

BPMN and Business Process Management

Introduction to the New Business Process Modeling Standard

By Martin Owen and Jog Raj, Popkin Software

Executive Summary	3
Introducing BPMN	4
BPMN Enables Business Process Management (BPM).....	4
BPMI.ORG Created to Establish BPM Standards	4
Rigor of the BPMI.ORG Standards	5
BPMN Enables Modeling of B2B and B2C	5
BPMN Maps to Business Execution Languages.....	5
BPMI.ORG Works with OASIS for e-Business Standards.....	5
BPMN Models Web Services	6
A First Look at BPMN	7
Modeling Business Events.....	8
More Complex Events -- Specifying Trigger Types	8
Business Processes, Sub-Processes, and Tasks	10
Decomposing Your Processes Into Hierarchies.....	10
Easily Viewing Process Complexities	12
Modeling the Sequence Flow of a Process.....	13
Modeling Decision Points with Gateways	13
Who Does What – Pools and Lanes	15
A Pool Can Represent Many Things.....	16
Modeling B2B Message Flows	16
Enforcing B2B Rules.....	17
Black Boxes and White Boxes.....	17
Understanding How Data Is Transformed	19
Annotating the Models with Text.....	20
BPMN Conclusion.....	20
Simulating Business Processes	21
Mapping to Business Execution Languages.....	22
BPMN Maps Directly to Execution Languages	22
BPMS's Orchestrate Web Services.....	23
How BPMN Fits In with UML	24
UML Is Alien to Most Business Analysts.....	24
UML Lacks an Implementation View of Business Models	24
UML Lacks Mathematical Foundation to Map to BPEL's	25
BPMN and UML Play Together.....	25
Conclusion	26
References.....	27

Executive Summary

BPMN stands for Business Process Modeling Notation. It is the new standard for modeling business processes and web service processes, as put forth by the Business Process Management Initiative (BPMI – www.BPMI.org). BPMN is a core enabler of a new initiative in the Enterprise Architecture world called Business Process Management (BPM). Business Process Management is concerned with *managing change to improve business processes*.

BPMN consists of one diagram – called the Business Process Diagram (BPD). The BPMN Business Process Diagram has been designed to be easy to use and understand, but also provides the ability to model complex business processes. It has also been designed specifically with web services in mind. BPMN is only one of three specifications that the BPMI has developed – the other two are a Business Process Modeling Language (BPML) and a Business Process Query Language (BPQL). All have been developed using a solid mathematical foundation, which enables a BPMN Business Process Diagram to map directly to BPML, in the same way that a physical data model maps directly to Data Definition Language (DDL).

There are competing standards to BPML, chief among them is the Business Process Execution Language For Web Services (BPEL4WS) created in a joint venture by BEA, IBM, Microsoft, and others. However, BPMI has created BPMN so that it maps readily to any business process execution language.

Business Process Execution Languages themselves are run, controlled, and orchestrated on a Business Process Management System (BPMS). OASIS (www.oasis-open.org) is a not-for-profit, global consortium that drives the development, convergence and adoption of e-business standards. Both BPEL4WS and BPMI's BPML have been submitted to OASIS to become a business process execution language standard. OASIS has created a subcommittee to decide upon a standard; the outcome of this committee is called Web Services – Business Execution Language (WS-BPEL). The OASIS WS-BPEL requires the development of new BPMS technologies as well.

BPMN provides a number of advantages to modeling business processes over the Unified Modeling Language (UML). First, it offers a process flow modeling technique that is more conducive to the way business analysts model. Second, its solid mathematical foundation is expressly designed to map to business execution languages, whereas UML is not. BPMN can map to UML, and provide a solid business modeling front end to systems design with UML.

This paper provides an in-depth introduction to the new BPMN standard, illustrating how it is used to model business processes and web services. The paper also provides greater detail on how BPMN fits within BPM, BPEL's, BPMS's, UML and other new industry standards and initiatives described above.

Introducing BPMN

The Business Process Modeling Notation (BPMN) is the new standard to model business process flows and web services. Created by the Business Process Management Initiative (BPMI), the first goal of BPMN is to provide a notation that is readily understandable by all business users. This includes 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.

A second, equally important goal is to ensure that XML languages designed for the execution of business processes, such as BPEL4WS (Business Process Execution Language for Web Services) and BPML (Business Process Modeling Language), can be visually expressed with a common notation.

