

Business Process Management

Using BPMN V2.0

By David Jones and Roderick Brown

Table of Contents

| | |
|---|---|
| Introduction..... | 1 |
| Business Process Model and Notation V2.0..... | 2 |
| Background | 2 |
| Types of BPMN Models..... | 2 |
| Understanding Swim Lanes | 3 |
| Overview | 3 |
| Pools..... | 3 |
| Lanes | 3 |
| Understanding Events | 4 |
| Overview | 4 |
| Start Events | 4 |
| Intermediate Events..... | 4 |
| Intermediate Boundary Events Triggers | 5 |
| End Triggers..... | 5 |
| Overview | 5 |
| Tasks..... | 6 |
| Sub-Processes | 6 |
| Additional Forms of Activity..... | 6 |

| | |
|--|---|
| Table 11: Types of Data | 8 |
| Table 12: Types of Artefacts | 8 |
| Table 13: Types of Connecting Objects..... | 8 |

List of Figures

| | |
|--|---|
| Figure 1: BPM Lifecycle..... | 2 |
| Figure 2: Descriptive Process Model sub-set of Core Element..... | 2 |
| Figure 3: Analytical Process Model sub-set of Core Element..... | 2 |
| Figure 4: Example Collaboration diagram..... | 3 |
| Figure 5: Example Process Diagram..... | 3 |
| Figure 6: Example Choreography diagram | 3 |
| Figure 7: Example Conversation Diagram..... | 3 |
| Figure 8: BPMN Symbol for a Pool..... | 3 |
| Figure 9: BPMN Symbol for Lanes and Sub-Lanes | 3 |
| Figure 10: Example Collaboration Diagram | 9 |
| Figure 11: Example Process Diagram..... | 9 |

Introduction

The importance of documenting Business Processes has been covered in many Management articles and texts.

Documenting business processes has been something organizations have commonly undertaken for more than 30 years. In fact many productivity and quality movements had their foundation before or during World War II. To illustrate the importance of documenting your business processes, virtually all Process Improvement approaches, such as Six Sigma and Lean, start with documenting the business processes being studied.

However, it is only more recently, such as over the last 10 to 15 years, that we have seen the new discipline of Business Process Management.

The well-known Business Process Management (BPM) related website, [BPTrends](#) defines BPM as referring:

“...to aligning processes with the organization's

strategic goals, designing and implementing process architectures, establishing process measurement systems that align with organizational goals, and educating and organizing managers so that they will manage processes effectively.”

Wikipedia defines to Business Process Management as referring:

“...to as a "holistic management" approach to aligning an organization's business processes with the wants and needs of clients. It promotes business effectiveness and efficiency while striving for innovation, flexibility, and integration with technology. BPM attempts to improve processes continuously.”

Business Process Management is based around lifecycle, which is usually defined as:

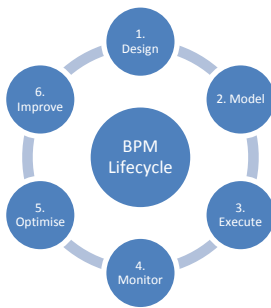


Figure 1: BPM Lifecycle

Although this lifecycle does vary slightly between different authors, for example some combine Design and Model and / or Optimize and Improve, the basic components of the life cycle are relatively consistent. Consequently, we will be concentrating on the Design and Modeling of Business Processes.

Business Process Model and Notation V2.0

Background

The lack of common diagramming approaches and standards were a barrier to developing Business Process Modeling as a more professional discipline. This need for a common process diagramming standard in late 1990's was a key driver in a number of different initiatives beginning in the early 2000's. BPMN V1.0 was first published in 2004 by the Business Process Management Initiative (BPMI). Shortly after, in 2005, the BPMI merged with a larger group, the Object Management Group (OMG). Since 2005, OMG has continued to develop BPMN, with the latest finalized version of BPMN Version 2.0 and was published in January 2012.

Types of BPMN Models

The BPMN 2.0 Specification defines three “sub-classes” (or types) of Process Models:

1. Descriptive Process Models

Descriptive models document Business Processes at an overview or high-level (i.e. minimal details) and use a limited sub-set of BPMN’s visible elements and attributes as shown below.

