uc BPMN - Notation Explained

Gateways

Gateways can define all types of business process Sequuence Flow behaviour: Decision/ branching (exclussive, inclussive and complex), merging, forking and joining.

Basic Gateways

- Exclussive Gateway – allows to express decissions
- Parallel Gateway – allows to express parllelism

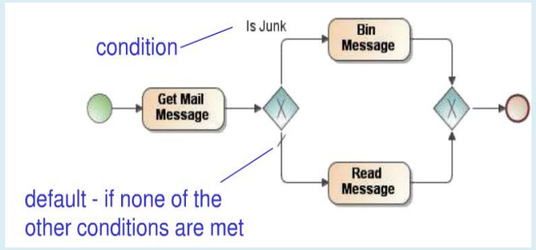
	Type	Input - merging	Output - splitting
	Exclusive	Passes on each incoming token	Single token to outgoing flow whose condition is true
	Parallel	Waits for token on *all* incoming flows	Token to *all* outgoing flows

Extended Gateways

	Type	Input - merging	Output - splitting
	Inclusive	Waits for token on one to all incoming flows depending on *flow* conditions	Token to one to all outgoing flows depending on flow conditions
	Complex	Waits for token on one to all incoming flows depending on *Gateway* condition	Token to one to all outgoing flows depending on *Gateway* condition

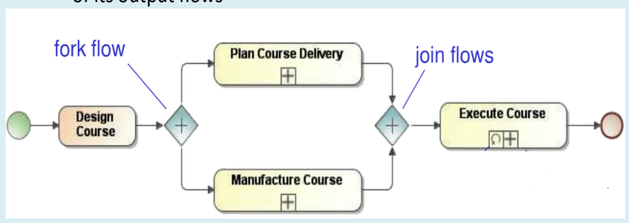
Exclusive Gateway

- Has one or more input flows, and two or more output flows.
- Each outputflow has a condition and set of conditions must be mutually exclussive; the first one that evaluates TRUE will determine the Sequence Flow to be taken
- only one gateway can be chosen;
- one of the Gates may be "default"



Parallel Gateway

- has one or more inputs and two or more otput flows
- it waitsfor a token on all input flows (join), then emits a token on all of its output flows

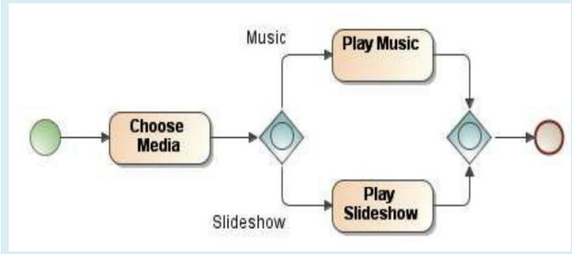


Inclusive Gateway

- Splitting – each output flow has a condition and a token is emitted on ALL the output flows whose conditions are true (this is a logical inclussive OR)
- Merging – all tokens generated by a corresponding upstream Inclussive Gateway split are synchronized and merged

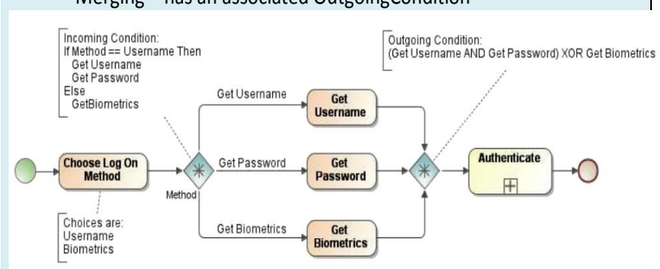
Example – Photo Player System:

- Choose Media (music, slideshow or both)
- Depending on what has been choosen, the system Play(s) Music, Play(s) Slideshow or both
- When ALL media has finished playing, the process ends



Complex Gateways

- Splitting – has an associated IncommingCondition
- Merging – has an associated OutgoingCondition



Inclusive Decision using Conditional Sequence Flow

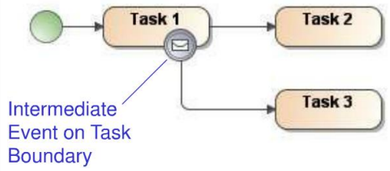
Decision situations can use - instead of Inclusive Gateway - a collection of conditional Sequence Flows, marked with mini-diamond marker.

