

Activities will be external to the Participant. Each Participant will have a different perspective as to which are internal and external. At runtime, the difference between internal and external **Activities** is important in how a Participant can view the status of the **Activities** or trouble-shoot any problems. However, the Diagram itself remains the same. Figure 7.3 displays a **Business Process** that has two points of view. One point of view is of a Patient, the other is of the Doctor's office. The Diagram shows the **Activities** of both participants in the **Process**, but when the **Process** is actually being performed, each Participant will only have control over their own **Activities**. Although the Diagram point of view is important for a viewer of the Diagram to understand how the behavior of the **Process** will relate to that viewer, **BPMN** will not currently specify any graphical mechanisms to highlight the point of view. It is open to the modeler or modeling tool vendor to provide any visual cues to emphasize this characteristic of a Diagram.

Understanding the Behavior of Diagrams

Throughout this document, we discuss how **Sequence Flows** are used within a **Process**. To facilitate this discussion, we employ the concept of a *token* that will traverse the **Sequence Flows** and pass through the elements in the **Process**. A *token* is a theoretical concept that is used as an aid to define the behavior of a **Process** that is being performed. The behavior of **Process** elements can be defined by describing how they interact with a *token* as it "traverses" the structure of the **Process**. However, modeling and execution tools that implement **BPMN** are NOT REQUIRED to implement any form of *token*.

A **Start Event** generates a *token* that MUST eventually be consumed at an **End Event** (which MAY be implicit if not graphically displayed). The path of *tokens* should be traceable through the network of **Sequence Flows**, **Gateways**, and **Activities** within a **Process**.

Note – A *token* does not traverse a **Message Flow** since it is a **Message** that is passed down a **Message Flow** (as the name implies).

7.2 BPMN Elements

It should be emphasized that one of the drivers for the development of **BPMN** is to create a simple and understandable 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 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 five basic categories of elements are:

1. Flow Objects
2. Data
3. Connecting Objects
4. Swimlanes
5. Artifacts

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

1. Events
2. Activities
3. Gateways

Data is represented with the four elements:

1. Data Objects
2. Data Inputs
3. Data Outputs
4. Data Stores

There are four ways of connecting the Flow Objects to each other or other information. There are four Connecting Objects:

1. Sequence Flows
2. Message Flows
3. Associations
4. Data Associations

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

1. Pools
2. Lanes

Artifacts are used to provide additional information about the **Process**. There are two standardized *Artifacts*, but modelers or modeling tools are free to add as many *Artifacts* as necessary. There could be additional **BPMN** efforts to standardize a larger set of *Artifacts* for general use or for vertical markets. The current set of *Artifacts* includes:

- Group
- Text Annotation

7.2.1 Basic BPMN Modeling Elements

Table 7.1 displays a list of the basic modeling elements that are depicted by the notation.

Table 7.1 - Basic Modeling Elements

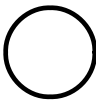

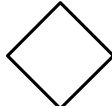
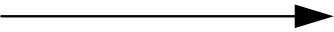
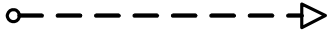
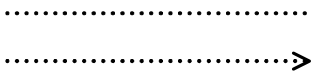





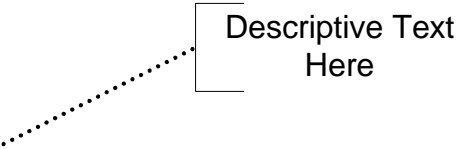
Element	Description	Notation
Event	An Event is something that “happens” during the course of a Process (see page 238) or a Choreography (see page 339). These Events affect the flow of the model and usually have a cause (<i>trigger</i>) or an impact (<i>result</i>). Events are circles with open centers to allow internal markers to differentiate different <i>triggers</i> or <i>results</i> . 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 (see page 151) in a Process. An Activity can be atomic or non-atomic (compound). The types of Activities that are a part of a Process Model are: Sub-Process and Task, which are rounded rectangles. Activities are used in both standard Processes and in Choreographies.	
Gateway	A Gateway is used to control the divergence and convergence of Sequence Flows in a Process (see page 145) and in a Choreography (see page 344). 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 (see page 97) and in a Choreography (see page 320).	
Message Flow	A Message Flow is used to show the flow of Messages between two <i>Participants</i> that are prepared to send and receive them (see page 120). In BPMN, two separate Pools in a Collaboration Diagram will represent the two <i>Participants</i> (e.g., PartnerEntities and/or PartnerRoles).	
Association	An Association is used to link information and Artifacts with BPMN graphical elements (see page 67). Text Annotations (see page 71) and other Artifacts (see page 66) can be Associated with the graphical elements. An arrowhead on the Association indicates a direction of flow (e.g., data), when appropriate.	