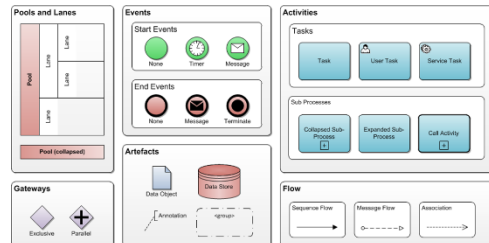


Figure 2: Descriptive Process Model sub-set of Core Element

2. Analytical Process Models

Analytical models provide more detail than the Descriptive type with an expanded sub-set of BPMN’s visible elements and attributes, about half the complete set.

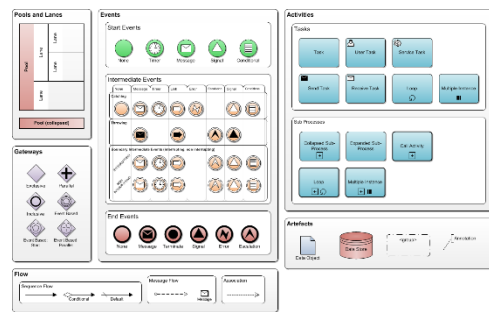


Figure 3: Analytical Process Model sub-set of Core Element

3. Common Executable.

Common Executable models focus on the detail needed to explain how Business Processes can be executed and use the complete set of BPMN’s visible Elements and attributes.



TIP:

Descriptive Process Models are typically used for the process models requiring minimal levels of detail that do not need significant detail, whereas Analytical Process Models are normally used to document the more detailed process models.

The BPMN 2.0 Specification also defines that Process Models must, allow Business Processes to be documented in the form of one of 4 different types of Process diagrams:

1. Collaboration diagrams

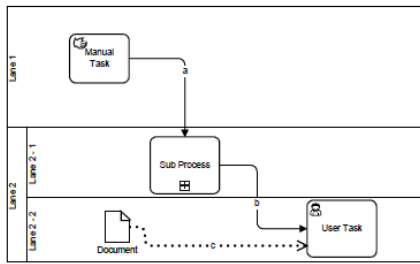


Figure 4: Example Collaboration diagram

2. Process diagrams;

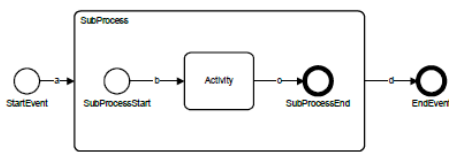


Figure 5: Example Process Diagram

3. Choreography diagrams; and

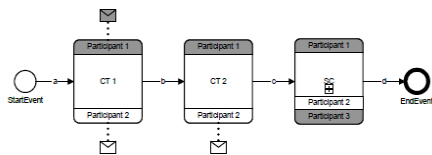


Figure 6: Example Choreography diagram

4. Conversation diagrams.

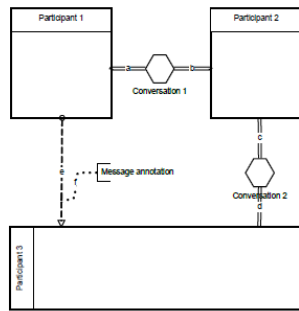


Figure 7: Example Conversation Diagram

Only the Collaboration and Process diagrams are routinely used.



TIP:

Collaboration Models are frequently defined using the Descriptive Process Modeling set of BPMN’s visible elements. Similarly Process diagrams are usually defined using the Analytically Process Modeling set of BPMN’s visible elements.

Understanding Swim Lanes

Overview

Swim Lanes are used to represent the various Parties and Roles involved in a Process. They are represented on the Collaboration and Process Diagrams by Pools and Lanes.

Pools

A Pool represents a Party involved in a process.

The Pool is often used to represent a Party external to the organization being modeled or alternatively different areas within the organization.



Figure 8: BPMN Symbol for a Pool

Under some circumstances, a Pool can also be used to show a Process. However, this should be seen as the exception rather than the norm.

