

In this session we will try get overview of BPMN 2.0 standard.

1. What BPMN is ?

Business Process Model and Notation (BPMN) is a standard for business process modeling that provides a graphical notation for specifying business processes in a Business Process Diagram (BPD). The objective of BPMN is to support business process management, for both technical users and business users, by providing a notation that is intuitive to business users, yet able to represent complex process semantics. The BPMN specification also provides a mapping between the graphics of the notation and the underlying constructs of execution languages, particularly Business process execution language (BPEL).

2. BPMN History

2004 : BPMN was originally developed by the Business Process Management Initiative (BPMI). They released a version 1.0 to the public in May, 2004.

The Business Process Management Initiative (BPMI.org) works on standards for the management of business processes that span multiple applications, corporate departments, and business partners.

2005 : In June 2005, BPMI merged with OMG, the Object Management Group. A BPMN Specification document was released by OMG in February, 2006.

2010 - 2013 : Version 2.0 of BPMN was developed in 2010, and the actual version of the specification was released in December 2013. The latest version (2.0.2) has been formally published by ISO as the 2013 edition standard: ISO/IEC 19510.

Over the last few years, BPMN rapidly became the de facto standard for process modeling. Part of its success, is due to the fact that BPMN offers a familiar look and feel to business analysts, while providing powerful business process model expressiveness. BPMN 2.0 models can be used to communicate, and

interchange the business requirements of a business process, as well as providing the underpinning of the actual process implementation. BPMN 2.0 is a fundamental evolution of the original standard. It extends the scope and capabilities of BPMN in several areas:

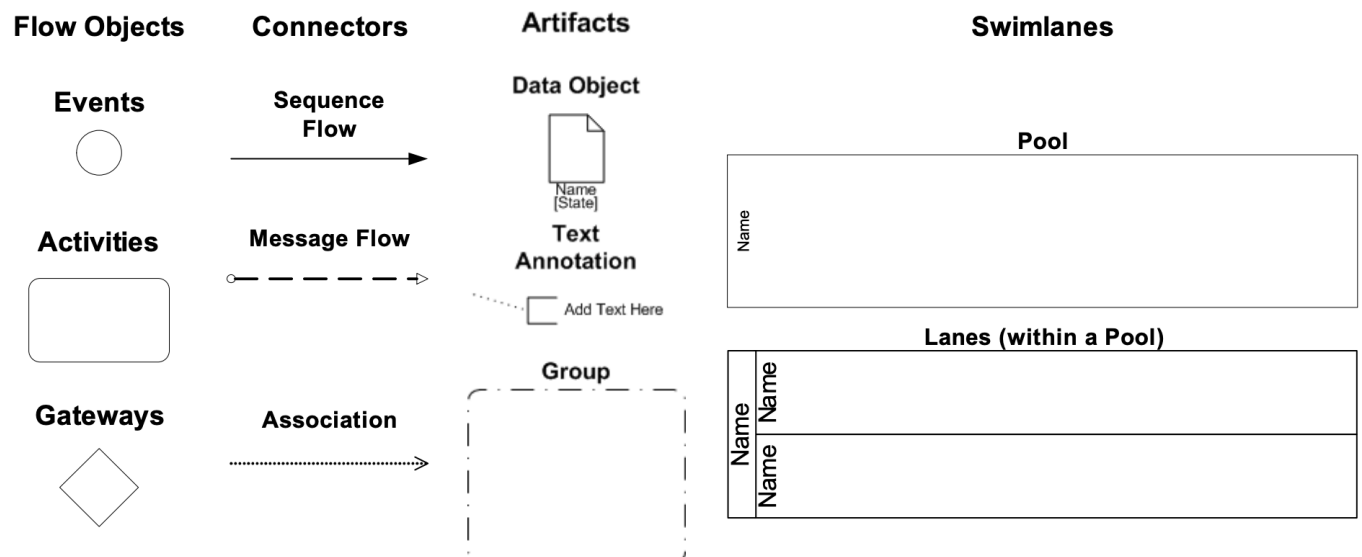
- It formalizes the execution semantics for all BPMN elements.
- It defines extensibility mechanism for both Process model extensions, and graphical extensions.
- It refines event composition and correlation.
- It extends the definition of human interactions.
- It defines a Choreography model, and a Conversation View for a Collaboration diagram.

The primary goal of BPMN is to provide a notation that is readily understandable by all business users, including:

- The business analysts, that create the initial drafts of the processes;
- The technical developers, responsible for implementing the technology, that will perform those processes;
- The business people, who will manage and monitor those processes.