Table 7.1 - Basic Modeling Elements

Pool	A Pool is the graphical representation of a <i>Participant</i> in a Collaboration (see page 112). It also acts as a “swimlane” and a graphical container for partitioning a set of Activities from other Pools, usually in the context of B2B situations. A Pool MAY have internal details, in the form of the Process that will be executed. Or a Pool MAY have no internal details, i.e., it can be a “black box.”	
Lane	A Lane is a sub-partition within a Process, sometimes within a Pool, and will extend the entire length of the Process, either vertically or horizontally (see on page 305). Lanes are used to organize and categorize Activities.	
Data Object	Data Objects provide information about what Activities require to be performed and/or what they produce (see page 205), Data Objects can represent a singular object or a collection of objects. Data Input and Data Output provide the same information for Processes.	
Message	A Message is used to depict the contents of a communication between two <i>Participants</i> (as defined by a business PartnerRole or a business PartnerEntity—see on page 93).	
Group (a box around a group of objects within the same category)	A Group is a grouping of graphical elements that are within the same Category (see page 70). This type of grouping does not affect the Sequence Flows within the Group. The Category name appears on the diagram as the group label. Categories can be used for documentation or analysis purposes. Groups are one way in which Categories of objects can be visually displayed on the diagram.	
Text Annotation (attached with an Association)	Text Annotations are a mechanism for a modeler to provide additional text information for the reader of a BPMN Diagram (see page 71).	

7.2.2 Extended BPMN Modeling Elements

Table 7.2 displays a more extensive list of the **Business Process** concepts that could be depicted through a business process modeling notation.

Table 7.2 - BPMN Extended Modeling Elements





Element	Description	Notation
Event	An Event is something that “happens” during the course of a Process (see page 238) or a Choreography (see page 339). These Events affect the flow of the model and usually have a cause (<i>Trigger</i>) or an impact (<i>Result</i>). Events are circles with open centers to allow internal markers to differentiate different <i>Triggers</i> or <i>Results</i> . There are three types of Events, based on when they affect the flow: Start, Intermediate, and End.	
Flow Dimension (e.g., Start, Intermediate, End)		
Start	As the name implies, the Start Event indicates where a particular Process (see page 238) or Choreography (see page 339) will start.	Start 
Intermediate	Intermediate Events occur between a Start Event and an End Event. They will affect the flow of the Process (see page 249) or Choreography (see page 341), but will not start or (directly) terminate the Process.	Intermediate 
End	As the name implies, the End Event indicates where a Process (see page 246) or Choreography (see page 343) will end.	End 

