

3. Business Process Diagrams

This section provides a summary of the BPMN graphical objects and their relationships. More details on the concepts will be provided in “Business Process Diagram Graphical Objects” on page 49 and “Connecting Objects” on page 119.

A goal for the development of BPMN is that the notation be simple and adoptable by business analysts. Also, there is a potentially conflicting requirement that BPMN provide the power to depict complex business processes and map to BPM execution languages. To help understand how BPMN can manage both requirements, the list of BPMN graphic elements is presented in two groups.

First, there is the list of core elements that will support the requirement of a simple notation. These are the elements that define the basic look-and-feel of BPMN. Most business processes will be modeled adequately with these elements. Second, there is the entire list of elements, including the core elements, which will help support requirement of a powerful notation to handle more advanced modeling situations. And further, the graphical elements of the notation will be supported by non-graphical attributes that will provide the remaining information necessary to map to an execution language or other business modeling purposes.

3.1 BPD Core Element Set

It should be emphasized that one of the drivers for the development of BPMN is to create a simple mechanism for creating business process models, while at the same time being able to handle the complexity inherent to business processes. The approach taken to do handle these two conflicting requirements was to organize the graphical aspects of the notation into specific categories. This provides a small set of notation categories so that the reader of a BPMN diagram can easily recognize the basic types of elements and understand the diagram. Within the basic categories of elements, additional variation and information can be added to support the requirements for complexity without dramatically changing the basic look and feel of the diagram. The four basic categories of elements are:

- Flow Objects
- Connecting Objects
- Swimlanes
- Artifacts

Flow objects are the main graphical elements to define the behavior of a Business Process. There are three Flow Objects:

- Events
- Activities
- Gateways

There are three ways of connecting the flow objects to each other or other information. There are three connecting objects:

- Sequence Flow
- Message Flow
- Association



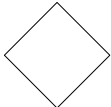
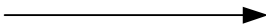
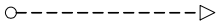
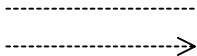

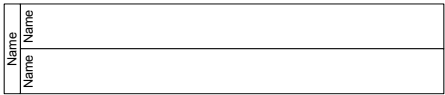
There are two ways of grouping the primary modeling elements through “Swimlanes:”

- Pools
- Lanes

Artifacts will provide additional information about the Process. There are four standardized Artifacts, but modelers or modeling tools are free to add as many Artifacts as required. There may be additional BPMN efforts to standardize a larger set of Artifacts for general use or for vertical markets. The current set of Artifacts include:

- Data Object
- Database
- Group
- Annotation

Table 1 displays a list of the core modeling elements that are depicted by the notation:

Element	Description	Notation
Event	An event is something that “happens” during the course of a business process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result). Events are circles with open centers to allow internal markers to differentiate different triggers or results. There are three types of Events, based on when they affect the flow: Start, Intermediate, and End.	
Activity	An activity is a generic term for work that company performs. An activity can be atomic or non-atomic (compound). The types of activities that are a part of a Process Model are: Process, Sub-Process, and Task. Tasks and Sub-Processes are rounded rectangles. Processes are either unbounded or a contained within a Pool.	
Gateway	A Gateway is used to control the divergence and convergence of Sequence Flow. Thus, it will determine branching, forking, merging, and joining of paths. Internal Markers will indicate the type of behavior control.	
Sequence Flow	A Sequence Flow is used to show the order that activities will be performed in a Process.	
Message Flow	A Message Flow is used to show the flow of messages between two entities that are prepared to send and receive them. In BPMN, two separate Pools in the Diagram will represent the two entities (participants).	
Association	An Association is used to associate information with flow objects. Text and graphical non-flow objects can be associated with the flow objects.	
Pool	A Pool is a “swimlane” and a graphical container for partitioning a set of activities from other Pools, usually in the context of B2B situations.	
Lane	A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize activities.	



