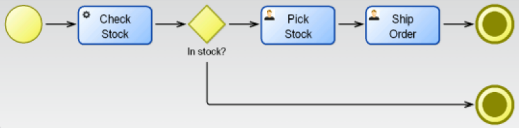
BPMN 2.0 – Business Process Model and Notation

Free BPMN tool: www.adonis-community.com

Business Process Diagram



The Business Process Diagram describes a **Process** i.e. a sequence or flow of activities in an organization with the objective of carrying out work. Processes can be defined at any level from enterprise-wide processes to processes performed by a single person. Low-level processes can be grouped together to achieve a common business goal.

There are several types of **Activities**:

A **Task** is an atomic activity within a process flow. It is used when the work in the process cannot be broken down to a finer level of detail.

A **Sub-Process** is an activity whose internal details have been modeled in a separate model.

A **Call Activity** identifies a point in the process where a global process or a Global Task is (re)used.

A **Transaction** is a specialized type of sub-process. It allows describing what happens if a whole set of activities is cancelled or interrupted by an error.

Event Sub-Processes allow to handle an event within the context of a given sub-process or process (either by interrupting the process or running in parallel to it). They are not part of a normal process flow, as they start only when their associated Start Event is triggered.

Different types of **Tasks** are visualized by markers (in top left corner):

- **Service** (automated e.g. by a web service)
- **User** ("workflow" task with human performer)
- **Send** (sends a Message)
- **Receive** (waits for a Message)
- **Manual** (performed without software aid)
- **Business Rule** (interacts with BR Engine)
- **Script** (executed by a business process engine)

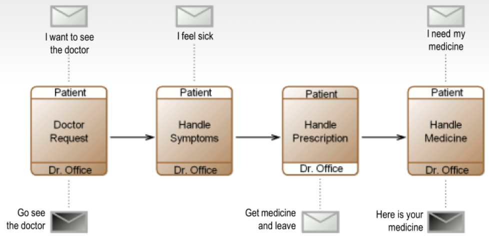
There are also markers that indicate characteristics of **Activities**:

Sub-Process **Loop** **Ad Hoc** **Compensation**

Multi-Instance: **Parallel** **Sequential**

Choreography Diagram

The Choreography Diagram defines a sequence of **interactions between Participants.**



A **Choreography Task** is an atomic Activity in a choreography process. It represents an interaction between two Participants

A **Sub-Choreography** is an atomic Activity in a choreography process. It represents an Interaction between two Participants

A **Call Choreography** identifies a point in the process where a Global Choreography or a Global Choreography Task is used.

Conversation Diagram

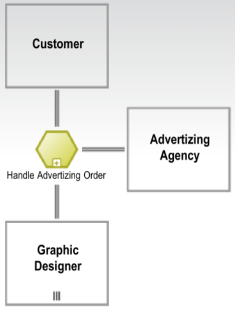
The Conversation Diagram shows logical relation of **message exchanges**. In general, it is a simplified version of Collaboration, but conversation diagrams do maintain all the features of a Collaboration. In particular, processes can appear within the Participants (Pools) of conversation diagrams, to show how Conversations and Activities are related.

A **Conversation** represents a set of Message Flows grouped together.

A **Sub-Conversation** is a conversation node that is a hierarchical division within the parent Collaboration.

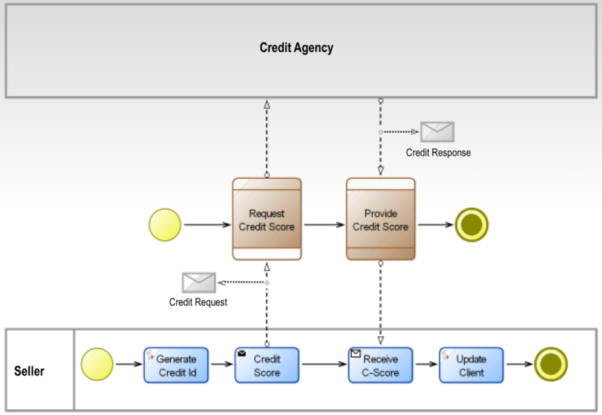
A **Call Conversation** identifies a place in the Conversation where a global Conversation is used.