Table 7.2 - BPMN Extended Modeling Elements

Type Dimension (e.g., None, Message, Timer, Error, Cancel, Compensation, Conditional, Link, Signal, Multiple, Terminate.)	<p>The Start and some Intermediate Events have “triggers” that define the cause for the Event (see section entitled “Start Event” on page 238 and section entitled “Intermediate Event” on page 249). There are multiple ways that these events can be triggered. End Events MAY define a “result” that is a consequence of a Sequence Flow path ending. Start Events can only react to (“catch”) a <i>trigger</i>. End Events can only create (“throw”) a <i>result</i>. Intermediate Events can catch or throw <i>triggers</i>. For the Events, <i>triggers</i> that catch, the markers are unfilled, and for <i>triggers</i> and <i>results</i> that throw, the markers are filled.</p> <p>Additionally, some Events, which were used to interrupt Activities in BPMN 1.1, can now be used in a mode that does not interrupt. The boundary of these Events is dashed (see figure to the right).</p>	<table><thead><tr><th></th><th>“Catching”</th><th>“Throwing”</th><th>Non-Interrupting</th></tr></thead><tbody><tr><td>Message</td><td></td><td></td><td></td></tr><tr><td>Timer</td><td></td><td></td><td></td></tr><tr><td>Error</td><td></td><td></td><td></td></tr><tr><td>Escalation</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>Conditional</td><td></td><td></td><td></td></tr><tr><td>Link</td><td></td><td></td><td></td></tr><tr><td>Signal</td><td></td><td></td><td></td></tr><tr><td>Terminate</td><td></td><td></td><td></td></tr><tr><td>Multiple</td><td></td><td></td><td></td></tr><tr><td>Parallel Multiple</td><td></td><td></td><td></td></tr></tbody></table>		“Catching”	“Throwing”	Non-Interrupting	Message				Timer				Error				Escalation				Cancel				Compensation				Conditional				Link				Signal				Terminate				Multiple				Parallel Multiple			
	“Catching”	“Throwing”	Non-Interrupting																																																			
Message																																																						
Timer																																																						
Error																																																						
Escalation																																																						
Cancel																																																						
Compensation																																																						
Conditional																																																						
Link																																																						
Signal																																																						
Terminate																																																						
Multiple																																																						
Parallel Multiple																																																						
Activity	An Activity is a generic term for work that company performs (see page 151) in a Process. An Activity can be atomic or non-atomic (compound). The types of Activities that are a part of a Process Model are: Sub-Process and Task, which are rounded rectangles. Activities are used in both standard Processes and in Choreographies.																																																					
Task (Atomic)	A Task is an atomic Activity that is included within a Process (see page 156). A Task is used when the work in the Process is not broken down to a finer level of Process detail.	<div>Task Name</div>																																																				
Choreography Task	A Choreography Task is an atomic Activity in a Choreography (see page 323). It represents a set of one (1) or more Message exchanges. Each Choreography Task involves two (2) <i>Participants</i> . The name of the Choreography Task and each of the <i>Participants</i> are all displayed in the different bands that make up the shape’s graphical notation. There are two (2) or more <i>Participant Bands</i> and one Task Name Band.	<div>Participant A</div> <div>Choreography Task Name</div> <div>Participant B</div>																																																				

Table 7.2 - BPMN Extended Modeling Elements

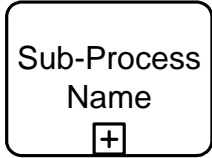
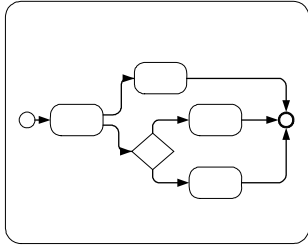
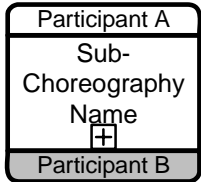
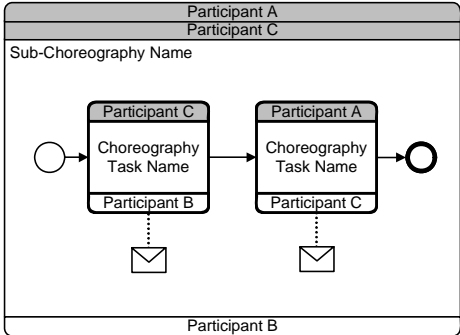
Process/Sub-Process (non-atomic)	A Sub-Process is a compound Activity that is included within a Process (see page 173) or Choreography (see page 328). It is compound in that it can be broken down into a finer level of detail (a Process or Choreography) through a set of sub-Activities.	See Next Four Figures
Collapsed Sub-Process	The details of the Sub-Process are not visible in the Diagram (see page 173). 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 Sub-Process	The boundary of the Sub-Process is expanded and the details (a Process) are visible within its boundary (see page 173). Note that Sequence Flows cannot cross the boundary of a Sub-Process.	
Collapsed Sub-Choreography	The details of the Sub-Choreography are not visible in the Diagram (see page 328). A “plus” sign in the lower-center of the Task Name Band of the shape indicates that the Activity is a Sub-Process and has a lower-level of detail.	
Expanded Sub-Choreography	The boundary of the Sub-Choreography is expanded and the details (a Choreography) are visible within its boundary (see page 328). Note that Sequence Flows cannot cross the boundary of a Sub-Choreography.	