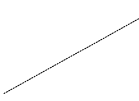
































































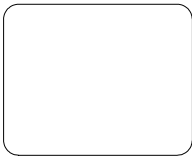
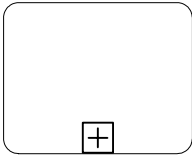
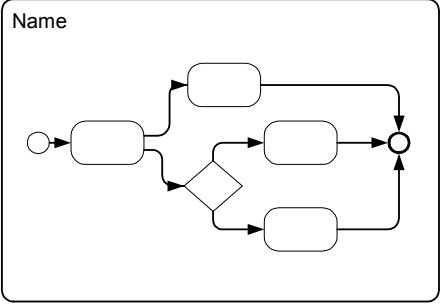
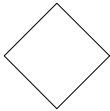






<p>Data Object</p>	<p>Data Objects are considered artifacts because they do not have any direct affect on the Sequence Flow or Message Flow of the Process, but they do provide information about what the Process does.</p>	 <p>Name</p>
<p>Group (a box around a group of objects for documentation purposes)</p>	<p>A grouping of activities that does not affect the Sequence Flow. The grouping can be used for documentation or analysis purposes. Groups can also be used to identify the activities of a distributed transaction that is shown across Pools.</p>	
<p>Text Annotation (attached with an Association)</p>	<p>Text Annotations are a mechanism for a modeler to provide additional information for the reader of a BPMN Diagram.</p>	 <p>Descriptive Text Here</p>

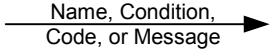
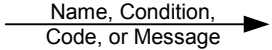
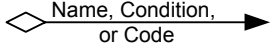
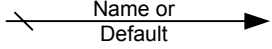
Table 1 BPD Core Element Set

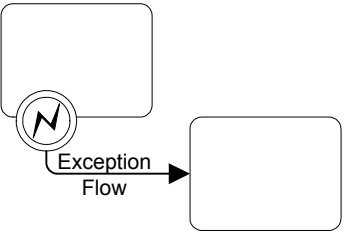
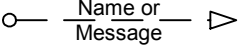
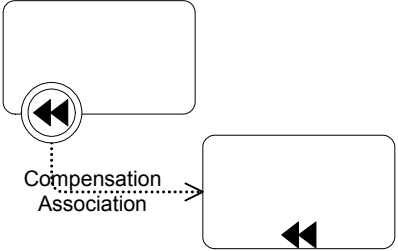
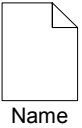
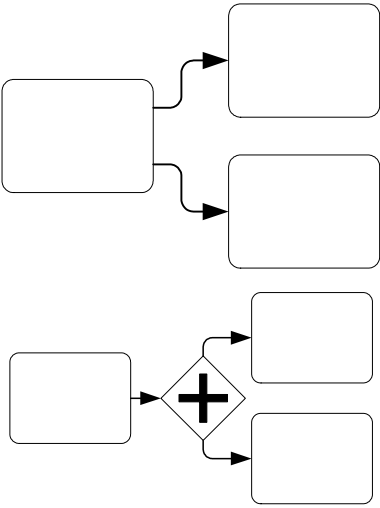
3.2 BPD Complete Set

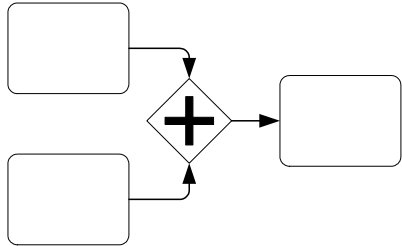
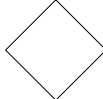


