Business Process Model and Notation—BPMN

Mark von Rosing, Stephen White, Fred Cummins, Henk de Man

INTRODUCTION

This chapter is intended to provide an overview and introduction to the Business Process Model and Notation (BPMN). We will describe BPMN and its historic development. In addition, we will provide the general context and usage of BPMN, layered upon the technical details defined in the BPMN 2.0 Specification. The basics of the BPMN notation will be described—that is, the types of graphical shapes, their purpose, and how they work together as part of a Business Process Model/Diagram. Also discussed will be the different uses of BPMN diagram types, including how levels of precision affect what a modeler will include in a diagram. Finally, the value in using BPMN as a standard notation will be defined.

It is vital to note that because both main authors and the additional four authors all officially work with the Object Management Group (OMG) to develop standards, this chapter and its content be based on the official OMG BPMN specification.[1]

WHAT IS BPMN?

Business Process Model and Notation (BPMN) is a standard for business process modeling that provides graphical notation for specifying business processes in a Business Process Diagram (BPD),[2] based on traditional flowcharting techniques. The objective of BPMN is to support business process modeling for both technical users and business users, by providing notation that is intuitive to business users, yet able to represent complex process semantics. The BPMN 2.0 specification also provides execution semantics as well as mapping between the graphics of the notation and other execution languages, particularly Business Process Execution Language (BPEL).[3]

BPMN is designed to be readily understandable by all business stakeholders. These include the business analysts who create and refine the processes, the technical developers responsible for implementing them, and the business managers who monitor and manage them. Consequently, BPMN serves as a common language, bridging the communication gap that frequently occurs between business process design and implementation.

The Complete Business Process Handbook. http://dx.doi.org/10.1016/B978-0-12-799959-3.00021-5

THE HISTORIC DEVELOPMENT OF BPMN

In 2001, the process-modeling marketplace was fragmented with many different modeling notations and viewpoints. It was in this context that members of Business Process Management Institute (BPMI), many of whom represented companies that contributed to the fragmented market, began discussing the idea of standardizing business-oriented techniques for visually representing process components and aligning the notation with an executable process language. The BPMN 1.0 specification was released to the public in May 2004. With this, the primary goal of the BPMN specification was to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. BPMN 1.0 was also supported with an internal model that was mapped to executable BPEL4WS.

It was February 6, 2006, when BPMI was subsumed by the OMG, who has since maintained and developed the BPMN standard. The BPMN 1.1 version was published in January 2008[4] and a year later version 1.2[5] was published. Work on the well-known version 2.0 took another two years, and it was published in January 2011.[6] This international standard represents the amalgamation of best practices within the business modeling community to define the notation and semantics of collaboration diagrams, process diagrams, and choreography diagrams. In doing so, BPMN will provide a simple means of communicating process information to other business users, process implementers, customers, and suppliers.

Another goal, but no less important, is to ensure that the models created by BPMN are executable. BPMN 1.x provided mappings to Extensible Markup Language (XML) designed for the execution of business processes, such as Web Services Business Process Execution Language (WSBPEL). The ability to execute BPMN via BPEL (BPEL, also known as WS-BPEL) breathed life into model-driven process execution. In essence, the equation Application = Computation + Coordination has become reality with network-addressable computation being provided by Web Services and BPMN graphically depicting the coordination logic. BPMN 2.0 provided its own execution semantics in addition to an updated mapping to BPEL. Thus, new process engines can directly execute BPMN models without the potential behavioral restrictions that might result in the complex mapping of the more free-form BPMN to the more structured BPEL.

Some of the main changes that the BPMN versions 2.0 brought with them are among others:

- The addition of a Choreography diagram.
- The addition of a Conversation diagram.

- Noninterrupting Events for a Process.
- Event Subprocesses for a Process.

The major technical changes include:

- A definition of the process execution semantics.
- A formal metamodel as shown through the class diagram figures.
- Interchange formats for abstract syntax model interchange in both XML Metadata Interchange (XMI) and XML Schema Definition (XSD).
- Interchange formats for diagram interchange in both XMI and XSD.
- Extensible Stylesheet Language Transformations (XSLT) between the XMI and XSD formats.

Other technical changes include:

- Reference Tasks are removed. These provided reusability within a single diagram, as compared to Global Tasks, which are resuable across multiple diagrams. The new Call Activity can be used to reference a Global Task or another Process to be used within a Process (instead of Reference Tasks).

Because of the version 2.0 updates, the number of elements more than doubled from 55 elements to 116. Many of these new elements were applied to modeling interactions between processes and/or entities, such as the new choreography diagram.

BPMN 2.0.2, released in December 2013,[7] included only minor modifications in terms of typo corrections and a change in clause 15.

THE BPMN NOTATIONS/SHAPES

A major goal for the development of BPMN was to create a simple and understandable notation for creating Business Process models, while providing the semantics and underlying mechanisms to handle the complexity inherent in 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. The various basic BPMN shapes are shown below (Table 1–6):

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. In the following sections, we will illustrate how the BPMN shapes are used in various end-to-end BPMN models.

Table 1 *BPMN Task Description*

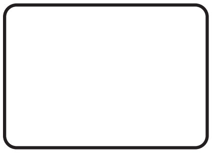
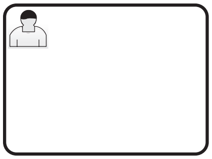
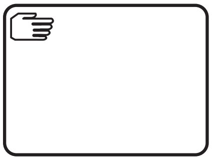
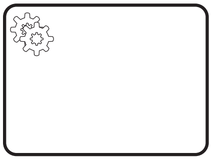
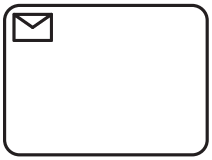
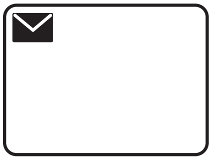
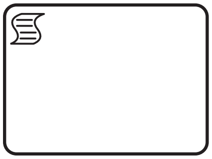
BPMN 2.0.2	Task Description
 None	No special task type is indicated.
 User Task	A User Task is a typical "workflow" task in which a human performer performs the task with the assistance of a software application and could be scheduled through a task list manager of some sort.
 Manual Task	A Manual Task is a task that is expected to be performed without the aid of any business process execution engine or application.
 Service Task	A Service Task is a task that uses some sort of service, which could be a web service or an automated application.
 Receive Task	A Receive Task is a simple task that is designed to wait for a message to arrive from an external participant (relative to the process).
 Send	A Send Task is a simple task that is designed to send a message to an external participant (relative to the process).
 Script	A Script Task is executed by a business process engine. The modeler or implementer defines a script in a language that the engine can interpret. When the task is ready to start, the engine will execute the script. When the script is completed, the task will also be completed.

