

BPMN® Poster Series #10

BPMN Diagrams – Must and Should Have

by Gregor Jošt

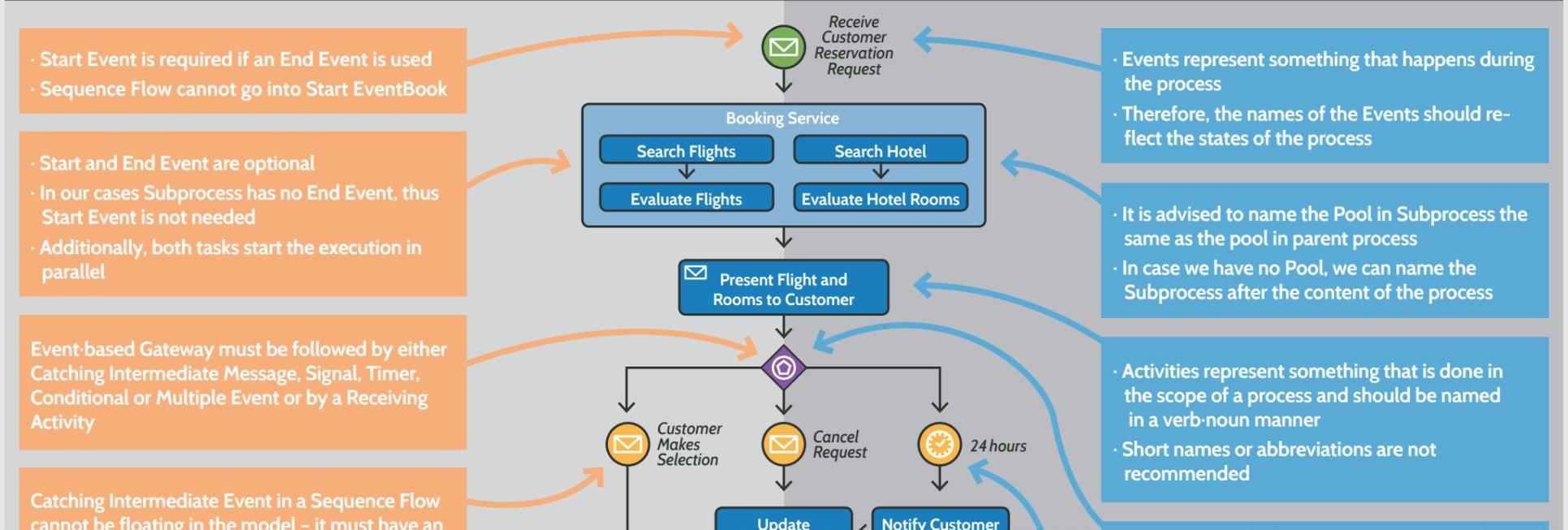
In this poster we will define which rules every BPMN modeler **MUST** follow and which suggestions every BPMN modeler **SHOULD** follow. We will achieve this by a case study, where we will analyze a BPMN diagram, taken from the BPMN's specification. The diagram is named "The Travel Booking Diagram" and represents the view of a travel booking agency. In order to represent as many different situations within a diagram as possible, some parts of the diagram have been slightly altered.

MUST

SHOULD

The BPMN specification provides a lot technical definitions and rules, such as connecting the elements, when and if we should use Start and End events, which elements must have both incoming and outgoing Sequence Flows and much more. Modelers must abide by these rules and follow them unconditionally in order to create syntactically sound diagrams.

The BPMN specification does not provide a mechanism that would reveal the process logic clearly and without the need for any additional documentation. Modelers can conduct syntactically and semantically correct diagrams, yet fail to be unambiguous. In this light, several good practices have been proposed, along with a basic methodology. Modelers should abide by these guidelines in order to increase the understandability of their diagrams.



Start Event is required if an End Event is used
Sequence Flow cannot go into Start Event

Start and End Event are optional
In our cases Subprocess has no End Event, thus Start Event is not needed
Additionally, both tasks start the execution in parallel

Event-based Gateway must be followed by either Catching Intermediate Message, Signal, Timer, Conditional or Multiple Event or by a Receiving Activity

Catching Intermediate Event in a Sequence Flow cannot be floating in the model – it must have an incoming and an outgoing Sequence Flow

Events represent something that happens during the process
Therefore, the names of the Events should reflect the states of the process

It is advised to name the Pool in Subprocess the same as the pool in parent process
In case we have no Pool, we can name the Subprocess after the content of the process

Activities represent something that is done in the scope of a process and should be named in a verb-noun manner
Short names or abbreviations are not recommended

Boundary Event (Timer in our case) must have one outgoing Sequence Flow and must be Catching, not Throwing

Gateways must have either 2 incoming (merge) or 2 outgoing (split) Sequence Flows
Gateways cannot send or receive Message Flows
Exclusive Gateway (XOR) can have a marker (X symbol) either displayed or hidden

There are three basic outcomes of a Task: successful completion, failed completion (Cancel Event) and

If Subprocess or Transaction Subprocess has a Start Event, this Start event must be None and an End Event is needed

End Event is required if a Start Event is used
Sequence Flow cannot go out of an End Event

Event-based Gateways (and Parallel Gateways) are conventionally not named

Timer Events should define a specific time/date or schedule in their names

Converging Exclusive Gateways are conventionally named

Exclusive Gateway should be named in question

The name should be in form of one verb, one

The name of the data object should be in form of a noun, related to the business domain

Events should represent the corresponding states of the process

100's of FREE Downloads Available!

Posters, Whitepapers, Case Studies, Videos, Quizzes, Articles

Download Now!

