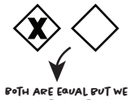
BPMN 2.0 FUNDAMENTALS

THE MASTER CHANNEL
www.themasterchannel.com

1 GATEWAYS

use the gateways to split & merge paths

ONE PATH



both are equal but we prefer the left one

A Data-Based Exclusive Gateway (XOR)

represents an exclusive decision that is based on data that is available in the process. The decision can be simple Yes/No or complex <5, >=5 & < 10, >= 10 but only one path will be activated after the gateway.
The gateway can be used to merge paths but has simple pass-through behavior.

An Event-Based Exclusive Gateway (XOR)

is for exclusive decisions that are based on events that may happen. The first event that happens triggers the path to follow. Often used with timer, message, or conditional intermediate events. Receive tasks can be used as well.
The gateway should not be used to merge paths.

MULTIPLE PATHS

A Parallel Gateway (AND)

is used to model concurrent (AND) behavior. All paths that follow the gateway are unconditionally activated.

The gateway can be used as a merge but waits for all paths to be completed before the outgoing flow is activated.

An Inclusive Gateway (OR)

can evaluate multiple conditions and the gateway activates all paths of which the conditions evaluate to True.

As a merge this gateway has complex behavior. It activates when all incoming paths which are active (active tokens) reach the gateway.

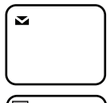
2 TASKS & SUBPROCESSES

are used to show work that must be performed

TASKS

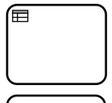
A Receive Task

waits for information from a participant outside the process. The process continues when the information is received.



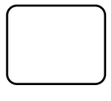
A Send Task

sends information to participants outside the process.



A Business Rule Task

evaluates a (business) rule and receives the outcome of the decision as data that can be used in the process.



An Abstract Task

represents work with an undefined type. All work is atomic: cannot be split, is small, & can be executed by one person.

this is valid for all task types

A Manual Task

is executed without the support of an information system or machine.

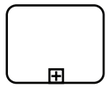
A User Task

is performed by a person and supported by an information system.

A Service Task

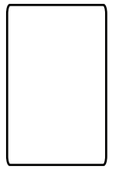
is executed by an information system.

SUBPROCESSES



A Collapsed Subprocess

serves as a pointer to a group of tasks and/or sub-processes that must be executed to achieve an intermediary result in the process. The content of the subprocess is hidden.



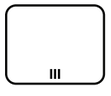
An Expanded Subprocess

shows a grouping of tasks and/or sub-processes that must be executed to achieve an intermediary result. The content of the subprocess is shown.

3 REPEATING BEHAVIOR

is used to execute tasks or subprocesses multiple times

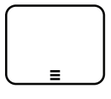
MULTI-INSTANCE



A Multi-Instance Parallel Task

is repeated based on a list of items or a clearly defined number. It is similar to a "for each" statement in programming.

Parallel means that the created instances are all triggered from the start and run in parallel.

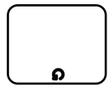


A Multi-Instance Sequential Task

is repeated based on a list of items or a clearly defined number. It is similar to a "for each" statement in programming.

Sequential means that the instances are executed in sequential order – one by one.

LOOP



A Loop Task

is repeated until a specified condition evaluates to true.

Each iteration is executed sequentially – one by one and the condition is evaluated before or after an iteration.

loops & multi-instances can be used on subprocesses as well

4 START EVENTS

are used to start a new process instance

A Message Start Event

starts the process when information is received from an external participant (e.g. an order).

A Timer Start Event

starts the process on a specific date, date-time or interval (e.g. every Monday).

A Conditional Start Event

starts the process when a condition evaluates to true. This can be a simple or a complex condition (e.g. Temperature > 200°C).

A None Start Event

represents that start of a subprocess and can be used to trigger a process without a clearly defined trigger.

5 END EVENTS

are used to show the end of a process path

BASIC END

A None End Event

ends the process(path) - a specific state is reached.

A Message End Event

ends the process(path) and a message is sent to an external participant.

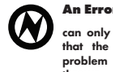
TERMINATE

A Terminate End Event

stops the (sub)process and stops all other active tasks in the (sub)process.

Be careful: only active tasks in the same subprocess and the same instance are terminated.

COMPLEX END



An Error End Event

can only be used in subprocesses to show that the subprocess has failed and this problem must be resolved. An error is thrown.

An Escalation End Event

is similar to an error end event – but in this case a human problem is escalated.

6 INTERMEDIATE EVENTS

THROWING "DO"

use like this

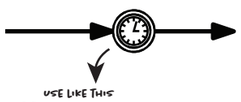
A Message Intermediate Event

sends information to an external participant.

