White Paper

The Role of Events in Modeling Processes Using BPMN 2.0

WP0041 | September 2012

David Jones and **Roderick Brown**

David Jones is a Senior Consultant with Enterprise Architects in Australia, specializing in Enterprise Architecture, particularly Business Architecture. He is also an experienced practitioner in business process improvement and simplification. David has worked with many sector clients, undertaking assignments in Financial Services, Telecommunications and Power Utilities.

Roderick Brown is a freelance Consultant working in Melbourne, Australia, specializing in Business Architecture, particularly in Process Architecture. He is also an experienced practitioner in business process improvement and simplification. Roderick has worked with many sector clients, undertaking assignments in Banking, Investment Management and Wealth Management.

David and Roderick are passionate about helping organizations understand and document their own business processes, using frameworks such as APQC's Process Classification Framework and standards such as BPMN as well as applying simple approaches to improve and simplify these business processes.

Events have been one of the distinct features of BPMN since the start and have provided a key point of difference in documenting business processes with BPMN and more traditional approaches to process Modeling and mapping.

Although Events have been previously used when documenting processes, Events were typically either documented as Attributes of Processes and Activities or were defined in the form of specialist diagrams such as State Transition diagrams.

Earlier versions of BPMN only had limited Event handling capability which meant Modeling using standard symbols that then required the use of a myriad of decision gateways. However, BPMN 2.0 has an extensive array of Events that can cater for nearly all possible circumstances.

This White Paper will demonstrate how Events are used when Modeling Processes in BPMN 2.0 by examining:

- The role of Events in BPMN 2.0 in terms of what they are, why we need them as well as what they are used for;
- The different types of Events available in BPMN 2.0 and how they are commonly used; and;
- Some best practice recommendations in how to name and use Events in your business process Modeling.

The Role of Events in BPMN 2.0

What are Events in BPMN 2.0?

The OMG's BPMN 2.0 Specification defines an Event as:

> *An Event is something that "happens" during the course of a Process. These Events affect the flow of the Process and usually have a cause or an impact and in general require or allow for a reaction.*

As is explained in the Specification, the term "event" is general enough to cover many things in a Process. For example, definition considers all the following as to be Events:

- The start of an Activity;
- The end of an Activity;
- The change of state of a document;
- A Message that arrives; and
- A Message that is sent.

The introduction of event-triggered behavior directly in the diagram is what separates BPMN 2.0 from other forms of process Modeling.

To provide further clarity around "What is an Event?" we have found a more complete definition that provides more clarity around events:

> *An Event is a BPMN 2.0 process modeling element representing something that "happens" during the course of a process. Events affect the flow of the process. Some events such as most start events and some intermediate events have a "trigger" which defines the cause for the event, while end events often define the "result" or consequence of the ending of a particular sequenced flow.*
>
> *Start Events can only "catch" a trigger. Similarly, End Events can only create "throw" a result. Intermediate events are less constrained and may catch triggers or throw results.i*

Understanding "Triggers" is an important and useful way of analysing processes, as it enables you to identify and articulate the circumstances of why an Activity needs to happen.

At its simplest, an Event shows something:

- Has happened; or
- (May be) About to happen.

Why do we need Events?

You may wonder why we need Events or what the point of events is? Events are used as they provide the means of explaining to the reader what has happened (or is about to happen) at the appropriate points within and across process.

While an Activity describes what is being done, the Event is used to define the circumstances of why the Activity needs to be undertaken. Not surprisingly, the use of Events helps to improve the overall completeness and quality of process documentation.

Prior to the arrival of BPMN 2.0, processes were commonly documented using simple symbols at the start and end of a process. However, these don't explain what starts or ends the process.

To address this, modellers often used State Transition diagrams to show what triggers each process, but this means using multiple types of diagrams rather than just one.

In BPMN 2.0, using Events allows process modellers to clearly document not only what the process is doing, but also what is causing it to happen.

For example, Events can be used to explain the path the process takes in the Event of an error or escalation requirement, alternatively that some Events only occur after a time trigger.

What do we use Events for?

It is because of Events that processes are now able to better reflect the real world and are models as opposed to just a drawing. It means a model can accurately capture when something needs to wait or what to do when there is an exception.

The introduction of Events into BPMN 2.0 provides standards for depicting Events in process models that cover not only the start and end but a range of other types of Events such as:

- Timing;
- Escalation; and
- Errors.