3. BPMN Elements

A Business Process Diagram is a simple diagram made up of a set of graphical elements that depicts a business process. There are four primary elements of BPD:



Flow Objects: Represent the core elements of the business process diagram these elements contains various symbols like Events, Activities (Tasks) and Gateways.

Connecting Objects: Used to connect the BPMN core objects (Flow Objects) and provide flow of process. We can say these are simply connectors.

Swim lanes: Mechanism to organize activities and responsibilities on a process diagram. These mainly contain Pool and Lanes.

We can have pool for e.g. Racing Track we can consider as one pool and one race track have different lanes where each participant (racer) have to run on his own lane. Similarly, we can consider Business Process as pool. In one Business Process can have list of Activities, Now from that list of activities some of activities are performed by one department of organization, while some of other

activities are performed by some other department of organization. So we can divide our business process(Pool) into different lanes according to departments like “one lane for one departments” now list of task performed by one department are put into that department lane and list of tasks performed by other department are put into other department lane.

So if process is complex and contain so many activities then by putting lanes and pool we can properly structure the business process so one can easily understand.

Artifacts : Allow process designers to extend the basic BPMN notation to include additional information about the process in the process diagram that will give extra information about activities and flow objects in process

So these are the Main primary Elements of Business Process.

3.1 Flow Objects

3.1.1 Events

There are generally four types of Events

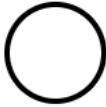






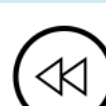
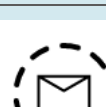
- Start event
- Intermediate event
- Boundary event
- End event





3.1.1.1 Start Events:

Start events define where a Process or Sub Process starts. The process engine supports different types of start events:

The engine requires at least one start event to instantiate a process. There can be a maximum of one blank or timer start event per process definition. There can be multiple message or signal start events.



Event	Notation	Description
-------	----------	-------------









Event	Notation	Description
Blank/None		A 'none' start event technically means that the trigger for starting the process instance is unspecified. This means that the engine cannot anticipate when the process instance must be started. The none start event is used when the process instance is started through the API
Timer		A timer start event is used to create process instance at a defined time.
Message		A message start event can be used to start a process instance using a named message.
Signal		A signal start event can be used to start a process instance using a named signal.
Conditional		A conditional start event can be used to start a process by evaluating some condition.
Escalation		An escalation start event can only be used to trigger an event sub-process - it cannot be used to start a process instance
Error		An error start event can only be used to trigger an Event Sub-Process - it cannot be used to start a process instance.
Compensation		A compensation start event can only be used to trigger an Event Sub-Process - it cannot be used to start a process instance.
Message(Non-Interrupting)		

Event	Notation	Description
Timer (Non-Interrupting)		
Conditional (Non-Interrupting)		
Signal (Non-Interrupting)		
Escalation (Non-Interrupting)		

3.1.1.1 Intermediate Events:


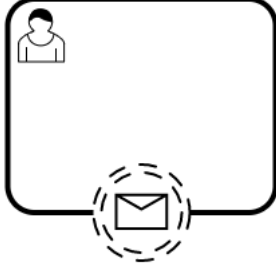
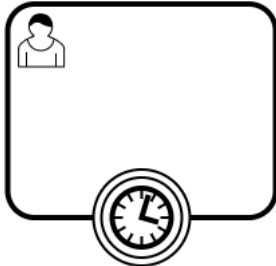
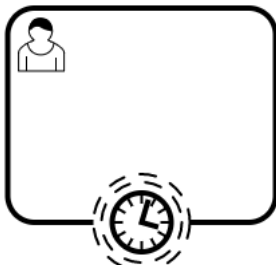
These are the Events can occur in the middle. Intermediate Events are always indicated by a double border. For e.g. Process Interrupted while running is an intermediate Event.

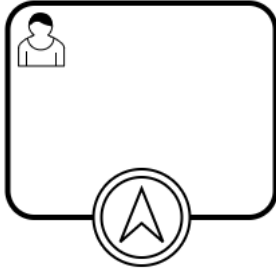



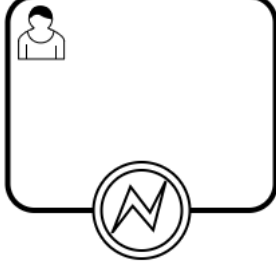
Event	Notation	Description
Message intermediate catch event		Message intermediate catching event will wait until a message with the proper name arrives.
Message intermediate throw event		A Message Intermediate Throwing event sends a message to an external service.