Table 1 *BPMN Task Description—Cont'd*

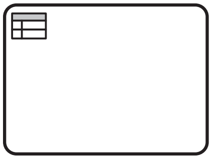
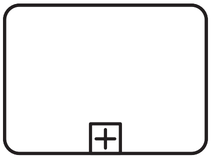
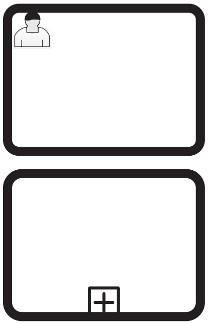
BPMN 2.0.2	Task Description
 Business Rule	A Business Rule Task provides a mechanism for the process to provide input to a Business Rules Engine and to get the output of calculations that the business rules engine might provide. The input/output specification of the task will allow the process to send data to and receive data from the Business Rules Engine.
 Sub-Process	A Sub-Process is a type of activity within a process, but it also can be "opened up" to show a lower-level process. This is useful for process decomposition or general process organization.
 Call Activity	A Call Activity is a type of activity within a process. It provides a link to reusable activities: for example, it will call a task into the Process (see upper figure on the left) or another Process (see lower figure on the left).

Table 2 *BPMN Flow Descriptio*

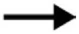
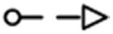
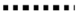
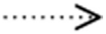
BPMN 2.0.2	Flow Description
 Sequence Flow	A Sequence Flow is represented by a solid line with a solid arrowhead and is used to show the order (the sequence) in which activities will be performed in a process or choreography diagram.
 Message Flow	A Message Flow is represented by a dashed line with an open arrowhead and is used to show the flow of messages between two separate process participants (business entities or business roles) that send and receive them.
 Association	An Association is represented by a dotted line, which may have a line arrowhead on one or both ends, and is used to associate text and other artifacts with flow objects.
 Data Association	A Data Association is represented by a dotted line with a line arrowhead and is used to associate data (electronic or nonelectronic) with flow objects. Data Associations are used to show the inputs and outputs of activities.

Table 3 *BPMN Marker Description*

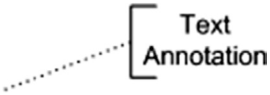
BPMN 2.0.2	Markers Description
↻ Loop Marker	A Loop Marker is used to represent an activity that will be executed multiple times until the condition is satisfied. The condition can be validated either at the start or end of the activtiy.
||| Parallel Multiple Instance Marker	A Parallel Multi-Instance Marker is used to represent an activity that can be executed as multiple instances performed in parallel. The number of instances will be determined through a condition expression that is evaluated at the start of the activity. All the instances will start in parallel and each instance can have different input parameters. The activity, as a whole, is completed after all the instances are completed. However, another expression, if it becomes true, will stop all instances and complete the activity.
≡ Sequential Multiple Instance Marker	A Sequential Multi-Instance Marker represents an activity that is similar to a Parallel Multi-Instance activity, but its instances will be executed in sequence. The second instance will wait until the first instance is completed and so on.
~ Adhoc Marker	The Adhoc Marker is a tilde symbol and used to mark a Sub-Process for which the normal sequence patterns are relaxed and its activities can be performed in any order at the discretion of the users. Tasks can start any time without any direct dependency on other tasks.
 Annotation Marker	An Annotation Marker is a mechanism for a modeler to provide additional text information (i.e., notes) for the reader of a BPMN diagram. Annotations can be connected to other objects through an Association (see above).

Table 4 *BPMN Data Object Description*

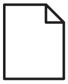
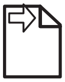
BPMN 2.0.2	Data Description
 Data Object	A Data Object represents the data that are used as inputs and outputs to the activities of a process. Data Objects can represent singular objects or collections of objects.
 Data Input	A Data Input is an external data input for the entire process. It is a kind of input parameter.

Table 4 *BPMN Data Object Description—Cont'd*

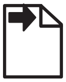
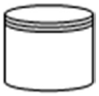
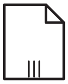
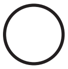
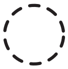
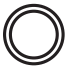
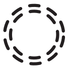
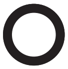
BPMN 2.0.2	Data Description
 Data Output	A Data Output is the data result of the entire process. It is a kind of output parameter.
 Data Store	A Data Store is a place where the process can read or write data (e.g., a database or a filing cabinet). It persists beyond the lifetime of the process instance.
 Collection of Data Objects	A Collection of Data Objects represents a collection of data elements related to the same data entity (e.g., a list of order items).

Table 5 *BPMN Event Description*

BPMN 2.0.2	Event Description
 Event: Start	Start Events indicate the instance or initiation of a process or an Event Sub-Process and have no incoming sequence flow. A Process can have more than one Start Event, but an Event Sub-Process only have one Start Event.
 Event: Event Sub-Process non-interrupting	Non-interrupting Start Events can be used to initiate an Event Sub-Process without interfering with the main process flow.
 Event: Intermediate and Boundary	Intermediate Events indicate something that occurs or may occur during the course of the process, between Start and End. Intermediate Catching Events can be used to catch the event trigger and can be in the flow or attached to the boundary of an activity. Intermediate Throwing Events can be used to throw the event trigger.
 Event: Boundary non-interrupting	Non-interrupting Boundary Events can be attached to the boundary of an activity. When they are triggered, flow will be generated from them, but the source activity will continue to be performed.
 Event: End	The End Event indicates where a path in the Process will end. A Process can have more than one end. The Process ends when all active paths have ended. End Events have no outgoing sequence flows.