Importantly events have been expanded to include boundary events. This is an important addition as it allows modellers to define an alternate path when certain conditions are not met, such as the need for handling of escalations or errors.

Boundary events can be used to depict scenarios such as what to do when the activity has not received a response in the required time or what to do when the activity breaches set parameters. For example, an interrupting boundary message event can be used to handle the scenario where an order breaches a credit limit.

Types of Events in BPMN 2.0

Overview

In explaining the different types of events, the Object Management Group's BPMN 2.0 specification has been referenced, rather than be too influenced by personal preferences. The specification published by the Object Management Group in January 2011 sets out the representation for each event type and the behavior of each event type. This paper goes a step further in that the paper explains how best to use the events in Modeling.

Start Events Triggers

A Start Event can only occur at the start of a process. It is the means by which a Process is started (hence the name) or initiated. A Start Event is graphical represented in BPMN 2.0 as a circle with a single thin outside border, for example:

Figure 1: BPMN graphical symbol for Start Event

A start event is typically triggered by:

- **None** – which has no specific defined trigger;
- **Message** – is triggered by the arrival of a message from another process or participant;
- **Timer** – is triggered by the arrival of a specific date (and time) or after a cycle, e.g. 9am on Monday or 1st of the Month; and
- **Conditional** – is triggered when a specified condition is reached, e.g. credit limit is exceeded;
- **Escalation** – occurs when escalation is identified and has been triggered;
- **Error** – is triggered when some form of error state has been previously identified and triggered; and
- **Signal** – is triggered by the arrival of a signal broadcast by another activity or process.

The most commonly used Start Events include:

Figure 2: Common types of BPMN Start Events

Intermediate Events Triggers

Overview

Intermediate Event occurs anywhere within a Process between the start and end events. An Intermediate Event is graphical represented in BPMN as a circle with twin thin outside borders:

Figure 3: BPMN graphical symbol for Intermediate Symbol

An intermediate event is typically triggered or impacted by:

- **None** – which has no specific defined trigger;
- **Message** – by the arrival of a message from another process or participant (a Catch event) or sending of a message to another process or participant (a Throw event);
- **Timer** – is triggered by the need to wait for a specific date (and time), cycle or delay, e.g. Monday at 9am, End of the Month or 12 hours;
- **Escalation** – occurs when escalation is identified and needs to be triggered;
- **Compensation** - occurs when a process has finished, but that compensation is needed;
- **Conditional** – is triggered when a specified condition is reached, e.g. credit limit is exceeded;
- **Link** – is used to connect two parts of the process, usually across page boundaries or to remove the need for crossed lines; and
- **Signal** – is triggered by the arrival of a signal broadcast by another activity or process.

The most commonly used Intermediate Events are:

Figure 4: Common types of BPMN Intermediate Events

A number of the trigger types of Intermediate Events occur in two forms:

- **Throwing** – which occurs when the Intermediate Event is causing something to happen, for example a 'throwing' Intermediate Message Event represents the sending (or throwing) of a message; and

- **Catching** – which occurs when the Intermediate Event is listening and waiting for something to happen, for a catching Intermediate Message Event represents the receiving (or catching) of a message.

Intermediate Boundary Events Triggers

Intermediate Boundary Events are a special form of Intermediate Events. A Boundary Event is attached to an Activity and is consequently triggered during the execution of an Activity if the specific event trigger occurs.

Boundary Events can be defined in two forms:

- **Interrupting Boundary Events** – which interrupts the execution of the Activity, i.e. the Activity is cancelled, when the event trigger occurs; and
- **Non-interrupting Boundary Events** – which does not interrupt the execution of the Activity, i.e. the Activity continues execution, when the event trigger occurs.

An Interrupting Intermediate Boundary Event is graphically represented in BPMN using the same symbol as a normal Intermediate event, i.e. a circle with twin thin outside borders, but shown on the side of an Activity (either Task or Sub-Process, but more commonly on a Sub-Process), for example:

Figure 5: BPMN graphical Symbol for Intermediate Boundary Event

A non-Interrupting Intermediate Boundary Event is graphically represented in BPMN using the same symbol as a normal Intermediate event, i.e. a circle with twin thin outside borders that are dashed, but shown on the side of an Activity (either Task or Sub-Process, but more commonly on a Sub-Process) as shown below:

