Business Process Modelling with BPMN

Knut Hinkelmann
**BPMN – Business Process Model and Notation**

- BPMN is a graphical modeling notation for business processes that is independent of a specific implementation environment.

- BPMN was officially adopted as an OMG specification in 2006, updated in 2008 and now available in version 2.0 (http://www.omg.org/spec/BPMN/2.0/)

- BPMN provides a standardized bridge for the gap between the *business process design* and *process implementation*.
Elements of BPMN

Elements of BPMN can be divided into 4 categories:

- **Flow Objects**
  - Events
  - Activities
  - Gateways

- **Connectors**
  - Sequence Flow
  - Message Flow
  - Association

- **Artefacts**
  - Data Object
  - Text Annotation

- **Swimlanes**
  - Pool
  - Lanes (within a Pool)
Activities

- An activity is work that is performed within a business process.
- Typically an activity is one step of a larger business process.
- Activities are rounded rectangles (some tools use colors)
- There are two types of activities:
  - A **Task** is a unit of work, the job to be performed.
  - When marked with a [+], it indicates a **Sub-Process**, an activity that can be refined.
Sub-Processes

- A Sub-Process is a compound activity that is included within a Process.
  - A process can be broken down into a finer level of detail through a set of sub-activities.

- Two kinds of representation
  - Collapsed: the details of the Sub-Process are not visible in the Diagram. A “plus” sign in the lower-center of the shape indicates that the activity is a Sub-Process and has a lower-level of detail.
  - Expanded: the details (a Process) are visible within its boundary.
**Task Types**

- Send Task
- Receive Task
- User Task
- Manual Task
- Business Rule Task
- Service Task
- Script Task

- Types specify the nature of the action to be performed.
- They can be identified by a symbol inside the object.
Activity Markers

Markers indicate execution behavior of activities / subprocesses:

- Sub-Process Marker
- Loop Marker
- Parallel MI Marker
- Sequential MI Marker
- Ad Hoc Marker
- Compensation Marker
**Sequence Flow**

- A Sequence Flow is used to show the order that activities will be performed in a Process.
- The source and target must be one of the following objects:
  - Events
  - Activities
  - Gateways
- In a sequence of activities, the subsequent activity is performed after the previous activities is finished.

(Bridgeland & Zahavi 2009, p. 106)
Events are states that affect the flow of the process
- they start, interrupt and finish the flow
- they can trigger an activity or are its result

Events are represented as circles. The type of boundary determines the type of Event
- Start Event
- Intermediate Event
- End-Event

Events can have descriptions, just as tasks.
Example: A simple End-to-End Process

- A process begins with a start event and ends with an end event
  - **Diner Arrives** is the start event
  - **Diner Seated** is the end event

- An intermediate event happens after the process starts and before it ends
  - **Party Arrived** is a catching intermediate event that models a delay: When the first diner of a party arrives the host checks the reservations but does not seat the diner until the rest of the party arrives.

- Note that the difference in the naming of events and activities:
  - Activity names are typically imperative sentences, they sound like command. The verb is at the beginning of the name.
  - Event names are typically declarative sentences, describing a state or something that happens

(Bridgeland & Zahavi 2009, p. 108f)
### Event-Types

<table>
<thead>
<tr>
<th>Event Type</th>
<th>None</th>
<th>Message</th>
<th>Timer</th>
<th>Conditional</th>
<th>Signal</th>
<th>Escalation</th>
<th>Error</th>
<th>Compensation</th>
<th>Parallel</th>
<th>Multiple</th>
<th>Link</th>
<th>Cancel</th>
<th>Terminate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start Events</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top-Level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Sub-Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interrupting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Sub-Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Interrupting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Intermediate Events</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catching</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Throwing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interrupting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Interrupting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>End Events</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Business Process Modeling, BPMN
Event Types

- **None**: Untyped events, indicate start point, state changes or final states.
- **Message**: Receiving and sending messages.
- **Timer**: Cyclic timer events, points in time, time spans or timeouts.
- **Conditional**: Reacting to changed business conditions or integrating business rules.
- **Signal**: Signalling across different processes. A signal thrown can be caught multiple times.
- **Escalation**: Escalating to an higher level of responsibility.
- **Error**: Catching or throwing named errors.
- **Compensation**: Handling or triggering compensation.
- **Multiple**: Catching one out of a set of events. Throwing all events defined
- **Parallel Multiple**: Catching all out of a set of parallel events.
- **Link**: Off-page connectors. Two corresponding link events equal a sequence flow.
- **Cancel**: Reacting to cancelled transactions or triggering cancellation
- **Terminate**: Triggering the immediate termination of a process.
**Properties of Events**

- **Start-Events:**
  - Top-level
  - Event Sub-Process Interrupting
  - Event Sub-Process Non-Interrupting

- **End-Event**

- **Intermediate Events**
  
  Between Activities:
  - Throwing
  - Catching

  On the boundary of activities
  - Boundary Interrupting
  - Boundary Non-Interrupting
Intermediate Events

- Events that are placed within the process flow represent things that happen during the normal operations of the process. They can represent …
  …a «trigger» that initiates an activity – catching
  …the result of an activity – throwing

- Events that are attached to the boundary of an activity can occur during the activity. They can …
  …interrupt the activity (solid lines)
  …open an additional path without interrupting (modelled with dashed line)
Catching and Throwing Events

- **A throwing** intermediate event, with the black icon inside, means the process generates the trigger signal.