Continued

Table 5 *BPMN Event Description—Cont'd*

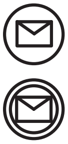
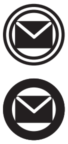
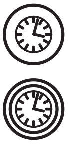
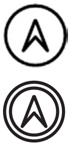
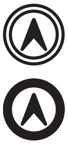
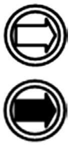
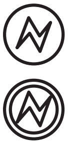
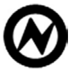
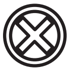
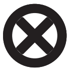
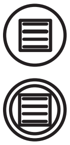
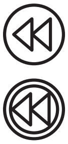
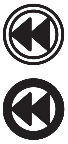
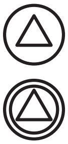
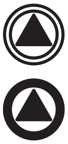
BPMN 2.0.2	Event Description
 Message (receive)	Receive messages to start a Process or in the middle of a Process, either in the flow or attached to the boundary of an activity.
 Message (send)	Send messages in the middle or at the end of a Process path.
 Timer (catch)	A Timer Event is always of catch type and used to signify waiting for a specific time condition to evaluate to true, which will start a Process, start an Event Sub-Process, wait in the middle of a flow, or wait as a Boundary Event.
 Escalation (catch)	An Escalation Event handles escalation conditions, triggering the start of an Event Sub-Process or a Boundary Event.
 Escalation (throw)	A throw Escalation Event will cause the escalation conditions that will trigger the catch Events.
 Link (throw and catch)	A Link Event has no significance related to how the Process is performed, but facilitates the diagram-creation process. For example, you can use two associated links as an alternative to a long sequence flow. There is a throwing Link Event as the "exit point," and a catching Link Event as the "entrance point," and the two events are marked as a pair.
 Error (catch)	A catch Error Event is used to capture errors and to handle them. This event can only be used as the start an Event Sub-Process or as a Boundary Event. These events can catch errors thrown by the throw Error Events or errors thrown by a BPM system or services used by the Process.

Table 5 *BPMN Event Description—Cont'd*

BPMN 2.0.2	Event Description
 Error (throw)	A throw Error Event is used set an error to be handled. This event can only be used as an End Event (i.e., never as an Intermediate Event).
 Cancel (catch)	Cancel Events can only be used in the context of the transactions. The catch Cancel Events are used as Boundary Events for the transaction Sub-Process, and will trigger the roll back of the transaction (i.e., the Activities of the Sub-Process).
 Cancel (throw)	Cancel Events can only be used in the context of the transactions. The throw Cancel Events are only used within a transaction Sub-Process.
 Conditional (catch)	Conditional Events are used to determine whether to start (or continue) only if a certain condition is true. Like the Timer Event, the Conditional Event can only exist as a catching event. They can be used at the start of a Process or an Event Sub-Process, in the middle of the flow, or as a Boundary Event.
 Compensation (catch)	A Compensation Event is used to handle compensation in the process. The catching Compensation Event be triggered as an Event Sub-Process Start Event, or as a Boundary Event.
 Compensation (throw)	A Compensation Event is used to handle compensation in the process. The throwing Compensation Event can be used in the middle or end of a Process path.
 Signal (start)	Catching Signal Events are used for receiving signals. They are a generic, simple form of communication and exist within pools (same participant), across pools (different participants), and across diagrams. They can be used at the start of a Process or an Event Sub-Process, in the middle of the flow, or as a Boundary Event.
 Signal (end)	Throwing Signal Events are used for sending signals. They are a generic, simple form of communication and exist within pools (same participant), across pools (different participants), and across diagrams. They can be used in the middle or end of a Process path.

Continued

Table 5 *BPMN Event Description—Cont'd*

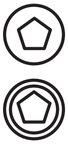
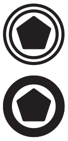
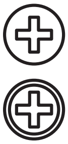
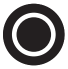
BPMN 2.0.2	Event Description
 Multiple (catch)	The Multiple Event is used to summarize several event types with a single symbol. The event is triggered if any one of those types is satisfied. They can be used at the start of a Process or an Event Sub-Process, in the middle of the flow, or as a Boundary Event.
 Multiple (throw)	The Multiple Event is used to summarize several event types with a single symbol. When this is event is reached, then all the event types are thrown. They can be used in the middle or end of a Process path.
 Parallel Multiple (catch)	The Parallel Multiple Event is used to summarize several event types with a single symbol. The difference between this event and the Multiple Event is that the Parallel Multiple is only triggered if *all* of those types are satisfied. They can be used at the start of a Process or an Event Sub-Process, in the middle of the flow, or as a Boundary Event.
 Terminate (throw)	The Terminate End Event is the "stop everything" event. When a Terminate End Event is reached, the entire process is stopped, including all parallel activities.

Table 6 *BPMN Gateway Description*

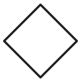
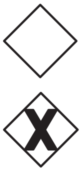
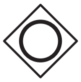
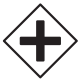
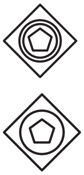
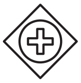
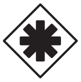
BPMN 2.0.2	Gateway Description
 Gateway	Gateways are used to control how process paths converge and diverge within a process.
 Exclusive Gateway	The Event Gateway, when splitting, routes sequence flow to only one of the outgoing branches, based on conditions. When merging, it awaits one incoming branch to complete before continuing the flow. The Gateway can be displayed with or without the "X" marker, but the behavior is the same.
 Inclusive Gateway	The Inclusive Gateway, when splitting, allows one or more branches to be activated, based on conditions. All active incoming branches must complete before merging.

Table 6 *BPMN Gateway Description*

BPMN 2.0.2	Gateway Description
 Parallel Gateway	The Parallel Gateway, when splitting, will direct the flow down all the outgoing branches. When merging, it awaits all the in branches to complete before continuing the flow.
 Event-based Gateway	The Event Gateway is always followed by catching events or receive tasks. The flow of the Process is routed to the subsequent event/task which happens first. When merging, it behaves like an Event Gateway. This Gateway can be configured such that it can be used to start a Process, based on the first event that follows it (see the lower figure on the left).
 Parallel Event-based Gateway	The Parallel Event Gateway is only used for starting a Process. It is configured like a regular Event Gateway, but *all* of the subsequent events must be triggered before a new process instance is created.
 Complex Gateway	The Complex Gateway defines behavior that is not captured by other gateways. Expressions are used to determine the merging and splitting behavior.

BPMN DIAGRAMS

Business Process Modeling is used to communicate a wide variety of process configurations to a wide variety of audiences. Thus, BPMN was designed to cover many types of modeling and allow the creation of end-to-end Business Processes. The structural elements of BPMN allow the viewer to be able to easily differentiate between sections of a BPMN Diagram. There are three basic types of submodels within a BPMN modeling environment:

1. Processes (*Orchestration*), including:
   - **a.** *Private non-executable* (internal) Business Processes.
   - **b.** *Private executable* (internal) Business Processes.
   - **c.** *Public* Processes.
2. Choreographies.
3. Collaborations, which can include Processes and/or Choreographies.
   - **a.** A view of Conversations.

