the assistance of a software application
* The task is scheduled through a task list manager of some sort

Examples:

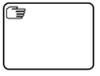
* Phone operator updates customer record
* User changing their password

**Manual Task**

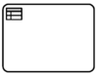
* Is expected to be performed without the aid of any business process execution engine or any application

Examples:

* A telephone technician installing a telephone at a customer location

[](http://training-course-material.com/training/File:Figure10-18-manual-task-object.png)

**Business Rule Task**

[](http://training-course-material.com/training/File:Figure10-19-business-rule-task-object.png)

* Provides a mechanism for the Process to provide input to a Business Rules Engine and to get the output of calculations that the Business Rules Engine might provide

**Script Task**

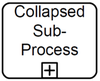
[](http://training-course-material.com/training/File:Figure10-20-script-task-object.png)

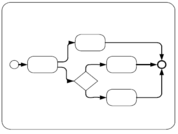
* Is executed by a business process engine
* The modeler or implementer defines a script in a language that the engine can interpret
* When the Task is ready to start, the engine will execute the script
* When the script is completed, the Task will also be completed.

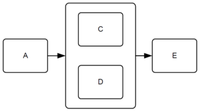
Sub-process Markers

* A Sub-Process is an Activity whose internal details have been modeled using Activities, Gateways, Events, and Sequence Flows
* A Sub-Process is a graphical object within a Process, but it also can be “opened up” to show a lower-level Process

Types of Sub-Processes

[](http://training-course-material.com/training/File:Figure10-25-sub-process-object-collapsed.png)

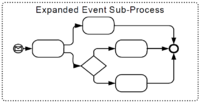
[](http://training-course-material.com/training/File:Figure10-26-sub-process-object-expanded.png)

[](http://training-course-material.com/training/File:Figure10-27-expanded-sub-process-used-as-parallel-box.png)

Sub-process Markers

[](http://training-course-material.com/training/File:Figure10-28-collapsed-sub-process-markers.png)

**Event Sub-Process**

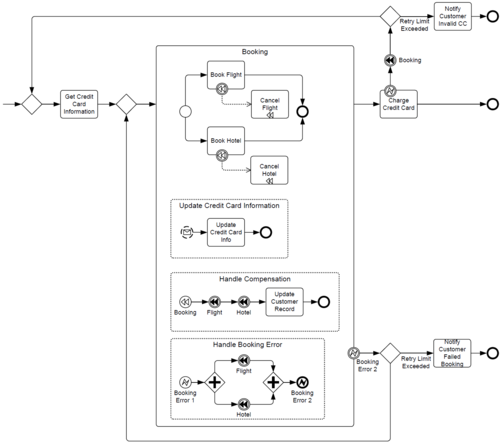
[Figure10-30-event-sub-process-object-collapsed.png](http://training-course-material.com/training/File:Figure10-30-event-sub-process-object-collapsed.png) [](http://training-course-material.com/training/File:Figure10-31-event-sub-process-object-expanded.png)

* An Event Sub-Process is a specialized Sub-Process that is used within a Process (or Sub-Process)
* The triggeredByEvent attribute is set to true
* An Event Sub-Process is not part of the normal flow of its parent Process—there are no incoming or outgoing Sequence Flows.
* It MAY occur many times.
* Event Sub-Process has a Start Event with a trigger:
  + Message, Error, Escalation, Compensation, Conditional, Signal, and Multiple
* An Event Sub-Process object shares the same basic shape as the Sub-Process object, which is a rounded rectangle

When an Event Sub-Process is triggered the parent Process:

1. can be interrupted
2. can continue its work (not interrupted)

This is determined by the type of Start Event that is used

[](http://training-course-material.com/training/File:Figure10-32-example-that-includes-event-sub-processes.png)

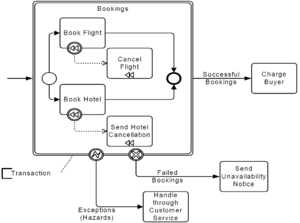
Transaction

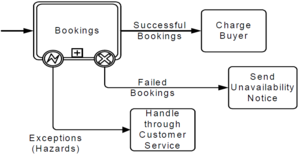
[](http://training-course-material.com/training/File:Element_transaction.png)