BPMN Enables Business Process Management (BPM)

“To improve is to change;
to be perfect is to change often.”
-- Winston Churchill

BPMN is a core enabler for a new initiative in the Enterprise Architecture world – Business Process Management (BPM).

BPM is concerned with *managing change to improve business processes*. BPM is unifying the previously distinct disciplines of Process Modeling, Simulation, Workflow, Enterprise Application Integration (EAI), and Business-to-Business (B2B) integration into a single standard.

The fact that Business Process Management is a new initiative might lead you to believe that business processes have not been managed previously. This is of course not true – many organizations have modeled and managed their business processes for years, using an eclectic mixture of tools and techniques.

These techniques have only been partially successful, or failed outright, because there has been a lack of standards and a complete lifecycle to control and guide the design and execution of business processes. Managing the process of change cannot be an ad-hoc process – it requires management to exercise control over the discovery, architecture, design, and deployment of processes. For management to understand the architecture, design, and deployment of processes, you need business modeling and business execution language standards.

BPMI.ORG Created to Establish BPM Standards

The Business Process Management Initiative (BPMI – www.bpmi.org) has been established to promote and develop the use of Business Process Management (BPM) through the use of standards for process design, deployment, execution, maintenance, and optimization of processes.

BPMI has developed, or is in the process of developing, three standards to facilitate BPM:

- BPMN, as a standard for modeling business processes,
- Business Process Modeling Language (BPML), as the standard business execution language, and
- Business Process Query Language (BPQL), a standard management interface for the deployment and execution of e-Business processes.

Rigor of the BPMI.ORG Standards

A very crucial and fundamental distinguishing feature of BPMI standards is that they have been developed with a solid mathematical foundation. The Pi-Calculus branch of Process Calculi has been used. This is a formal method of computation that forms the foundation for dynamic and mobile processes. This makes BPMI standards analogous to the mathematical foundation of relational theory that underpins relational database management systems (RDBMS's). It means that business processes designed using the BPMN standard can be manipulated directly and executable language created and made available for immediate execution. Again, this is analogous to the functionality of relational data models and the generation of SQL/DDDL statements. The business process modeling language (BPML) is designed by BPMI.org to be a Pi-Calculus-based standard description of a business process.

BPMN Enables Modeling of B2B and B2C

Unlike previous business process diagram types, the BPMN business process diagram was created with business execution languages and web services in mind. Special notations have been added to the diagram to depict message-based events and message passing between organizations.

BPMN Maps to Business Execution Languages

What's more, BPMN has been specified to map directly to the BPML standard, and any other rival business execution languages that are introduced, such as BPEL4WS, developed by BEA, IBM, Microsoft, and others.

BPMI.ORG Works with OASIS for e-Business Standards

OASIS (www.oasis-open.org) is a not-for-profit, global consortium that drives the development, convergence and adoption of e-business standards. OASIS produces worldwide standards for security, web services, XML conformance, business transactions, electronic publishing, topic maps and interoperability within and between marketplaces.

Both BPML (from BPMI.ORG), and BPEL4WS (from Microsoft, IBM, and others) have been submitted to OASIS, which has formed a technical committee to create a Business Process Execution language standard. The output from this committee is currently called Web Services – Business Process Execution Language (WS-BPEL). BPML's current status is that it is submitted as an influencing specification to the WS-BPEL committee.

BPMN Models Web Services

Making web services work is a four-stage process, as follows:

1. Design the processes using BPMN.
2. Simulate the processes and modify them for efficiency.
3. Make the services available by publishing them using a Business Process Execution language.
4. Orchestrate the web services into end-to-end business flows by assembling them and coordinating their behavior. Business Process Management Systems (BPMS) are employed for this stage.

The OASIS WS-BPEL requires the development of new BPMS technologies where these functionalities will reside and be executed from.

We will examine each of the steps above in this whitepaper, beginning with a first look at the new BPMN standard. Finally, we'll take a look at BPMN in relation to a popular industry standard for systems analysis, the Unified Modeling Language (UML).

A First Look at BPMN

BPMN specifies a single business process diagram, called the Business Process Diagram (BPD). This diagram was designed to do two things well. First, it is easy to use and understand. You can use it to quickly and easily model business processes, and it is easily understandable by non-technical users (usually management). Second, it offers the expressiveness to model very complex business processes, and can be naturally mapped to business execution languages.

To model a business process flow, you simply model the events that occur to start a process, the processes that get performed, and the end results of the process flow. Business decisions and branching of flows is modeled using gateways. A gateway is similar to a decision symbol in a flowchart.