- **A catching** intermediate event, with the wide icon inside, means the process waits for the trigger signal.
  - A catching event interrupts a process and waits for the trigger signal to arrive.
  - When the trigger signal arrives, the process resumes on the sequence flow out of the event.
Swimlanes – Pools and Lanes

- A pool is a container for a business process or a participant in a collaboration.
- A lane is an optional subdivision of a process level. They are typically used to associate process activities with particular actors.
  - Each participant that performs activities in a business process has a lane.
  - A lane can represent a role, an organizational unit, or a system.

(Bridgeland & Zahavi 2009, p. 110f)
Gateways

- **Gateways** model sequence flow alternatives, i.e. they represent points of control.
- They split and merge the flow of a Process.
- All types of Gateways are diamonds.
- The underlying idea is that Gateways are unnecessary if the Sequence Flow does not require controlling.
Gateways – Splitting and Merging

**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.

**Event-based Gateway**: Sequence flow is routed to the subsequent event/task which happens first.

**Parallel Gateway (AND)**: 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 (OR)**: When splitting, one or more branches are activated. All active incoming branches must complete before merging.
Exclusive Gateways

- For exclusive Gateways exactly one of the following sequence flows is selected

- The name of the gateway is a question with the alternative answers to the questions as labels on the outgoing sequence flows
Exclusive Gateways based on Data

- The Gateway (Decision) creates alternative paths based on defined conditions.
- Exclusive Gateways based on Data are the most commonly used Gateways.
- They can be shown with or without an internal „X“ marker. Without is the most common use.
Exclusive Gateways based on Events

- Alternatives in this Decision are based on events that occur at the point in the process rather than conditions.
- The Multiple Intermediate Event is used to identify this Gateway.
- The Events that follow the Gateway Diamond determine the chosen path:
  - The first Event triggered wins.
A parallel gateway

- starts parallel work, i.e. two (or more) sequence flows that then progress at the same time
- parallel flows are joined back together by another parallel gateway

(Bridgeland & Zahavi 2009, p. 114f)
Inclusive Gateway

- An inclusive gateway allows either of the outgoing sequence flow to be taken or several in parallel.
- They usually are followed by a corresponding merging Inclusive Gateway.
- Example: The following process shows a process where the guests do not have both appetizers and entrees but can have only one of them.

(Bridgeland & Zahavi 2009, p. 114f)
Default Sequence Flow and Conditional Sequence Flow

- One of the outgoing sequence flows from a gateway can be marked as default – the one that is taken if there is no reason to take another sequence flow.

- The default is modeled with a short line crossing the sequence flow.

- The same can be modeled without a gateway using a conditional sequence flow.

- A conditional sequence flow is a sequence flow that includes a condition.

Example: Identical process with a gateway and with conditional sequence flow

(Bridgeland & Zahavi 2009, p. 116)
Artifacts

- Artifacts provide the capability to show information beyond the basic flow-chart structure of the Process
- There are currently three standard Artifacts in BPMN:
  - Data Objects
  - Groups
  - Annotations
- A modeler or tool can extend BPMN by defining new Artifacts
Text Annotations and Data Objects

- Text Annotations are a mechanism for a modeler to provide additional information about a Process.
- Text Annotations can be connected to a specific object on the Diagram with an Association.
- Data Objects can be used to define inputs and outputs of activities.
- Data Objects can be given a “state” that shows how a document may be changed or updated within the Process.
Data Elements in BPMN

- BPMN 2.0 contains new graphical elements to represent data
  - Data Associations: connecting Data Objects to Activities
  - Data Inputs and Outputs can be visualized
  - Data Stores represent repositories or databases
  - Collections, marked by [+] , represent groups of Data Objects
Swimlanes partition and organise activities

There are two main types of swimlanes: Pool and Lane

- Pools represent Participants in an interactive (B2B) Business Process Diagram
- Lanes represent sub-partitions for the objects within a Pool – they represent participants of a process

(Bridgeland & Zahavi 2009, p. 123)
A Pool may be a „black box“ or may contain a Process

Interaction between Pools is handled through **Message Flow**

A Message Flow can connect to the boundary of the Pool or to an object within the Pool

Message Flows are not allowed between objects within a single Pool

Sequence Flow must not cross the boundary of a Pool (i.e. a Process is fully contained within a Pool)
Data Transfer with Message Flow and Associations

Message Flow between pools:

Insurance

Request medical certificate

Physician

Write certificate

Data transfer inside a pool MUST NOT be modeled with Message Flow but with Associations:
Groups

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

- A Group can stretch across the boundaries of a Pool, often to identify Activities that exist within a distributed business-to-business transaction.