TO POINT (1) PRIVATE (INTERNAL) BUSINESS PROCESSES

Private Business Processes are those internal to a specific organization. These Processes have been generally called workflow or BPM Processes (see Figure 1). Another synonym typically used in the Web services area is the *Orchestration* of services. There are two types of *private* Processes: *executable* and *non-executable*.

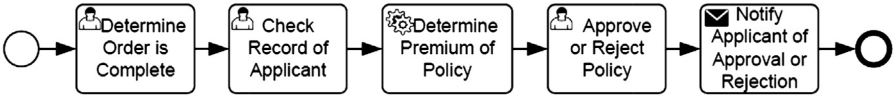

Determine Order is Complete
Check Record of Applicant
Determine Premium of Policy
Approve or Reject Policy
Notify Applicant of Approval or Rejection

FIGURE 1

Example of private process.

An *executable* Process is a Process that has been modeled for being executed according to the defined BPMN execution semantics. Of course, during the development cycle of the Process, there will be stages in which the Process does not have enough detail to be "executable."

A non-executable Process is a *private* Process that has been modeled for documenting Process behavior at a modeler-defined level of detail. Thus, information needed for execution, such as formal condition expressions are typically not included in a *non-executable* Process.

If a swim lanes-like notation is used (e.g., a Collaboration, see below) then a *private* Business Process will be contained within a single Pool. The Process flow is therefore contained within the Pool and cannot cross the boundaries of the Pool. The flow of Messages can cross the Pool boundary to show the interactions that exist between separate *private* or *public* Business Processes.

PUBLIC PROCESSES

A *public* Process represents the interactions to and from another Process or *Participant* (see Figure 2). Only those Activities and Events that are used to communicate to the other *Participants* are included in the *public* Process. These Activities and Events can be considered the "touch-points" between the participants. All other "internal" Activities of the *private* Business Process are not shown in the *public* Process. Thus, the *public* Process shows to the outside world the Message Flows and the order of those Message Flows that is needed to interact with that Process. *Public* Processes can be modeled separately

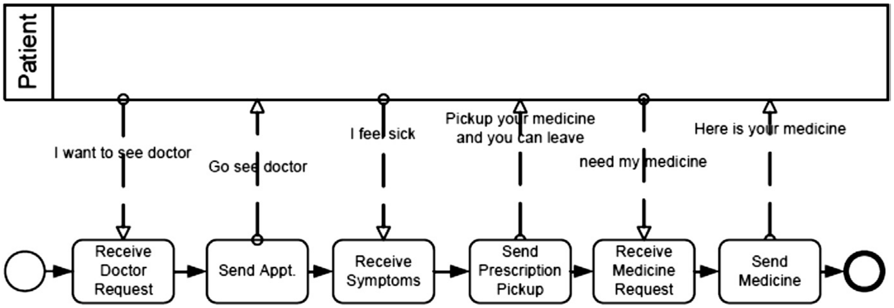


FIGURE 2

Example of public process.

or within a Collaboration to show the directional flow of Messages. Note that the *public* type of Process was named “abstract” in BPMN 1.2 (2009 release).

COLLABORATIONS

A Collaboration depicts the interactions between two or more business entities. A Collaboration usually contains two or more Pools, representing the *Participants* in the Collaboration. The Message exchange between the *Participants* is shown by a Message Flow that connects two Pools (or the objects within the Pools). The Messages associated with the Message Flows can also be shown graphically. The Collaboration can be shown as two or more *public* and/or *private* Processes communicating with each other (see Figure 3). Or a Pool MAY be empty, a “black box.” Choreography elements MAY be shown “in between” the Pools as they bisect the Message Flows between the Pools. All combinations of Pools, Processes, and a Choreography are allowed in a Collaboration.

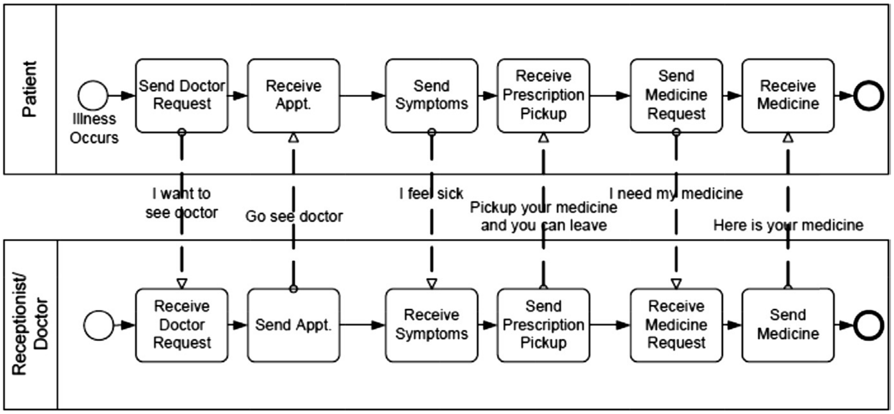


FIGURE 3

Example of a Collaboration.

TO POINT (2) CHOREOGRAPHY

A self-contained Choreography (no Pools or *Orchestration*) is a definition of the expected behavior, basically, a procedural contract between interacting *Participants*. Although a normal Process exists within a Pool, a Choreography exists between Pools (or *Participants*).

The Choreography looks similar to a *private* Business Process because it consists of a network of Activities, Events, and Gateways (see Figure 4). However, a Choreography is different in that the Activities are interactions that represent a set (one or more) of Message exchanges, which involves two or more *Participants*. In addition, unlike a normal Process, no central controller, responsible entity, or observer of the Process exists.

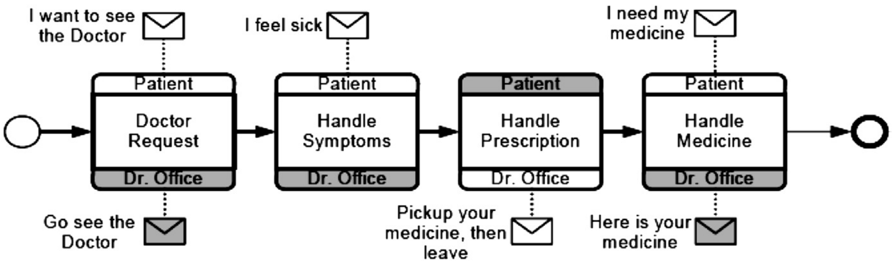


FIGURE 4

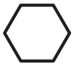
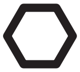
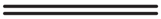
Example of a Choreography.