Furthermore, a process in the flow can contain sub-processes, which can be graphically shown by another Business Process Diagram connected via a hyperlink to a process symbol. If a process is not decomposed by sub-processes, it is considered a task – the lowest-level process. A '+' mark in the process symbol denotes that the process is decomposed; if it doesn't have a '+' mark, it is a task.

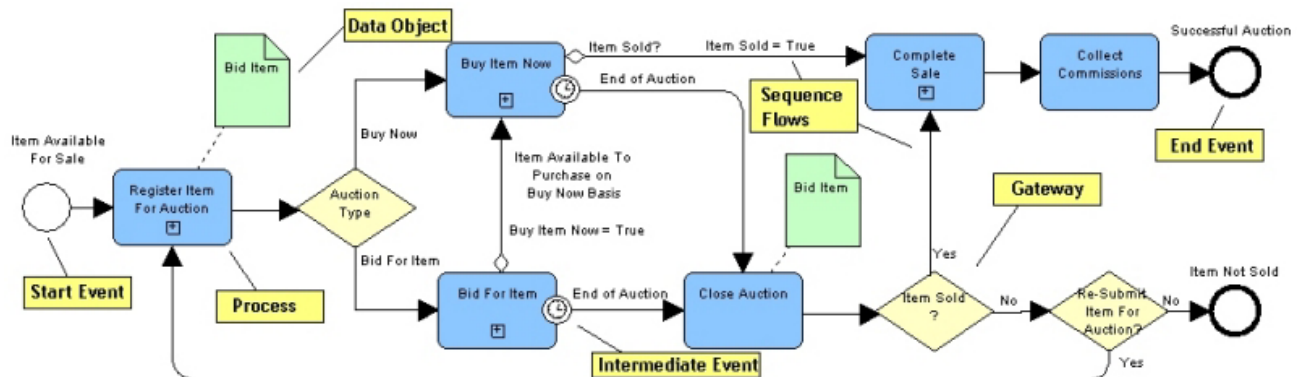


Figure 1. Simple BPMN Business Process Diagram for an on-line auction system.

As you drive further into business analysis, you can specify 'who does what' by placing the events and processes into shaded areas called pools that denote who is performing a process. You can further partition a pool into lanes. A pool typically represents an organization and a lane typically represents a department within that organization (although you may make them represent other things such as functions, applications, and systems).