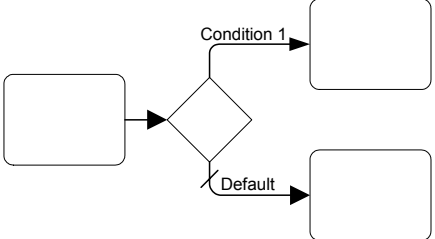
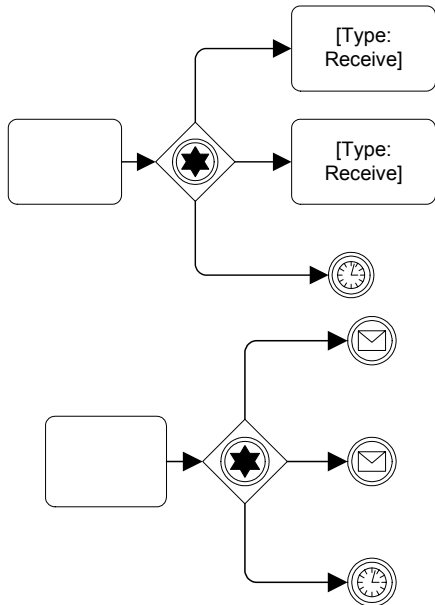
Table 2 displays a more extensive list of the business process concepts that could be depicted through a business process modeling notation.

Element	Description	Notation																																				
Event	<p>An event is something that “happens” during the course of a business process. These events affect the flow of the process and usually have a cause (trigger) or an impact (result). There are three types of Events, based on when they affect the flow: Start, Intermediate, and End.</p>	 Name or Source																																				
Flow Dimension (e.g., Start, Intermediate, End) Start (None, Message, Timer, Rule, Link, Multiple) Intermediate (None, Message, Timer, Exception, Cancel, Compensation, Rule, Link, Multiple, Branching) End (None, Message, Exception, Cancel, Compensation, Link, Terminate, Multiple)	<p>As the name implies, the Start Event indicates where a particular process will start.</p> <p>Intermediate Events occur between a Start Event and an End Event. It will affect the flow of the process, but will not start or (directly) terminate the process.</p> <p>As the name implies, the End Event indicates where a process will end.</p>	<p>Start </p> <p>Intermediate </p> <p>End </p>																																				
Type Dimension (e.g., Message, Timer, Exception, Cancel, Compensation, Rule, Link, Multiple, Terminate)	<p>Start and Intermediate Events have “Triggers” that define the cause for the event. There are multiple ways that these events can be triggered. End Events may define a “Result” that is a consequence of a Sequence Flow ending.</p>	<table border="0"> <tr> <td>Message</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Timer</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Exception</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Cancel</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Compensation</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Rule</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Link</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Multiple</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Terminate</td> <td></td> <td></td> <td></td> </tr> </table>	Message				Timer				Exception				Cancel				Compensation				Rule				Link				Multiple				Terminate			
Message																																						
Timer																																						
Exception																																						
Cancel																																						
Compensation																																						
Rule																																						
Link																																						
Multiple																																						
Terminate																																						



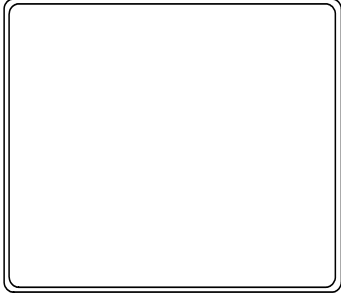

<p>Task (Atomic)</p>	<p>A Task is an atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail.</p>	
<p>Process/Sub-Process (non-atomic)</p>	<p>A Sub-Process is a compound activity that is included within a Process. It is compound in that it can be broken down into a finer level of detail (a Process) through a set of sub-activities.</p>	<p>See Next Two Figures</p>
<p>Collapsed Sub-Process</p>	<p>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.</p>	
<p>Expanded Sub-Process</p>	<p>The boundary of the Sub-Process is expanded and the details (a Process) are visible within its boundary. Note that Sequence Flow cannot cross the boundary of a Sub-Process.</p>	<p>Name</p> 
<p>Gateway</p>	<p>A Gateway is used to control the divergence and convergence of multiple Sequence Flow. Thus, it will determine branching, forking, merging, and joining of paths.</p>	
<p>Gateway Control Types</p>	<p>Icons within the diamond shape will indicate the type of flow control behavior. The types of control include:</p> <ul style="list-style-type: none"> • XOR -- exclusive decision and merging. Both Data-Based and Event-Based. Data-Based can be shown with or without the “X” marker. • OR -- inclusive decision • Complex -- complex conditions and situations (e.g., 3 out of 5) • AND -- forking and joining <p>Each type of control affects both the incoming and outgoing Flow.</p>	<p>Exclusive (XOR) Data-Based  or  Event-Based  Inclusive (OR)  Complex  Parallel (AND) </p>