TO POINT (3) CONVERSATIONS

The Conversation diagram is a particular usage and an informal description of a Collaboration diagram. However, the Pools of a Conversation diagram usually do not contain a Process, and a Choreography is usually not placed between the Pools of a Conversation diagram. An individual Conversation (within the diagram) is the logical relation of Message exchanges. The logical relation, in practice, often concerns a business object(s) of interest, for example, “Order,” “Shipment and Delivery,” or “Invoice.”

Thus, the Conversation diagram is a high-level modeling diagram that depicts a set of related Conversations that reflect a distinct business scenario Table 7. For example, in logistics, stock replenishments involve the following type of scenarios: creation of sales orders, assignment of carriers for shipments combining different sales orders, crossing customs/quarantine, processing payment, and investigating exceptions. Thus, a Conversation diagram, as shown in Figure 5, shows Conversations (as hexagons) between *Participants* (Pools). This provides a “bird’s eye” perspective of the different Conversations that relate to the domain.

Table 7 *BPMN Conversation Description*

BPMN 2.0.2	Conversations Description
 Conversation	A Conversation defines a set of logically related Message Flows. When marked with a (+) symbol it indicates a Sub-Conversation, a compound conversation element.
 Call Conversation	A Call Conversation is a wrapper for a globally defined, re-usable Conversation or Collaboration. A call to a Collaboration is marked with a (+) symbol.
 Conversation Link	Connects Conversations and Participants.

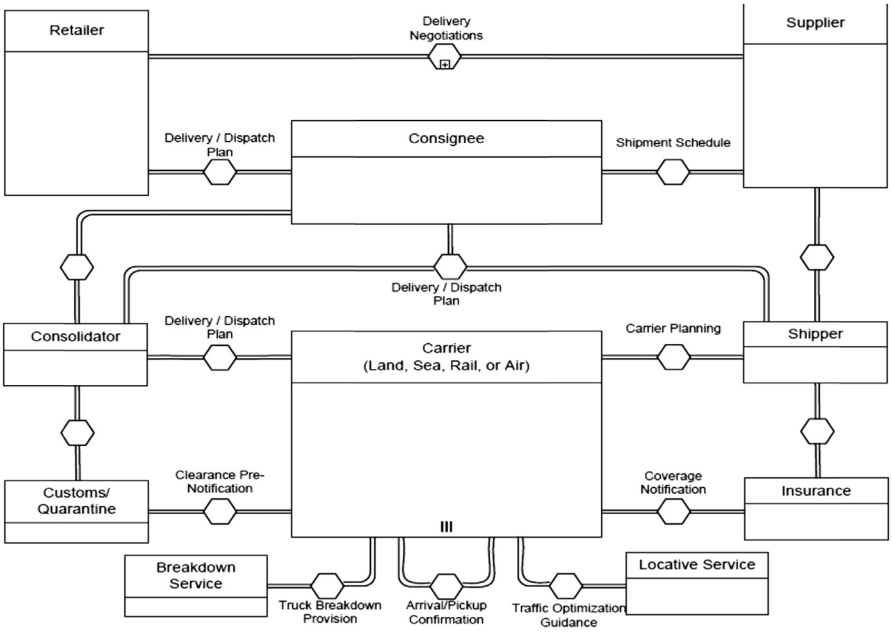


FIGURE 5

Example of a conversation diagram.

BPMN USAGE

We have just illustrated the three basic BPMN models of Processes—*private* Processes (both *executable* and *non-executable*), *public* Processes - Collaborations (including Conversations), and Choreographies. Within and between these BPMN sub-models, many types of Diagrams can be created.

The following are examples of Business Processes that can be modeled:

- 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.

One of the benefits of BPMN, among others, is that it has the flexibility to allow the development of all the above examples of business processes. However, the ways that different submodels are combined within a specific tool is a choice of the vendors and can vary quite a bit.

DIAGRAM POINT OF VIEW

Because a BPMN diagram may depict the processes of different participants, each participant could view the diagram differently. That is, the participants have different points of view regarding how the processes will apply to them. Some of the activities will be internal to a participant (that is, they are performed by or under control of that participant) and other activities will be external to that participant. Each participant will have a different perspective as to which are internal and external. At run time, the difference between internal and external activities is important in how a participant can view the status of the activities or troubleshoot any problems. However, the diagram itself remains the same. Figure 3, above, 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 may show 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

So far, we have mentioned how sequence flows are used within a process. To facilitate the understanding of process behavior, 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. However, modeling and execution tools that implement BPMN are NOT REQUIRED to implement any form of *token*.

Process elements can be defined by describing how they interact with a *token* as it moves through the structure of the Process. 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 a *token* should be traceable through the network of Sequence Flows, Gateways, Events, 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).

BPMN EXAMPLE

The following is an example of a manufacturing process from different perspectives (Figure 6–8).

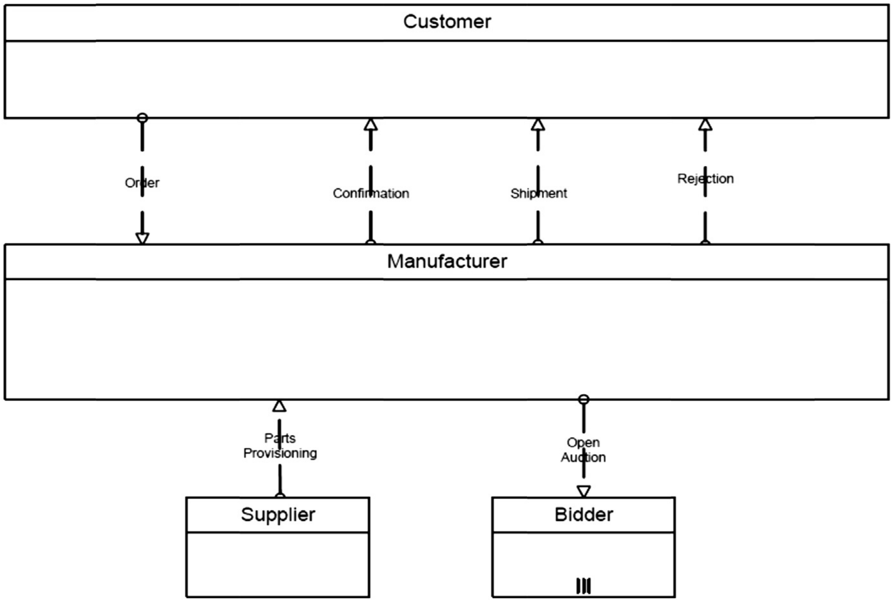


FIGURE 6

An example of a Collaboration diagram with black-box Pools.