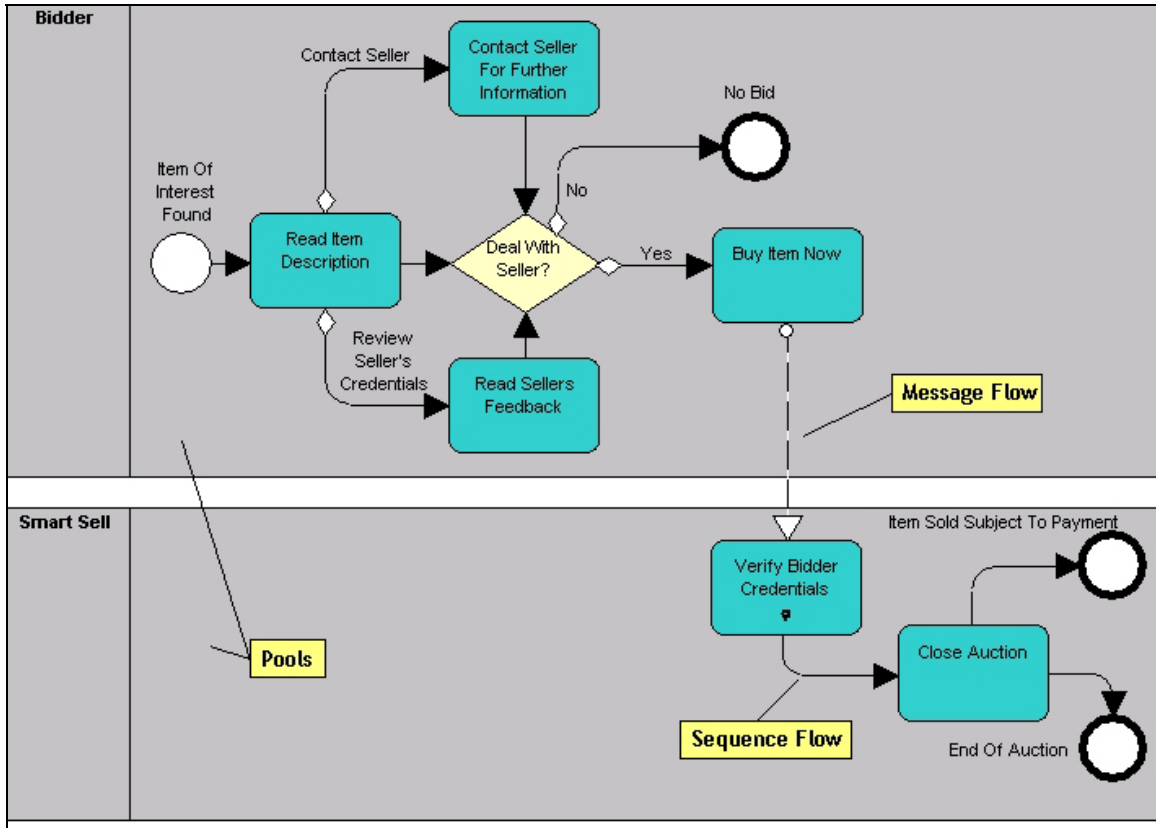


Figure 2. BPMN Business Process Diagram with processes drawn in pools.

We take a closer look at each of the elements that can be drawn on a BPMN Business Process diagram in the next section.

Modeling Business Events

During business process modeling, you model the events that happen in the business, and show how they affect process flows. An event either kicks off a process flow, or happens during a process flow, or ends a process flow. BPMN provides a distinct notation for each of these types of events, shown in the table below.

Table 1: Basic event types in BPMN and their notations.
















Start Event		Intermediate Event		End Event	
Start Event	Starts a process flow.	Event	Happens during the course of a process flow.	End Event	Ends a process flow.





More Complex Events -- Specifying Trigger Types

When you model more complex process flows, such as B2B web services, you need to model more complex business events, such as messages, timers, business rules, and error conditions. BPMN enables you to specify the trigger type of the event, and denote it with a representative icon, as specified in Table 2.

Specifying a trigger type to an event puts certain constraints on the process flow that you are modeling, which are explained in the table. For example, a timer cannot end a process flow. You can only draw message flows from and to message events. These types of modeling rules, which are actually kinds of business rules, should be enforced automatically by the modeling tool providing support for BPMN.

Table 2: Event Trigger Types.

Start Events	Intermediate Events	End Events	Description
Start Message 	Message 	End Message 	A start message arrives from a participant and triggers the start of the process, or continues the process in the case of an intermediate event. An end message denotes a message generated at the end of a process.
Start Timer 	Timer 	A Timer cannot be an End Event.	A specific time or cycle (for example every Monday at 9am) can be set to trigger the start of the process, or continue the process in the case of an intermediate event.
Start Rule 	Rule 	A Rule cannot be an End Event.	Triggers when the conditions for a rule become true, such as "Stock price changes by more than 10% since opening."
Start Link 	Link 	End Link 	A link is a mechanism for connecting the end event of one process flow to the start event of another process flow.
Start Multiple 	Multiple 	End Multiple 	For a start multiple event, there are multiple ways of triggering the process, or continuing the process in the case of the intermediate event. Only one of them is required. The attributes of the event define which of the other types of triggers apply. For end multiple, there are multiple consequences of ending the process, all of which will occur (for example, multiple messages sent).
An Exception cannot be a Start event.	Exception 	End Exception 	An end exception event informs the process engine that a named error should be generated. This error will be caught by an intermediate exception event.

Start Events	Intermediate Events	End Events	Description
A Compensation event cannot be a Start event.	Compensation 	End Compensation 	An end compensation event informs the process engine that a compensation is necessary. This compensation identifier is used by an intermediate event when the process is rolling back.
An End event cannot be a Start event.	An End event cannot be an Intermediate event.	End Cancel 	An end event means that the user has decided to cancel the process. The process is ended with normal event handling.
An End Kill event cannot be a Start event.	An End Kill event cannot be a Intermediate event.	End Kill 	An end kill event means that there is a fatal error and that all activities in the process should be immediately ended. The process is ended without compensation or event handling.

Oftentimes an event happens while a particular process is being performed, causing an interrupt to the process, and triggering a new process to be performed. Or, a process will complete, causing an event to start, and a new process to be performed. You can model these intermediate events by placing an event symbol directly on the process that it is associated with. In Figure 3, you can see a message event being triggered when a process Check Inbox completes, causing a message Password Request to be sent to the Send Password process. This type of BPMN notation makes it crystal clear to the reader that the Check Inbox process generates a message event that sends a message to another process.