Sequence Flow	A Sequence Flow is used to show the order that activities will be performed in a Process.	See next seven figures
Normal flow	Normal Sequence Flow refers to the flow that originates from a Start Event and continues through activities via alternative and parallel paths until it ends at an End Event.	
Uncontrolled flow	Uncontrolled flow refers to flow that is not affected by any conditions or does not pass through a Gateway. The simplest example of this is a single Sequence Flow connecting two activities. This can also apply to multiple Sequence Flow that converge on or diverge from an activity. For each uncontrolled Sequence Flow a "Token" will flow from the source object to the target object.	
Conditional flow	Sequence Flow can have condition expressions that are evaluated at runtime to determine whether or not the flow will be used. If the conditional flow is outgoing from an activity, then the Sequence Flow will have a mini-diamond at the beginning of the line (see figure to the right). If the conditional flow is outgoing from a Gateway, then the line will not have a mini-diamond (see figure in the row above).	
Default flow	For Data-Based Exclusive Decisions, one type of flow is the Default condition flow. This flow will be used only if all the other outgoing conditional flow is not true at runtime. These Sequence Flow will have a diagonal slash will be added to the beginning of the line (see the figure to the right). Note that it is an Open Issue whether Default Conditions will be used for Inclusive Decision situations.	

<p>Exception flow</p>	<p>Exception flow occurs outside the normal flow of the Process and is based upon an Intermediate Event that occurs during the performance of the Process.</p>	
<p>Message Flow</p>	<p>A Message Flow is used to show the flow of messages between two entities that are prepared to send and receive them. In BPMN, two separate Pools in the Diagram will represent the two entities.</p>	
<p>Compensation Association</p>	<p>Compensation Association occurs outside the normal flow of the Process and is based upon an event (a Cancel Intermediate Event) that is triggered through the failure of a Transaction or a Compensate Event. The target of the Association must be marked as a Compensation Activity.</p>	
<p>Data Object</p>	<p>Data Objects are considered artifacts because they do not have any direct affect on the Sequence Flow or Message Flow of the Process, but they do provide information about what the Process does.</p>	
<p>Fork (AND-Split)</p>	<p>BPMN uses the term “fork” to refer to the dividing of a path into two or more parallel paths (also known as an AND-Split). It is a place in the Process where activities can be performed concurrently, rather than sequentially. There are two options: Multiple Outgoing Sequence Flow can be used (see figure top-right). This represents “uncontrolled” flow is the preferred method for most situations. A Parallel (AND) Gateway can be used (see figure bottom-right). This will be used rarely, usually in combination with other Gateways.</p>	