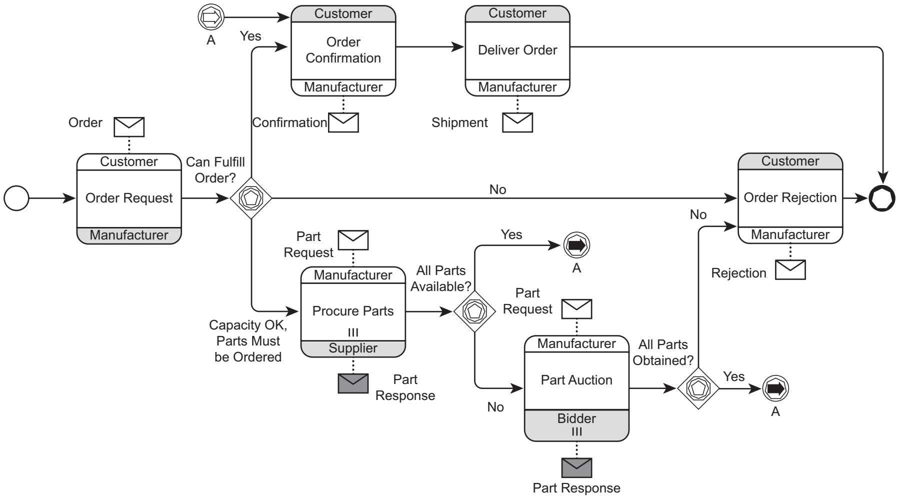


FIGURE 7

An example of a standalone Choreography diagram.

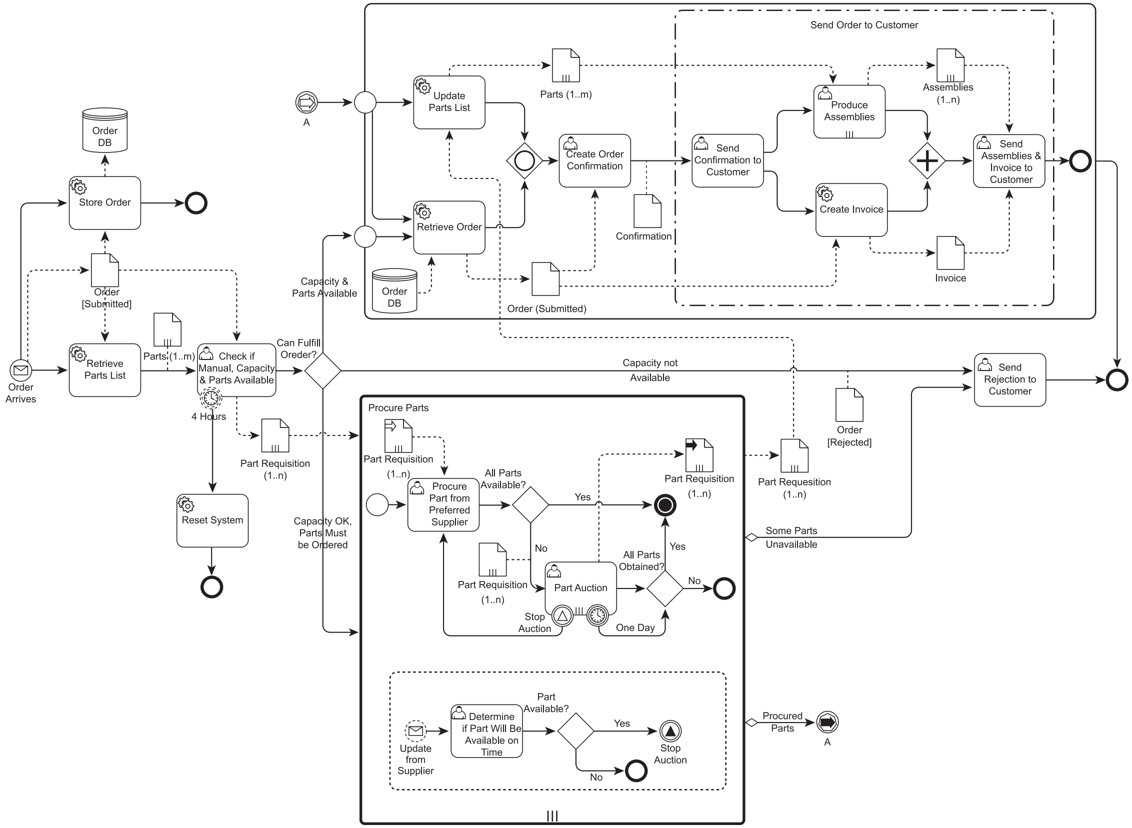


FIGURE 8

An example of a standalone Process (Orchestration) diagram.

BPMN CAVEATS

The focus of BPMN is to enhance primary process modeling capabilities. It does not attempt to model other business models, such as organization, strategic direction, business functions, rules/compliance aspects, etc. Therefore, it is vital to understand that other types of modeling done by organizations outside the primary process purposes are out of scope for BPMN, but they all fit within larger BPM solutions. Below is therefore a specification of modeling principles and concepts excluded from BPMN:

- The linking of business strategies, critical success factors, and value drivers to processes.
- The relation between organizational structures, including business competencies, capabilities, and resources to processes.
- Functional breakdowns of business functions into process tasks.
- Arrangement of business objects such as product, machine, warehouse, and so on, throughout the process models.
- Specification of information objects and thereby information flow within the process models.
- The ability to illustrate or model business measurement that is, Key Performance Indicators or Process Performance Indicators (PPIs) within the process.
- Data models, whereas BPMN shows the flow of data (messages), and the association of data artifacts to activities, it is not a data model or even a data flow diagram.
- Even though the data objects are specified within the process, real-time process monitoring in terms of Scorecards, Dashboards, and/or Cockpits.
- The support for Business Rules Modeling, in terms of business rules, rule script, flow rule, decision table, report, and thereby decision-making support.
- The ability to run process ownership gap analysis, that is, to both process and processes rules or process measurements.