Table 7.2 - BPMN Extended Modeling Elements

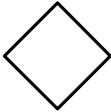








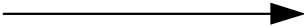
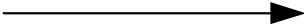
Gateway	A Gateway is used to control the divergence and convergence of Sequence Flows in a Process (see page 287) and in a Choreography (see page 344). Thus, it will determine branching, forking, merging, and joining of paths. Internal markers will indicate the type of behavior control (see below).	
Gateway Control Types	<p>Icons within the diamond shape of the Gateway will indicate the type of flow control behavior. The types of control include:</p> <ul style="list-style-type: none"> • Exclusive decision and merging. Both Exclusive (see page 290) and Event-Based (see page 297) perform exclusive decisions and merging. Exclusive can be shown with or without the “X” marker. • Event-Based and Parallel Event-based gateways can start a new instance of the Process. • Inclusive Gateway decision and merging (see page 292). • Complex Gateway -- complex conditions and situations (e.g., 3 out of 5; page 295). • Parallel Gateway forking and joining (see page 293). <p>Each type of control affects both the incoming and outgoing flow.</p>	<p>Exclusive  or </p> <p>Event-Based  </p> <p>Parallel Event-Based </p> <p>Inclusive </p> <p>Complex </p> <p>Parallel </p>
Sequence Flow	A Sequence Flow is used to show the order that Activities will be performed in a Process (see page 97) and in a Choreography (see page 320).	See next seven figures
Normal Flow	<i>Normal flow</i> refers to paths of Sequence Flow that do not start from an Intermediate Event attached to the boundary of an Activity.	
Uncontrolled flow	<i>Uncontrolled flow</i> 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 Flows that converge to or diverge from an Activity. For each uncontrolled Sequence Flows a <i>token</i> will flow from the source object through the Sequence Flows to the target object.	

Table 7.2 - BPMN Extended Modeling Elements

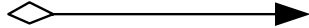
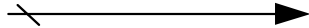
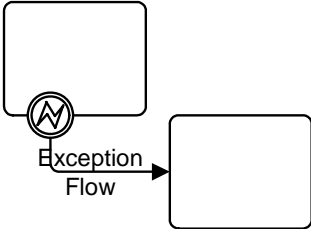

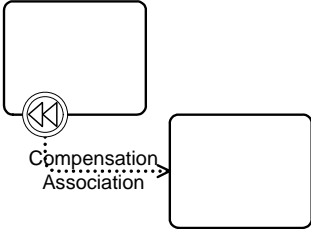
Conditional flow	A Sequence Flow can have a condition Expression that are evaluated at runtime to determine whether or not the Sequence Flow will be used (i.e., will a <i>token</i> travel down the Sequence Flow – see page 97). If the <i>conditional flow</i> is outgoing from an Activity, then the Sequence Flow will have a mini-diamond at the beginning of the connector (see figure to the right). If the <i>conditional flow</i> 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 Gateways or Inclusive Gateways, one type of flow is the Default <i>condition flow</i> (see page 97). This flow will be used only if all the other outgoing <i>conditional flow</i> is not <i>true</i> at runtime. These Sequence Flows will have a diagonal slash added to the beginning of the connector (see the figure to the right).	
Exception Flow	<i>Exception flow</i> occurs outside the <i>normal flow</i> of the Process and is based upon an Intermediate Event attached to the boundary of an Activity that occurs during the performance of the Process (see page 287).	
Message Flow	A Message Flow is used to show the flow of Messages between two <i>Participants</i> that are prepared to send and receive them (see page 120). In BPMN, two separate Pools in a Collaboration Diagram will represent the two <i>Participants</i> (e.g., <i>PartnerEntities</i> and/or <i>PartnerRoles</i>).	
Compensation Association	<i>Compensation Association</i> occurs outside the <i>normal flow</i> of the Process and is based upon a Compensation Intermediate Event that is triggered through the failure of a <i>transaction</i> or a <i>throw Compensation Event</i> (see page 302). The target of the Association MUST be marked as a Compensation Activity.	