A **Conversation Link** is used to connect to and from Participants.



Collaboration Diagram & Combined Views

The Collaboration package contains classes which are used for modeling Collaborations, which is a collection of Participants shown as Pools, their interactions as shown by Message Flows, and MAY include processes within the Pools and/or Choreographies between the Pools.



Events

		None	Message	Timer	Con-ditional	Signal	Escalation	Error	Com-pensation	Multiple	Parallel Multiple	Link	Cancel	Terminate
Start Events	Top-Level	○	○	○	○	○				○	○			
	Event Sub-Process Interrupting		○	○	○	○	○	○	○	○	○			
	Event Sub-Process Non-Interrupting		○	○	○	○	○			○	○			
Inter-mediate Events	Catching		○	○	○	○				○	○	○		
	Throwing	○	○			○	○		○	○		○		
	Boundary Interrupting		○	○	○	○	○	○	○	○	○		○	
	Boundary Non-Interrupting		○	○	○	○	○			○	○			
End Events		○	○			○	○	○	○	○			○	○

An Event is something that "happens" during the course of a process. Events affect the flow of the process and usually have a cause or an impact.

A **Start Event** indicates place where a process starts (e.g. after receiving message, when condition is met, or at a scheduled time).

An **Intermediate Event** happens between start and end of a process. It may be part of a process **flow** (either throwing or catching triggers such as Message) or represent exception/compensation handling (when placed on the Task/Sub-Process **boundary**).

An **End Event** indicates a process end. It may show e.g. that a certain process path ends with an error or sends a signal for other processes.

The **new types of Events** in the BPMN 2.0 are the types Escalation and Parallel Multiple as well as the whole categories Event Sub-Process (Interrupting and Non-Interrupting) and Boundary Non-Interrupting.

Connectors

Sequence Flow – defines the execution order of activities.

Conditional Flow - has a condition that defines whether or not the flow is used.

Default Flow - the default branch to be chosen of all other conditions evaluate to false.

Data Association - used to move data between Data Objects, Properties, and inputs and outputs of Activities, Processes, and Global Tasks.

Message Flow - used to show the flow of Messages between two Participants that are prepared to send and receive them.

Data

Task

Data Object Collection

initiating non-initiating

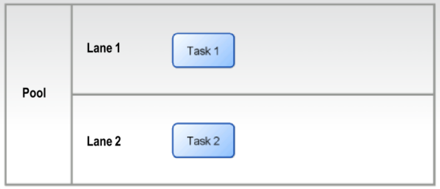
A **Data Input** is an external input for the entire Process. It can be read by an Activity. A **Data Output** is a variable available as result of the entire Process.

A **Data Object** represents information flowing through the Process, such as business documents, e-mails, letters. A **Collection Data Object** represents a collection of information (e.g. a list of order items).

A **Data Store** is a place where the Process can read or write data, e.g. a database or a filling cabinet. It persists beyond the lifetime of the Process instance.

A **Message** is used to depict the contents of a communication between two Participants.

Pools & Artifacts



A **Pool** is the graphical representation of a Participant in a Collaboration. A **Lane** is a sub-partition within a Process (often within a Pool) and will extend the entire length of the Process level.

An **Association** is used to connect information and Artifacts with Flow Objects.

The **Group** object is an Artifact that provides a visual mechanism to group elements of a diagram informally.

Text Annotations are a mechanism for a modeler to provide additional information for the reader of a BPMN Diagram.

Gateways

Gateways are used to control how the Process flows (how tokens flow) through Sequence Flows as they converge and diverge within a Process. The Gateway controls the flow of both diverging and converging Sequence Flows.

Event-based Gateway - is always followed by catching events or receive tasks. Sequence flow is routed to the subsequent event/task which happens first.

Parallel Gateway - when used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.

Inclusive Gateway - when splitting, one or more branches are activated. All active incoming branches must complete before merging.

Complex Gateway - Complex merging and branching behavior that is not captured by other gateways.

Exclusive Gateway - when splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.

Exclusive Event-based Gateway (instantiate) - Each occurrence of a subsequent event starts a new process instance.

Parallel Event-based Gateway (instantiate) - The occurrence of all subsequent events starts a new process instance.