Figure 3. A message event is triggered at the end of the *Check Inbox* process, sending the *Password Request* message to the *Send Password* process.

Business Processes, Sub-Processes, and Tasks

At the core of business process modeling are the processes themselves. There are three types of processes – the **process**, the **sub-process**, and the **task**. Each is graphically depicted by the same rounded rectangular symbol; the use of different nouns simply reflects the hierarchical relationships between them.

Decomposing Your Processes Into Hierarchies

A **process** is a network of 'doing things'. You draw it as a rounded rectangle on your top-level BPMN Business Process diagram. You can specify the inner details of a

process by creating or attaching another Business Process diagram to it. The sub-diagram is considered a 'child' diagram. A process that has a child diagram gets a '+' marker in its body.

Graphically showing the details of a process with another Business Process diagram is considered 'decomposing' the process. You can continue to decompose a process without any restriction -- creating a child diagram for a process, and child diagrams for the processes on the first child diagram, and so forth. Processes that you draw on 'child' diagrams are considered **sub-processes**. The lowest-level process, which you do not decompose further, is considered a **task**.

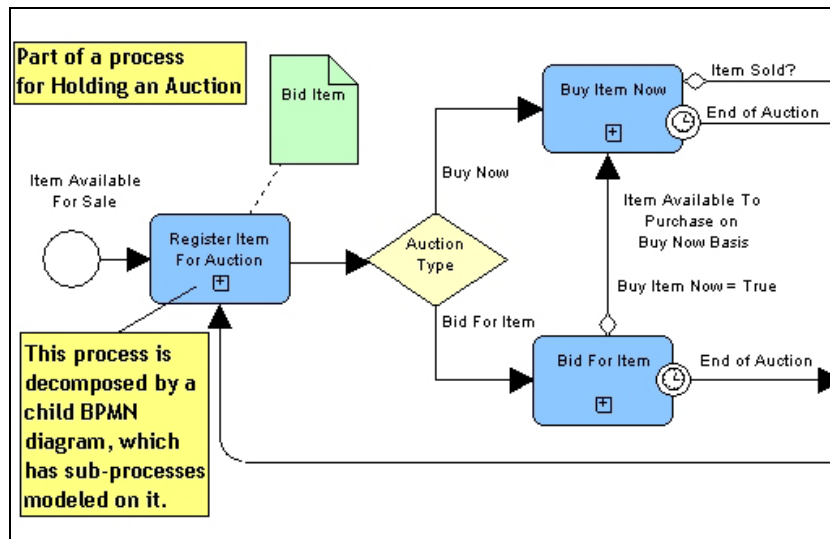


Figure 4. Part of a BPMN Business Process diagram for an on-line auction system.

Figure 4 shows a BPMN business process diagram on which the process Register Item for Auction has been modeled. The '+' mark in the process's body tells you that there is at least one 'child' business process diagram hyperlinked to this process, and on that diagram is a graphical depiction of the details of this process.

Figure 5 shows part of the 'child' BPMN Business Process diagram to the *Register Item for Auction* process. Since they are on a 'child' diagram, the processes are considered sub-processes. Processes on this diagram that are not further decomposed (no '+' mark in their center) are considered tasks. As you can see, it is easy to pick out a task on a diagram – simply those rounded rectangles without a '+' mark at their center.