Table 7.2 - BPMN Extended Modeling Elements






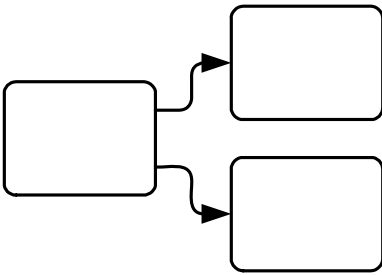
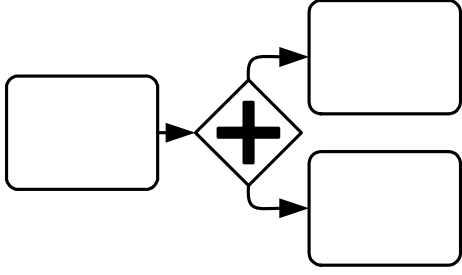
Data Object	<p>Data Objects provide information about what Activities require to be performed and/or what they produce (see page 205), Data Objects can represent a singular object or a collection of objects. Data Input and Data Output provide the same information for Processes.</p>	<p>Data Object</p>  <p>Data Object (Collection)</p>  <p>Data Input Data Output</p>  
Message	<p>A Message is used to depict the contents of a communication between two <i>Participants</i> (as defined by a business <i>PartnerRole</i> or a business <i>PartnerEntity</i>—see on page 93).</p>	
Fork	<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.</p> <p>There are two options:</p> <ul style="list-style-type: none"> • Multiple Outgoing Sequence Flows can be used (see figure top-right). This represents “uncontrolled” flow is the preferred method for most situations. • A Parallel Gateway can be used (see figure bottom-right). This will be used rarely, usually in combination with other Gateways. 	 

Table 7.2 - BPMN Extended Modeling Elements

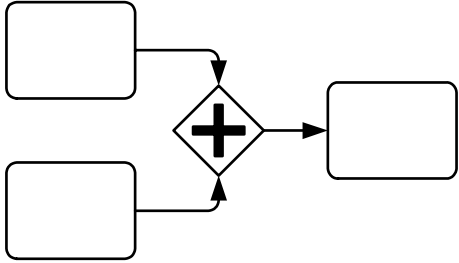
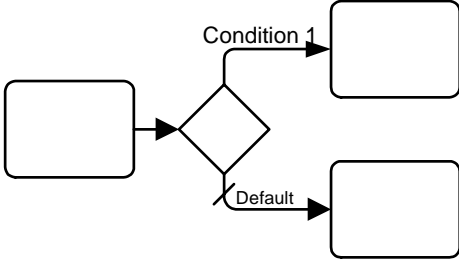
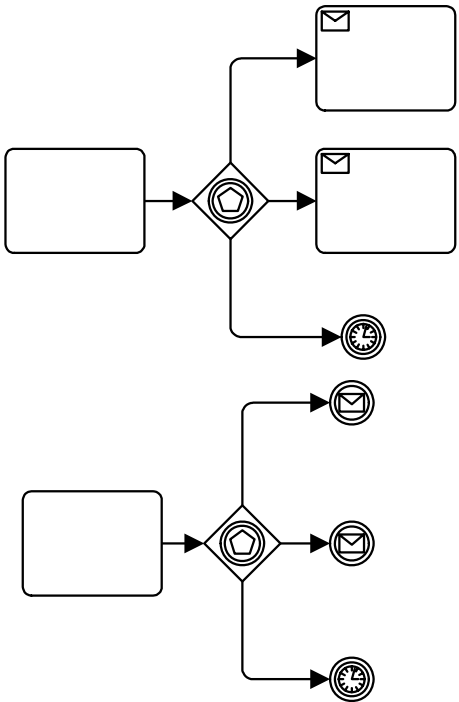
Join	<p>BPMN uses the term “join” to refer to the combining of two or more parallel paths into one path (also known as an AND-Join or synchronization).</p> <p>A Parallel Gateway is used to show the joining of multiple Sequence Flows.</p>	 <pre> graph LR A[] --> C{+} B[] --> C C --> D[] </pre>
Decision, Branching Point	<p>Decisions are Gateways within a Process (see page 287) or a Choreography (see page 344) where the flow of control can take one or more alternative paths.</p>	<p>See next five rows.</p>
Exclusive	<p>This Decision represents a branching point where Alternatives are based on conditional Expressions contained within the <i>outgoing</i> Sequence Flows (see page 290 or page 345). Only one of the Alternatives will be chosen.</p>	 <pre> graph LR A[] --> C{ } C -- "Condition 1" --> B1[] C -- "Default" --> B2[] </pre>
Event-Based	<p>This Decision represents a branching point where Alternatives are based on an Event that occurs at that point in the Process (see page 297) or Choreography (see page 350). 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:</p> <ul style="list-style-type: none"> • Tasks of Type Receive can be used (see figure top-right). • Intermediate Events of Type Message can be used (see figure bottom-right). 	 <pre> graph LR subgraph Top A1[] --> C1{ } C1 --> B1[] C1 --> B2[] C1 --> B3(()) end subgraph Bottom A2[] --> C2{ } C2 --> B4(()) C2 --> B5(()) C2 --> B6(()) end </pre>