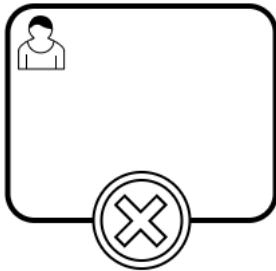
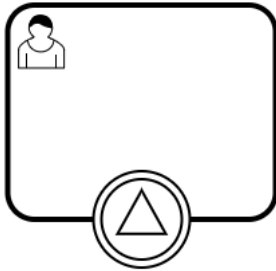
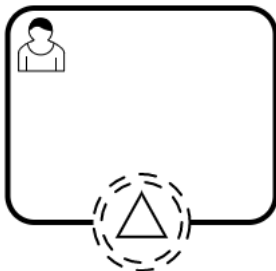
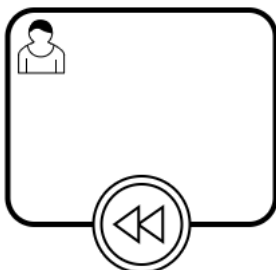
Event	Notation	Dscription
Timer intermediate catch event		A timer intermediate event acts as a stopwatch. It will stop process execution for defined time.
Escalation intermediate throw event		It will throw named escalation and same can be caught by start escalation event in event sub processor or boundary escalation event.
Conditional intermediate catch event		An intermediate conditional event is like a wait until the condition is true.
Link intermediate catch event		A Link Catching Intermediate Event provides the capability to connect two sections of a process.
Link intermediate throw event		A Link Catching Intermediate Event provides the capability to connect two sections of a process.
Compensation intermediate throw event		An intermediate throwing compensation event can be used to trigger compensation.
Signal intermediate catch event		It will wait until a signal with the proper name arrives.
Signal intermediate throw event		It throws a signal event for a defined signal.

3.1.1.2 Boundary Events:

Some events can happen on the Boundary of an activity. This event is used to divert the sequence flow from the “normal” flow to another flow. Interruptive Boundary Event represented with double border like intermediate event and non interruptive boundary event represented with double dotted border.








Events	Notation	Description
Message boundary event		Message boundary event is listening for named message. The activity is interrupted and the sequence flow going out of the event is followed.
Message boundary event(non-interrupting)		Message boundary event is listening for named message. The activity will not be interrupted and the sequence flow going out of the event is followed.
Timer boundary event		Timer boundary event defines deadline for activity. The activity is interrupted upon timer fired and the sequence flow going out of the event is followed.
Timer boundary event (non-interrupting)		Timer boundary event defines deadline for activity. The activity will not interrupt upon timer fired and the sequence flow going out of the event is followed.


Events	Notation	Description
Escalation boundary event		Catches escalations that are thrown within the scope of the activity on which it is defined. The activity is interrupted upon escalation caught.
Escalation boundary event(non-interrupting)		Catches escalations that are thrown within the scope of the activity on which it is defined. The activity will not be interrupted upon escalation caught.
Conditional boundary event		It will wait for the condition to become true while the activity is active on which it is attached, and it will interrupt the activity.
Conditional boundary event(non-interrupting)		It will wait for the condition to become true while the activity is active on which it is attached, and it will not interrupt the activity.
Error boundary event		Catches errors that are thrown within the scope of the activity on which it is defined. The activity is interrupted upon error caught.

Events	Notation	Description
Cancel boundary event		<p>A cancel boundary event, is triggered when a transaction is canceled. When the cancel boundary event is triggered, it first interrupts all active executions in the current scope. Next, it starts compensation of all active compensation boundary events in the scope of the transaction.</p>
Signal boundary event		<p>The signal boundary event catches signals with the proper name. It will interrupt the current activity.</p>
Signal boundary event(non-interrupting)		<p>The signal boundary event catches signals with the proper name. It will not interrupt the current activity.</p>
Compensation boundary event		<p>A compensation boundary event, can be used to attach a compensation handler to an activity or an embedded subprocess</p>

3.1.1.3 End Events:

These are the Events that can occur at the end. End Events always indicated by a single thick or bold border. For e.g. Process Stopped is an End event.

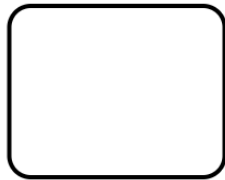
Events	Notation	Description
End event		Ends current path of execution only.
Message end event		Current path of execution is ended and a message is sent.
Escalation end event		Current path of execution is ended and a named escalation is thrown.
Error end event		When process execution arrives at an error end event, the current path of execution is ended and an error is thrown.
Compensation end event		A compensation end event triggers compensation and the current path of execution is ended.
Signal end event		A signal end event throws a signal event for a defined signal and the current path of execution is ended
Terminate end event		A terminate event ends the complete scope where the event is raised and all inner scopes.