So although we realize that many BPM teams wish the ability to relate process models to other vital aspects of enterprise modeling, that is, business modeling, value modeling, performance management, and enterprise architecture (e.g., business architecture, allocation/information systems architecture, and technology architecture). The scope of BPMN does not provide such modeling capabilities, but a robust BPM modeling environment could provide the linkages between the various BPM modeling domains.

THE FUTURE OF BPMN

At some point, the OMG will update BPMN to version 3.0. Although some discussions have occurred on this topic, no certain timeline exists as to when this will happen. BPMN versions 1.0 and 2.0 did not cover the wide landscapes and complexities that exist in the process-modeling domain. Thus, certain topics and capabilities could and should be addressed in BPMN 3.0. However, note that the material presented in this section is solely the opinion of the authors of this chapter. The OMG membership, which does include the authors, will determine what will be included in the next version of BPMN.

FULFILLING THE BPMN VISION

In a presentation introducing BPMN to the Business Process Management Initiative (BPMI) in April 2002, the following statement was made: "The BPMN will provide businesses with the capability of understanding their internal and external business procedures with graphical notation and will give organizations the ability to communicate these procedures in a standard manner."

Business Process types cover a wide range that is required for normal operations of most organizations. In the first two versions of BPMN, the standard has focused on more controlled, prescriptive types of internal processes as well as external processes modeled through Collaboration, Conversation, and Choreography. Nevertheless, BPMN does not yet have the built-in capabilities to easily model the entire range of process types that organizations require to run their businesses.

To fulfill this vision, BPMN eventually must be able to cover the entire range of processes that occur in the real world. This range is bounded on one side by very structured processes and on the other side by very unstructured (ad hoc) processes (see Figure 9). Potential work is available on both ends of the spectrum.

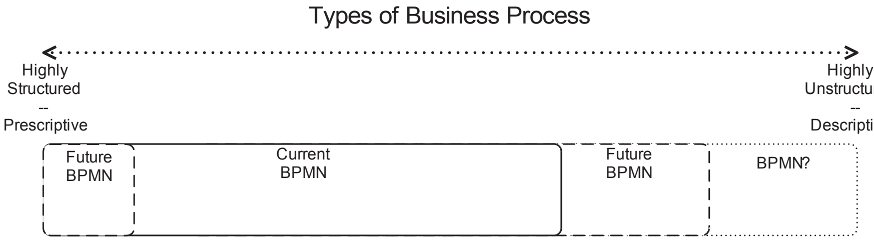


FIGURE 9

A diagram representing the range of process types that are performed by organizations.

There are different areas where future work can be applied to BPMN, including:

- Collaboration, Choreography, and Conversation.
- Metamodel changes.
- Implementation Level Modeling.
- Case Management.

We don't expect much work to be done on Collaboration, Choreography, or Conversation. More vendor/customer experience and feedback is required.

In terms of metamodel work, the following could be applied:

- Various extensions could be added.
- Separate ad hoc processes for better case management support.
- Inherent support for element substitution.
- Allowing different levels of detail or local variations of detail based on single model.

The next two sections discuss the two other major topics that could be added to the BPMN standard.

IMPLEMENTATION LEVEL MODELING

This type of modeling involves highly structured diagrams and fits on the left side of Figure 9 (above). BPMN allows multiple levels of process detail through subprocesses and tasks. But tasks are the lowest level of detail that can be modeled in BPMN. However, some BPM tools provide modelers of executable BPMN models with additional modeling capabilities for modeling the execution details of tasks, which are provided by the services that implement the tasks. These details include the sequence of steps or user interface screens in a service (sometimes called screen flow).

Thus, a process-like level of modeling exists at the service or implementation level. The layout of these models looks very similar to standard BPMN processes, but they are not, at this point, BPMN processes. They have slightly different semantics and visualizations. Figure 10 shows how a BPMN user task could be broken down to an implementation flow.

Some of the characteristics of service flow models include (for example):

- No Lanes. They exist fully within the lane of their parent task.
- Only one Start Event. This Start Event does not have a trigger. Control is always passed from the parent task.
- There are no parallel paths.
- Gateways are allowed.
- They can nest lower level service flow models.
- Semantics of the user events in a service level.
  - They do not interrupt activity in normal sense.
  - They represent a normal completion of the activity.
  - For example, through the clicking of a screen button.
  - User Event notation: User (like a User Task) or a button icon.

Given that modeling tools already exist that provide modeling at the implementation level, this type of diagram could easily be built into the BPMN standard.