Table 7.2 - BPMN Extended Modeling Elements

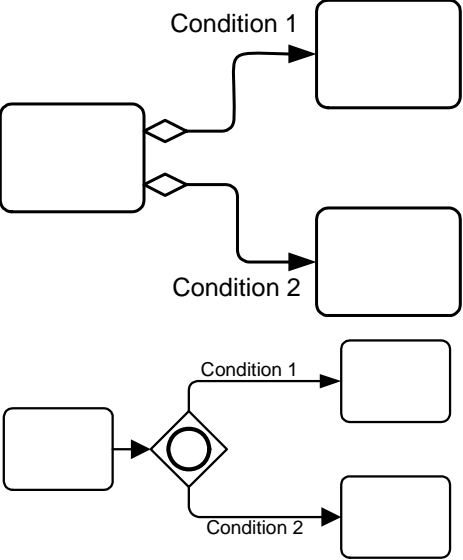
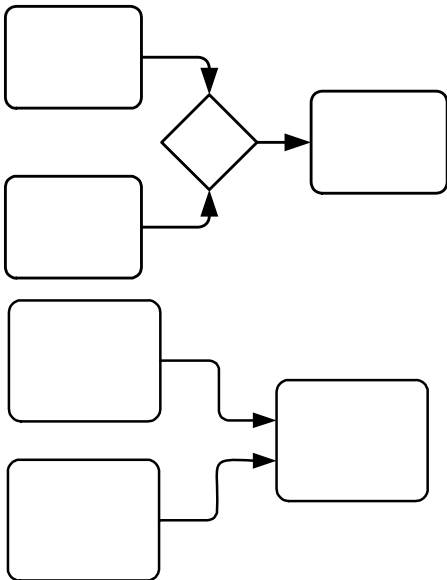
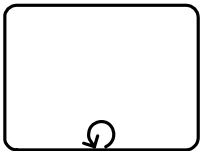
<p>Inclusive</p>	<p>This Decision represents a branching point where Alternatives are based on conditional Expressions contained within the outgoing Sequence Flows (see page 292). In some sense it is a grouping of related independent Binary (Yes/No) Decisions. Since each path is independent, all combinations of the paths MAY be taken, from zero to all. However, it should be designed so that at least one path is taken. A Default Condition could be used to ensure that at least one path is taken.</p> <p>There are two versions of this type of Decision:</p> <ul style="list-style-type: none"> • The first uses a collection of conditional Sequence Flows, marked with mini-diamonds (see top-right figure). • The second uses an Inclusive Gateway (see bottom-right picture). 	
<p>Merging</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 OR-Join). A Merging Exclusive Gateway is used to show the merging of multiple Sequence Flows (see upper figure to the right). If all the incoming flow is alternative, then a Gateway is not needed. That is, uncontrolled flow provides the same behavior (see lower figure to the right).</p>	
<p>Looping</p>	<p>BPMN provides two mechanisms for looping within a Process.</p>	<p>See Next Two Figures</p>
<p>Activity Looping</p>	<p>The attributes of Tasks and Sub-Processes will determine if they are repeated or performed once (see page 190). 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>	

Table 7.2 - BPMN Extended Modeling Elements

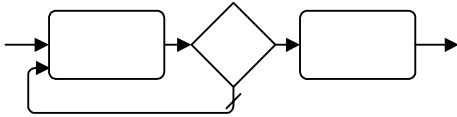
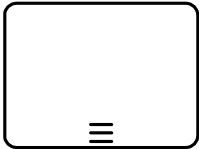
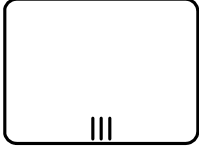
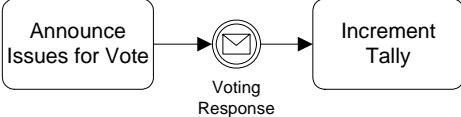
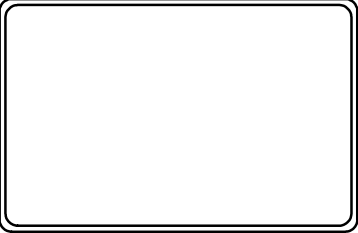
Sequence Flow Looping	Loops can be created by connecting a Sequence Flow to an “upstream” object. An object is considered to be upstream if that object has an outgoing Sequence Flow that leads to a series of other Sequence Flows, the last of which is an incoming Sequence Flow for the original object.	
Multiple Instances	The attributes of Tasks and Sub-Processes will determine if they are repeated or performed once (see page 191). A set of three horizontal lines will be displayed at the bottom-center of the activity for sequential Multi-Instances (see upper figure to the right). A set of three vertical lines will be displayed at the bottom-center of the activity for sequential Multi-Instances (see lower figure to the right).	<p>Sequential</p>  <p>Parallel</p> 
Process Break (something out of the control of the process makes the process pause)	A Process Break is a location in the Process that shows where an expected delay will occur within a Process (see page 249). 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.	
Transaction	A transaction is a Sub-Process that is supported by a special protocol that insures that all parties involved have complete agreement that the activity should be completed or cancelled (see page 178). 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.	