An Escalation Intermediate Event

signals a human problem.

CATCHING "WAIT"



use like this

A None Intermediate Event

shows that a certain state is reached. The process will continue without waiting.

A Message Intermediate Event

waits until information from an external participant is received.

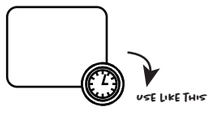
A Timer Intermediate Event

waits for a specific duration or date.

A Conditional Intermediate Event

waits for a specific condition to become true.

INTERRUPTING "LISTEN & STOP"



use like this

A Message Intermediate Event

stops the activity when the message (external information) is received.

A Timer Intermediate Event

stops the activity if it takes longer than the specified time or if the task is not completed before a specific date.

An Error Intermediate Event

stops the subprocess if an error is thrown.

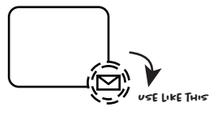
An Escalation Intermediate Event

stops the subprocess if a (human) problem happens.

A Conditional Intermediate Event

stops the activity when a condition evaluates to true.

NON-INTERRUPTING "LISTEN & CONTINUE"



use like this

A Message Intermediate Event

activates an additional path if the message is received during the execution of the underlying task or subprocess.

A Timer Intermediate Event

activates an additional path if the task takes too long (duration or date).

A Escalation Intermediate Event

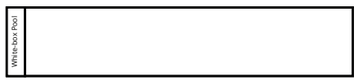
activates an additional path if a human problem is raised.

A Conditional Intermediate Event

activates an additional path if a condition evaluates to true.

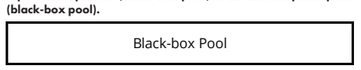
7 POOLS & LANES

POOL



A Pool

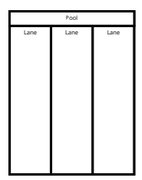
represents a process (white-box pool) or an external participant (black-box pool).



LANE

Lanes

show performers in the process. But that's not all. Lanes can categorize tasks any way you like.



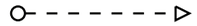
8 CONNECTORS

SEQUENCE FLOW

A Sequence Flow

is used to connect tasks, sub-processes, gateways and events in the same pool.

MESSAGE FLOW



A Message Flow

shows communication between different pools. Only tasks, sub-processes, events and pools can be connected

9 EXTRA

ARTIFACTS

A Text Annotation

is a piece of documentation shown in the diagram.

A Group

is a way to visually group elements in the model.

BPMN 2.0 PRO

THE MASTER CHANNEL
www.themasterchannel.com

1 GATEWAYS

USE THE GATEWAYS TO SPLIT & MERGE PATHS

A Complex Gateway
can evaluate multiple conditions and the gateway activates all paths of which the conditions evaluate to True.

As a merge this gateway can be configured using an activation condition to specify when the gateway is activated. It has a waitingForStart attribute that can be used on the activation condition.

THESE GATEWAYS CAN BE USED TO START A PROCESS

An Instantiating Event-Based Gateway (XOR)
is basically a way to start a process based on one or more events. The first event that happens starts the process.

Often used with timer, message, or conditional intermediate events. Receive tasks can be used as well.

The gateway should not be used to merge paths.

An Instantiating Parallel Event-Based Gateway
is basically a way to start a process based on a combination of message events. All of these events must be triggered before the process is activated.

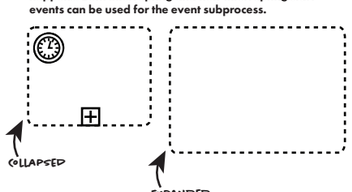
The gateway should not be used to merge paths.

2 TASKS & SUBPROCESSES

ARE USED TO SHOW WORK THAT MUST BE PERFORMED

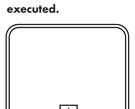
An Event Subprocess
is a special subprocess that is part of a pool or a subprocess. The event subprocess is not connected to the other elements in the (sub)process.

The event subprocess is triggered when the specified event happens. Both interrupting and non-interrupting start events can be used for the event subprocess.

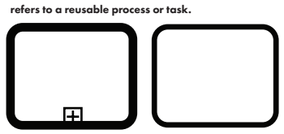


A Transaction Subprocess
forces transaction semantics on a set of tasks. This means a transaction subprocess has 3 basic outcomes:
- Successfull completion,
- failed completion (shown with a cancel intermediate event) and
- hazard (shown by an error end event).

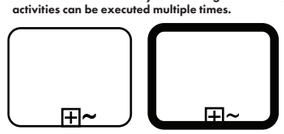
When a transaction fails the compensation activities are executed.



A Call-Activity
refers to a reusable process or task.



An Ad-hoc Subprocess
contains a number of activities and is intended to be executed with a more flexible ordering. The order of the activities is determined by a knowledge worker. Also the activities can be executed multiple times.