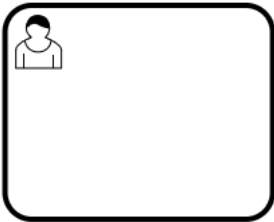
Events	Notation	Description
Cancel end event		The cancel end event can only be used in combination with a transaction sub process. When the cancel end event is reached, a cancel event is thrown which must be caught by a cancel boundary event. The cancel boundary event then cancels the transaction and triggers compensation.


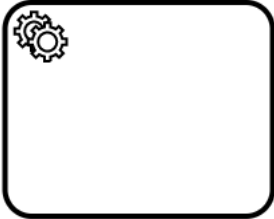
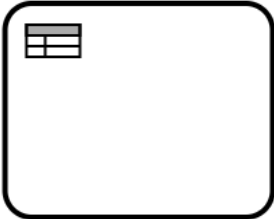

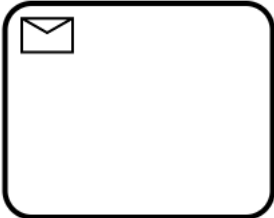
3.1.2 Activities

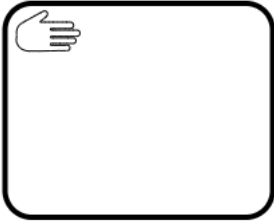
Activities describes, the kind of work that is being done in a particular process instance. There are four activity types, they are tasks, sub-processes, transactions, and call activities.

3.1.2.1 Tasks






Tasks	Notation	Description
User task		A user task is used to model work that needs to be done by a human actor.


Tasks	Notation	Description
Script task		A script task defines a script in a language that the engine can interpret
Service task		A Service Task is a task that uses some sort of service, which could be a Web service or an automated application.
Business rule task		A Business Rule Task provides a mechanism for a process to provide inputs to a business rules engine and to get the output of calculations that the business rules engine might provide.
Send task		A Send Task is a simple task that is designed to send a message to an external participant. Once the message has been sent, the task is completed.
Receive task		A Receive Task is a simple task that is designed to wait for a message to arrive from an external participant (relative to the Process). Once the message has been received, the task is completed.

Tasks	Notation	Description
Manual task		A Manual Task is a task that is expected to be performed without the aid of any business process execution engine or application, for example, installing a telephone at a customer location.

3.1.2.2 Gateways

A Gateway allows you to control the flow of a process through a sequence flow. The term Gateway implies that there is a gating mechanism that either allows or disallows passage through the Gateway. Tokens that arrive at the gateway can be merged as inputs and/or split as outputs.

Gateways	Notation	Description
Exclusive gateway or XOR gateway or exclusive data-based gateway		Exclusive Gateway (Decision) is used to create alternative paths within a process flow. This is basically the diversion point in the road for a process. Only one alternative path can be taken for a given instance of the process
Parallel gateway		A Parallel Gateway is used to synchronize (combine) and create parallel flows.
Inclusive Gateway		The Inclusive Gateway can be seen as a combination of an exclusive and a parallel gateway.

Gateways	Notation	Description
Event Based Gateway		The event-based Gateway allows you to make a decision based on events. Each outgoing sequence flow of the gateway needs to be connected to an intermediate catching event. When process execution reaches an event-based Gateway, the gateway acts like a wait state

3.1.2.1 Sub processes

A sub-process is a group of tasks that fit together well. There are two different views of sub-process. One is the 'collapsed view', that has an expandable plus sign, to show 'more details'. The other view is an 'expanded sub-process view', which is large enough to house, all the tasks that describe the sub-process fully.

A sub-process used for Reusability and grouping of elements in process.

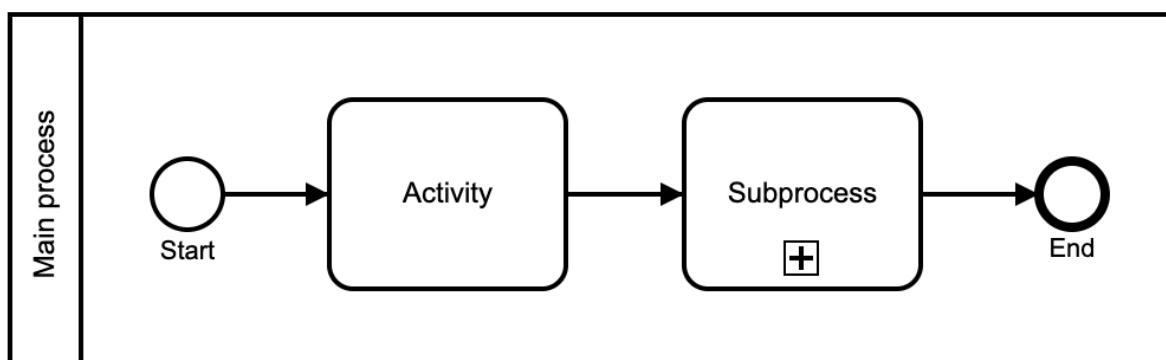
Different types of SubProcesses used in business process modeling are as follows:

- ▶ SubProcess
- ▶ Event SubProcess
- ▶ Transaction

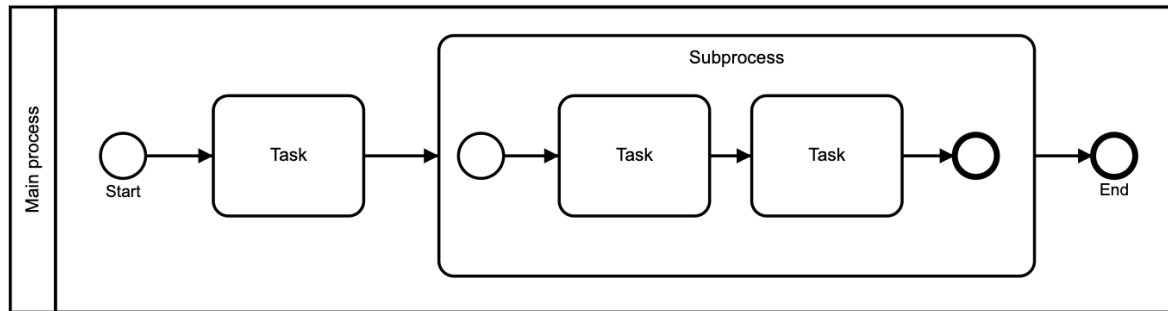
SubProcess (Embedded sub process)

A subprocess is an activity that contains other activities, gateways, events, etc., which itself forms a process that is part of a bigger process. A subprocess is completely defined inside a parent process (that's why it's often called an embedded Subprocess).

Collapsed view:

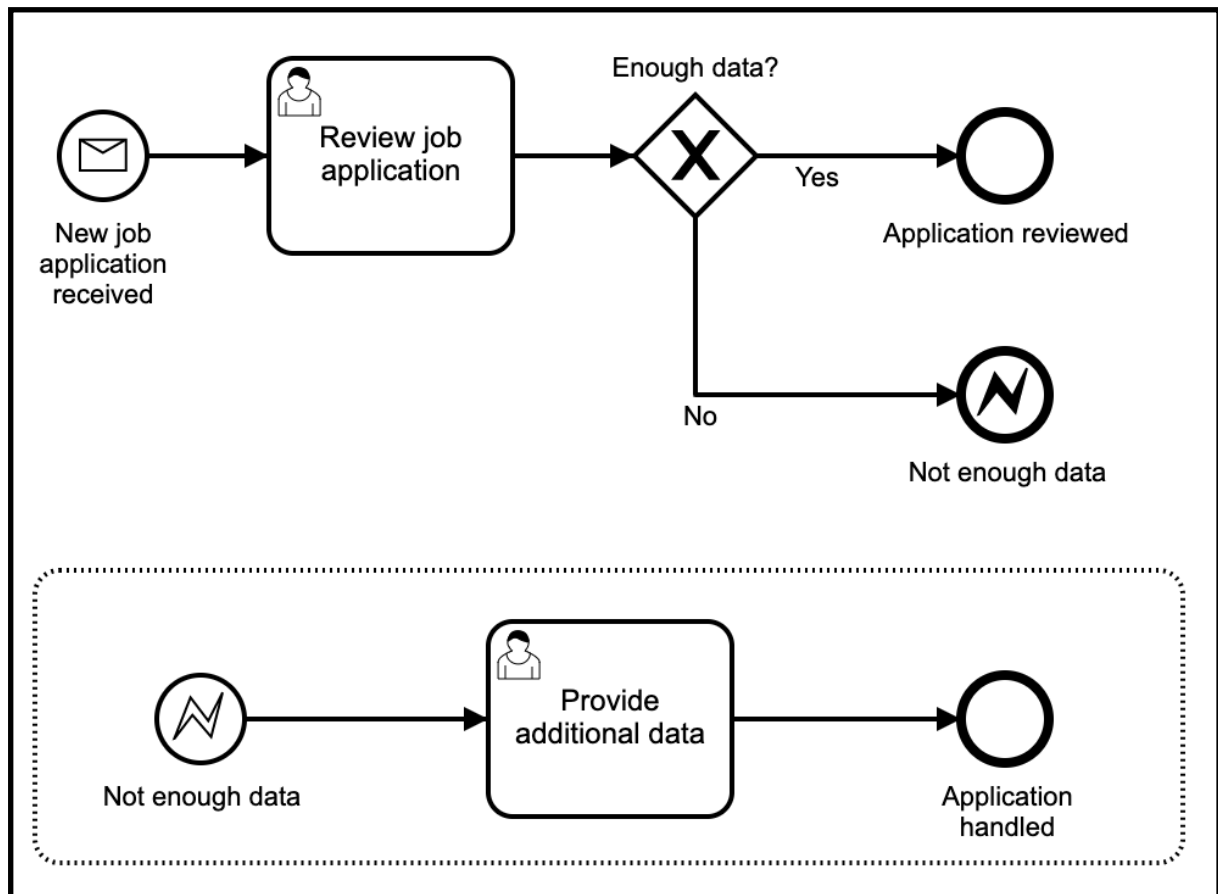


Expanded view:

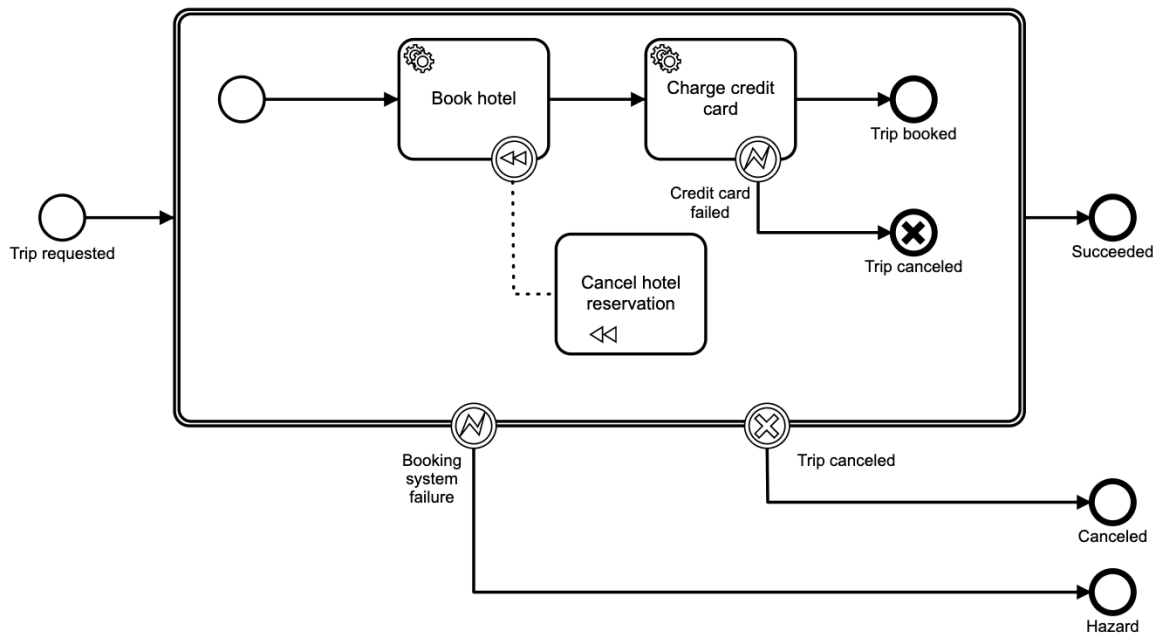


Event SubProcess

The event subprocess is a subprocess that is triggered by an event. An event subprocess can be added at the process level or at any subprocess level.



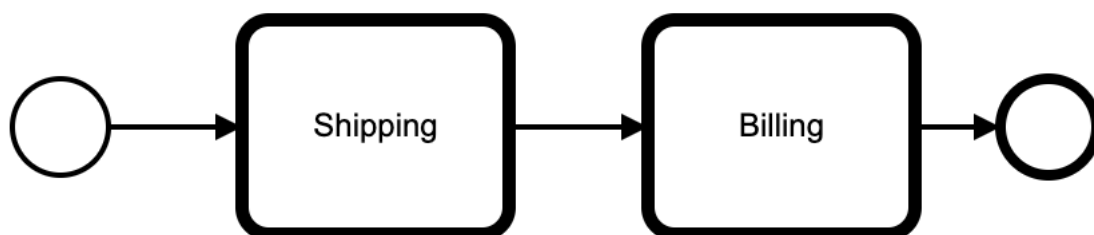
Transaction SubProcess



transaction subprocess is an embedded subprocess which can be used to group multiple activities to a transaction. A transaction is a logical unit of work which allows grouping of a set of individual activities, so that they either succeed or fail collectively.

3.1.2.2 Call Activity

The call activity references a process that is external to the process definition.