* Is a specialized type of Sub-Process that will have a special behavior that is controlled through a transaction protocol (such as WS-Transaction)
* The boundary of the Sub-Process will be double-lined to indicate that it is a Transaction
* The behavior at the end of a successful Transaction Sub-Process is slightly different than that of a normal Sub-Process
* When each path of the Transaction Sub-Process reaches a non-Cancel End Event(s), the flow does not

immediately move back up to the higher-level parent Process, as does a normal Sub-Process

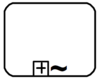
* First, the transaction protocol needs to verify that all the Participants have successfully completed their end of the Transaction
* Most of the time this will be true and the flow will then move up to the higher-level Process
* But it is possible that one of the Participants can end up with a problem that causes a Cancel or a Hazard
* In this case, the flow will then move to the appropriate Intermediate Event, even though it had apparently finished successfully

[](http://training-course-material.com/training/File:Figure10-33-transaction-sub-process.png)

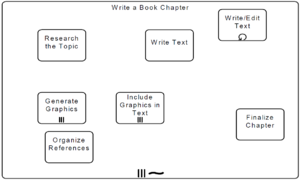
[](http://training-course-material.com/training/File:Figure10-34-collapsed-transaction-sub-process.png)

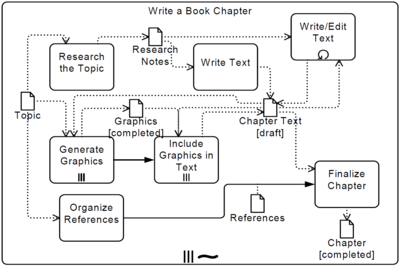
Ad-Hoc

* A group of Activities that have no REQUIRED sequence relationships
* A set of Activities can be defined for the Process, but the sequence and number of performances for the Activities is determined by the performers of the Activities.

[](http://training-course-material.com/training/File:Figure10-35-collapsed-ad-hoc-sub-process.png)

[](http://training-course-material.com/training/File:Figure10-36-expanded-ad-hoc-sub-process.png)

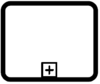
[](http://training-course-material.com/training/File:Figure10-37-ad-hoc-sub-process-for-writing-book-chapter.png)

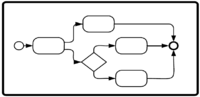
[](http://training-course-material.com/training/File:Figure10-38-ad-hoc-sub-process-with-data-and-sequence-dependencies.png)

Call Activity

* A Call Activity identifies a point in the Process where a global Process or a Global Task is used
* The Call Activity acts as a ‘wrapper’ for the invocation of a global Process or Global Task within the execution
* The activation of a call Activity results in the transfer of control to the called global Process or Global Task

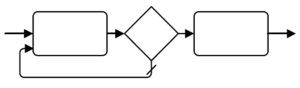
[](http://training-course-material.com/training/File:Figure10-39-call-activity-object-calling-global-task.png)

[](http://training-course-material.com/training/File:Figure10-40-call-activity-object-calling-process-collapsed.png)

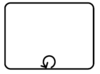
[](http://training-course-material.com/training/File:Figure10-41-call-activity-obejct-calling-process-expanded.png)

Loops

**Upstream Sequence Flow**

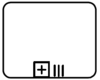
[](http://training-course-material.com/training/File:Element_sequence_flow_looping.png)

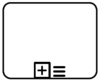
**Standard Loop**[[edit](http://training-course-material.com/index.php?title=BPMN_2.0_Activities&action=edit&section=17)]

[](http://training-course-material.com/training/File:Element_activity_looping.png) [](http://training-course-material.com/training/File:Figure10-47-sub-process-object-with-standard-loop-marker.png)

* The Activity will loop as long as the boolean condition is true
* The condition is evaluated for every loop iteration, and MAY be evaluated at the beginning or at the end of the iteration
* In addition, a numeric cap can be optionally specified
* The number of iterations MAY NOT exceed this cap

**Multi-Instance**

[](http://training-course-material.com/training/File:Element_multiple_instances_parallel.png) [](http://training-course-material.com/training/File:Figure10-48-activity-multi-instance-marker-for-parallel-instances.png)

[](http://training-course-material.com/training/File:Element_multiple_instances_sequential.png) [](http://training-course-material.com/training/File:Figure10-49-activity-multi-instance-marker-for-sequential-instances.png)

* The instances MAY execute in parallel or MAY be sequential.
* Either an Expression is used to specify or calculate the desired number of instances or a data driven setup can be used
* In that case a data input can be specified, which is able to handle a collection of data
* The number of items in the collection determines the number of Activity instances
* This data input can be produced by an input Data Association
* The modeler can also configure this loop to control the tokens produced.