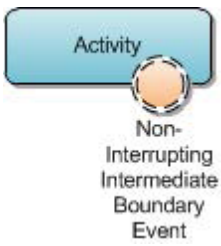

Figure 6: BPMN graphical symbol for Non-interrupting Boundary Event

The set of Boundary Event used in preparing Analytical type of Process Models are:

- Message Intermediate Boundary Event (both Interrupting and Non-Interrupting);
- Timer Intermediate Boundary Event (both Interrupting and Non-Interrupting);
- Error Intermediate Boundary Event (Interrupting only);
- Escalation Intermediate Boundary Event (Non-Interrupting only);
- Signal Intermediate Boundary Event (both Interrupting and Non-Interrupting); and
- Conditional Intermediate Boundary Event (both Interrupting and Non-Interrupting).

The most commonly used Interrupting Boundary Intermediate Events are:

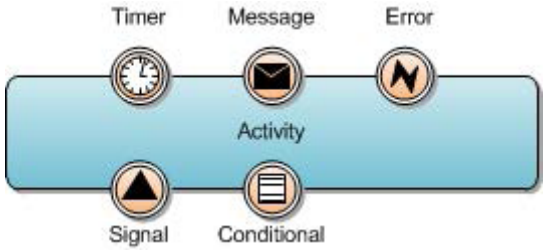

Figure 7: Commonly used Interrupting Boundary Events

The most commonly used non-Interrupting Boundary Intermediate Events are:

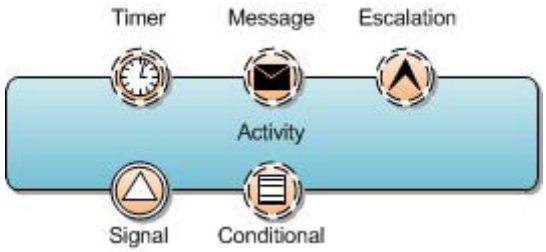

Figure 8: Commonly used non-Interrupting Boundary Events

End Events Triggers

An End Event can only occur at the end of a process and is used to define the exit point of a process and be the trigger for why the process ended. An End Event is graphically represented in BPMN 2.0 as a circle with either a single (or twin) thick outside borders, as shown below:

Figure 9: BPMN graphical symbol for End Event

An end event is typically triggered or impacted by:

- **None** – which has no specific defined trigger;
- **Message** – is triggered by the sending of a message to another process or participant;
- **Error** – is impacted when some form of error state is identified and a named Error should be generated and triggered;
- **Escalation** – occurs when escalation is identified and needs to be triggered;
- **Signal** – is triggered by the arrival of a signal broadcast by another activity or process;
- **Compensation** - occurs when a process has finished, and compensation is necessary; and
- **Terminate** – indicates that all Activities in the Process should be immediately ended.

The most commonly used End Events are:

Figure 10: Commonly used End Events

Send and Receive Task Types

Although the Send and Receive type of Tasks are not really Events, we have included them here as they both effectively include an implied or implicit Message type Intermediate Event.

The Send type Task includes an implied throw Intermediate Message Event and the Receive type Task includes an implied catch Intermediate Message Event. Although these are both useful, we recommend that modellers new to BPMN 2.0 should refrain from using them and explicitly use the Message type Intermediate Event with either the User or Service types of Task types.

Demonstrating the Role of Events

Overview

The best way to demonstrate the capabilities and importance of events is to look at sample process models. We have taken the examples shown here from the BPMN 2.0 by Example document published by the Object Management Group (OMG)ii.