<p>Join (AND-Join)</p>	<p>BPMN uses the term “join” to refer to the combining of two or more parallel paths into one path (also known as an AND-Join or synchronization). A Parallel (AND) Gateway is used to show the joining of multiple flows.</p>	
<p>Decision, Branching Point; (OR-Split)</p>	<p>Decisions are Gateways within a business process where the flow of control can take one or more alternative paths.</p>	<p>See next five rows.</p>
<p>Exclusive</p>	<p>An Exclusive Gateway (XOR) restricts the flow such that only one of a set of alternatives may be chosen during runtime. There are two types of Exclusive Gateways: Data-based and Event-based.</p>	<p>Data-Based  or  Event-Based </p>
<p>Data-Based</p>	<p>This Decision represents a branching point where Alternatives are based on conditional expressions contained within the outgoing Sequence Flow. Only one of the Alternatives will be chosen.</p>	
<p>Event-Based</p>	<p>This Decision represents a branching point where Alternatives are based on an Event that occurs at that point in the Process. The specific Event, usually the receipt of a Message, determines which of the paths will be taken. Other types of Events can be used, such as Timer. Only one of the Alternatives will be chosen.</p> <p>There are two options for receiving Messages: Tasks of Type Receive can be used (see figure top-right). Intermediate Events of Type Message can be used (see figure bottom-right).</p>	

<p>Inclusive</p>	<p>This Decision represents a branching point where Alternatives are based on conditional expressions contained within the outgoing Sequence Flow.</p> <p>In some sense it is a grouping of related independent Binary (Yes/No) Decisions. Since each path is independent, all combinations of the paths may be taken, from zero to all. However, it should be designed so that at least one path is taken.</p> <p>There are two versions of this type of Decision:</p> <p>The first uses a collection of conditional Sequence Flow, marked with mini-diamonds (see top-right figure).</p> <p>The second uses an OR Gateway, usually in combination with other Gateways (see bottom-right picture).</p>	
<p>Merging (OR-Join)</p>	<p>BPMN uses the term “merge” to refer to the exclusive combining of two or more paths into one path (also known as an a OR-Join).</p> <p>A Merging (XOR) Gateway is used to show the merging of multiple flows.</p> <p>If all the incoming flow is alternative, then a Gateway is not needed. That is, uncontrolled flow provides the same behavior.</p>	
<p>Looping</p>	<p>BPMN provides 2 (two) mechanisms for looping within a Process.</p>	<p>See Next Two Figures</p>
<p>Activity Looping</p>	<p>The properties of Tasks and Sub-Processes will determine if they are repeated or performed once. There are two types of loops: Standard and Multi-Instance. A small looping indicator will be displayed at the bottom-center of the activity.</p>	
<p>Sequence Flow Looping</p>	<p>Loops can be created by connecting a Sequence Flow to an “upstream” object. An object is considered to be upstream if that object has an outgoing Sequence Flow that leads to a series of other Sequence Flows, the last of which is an incoming Sequence Flow for the original object.</p>	

<p>Multiple Instances</p>	<p>The attributes of Tasks and Sub-Processes will determine if they are repeated or performed once. A small parallel indicator will be displayed at the bottom-center of the activity.</p>	
<p>Process Break (something out of the control of the process makes the process pause)</p>	<p>A Process Break is a location in the Process that shows where an expected delay will occur within a Process. An Intermediate Event is used to show the actual behavior (see top-right figure). In addition, a Process Break artifact, as designed by a modeler or modeling tool, can be associated with the Event to highlight the location of the delay within the flow.</p>	
<p>Transaction</p>	<p>A transaction is a Sub-Process that is supported by special protocol that insures that all parties involved have complete agreement that the activity should be completed or cancelled. The attributes of the activity will determine if the activity is a transaction. A double-lined boundary indicates that the Sub-Process is a Transaction.</p>	
<p>Nested Sub-Process (Inline Block)</p>	<p>A nested Sub-Process is an activity that shares the same set of data as its parent process. This is opposed to a Sub-Process that is independent, reusable, and referenced from the parent process. Data needs to be passed to the referenced Sub-Process, but not to the nested Sub-Process.</p>	<p>There is no special indicator for nested Sub-Processes</p>
<p>Group (a box around a group of objects for documentation purposes)</p>	<p>A grouping of activities that does not affect the Sequence Flow. The grouping can be used for documentation or analysis purposes. Groups can also be used to identify the activities of a distributed transaction that is shown across Pools.</p>	