Table 7.2 - BPMN Extended Modeling Elements


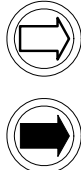

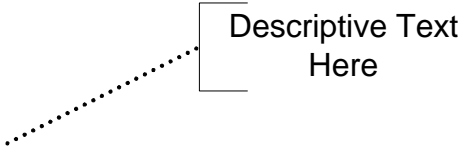

Nested/Embedded Sub-Process (Inline Block)	A nested (or embedded) Sub-Process is an activity that shares the same set of data as its parent process (see page 173). This is opposed to a Sub-Process that is independent, re-usable, and referenced from the parent process. Data needs to be passed to the referenced Sub-Process, but not to the nested Sub-Process.	There is no special indicator for nested Sub-Processes
Group (a box around a group of objects within the same category)	A Group is a grouping of graphical elements that are within the same <i>Category</i> (see page 68). This type of grouping does not affect the Sequence Flows within the Group. The <i>Category</i> name appears on the diagram as the group label. <i>Categories</i> can be used for documentation or analysis purposes. Groups are one way in which <i>Categories</i> of objects can be visually displayed on the diagram.	
Off-Page Connector	Generally used for printing, this object will show where a 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 link information and Artifacts with BPMN graphical elements (see page 67). Text Annotations (see page 71) and other Artifacts (see page 66) can be Associated with the graphical elements. An arrowhead on the Association indicates a direction of flow (e.g., data), when appropriate.	
Text Annotation (attached with an Association)	Text Annotations are a mechanism for a modeler to provide additional text information for the reader of a BPMN Diagram (see page 71).	
Pool	A Pool is the graphical representation of a <i>Participant</i> in a Collaboration (see page 112). It also acts as a "swimlane" and a graphical container for partitioning a set of Activities from other Pools, usually in the context of B2B situations. A Pool MAY have internal details, in the form of the Process that will be executed. Or a Pool MAY have no internal details, i.e., it can be a "black box."	

Table 7.2 - BPMN Extended Modeling Elements

Lanes	A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally (see on page 305). Lanes are used to organize and categorize Activities.	<table><tr><td>Name</td><td></td></tr><tr><td>Name</td><td></td></tr></table>	Name		Name	
Name						
Name						

7.3 BPMN Diagram Types

The **BPMN 2.0** aims to cover three basic models of **Processes**: *private Processes* (both *executable* and *non-executable*), *public Processes*, and **Choreographies**. Within and between these three **BPMN** sub-models, many types of Diagrams can be created. The following are examples of **Business Processes** that can be modeled using **BPMN 2.0**:

- High-level *non-executable* **Process Activities** (not functional breakdown)
- Detailed executable **Business Process**
- As-is or old **Business Process**
- To-be or new **Business Process**
- A description of expected behavior between two (2) or more business *Participants*—a **Choreography**.
- Detailed *private Business Process* (either *executable* or *non-executable*) with interactions to one or more external *Entities* (or “Black Box” **Processes**)
- Two or more detailed *executable Processes* interacting
- Detailed *executable Business Process* relationship to a **Choreography**
- Two or more *public Processes*
- *Public Process* relationship to **Choreography**
- Two or more detailed *executable Business Processes* interacting through a **Choreography**

BPMN is designed to allow describing all above examples of **Business Processes**. However, the ways that different sub-models are combined is left to tool vendors. A **BPMN 2.0** compliant implementation could RECOMMEND that modelers pick a focused purpose, such as a *private Process*, or **Choreographies**. However, the **BPMN 2.0** specification makes no assumptions.

7.4 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 a **BPMN** Diagram.

- ◆ BPMN elements (e.g. Flow objects) 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 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). However,