Example 1

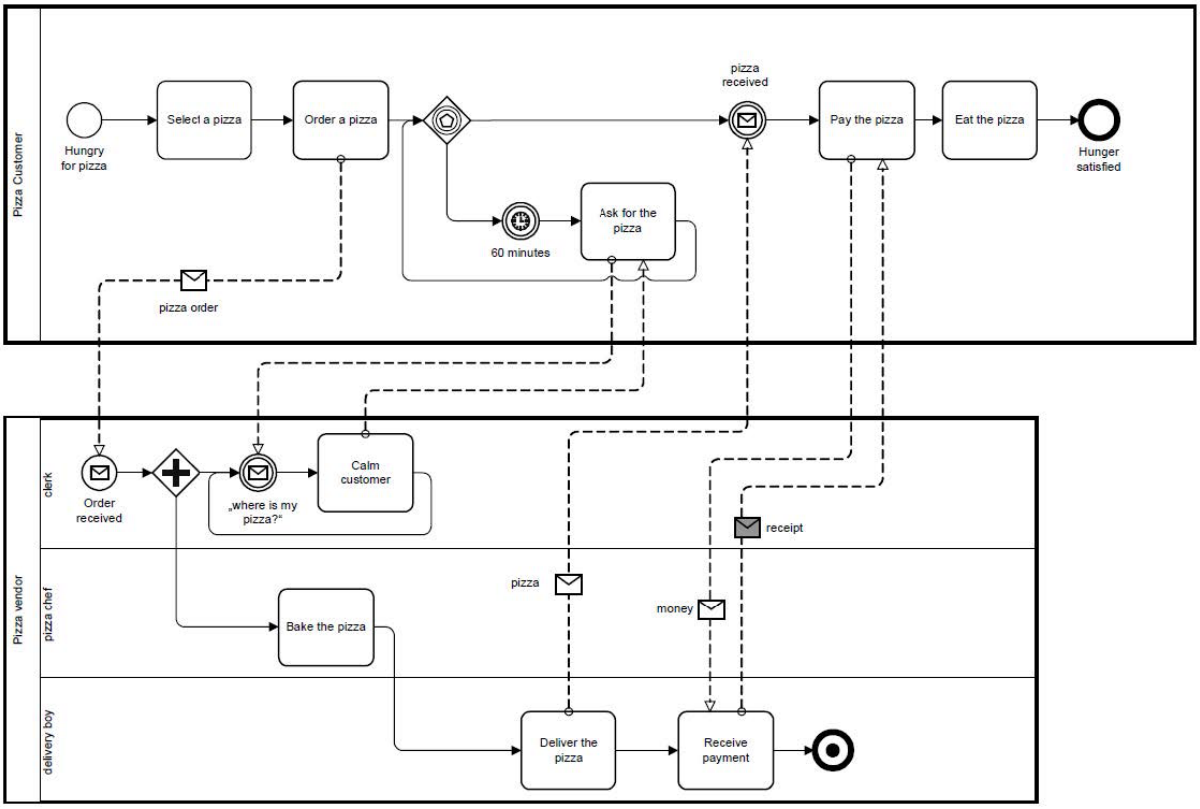

Figure 11: Pizza Order and Delivery Example

Example 2

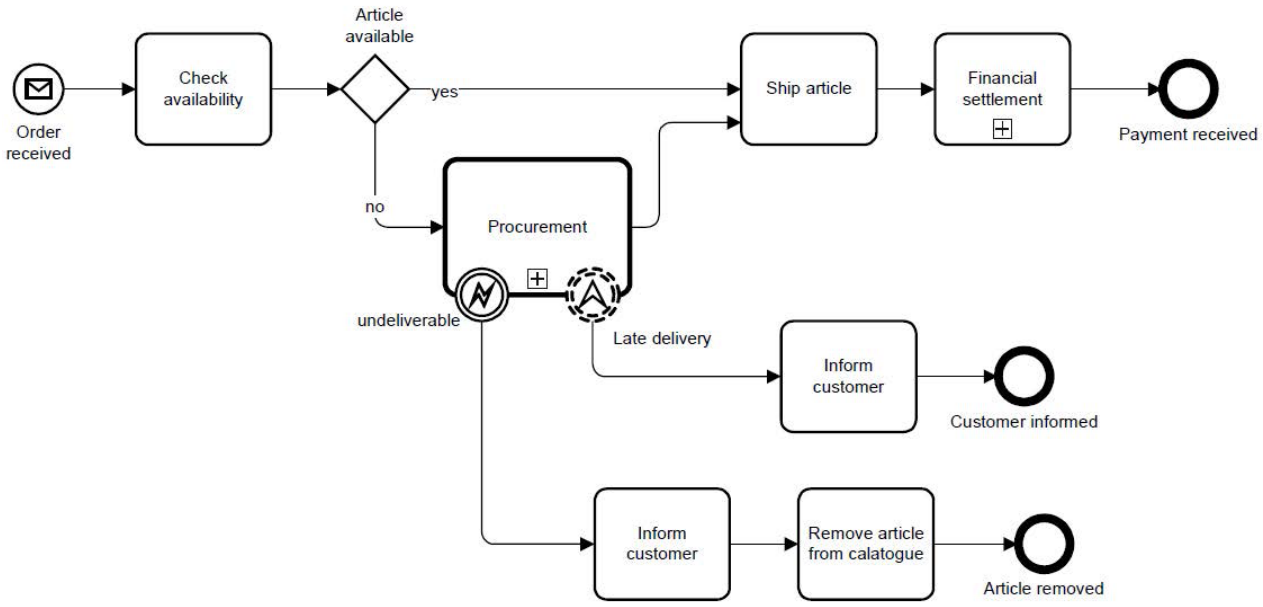

Figure 12: Order Fulfilment and Procurement Example

Best Practice Recommendations

Recommendations for Event Naming

Events should be named using a <Qualifier> <Data Object> <Verb> approach. The Verb is usually derived from the preceding or following action to be undertaken, taking the past tense or future tense of the Verb used. (Note the OMG Examples used do not follow this recommended approach)

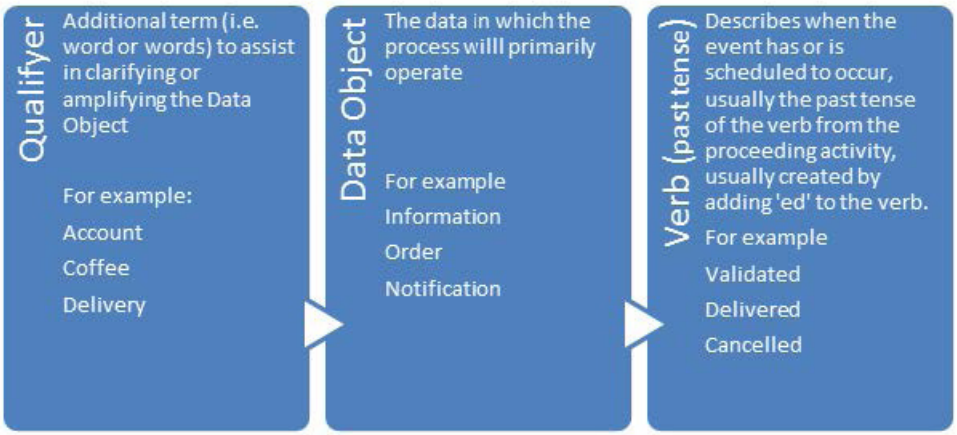

Figure 13: Recommended Event Naming Approach

It is important to avoid ambiguity so your readers can understand the Event and its trigger.

Recommendations for Start Events

The most commonly used trigger types for Start Events are the None, Message and Timer. However, there are also occasions when the Error, Escalation and Conditional trigger are not only useful but appropriate.

Recommendations for Intermediate Events

The None and Message and Timer triggers are again the most commonly used triggers when documenting processes, although there are occasions when the Escalation, Compensation, Conditional and Error can be useful. The Link type of an Intermediate Event is a special form of Intermediate Event that allows linking between pages.

Recommendations for Boundary Intermediate Events

Knowing when to use Boundary Intermediate Events is difficult. The common uses for Boundary intermediate Events are to handle Error and Escalation triggers within Activities, i.e. Sub-Processes or Tasks.