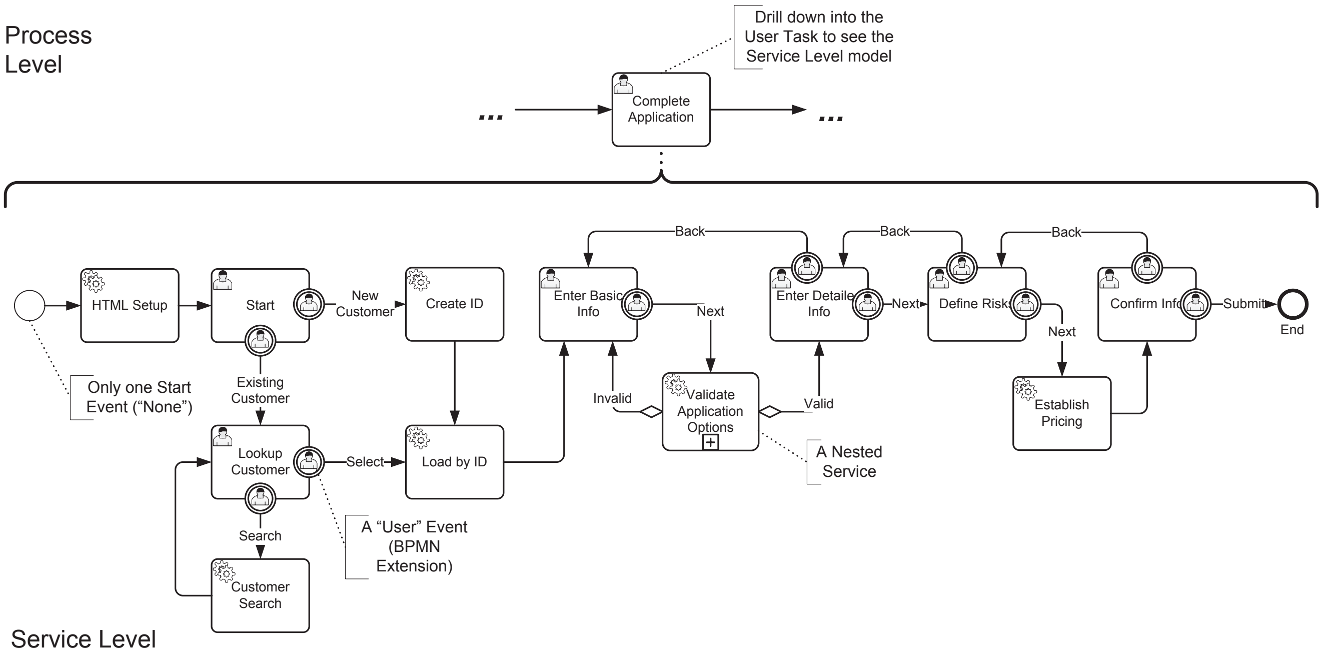


FIGURE 10

An example of an Implementation Level Diagram.

CASE MANAGEMENT MODELING

Case Management is a hot topic in BPM. This type of modeling involves highly unstructured diagrams and fits on the right side of Figure 9 (above). However, not all businesses have the same understanding of what Case Management is or how it works. Sometimes a case involves mainly straight-through prescriptive processes, with some trouble-shooting. However, most of the time a case involves mainly free-form descriptive processes.

BPMN 2.0 has incomplete support for Case Management (unstructured) Processes. BPMN mainly defines "Structured" Processes—those processes that have a well-defined sequence flow. But BPMN does provide for "Unstructured" Processes—The Ad Hoc Sub-Process. However, additional descriptive process types and behavior are required to fully handle all the unique aspects of unstructure processes.

When BPMN 1.0 was first developed, there was an understanding that descriptive processes were an important part of the process landscape. However, the initial focus of BPMN was to create a business process modeling language for business people that could also be executed by the available BPMSs. The Ad Hoc Process was included in BPMN as a placeholder that provides many of the capabilities required for modeling descriptive processes. It is expected as BPMN evolves, the Ad Hoc Process will also evolve to handle all Case Management Process requirements, which include:

- No predefined sequence flow exists.
- Activities can occur in any order or any frequency.
- But some sequence flow and data flow can be shown.

Unstructured Processes have additional requirements, such as:

- Milestones—e.g., a Case state life cycle.
- New types of events.
- For example, the Case state (life cycle) changes, document updates, and so on.
- Preconditions, dependencies.
- Activities that can be started manually or automatically.
- Activities that are optional.
- Activities that can be repeated.

Figure 11 displays some potential notational updates to BPMN elements that would allow the standard to provide the modeling of more sophisticated unstructured processes.

The OMG has been developing a Case Management Modeling Notation (See the CMMN Chapter) standard. It is focused on a specification for tools that specialize in free-form Case Management behaviors. Because both CMMN and BPMN provide modeling, the authors believe that the best course is that the OMG consolidate the two specifications in the next update to BPMN (version 3.0).

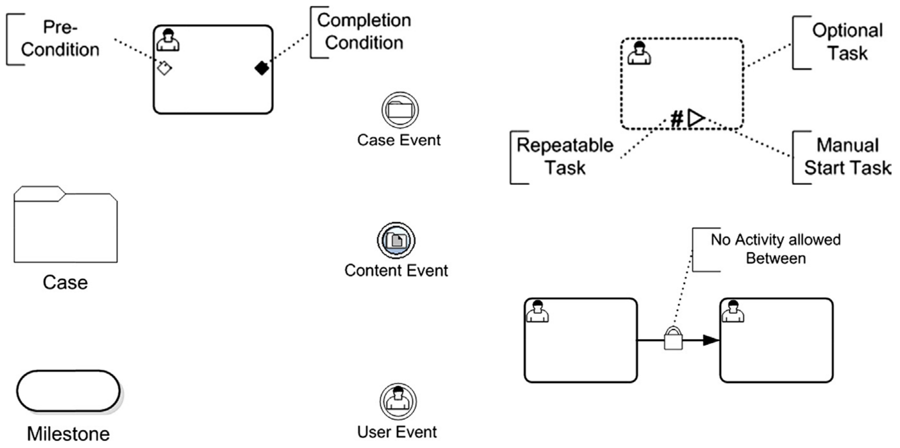


FIGURE 11

Potential notation updates for BPMN 3.0.

CONCLUSIONS

This chapter provides an overview and introduction to the Business Process Model and Notation (BPMN), what it is, and how it is used. We illustrated the primary goal of BPMN and how it provides a standard notation readily understandable by various stakeholders. Further, through its model types, BPMN provides the flexibility to integrate various views from business to technical perspectives. However, as we talk with many organizations about how BPMN can and cannot be used, we have discovered that BPMN has been, by choice, constrained to support only the concepts of modeling applicable to traditional business processes. Therefore, extended business process modeling aspects such as linking processes to business goals, the ability to do Value-Oriented Process Modeling, defining relationships between business competencies and processes, specifying measurements and reporting aspects, or defining rule sets (business, application, etc.), while all relevant, are not the focus of BPMN. It was, however, more vital to have a standard in the marketplace that enables all to have a common platform, than having the ability to do extended business process modeling. It is the start of a great journey, one that enables organizations and BPM teams around the world to analyze, design, build, and implement their processes. More will come!

End Notes

1. Object Management Group, "BMI Standard Specification 2.0.2," (2013), http://www.omg.org/spec/BPMN//2.0.2/PDF.
2. Simpson S., "An XML Representation for Crew Procedures, Final Report NASA Faculty Fellowship Program (Johnson Space Center)," (2004).

3. White S., *"Business Process Modeling Notation v1.0." for the Business Process Management Initiative (BPMI)* (May 2004).
4. Object Management Group, "BMI Standard Specification 1.1," (2008), http://www.omg.org/spec/BPMN/1.1/.
5. Object Management Group, "BMI Standard Specification 1.2," (2009), http://www.omg.org/spec/BPMN/1.2/.
6. Object Management Group, "BMI Standard Specification 2.0," (2011), http://www.omg.org/spec/BPMN/2.0/.
7. Object Management Group, "BMI Standard," http://www.omg.org/spec/BPMN/2.0.2/.