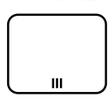


Compensation Task
is concerned with undoing steps that were already successfully completed, because their results are no longer desired and need to be reversed. They are triggered by a compensation event and linked with an association.

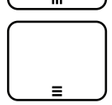
3 REPEATING BEHAVIOR

IS USED TO EXECUTE TASKS OR SUBPROCESSES MULTIPLE TIMES

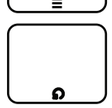
SHORT RECAP



Multi-Instance Parallel
- the number of iterations is known
- based on a list of items
- parallel execution



Multi-Instance Sequential
- the number of iterations is known
- based on a list of items
- sequential execution



Loop
- the number of iterations is unknown
- based on a condition
- sequential execution

FOR MORE INFORMATION MAKE SURE TO CHECK OUT OUR BPMN 2.0 FUNDAMENTALS POSTER!

4 START EVENTS

ARE USED TO START A NEW PROCESS INSTANCE

A Signal Start Event
starts a process based on an incoming signal. A signal is similar to a message, but follows the broadcasting principle.

SAME BEHAVIOR

A Multiple Start Event
is basically a way to start a process based on one or more events. The first event that happens starts the process.

Often used with timer, message, or conditional intermediate events.

SAME BEHAVIOR, BUT DIFFERENT TRIGGERS

A Parallel Multiple Start Event
is basically a way to start a process based on multiple events. These multiple events both need to happen to trigger the process.

5 END EVENTS

ARE USED TO SHOW THE END OF A PROCESS PATH

A Signal End Event
ends the process(path) - and broadcasts a signal.

A Multiple End Event
ends the process(path) and throws a number of triggers that are combined in one event

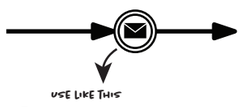
TRANSACTIONS

A Cancel End Event
can be used in a transaction subprocess to cancel the complete transaction and trigger the compensation tasks.

A Compensate End Event
can be used to trigger a compensation task.

6 INTERMEDIATE EVENTS

THROWING "DO"



A Compensation Intermediate Event
triggers a compensation interrupting event to undo work that was previously done.

A Link Intermediate Event
continues its execution from the catching link intermediate event.

A Signal Intermediate Event
broadcasts a signal.

A Multiple Intermediate Event
throws a number of triggers that are combined in one event (all triggers are activated).

CATCHING "WAIT"

A Link Intermediate Event
is activated by a throwing link intermediate event and continues its path here.

A Signal Intermediate Event
waits until a broadcasted signal is captured.

A Multiple Intermediate Event
waits for one of the defined events to happen. When one happens the process continues.

A Parallel Multiple Intermediate Event
waits for all defined events to happen.

INTERRUPTING "LISTEN & STOP"

A Compensation Intermediate Event
triggers the compensation task when triggered

A Signal Intermediate Event
stops the activity if a signal is captured.

A Multiple Intermediate Event
stops the activity when one of the defined events occurs.

A Paralle Multiple Intermediate Event
stops the subprocess when all events occur.

A Cancel Intermediate Event
can only be triggered when attached to a transaction subprocess. When activated the cancel event triggers all compensation activities in the transaction.

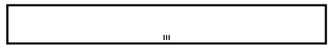
NON-INTERRUPTING "LISTEN & CONTINUE"

A Signal Intermediate Event
activates an additional path if a signal is captured.

A Multiple Intermediate Event
activates an additional path if one of the events happens.

A Parallel Multiple Intermediate Event
activates an additional path if all events happen.

7 POOLS & LANES



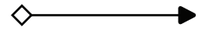
A Multi-instance Pool
is used to indicate that multiple instances of the pool may exist. This means there are multiple participants of the same role.
E.g. we request a quote from multiple suppliers.

8 CONNECTORS

SEQUENCE FLOW

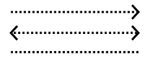
A Default Flow
is used in combination with the data-based XOR and OR gateways – and represents a fallback when no condition is true and the gateway has a risk to deadlock.

A Conditional Flow
can be attached to a task or subprocess to show that a path is conditionally activated (only when the condition is true).



An Association
connects artifacts (data object, text annotation, ...) to other elements in the diagram.

It is used to link compensation tasks to compensation events as well.



9 EXTRA

DATA

A Data Object
is a data element that is used in tasks or is generated by tasks. The data element is destroyed once the process is finished.

Output
shows the outputs of a process.

Input
shows the inputs to the top-level process.

Collection
shows multiple instances of a data object.

A Message
Data that is communicated.

A Data Store
is persistent data – data that is stored even after the process reaches an end.