It is more likely that Boundary Intermediate Events will be used on Sub-Processes than Tasks, as they are used as a means of catching specific event triggers during the execution of a Sub-Process.

Recommendations for End Events

None and Message triggers are the commonly used triggers when documenting processes, although there are occasions when the Escalation, Conditional, Error and Terminate can be useful.

Conclusion

Events are an important part of process Modeling and provide the means of explaining to the reader what has happened (or is about to happen) at the appropriate points within and across process. It is because of BPMN 2.0 and the expansion of Events that processes are now able to better reflect the real world and processes are now models as opposed to just drawings. It means a model can accurately capture when something needs to wait or what needs to happen when there is an exception.

It is because of this that BPMN 2.0 has also become the notation of choice for Business Process Management Systems (BPMS), as process execution needs to be able to handle all possible triggering Events.

The suggested naming conventions will help you avoid ambiguity when Modeling and assist in keeping your Modeling repository tidy. Having the ability to validate your BPMN 2.0 model, such as that provided in Orbus Software's new BPA Solution will also assist you.

Remember too that just because there are many Event types available one does not need to use them all when Modeling. The recommendations have set out the most commonly used and these should suffice for most circumstances.

Orbus Software
3rd Floor
111 Buckingham Palace Road
London
SW1W 0SR
United Kingdom

+44 (0) 870 991 1851
enquiries@orbussoftware.com
www.orbussoftware.com

orbus software