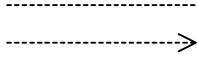


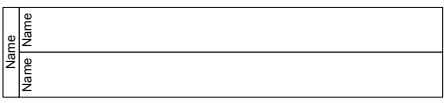
Off-Page Connector	Generally used for printing, this object will show where the Sequence Flow leaves one page and then restarts on the next page. A Link Intermediate Event can be used as an Off-Page Connector.	
Association	An Association is used to associate information with flow objects. Text and graphical non-flow objects can be associated with the flow objects.	
Text Annotation (attached with an Association)	Text Annotations are a mechanism for a modeler to provide additional information for the reader of a BPMN Diagram.	
Pool	A Pool is a “swimlane” and a graphical container for partitioning a set of activities from other Pools, usually in the context of B2B situations.	
Lanes	A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize activities within a Pool.	

Table 2 BPD Complete Element Set

3.3 Use of Text, Color, Size, and Lines in a Diagram

Text Annotation objects can be used by the modeler to display additional information about a Process or attributes of the objects within the Process.

- ❖ Flow objects and Flows MAY have labels (e.g., its name and/or other attributes) placed inside the shape, or above or below the shape, in any direction or location, depending on the preference of the modeler or modeling tool vendor.
- ❖ The fills that are used to for the graphical elements MAY be white or clear.
 - ❖ The notation MAY be extended to use other fill colors to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute).
- ❖ Flow objects and markers MAY be of any size that suits the purposes of the modeler or modeling tool.
- ❖ The lines that are used to draw the graphical elements MAY be black.
 - ❖ The notation MAY be extended to use other line colors to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute).
 - ❖ The notation MAY be extended to use other line styles to suit the purpose of the modeler or tool (e.g., to highlight the value of an object attribute) with the condition that the line style MAY NOT conflict with any current BPMN defined line style. Thus, the line styles of Sequence Flows, Message Flows, and Associations MAY NOT be modified.

3.4 Flow Object Connection Rules

An incoming Sequence Flow can connect to any location on a flow object (left, right, top, or bottom). Likewise, an outgoing Sequence Flow can connect from any location on a flow object (left, right, top, or bottom). Message Flows also have this capability. BPMN allows this flexibility, however, we also recommend that modelers use judgment or best practices in how flow objects should be connected so that readers of the Diagrams will find the behavior clear and easy to follow. This is even more important when a Diagram contains Sequence Flows and Message Flows. In these situations it is best to pick a direction of Sequence Flow, either left to right or top to bottom, and then direct the Message Flow at a 90° angle to the Sequence Flow. The resulting Diagrams will be much easier to understand.

3.4.1 Sequence Flow Rules

Table 3 displays the BPMN flow objects and shows how these objects can connect to one another through Sequence Flows. The ↗ symbol indicates that the object listed in the row can connect to the object listed in the column. The quantity of connections into and out of an object is subject to various configuration dependencies are not specified here. Refer to the sections in the next chapter for each individual object for more detailed information on the appropriate connection rules. *Note that if a sub-process has been expanded within a Diagram, the objects within the sub-process cannot be connected to objects outside of the sub-process. Nor can Sequence Flows cross a Pool boundary.*













From\To						
		↗	↗	↗	↗	↗
		↗	↗	↗	↗	↗
		↗	↗	↗	↗	↗
		↗	↗	↗	↗	↗
		↗	↗	↗	↗	↗
						

Table 3 Sequence Flow Connection Rules

Note: Only those objects that can have incoming and/or outgoing Sequence Flow are shown in the table. Thus, Pool, Lane, Data Object, and Text Annotation are not listed in the table.