Pools can be shown as either a “White Box” or “Black Box”:

A “White Box” Pool contains the Process details; whereas

A “Black Box Pool” does not.



TIP:

Use Black Box Pools to represent External Parties, Organization Units outside the scope of the Process or an IT Application or system.
Use White Box Pools to represent Organization Units or possibly a Process.

Lanes

A Lane represents a position or role position within the Party defined by the Pool in which it contains.

Lanes can also be further divided into sub-Lanes.

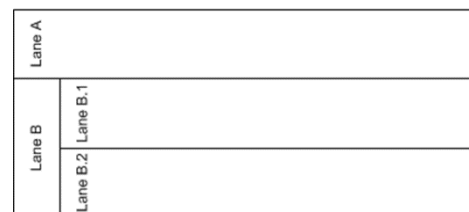


Figure 9: BPMN Symbol for Lanes and Sub-Lanes



TIP:

Use Lanes to represent smaller Organization Units, such as a Team, Positions or Roles.




Understanding Events

Overview

Events “happen” during the course of a Process and affect the flow. Events have a cause (trigger) or an impact (result).

There are three basic types of events:








Table 1: Type of Events

| Event Type | Description | BPMN Symbol |
|---------------------|---|---|
| Start Events | Occur at the start of a process and cause the initiation of a process. |  Start Event |
| Intermediate Events | Occur anywhere between the start and end events within the process. |  Intermediate Event |
| End Events | Occur at the end of a process and describe the results that are leading to the exit of the process. |  End Event |

Start Events

The typical Start Events used include:

Table 2: Start Event Triggers

| Trigger Type | Trigger cause | BPMN Symbol |
|--------------|---|---|
| None | Has no specific defined trigger. |  Start Event (None) |
| Message | Arrival of a message from another process or participant. |  Start Event (Message) |
| Timer | Arrival of a specific date (and time) or after a cycle, e.g. 9am on Monday or 1st of the Month. |  Start Event (Timer) |
| Conditional | When a specified condition is reached, e.g. credit limit is exceeded. |  Conditional Start Event |
| Signal | Arrival of a signal broadcast by another activity or process. |  Signal Start Event |
| Escalation | Need for escalation is identified. |  Start Event (Escalation) |
| Error | Error state has been identified. |  Start Event (Error) |



TIP:

The key types of Start Events commonly used are the None, Message, Timer and Error. However, Conditional and Escalation Start Events are also very useful.

Use of the Signal Start Event causes a lot of confusion with novices, so should be used with care.

Intermediate Events















Intermediate Events trigger types sometimes occur in two forms:

Throwing – when Intermediate Event causes something to happen; and

Catching – when Intermediate Event is listening and waiting for something to happen.

Typical Intermediate Events used include:

Table 3: Intermediate Event Triggers

| Trigger Type | Trigger cause | BPMN Symbol |
|--------------|--|---|
| None | Has no specific defined trigger. |  Intermediate Event (None) |
| Message | Arrival of a message (Catch) or sending a message (Throw). |   Intermediate Event (Message - Catching) Intermediate Event (Message - Throwing) |
| Timer | Wait for a specific date (and time), cycle or delay, e.g. Monday at 9am, End of the Month or 12 hours. |  Intermediate Event (Timer) |
| Compensation | When the need for compensation is identified. |   Intermediate Event (Compensation - Catch) Intermediate Event (Compensation - Throw) |
| Conditional | When a specified condition is reached, e.g. credit limit is exceeded. |  Intermediate Event (Conditional) |
| Signal | Arrival of a signal broadcast by another activity or sending of a signal broadcast by another activity. |   Intermediate Event (Signal - Catch) Intermediate Event (Signal - Throw) |
| Escalation | Need for escalation is identified. |   Intermediate Event (Escalation - Catch) Intermediate Event (Escalation - Throw) |
| Link | Connect two parts of the process, usually across page boundaries or to remove the need for crossed lines |   Intermediate Event (Link - Catch) Intermediate Event (Link - Throw) |
| Error | Error state has been identified. |  Intermediate Event (Error) |



TIP:

The key types of Intermediate Events commonly used are the None, Message, Timer and Error. However, Conditional, Compensation and Escalation Intermediate Events are also very useful. The Link Intermediate Event is used for on and off page connectors.