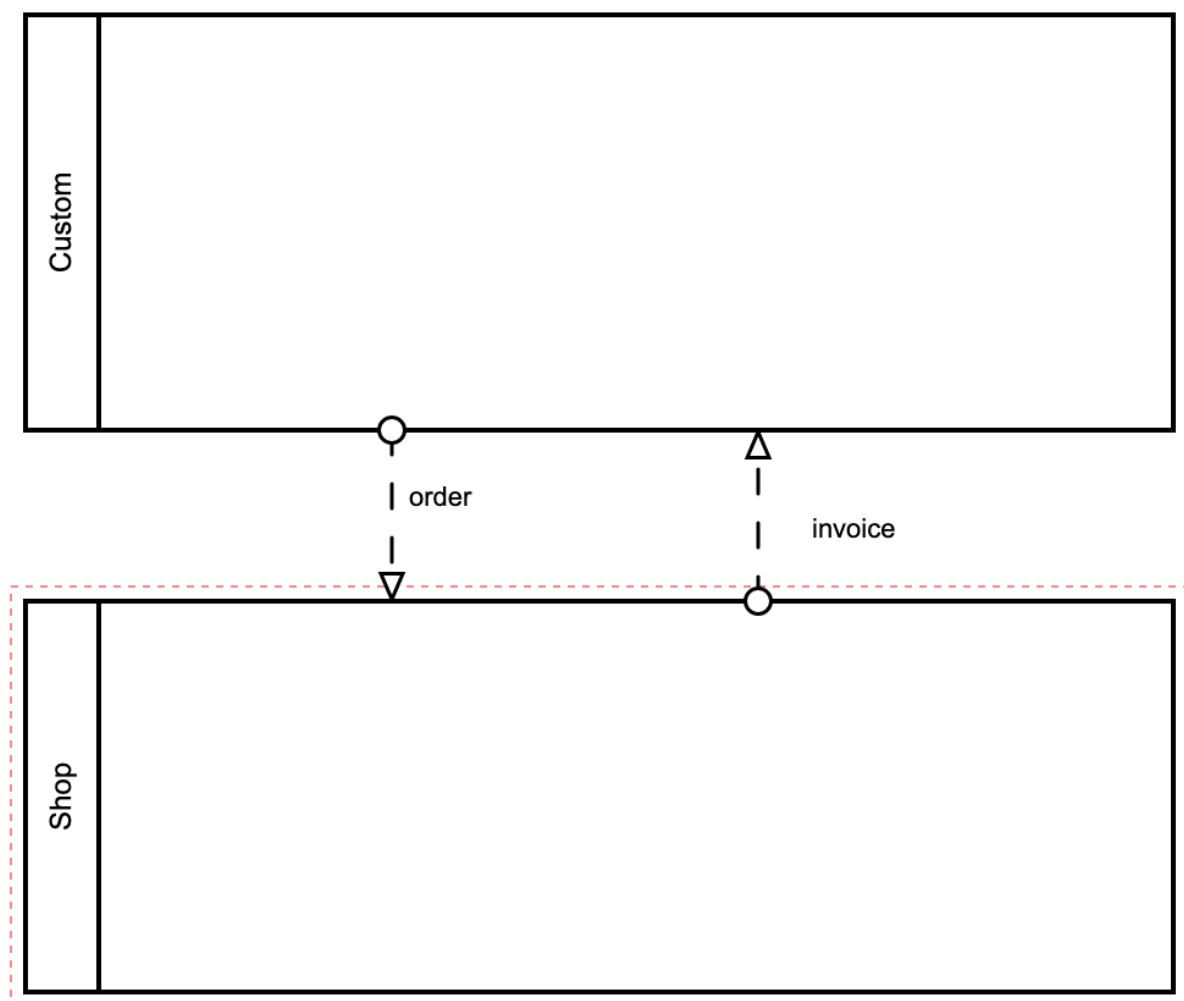


3.2 Flow Objects (Connectors)

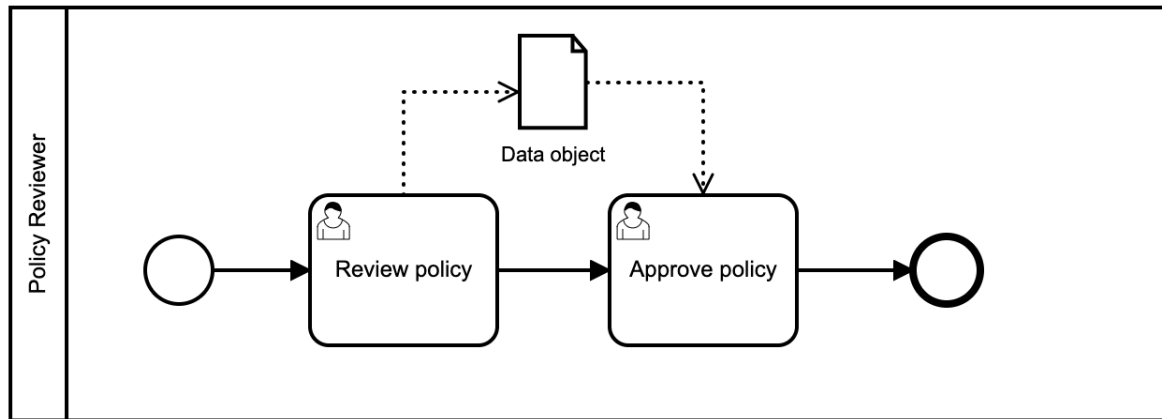
Sequence Flow : A Sequence Flow is used to show the order that activities will be performed in a Process.