3.4.2 Message Flow Rules

Table 4 displays the BPMN modeling objects and shows how these objects can connect to one another through Message Flows. The ↗ symbol indicates that the object listed in the row can connect to the object listed in the column. The quantity of connections into and out of an object is subject to various configuration dependencies are not specified here. Refer to the sections in the next chapter for each individual object for more detailed information on the appropriate connection rules. *Note that Message Flows cannot connect to objects that are within the same Participant Lane boundary.*













From\To						
						
	↗	↗	↗	↗	↗	
	↗	↗	↗	↗	↗	
	↗	↗	↗	↗	↗	
						
	↗	↗	↗	↗	↗	

Table 4 Message Flow Connection Rules

Note: Only those objects that can have incoming and/or outgoing Message Flow are shown in the table. Thus, Lane, Gateway, Data Object, and Text Annotation are not listed in the table.

3.5 Diagram Attributes

The following are attributes of a Business Process Diagram:

Attributes	Description
Id: ObjectId	This is a unique Id that distinguishes the Diagram from other Diagrams.
Name: String	Name is an attribute that is text description of the Diagram.
Version?: String	This defines the Version number of the Diagram.
Author?: String	This holds the name of the author of the Diagram.
Language?: String	This holds the name of the language in which text is written. The default is English.
ExpressionLanguage?: String	A Language MAY be provided so that the syntax of expressions used in the Diagram can be understood.
QueryLanguage?: String	A Language MAY be provided so that the syntax of queries used in the Diagram can be understood.
CreationDate?: Date	This defines the date on which the Diagram was create (for this Version).
ModificationDate?: Date	This defines the date on which the Diagram was last modified (for this Version).
Pool+: PoolId	A BDP SHALL contain one or more Pools. The boundary of one of the Pools MAY be invisible (especially if there is only one Pool in the Diagram).
Documentation?: String	The modeler MAY add optional text documentation about the Diagram.

Table 5 Business Process Diagram Attributes

3.5.1 Changes Since 1.0 Draft Version

These are the changes since the last publically release version:

- The Id attribute, within the set of Business Process Diagram attributes, was change to be of type ObjectId.
- The ModificationDate, ExpressionLanguage, and QueryLanguage attributes were added to the set of Diagram attributes.
- The multiplicity of the Version, Author, CreationDate, and ModificationDate attributes, within the set of Business Process Diagram attributes, was change to 0 to 1.
- The Process attribute was removed from the set of Diagram attributes. The attribute was redundant with the Pool attribute, which will refer to all the Processes in the Diagram.

3.6 Business Process

A **Process** is an activity performed within a company or organization. In BPMN a Process is depicted as a graph of flow objects, which are a set of other activities and the controls that sequence them. The concept of process is intrinsically hierarchical. Processes may be defined at any level from enterprise-wide processes to processes performed by a single person. Low-level processes may be grouped together to achieve a common business goal.

Note that BPMN defines the term Process fairly specifically and defines a Business Process more generically as a set of activities that are performed within an organization or across organizations. Thus a Business Process, as shown in a Business Process Diagram, may

contain more than one separate Process. Each Process may have its own Sub-Processes and would be contained within a Pool (refer to the section entitled “Pool” on page 87). The individual Processes would be independent in terms of Sequence Flow, but could have Message Flow connecting them.

3.6.1 Attributes

The following are attributes of a Process (Note that this is the complete set and it does not extend the set of common object attributes):