Use of the Signal Intermediate Event causes a lot of confusion with novices, so should be used with care.



TIP:

The key types of Intermediate Boundary Events commonly used are Message, Timer and Error. However, Conditional and Escalation Start Events are also very useful.

Use of the Signal Start Event causes a lot of confusion with novices, so should be used with care.

Intermediate Boundary Events Triggers

Intermediate Boundary Events are a special form of Intermediate Events, which are triggered during the execution of an Activity and are defined in two forms:

Interrupting Boundary Events – which interrupts the execution of the Activity when the trigger occurs; and Non-interrupting Boundary Events – which does not interrupt the execution of the Activity when the trigger occurs.

The typical Boundary Intermediate Events used include:

Table 4: Intermediate Boundary Event Triggers

| Trigger Type | Interrupting Boundary Events | Non-Interrupting Boundary Events |
|---|------------------------------|----------------------------------|
| Message Intermediate Boundary Event | | |
| Timer Intermediate Boundary Event | | |
| Conditional Intermediate Boundary Event | | |
| Error Intermediate Boundary Event | | |
| Escalation Intermediate Boundary Event | | |
| Signal Intermediate Boundary Event | | |

End Triggers

The typical End Events used include:

Table 5: End Event Triggers

| Trigger Type | Trigger cause | BPMN Symbol |
|--------------|---|-------------|
| None | Has no specific defined trigger. | |
| Message | Sending of a message to another process or participant. | |
| Compensation | When the need for compensation is identified. | |
| Signal | Sending of a signal broadcast by another activity. | |
| Escalation | Need for escalation is identified. | |
| Error | Error state has been identified. | |
| Terminate | Indicates that all Activities in the Process should be immediately ended. | |



TIP:

The key types of End Events commonly used are the None, Message, Timer and Error. However, Conditional, Escalation and Terminate End Events are also very useful.

Use of the Signal End Event causes a lot of confusion with novices, so should be used with care.

Understanding Activities

Overview

Activities are used to represent work undertaken within a Business Process. They are used to represent

the points in a Process flow where work is actually performed.

An Activity is represented, within a Process, as either:



An Atomic Activity – called a Task, is the simplest level form of Activity, i.e. cannot be broken down further; or

A non-Atomic Activity – called a Sub-Process, is the compound level form of Activity, i.e. able to be broken down further.

Tasks

Tasks are defined in one of two forms:

Table 6: Types of Tasks

| Type of Task | Description | BPMN Symbol |
|----------------------|--|---|
| Task (Standard form) | A Task shares the same basic outline shape in as the Sub-Process, which is a rectangle with rounded corners with a single thin line. |  |
| Call Activity Task | A Task used in its Call Activity form, i.e. a re-usable Task used by many Processes, the thin line is changed to a thick line |  |

A Task represents the work performed and generally is executed by:


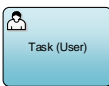
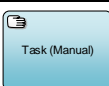

An end-user, who may optionally use an application; or



An application.

Tasks can also be divided into a number of sub-types, which identify specific details of how the Task is performed or who the Task is performed by.

For example, the following sub-types of Tasks are defined:

Table 7: Sub-Types of Tasks

| Sub-Type | Description | BPMN Symbol |
|----------|---|---|
| Service | A Service Task is where the Task is performed by a service, i.e. some form of automation, e.g. a Web service or an automated application |  |
| User | A User Task is a “workflow” Task performed by a person with the assistance of an application and is scheduled through a task list manager |  |
| Manual | A Manual Task performed by a person without the aid of an application, such as technician installing a telephone. |  |
| Send | A Send Task is designed to send a Message to an external Participant of the process |  |

| | | |
|---------------|--|---|
| Receive | A Receive Task is designed to receive a Message from an external Participant |  |
| Business Rule | A Business Rule Task is designed to evaluate a business rule or to integrate with a Business Rule engine |  |



TIP:

The key types of Tasks used include, Service, User, Manual and Business Rule.