Message Flow : A the communication between pools is achieved by the use of message. Message flow is used to show the flow of messages between pools or flow elements between pools.



Association : An Association is used to associate data, information and artifacts with flow objects



3.2 Artifacts

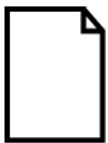
Artifacts represent information relevant to the model but not to individual elements within the process.

There are generally three types of Artifacts.

- Data object
- Text annotation
- Group

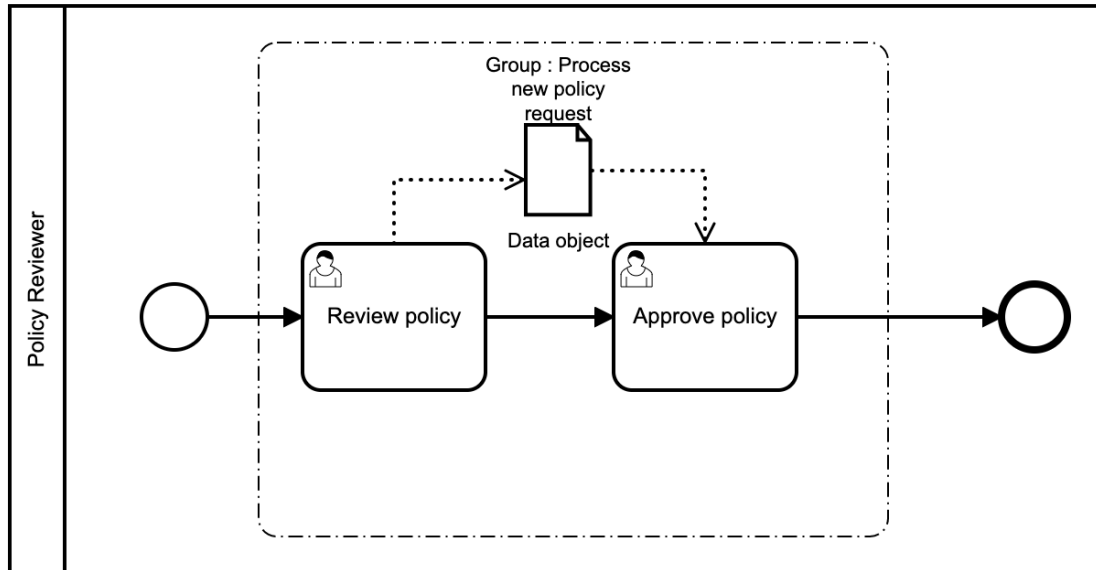
3.2.1 Data object

Represent data placed into the process, data resulting from the process, data that needs to be collected, or data that must be stored.



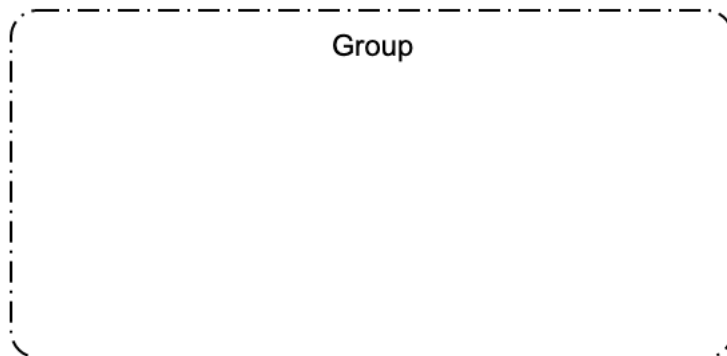
3.2.2 Text annotation

Text annotation allow the modeler to describe additional flow parts of the model or notation



3.2.3 Group

Group organize tasks or processes that have significance in the overall process.



3.3 Swimlanes

Pools and lanes define responsibilities within a business process. Hereby, a pool is a unit with clear organizational boundaries to its environment, such as a company or an organization. Lanes are always contained by a pool or another lane and communicate with other lanes within the same pool without any restrictions. They typically represent the different roles that execute a process – the process participants.