Attributes	Description
Id: ObjectID	This is a unique Id that identifies the object from other objects within the Diagram.
Name: String	Name is an attribute that is text description of the object.
ProcessType: (None Executable Abstract Collaboration): None	ProcessType is an attribute that provides information about which lower-level language the Pool will be mapped. By default, the ProcessType is None (or undefined). A Private ProcessType MAY be mapped to an executable BPEL4WS <i>process</i> . An Abstract ProcessType is also called the public interface of a process (or other web services) and MAY be mapped to an abstract BPEL4WS <i>process</i> . A Collaboration ProcessType will have two Lanes that represent business roles (e.g., buyer or seller) and will show the interactions between these roles. These pools MAY be mapped to languages such as ebXML or WS Choreography. However, these mappings are not provided in this version of the specification. If the Process is to be used to create a BPEL4WS document, then the attribute MUST be set to Executable or Abstract.
Status: (None Ready Active Cancelled Aborting Aborted Completing Completed) : None	The Status of a Process is determined when the Process is being executed by a process engine. The Status of a Process can be used within Assignment Expressions.
GraphicalElements*: ObjectID	The GraphicalElements attribute identifies all of the objects (e.g., Events, Activities, Gateways, and Artifacts) that are contained within the Business Process.
Assign*: Assignment	Zero or more assignment expressions MAY be made for the object. The Assignment SHALL be performed as defined by the AssignTime attribute (see below). The Assignment will be in the following format: To = From. Both sides of the Assignment are defined separately as defined in the section entitled “Assignment” on page 261. The type for each side of the Assignment MUST match.
AssignTime*: (Start End): Start	Each Assignment will have an AssignTime. A value of Start means that the assignment SHALL occur at the start of the Process. A value of End means that the assignment SHALL occur at the end of the Process.

Attributes	Description
Properties* : Property	Modeler-defined Properties MAY be added to a Process. These Properties are “local” to the Process. All Tasks, Sub-Process objects, and Sub-Processes that are embedded SHALL have access to these Properties. The fully delineated name of these properties are “<process name>.<property name>” (e.g., “Add Customer.Customer Name”). If a process is embedded within another Process, then the fully delineated name SHALL also be preceded by the Parent Process name for as many Parents there are until the top level Process. Further details about the definition of a Property can be found in the section entitled “Property” on page 259.
AdHoc : Boolean: False	AdHoc is a Boolean attribute, which has a default of False. This specifies whether the Process is Ad Hoc or not. The activities within an Ad Hoc Process are not controlled or sequenced in a particular order, their performance is determined by the performers of the activities. If set to True, then the Ad Hoc marker SHALL be placed at the bottom center of the Process or the Sub-Process shape for Ad Hoc Processes.
(AdHoc = True only) AdHocOrdering ? : (Sequential Parallel): Parallel	If the Process is Ad Hoc (the AdHoc attribute is True), then the AdHocOrdering attribute MUST be included. This attribute defines if the activities within the Process can be performed in Parallel or must be performed sequentially. The default setting is Parallel and the setting of Sequential is a restriction on the performance that may be required due to shared resources.
(AdHoc = True only) AdHocCompletionCondition ? : Expression	If the Process is Ad Hoc (the AdHoc attribute is True), then the AdHocCompletionCondition attribute MUST be included. This attribute defines the conditions when the Process will end.
Category* : String	This modeler MAY add one or more defined Categories that can be used for purposes such as reporting and analysis.
Documentation ? : String	The modeler MAY add text documentation about the Process.

Table 6 Process Attributes

3.6.2 Changes Since 1.0 Draft Version

These are the changes since the last publically release version:

- The set of Process attributes was decoupled from the set of Common Object attributes. This was done since some of the Common attributes, such as Pool and Lane, did not apply to a Process.
- The Id, Name, ProcessType, GraphicalElements, Assign, AdHocOrdering, Category, and Document attributes were added to the set of Process attributes.
- The Property attribute, within the set of Process Attributes, was renamed to Properties and defined as type Property.
- The multiplicity of the AssignTime attribute, within the set of Process attributes, was change to zero to many.
- The AdHocCompletionCondition attribute, within the set of Process attributes, was changed to an option attribute.
- The Name and Type attributes were removed from the set of Process attributes. These attributes can be found in the definition of a Property, which can be found in the section

3.6.2 Changes Since 1.0 Draft Version

entitled “Property” on page 259.

- The PassThrough attribute was removed from the set of Process attributes. Link type of Start, End, and Intermediate Events provide this functionality.