However, there are good reasons why you may want to also use the Send and Receive Tasks.





Although not commonly used, the Call Activity form of Task can be used for re-usable Tasks.

Sub-Processes

A Sub-Process is an Activity whose internal details have been modeled using Activities, Gateways, Events, and Sequence Flows in another Process Model.

Sub Processes are defined in one of two sub-types and also occur in one of two forms:

Table 8: Types of Sub-Process Activities

| Sub-Process Type | Description | BPMN Symbol | |
|------------------|--|---|---|
| | | Standard Form | Call Activity Form |
| Expanded | Details of the Sub-Process are embedded inside the Sub-Process symbol contained on the Process Model |  |  |
| Collapsed | Details of the Sub-Process are shown on a separate defined Process Model |  |  |



TIP:

Both the Expanded and Collapsed types of Sub-Processes are commonly used. The Collapsed type of Sub-Process is most commonly used when the detail needs to be defined separately.

Although less commonly used, the Call Activity form of Sub-Processes is used when a Sub-Process is going to be used in many places.

Additional Forms of Activity

In addition to the sub-types of Activity, there are additional markers that can be added to an Activity. The two Activity markers most commonly used are:

Table 9: Additional Activity Markers

| Activity Marker | Description | BPMN Symbol |
|-----------------|--|-------------|
| Loop | The Loop marker is used to indicate that an Activity is repeated sequentially until a defined condition is achieved. | |
| Multi-Instance | The Multi-Instance marker is used to show that (defined) instances of an Activity are created and executed either in Parallel or Sequentially. | |

TIP: Loop and Multi-Instance Activities markers are both used. Loops are typically used where the number of iterations is not known in advance, but the looping is undertaken until a known condition is met, whereas Multi-Instance Activities require the number of instances to be known in advance.

Understanding Gateways

Gateways are used to control how Sequence Flows interact as they converge and diverge within a Process.

Gateways do not evaluate expressions or events. Rather, they are simply the routing mechanism. Evaluation of any expression, for which a Gateway undertake routing, must be undertaken in the previous Activity.

Typical types of Gateways used include:

Table 10: Types of Gateways

| Type of Gateway | Description | BPMN Symbol |
|-------------------|---|-------------|
| Exclusive Gateway | Represent a branching point in the process that requires one and only one of the possible paths being chosen from the available options based on the evaluation of an Expression. | |

| | | |
|---------------------|--|--|
| Inclusive Gateway | Represent a branching point in the process that allows one or more (i.e. multiple) of the possible paths being chosen from the available options based on the evaluation of an Expression. | |
| Parallel Gateway | Represent a branching point in the process where all of the paths in the process must be followed. | |
| Event Based Gateway | Represents a branching point in the Process where the alternative paths that follow the Gateway are based on Events (defined after the Event Based Gateway) that can occur, rather than the evaluation of an Expression. | |
| Complex Gateway | Provide the ability to support complex merging and branching behaviour that are not captured by other Gateways. | |

TIP: Although the most commonly used type of Gateway is the Exclusive Gateway, each type 7 can be very useful in their own right. For example:

- Inclusive Events allow multiple paths from the Gateway to be executed because the appropriate condition of each path is true;
- Parallel Gateways provide a means of executing multiple paths in a process in parallel; and
- Event Based Gateways are used when a specific event is expected, such as receipt of a message, however another event type, such as a Timer, provides the exit point if no message is received.

Understanding Data

Data objects in BPMN provide the mechanism to show how data or information is created or consumed by a process.

There are three types of objects use to define Data:

Table 11: Types of Data

| Type of Data | Description | BPMN Symbol |
|--------------|--|--|
| Data Object | An item or collection of data that is either used by or created by an Activity that only exists for the current life of the Process. | <p>Data Object (Single)</p> <p>Data Object (Input - Single)</p> <p>Data Object (Output - Single)</p> <p>Data Object (Collection)</p> |
| Data Store | A collection of data that is either used or created by an Activity that exists beyond the current life of the Process. | <p>Data Store</p> |
| Messages | Content of a communication that flows between two participants. | <p>Message</p> |

TIP:

The commonly used forms of Data Objects are single and collection forms of Data Object and are used to show the inputs and outputs of an Activity. Data Stores are used to show that data is being stored beyond the life of the Process. Messages are used to allow the content of a message to be defined.