uc BPMN - Notation Explained

Intermediate Events

Name	Intermediate			Semantics
	Throw	Catch	Interrupt	
None				No Trigger - fires immediately
Timer			Yes	Fire when time condition becomes true
Message			Yes	Send/wait for a Message
Signal			Yes	Send/wait for a Signal

Intermediate Events as interrupts



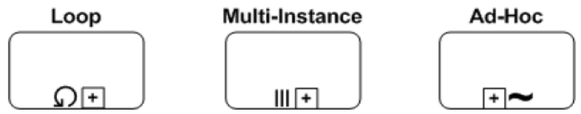
Activity

An Activity is a pice of work performed in a business process; there are two types of Activity:

- Tasks – are atomic, in the context od a particular process model
- Sub-Processes – may be decopmosed hierarchically into nested sub-processes and tasks; they can be Embedded (may be Ad-Hoc), Reusable or Reference

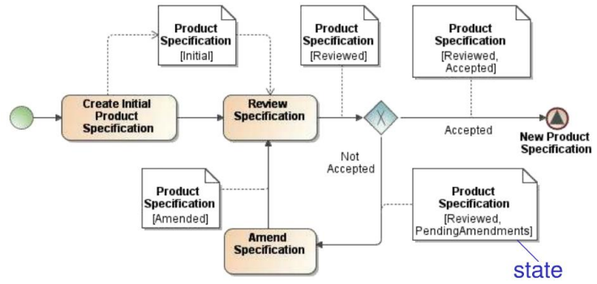
Activities

- Can Loop internally
- In a repetition, a new Activity instance is created each time; they may execute in sequence or in parallel



Data Objects

Data Objects indicate data flowing through the process and possibly being transformed by it; they may have a state



Gateways

Gateways can define all types of business process Sequuence Flow behaviour: Decision/ branching (exclusive, inclusive and complex), merging, forking and joining.

Basic Gateways

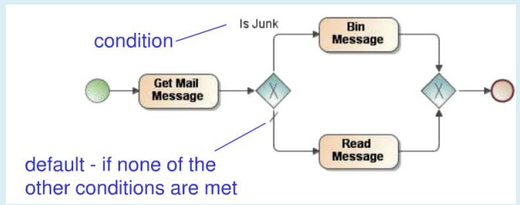
- Exclussive Gateway – allows to express decissions
- Parallel Gateway – allows to express parllelism

	Type	Input - merging	Output - splitting
	Exclusive	Passes on each incoming token	Single token to outgoing flow whose condition is true
	Parallel	Waits for token on *all* incoming flows	Token to *all* outgoing flows

Extended Gateways

Exclusive Gateway

- Has one or more input flows, and two or more output flows.
- Each outputflow has a condition and set of conditions must be mutually exclussive; the first one that evaluates TRUE will determine the Sequence Flow to be taken
- only one gateway can be chosen;
- one of the Gates may be “default”



Parallel Gateway

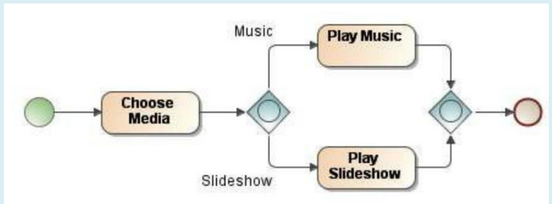
- has one or more inputs and two or more otput flows
- it waitsfor a token on all input flows (join), then emits a token on all of its output flows

Inclusive Gateway

- Splitting – each output flow has a condition and a token is emitted on ALL the output flows whose conditions are true (this is a logical inclussive OR)
- Merging – all tokens generated by a corresponding upstream Inclussive Gateway split are synchronized and merged

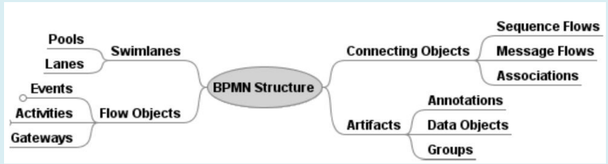
Example – Photo Player System:

- Choose Media (music, slideshow or both)
- Depending on what has been choosen, the system Play(s) Music, Play(s) Slideshow or both
- When ALL media has finished playing, the process ends

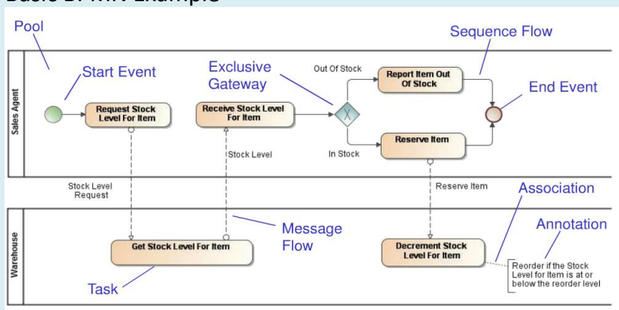


uc BPMN - Notation Explained

BPMN Structure

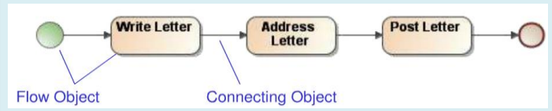


Basic BPMN Example



The Business Process Diagram (BRD)

BRD is a type of flowchart written in BPMN that comprises Flow Objects connected by Connecting Objects.

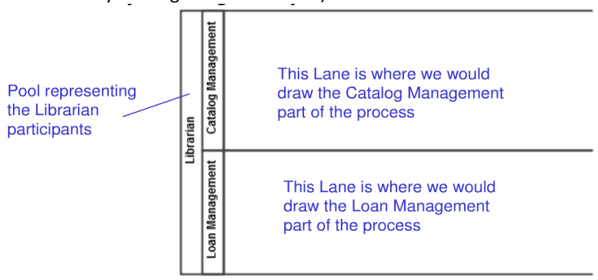


Swimlanes

Swimlanes organize the BPD into:

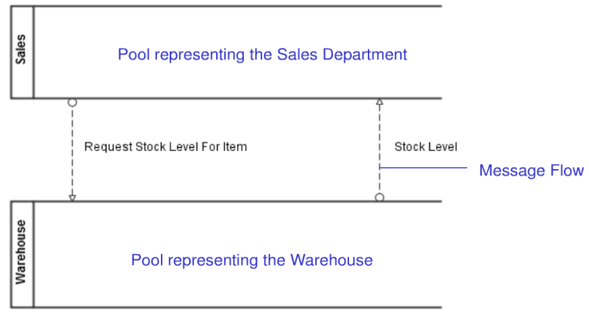
- Pools – representing a participant in the process
- Lanes – group related Activities

(Note: when the BPD has a single Pool and no Lanes, then the Pool boundary rectangle can be omitted)



Participants

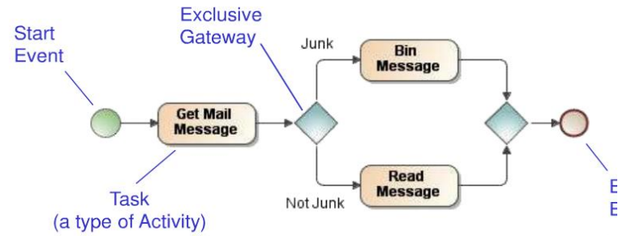
Pools are used in conjunction with Message Flows to show communication between two or more business participants.

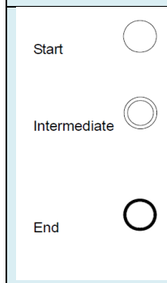


Flow Objects

A process is defined as a sequence of flow objects:

- Events – something that happens during the process
- Activities – work performed in the process
- Gateways – split/ merge flow through the proces





Events

- begin the process
- happen during process
- terminate process

Start Events

	Timer Start Event	Emits a token when the time condition becomes true
	Message Start Event	Emits a token when a specified Message is received
	Signal Start Event	Emits a token when a specified Signal is received

End Events

	Message End Event	Terminate thread and send a Message
	Signal End Event	Terminate thread and send a Signal
	Terminate End Event	Terminate *all* threads in the process