Understanding Artefacts

There are two different types of Artefact objects available:

Table 12: Types of Artefacts

| Type of Artefacts | Description | BPMN Symbol |
|-------------------|--|-------------------|
| Group Object | Allows flow objects to be grouped for purposes of documentation or analysis. | <p>Group</p> |
| Annotation | Provides additional information to elaborate an element within the process. | <p>Annotation</p> |

TIP:

Although using Group and Annotation Artefacts are very useful when adding documentation to a Process, take care not to add too many Artefacts and making the Process Diagram difficult to read.

Understanding Connecting Objects

There are five different types of Connecting objects available:

Table 13: Types of Connecting Objects

| Type of Artefacts | Description | BPMN Symbol |
|------------------------------|---|----------------------------------|
| Sequence Flows (normal form) | Shows the execution path (or order) of Process by connecting Flow Objects and is the normal and hence typical means of connecting two Flow Objects. | <p>Sequence Flow</p> |
| Default Sequence Flow | Defines a default path following an Exclusive, Inclusive or Complex Gateway. | <p>Default Sequence Flow</p> |
| Conditional Sequence Flow | Provides a pathway from an Activity that is only followed where a defined condition is true. | <p>Conditional Sequence Flow</p> |
| Message Flows | Shows the flow of information in the form of Messages between Participants. | <p>Message Flow</p> |
| Associations | Are used to attach Data objects or Artefact objects to Flow Objects, typically Activities. | <p>Association</p> |

TIP:

Obviously the most common type of Connecting Object used is the (normal) Sequence Flow. However, Message Flows and Associations are also commonly used for their required purposes. The Default and Conditional types of Sequence Flows are also useful in certain circumstances, but should not be over used. Rather their role should be used to highlight specific reasons for use. For example:

- Use Conditional Sequence Flow to illustrate where one path is always followed, but another may also be followed if a certain condition is true; and
- Use Default Sequence Flow in the situation where it is possible not to have a path with a valid condition, so the default path becomes the path followed in this situation.

Case Study

The best way to understand how to apply BPMN 2.0 for your own needs is to look at some examples.

The following Models are taken from our Orbus Software Publication, BPMN by Example: An Introduction to the Business Process Modelling Notation v2.0:

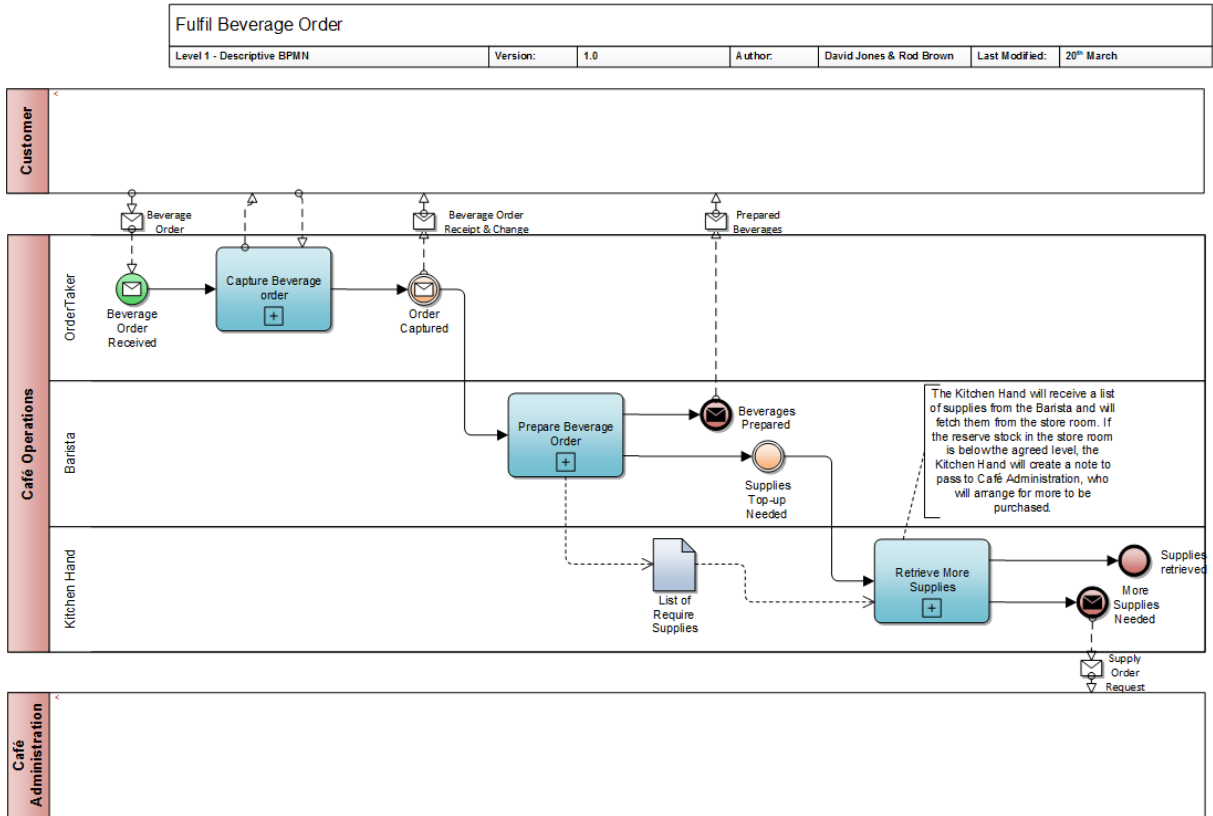


Figure 10: Example Collaboration Diagram

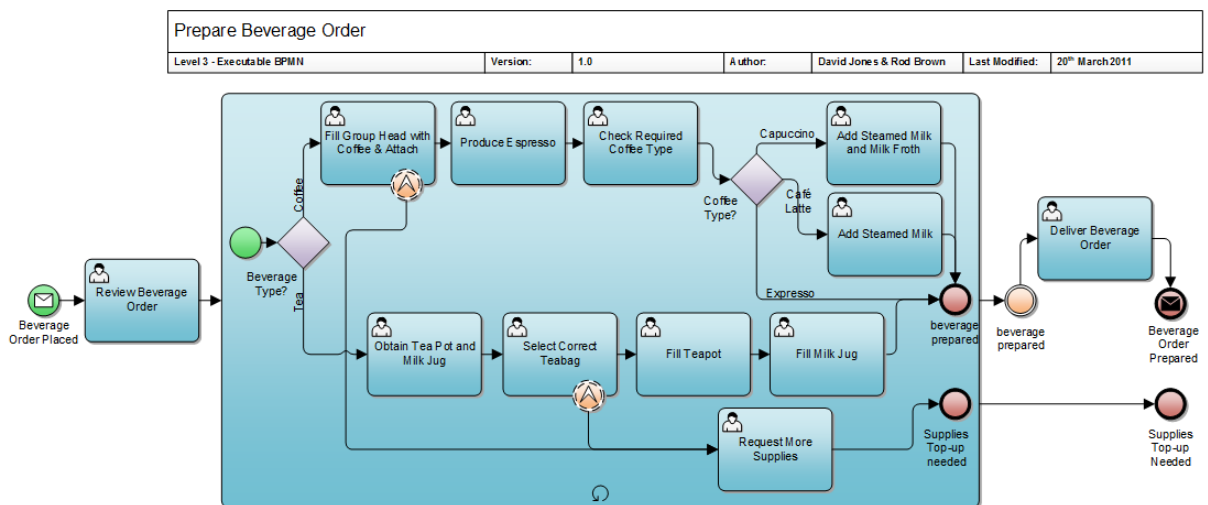


Figure 11: Example Process Diagram

© Copyright 2013 Orbus Software. All rights reserved.

No part of this publication may be reproduced, resold, stored in a retrieval system, or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

Such requests for permission or any other comments relating to the material contained in this document may be submitted to:

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

or by electronic mail to: marketing@orbussoftware.com