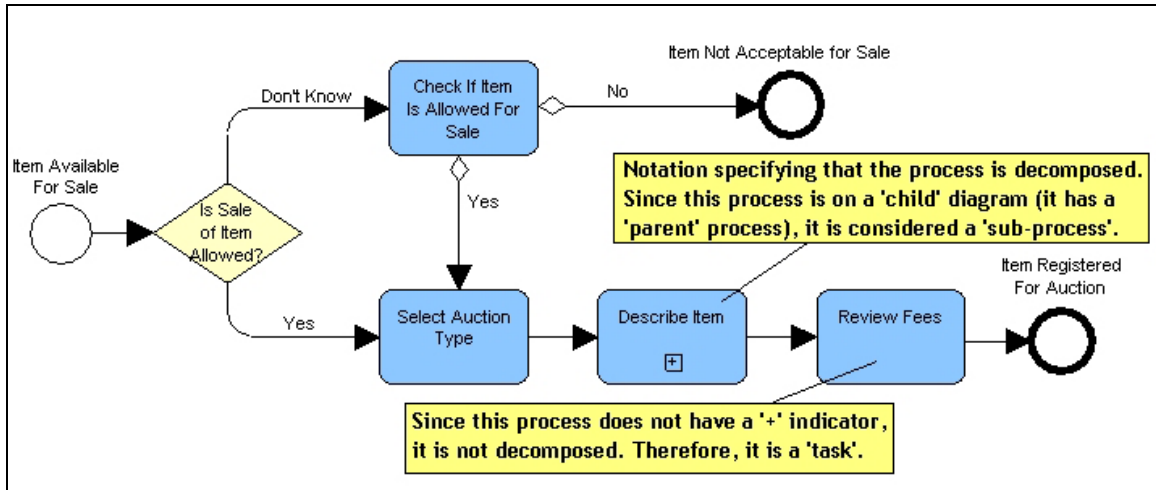


Figure 5. Sub-processes and tasks.

Easily Viewing Process Complexities

Again, the BPMN diagram is designed to be easily understood by viewers. To help readers understand the complexities of processes, you may graphically show a thumbnail of a 'child' process flow on a process symbol itself. In the modeling tool, this is done by clicking on the '+' mark in the process symbol's center, changing it to a '-' mark, and presenting the thumbnail sketch. In this way, you can view a BPMN Business Process diagram and quickly see which processes are complex, and that decompose to further levels.

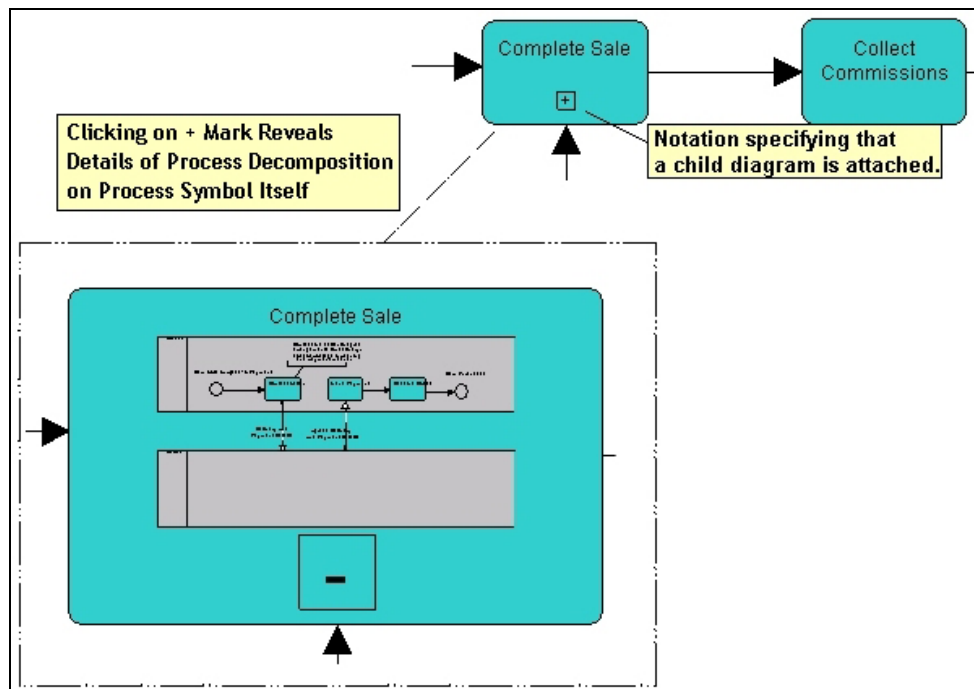


Figure 7. Showing a thumbnail sketch of the child diagram on a process.

Modeling the Sequence Flow of a Process

To show the order of execution of processes, you connect them with a Sequence Flow. A Sequence Flow is drawn as a line with a filled-in arrowhead (see Figures 4 and 5).

A Sequence Flow is used to show the sequence of processes *in* an organization or department. So if you have added pools or lanes to your diagram, you use Sequence Flow lines to connect events, processes, and gateways placed within the pools or lanes.

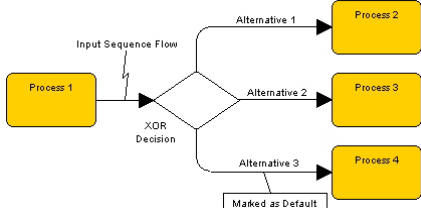
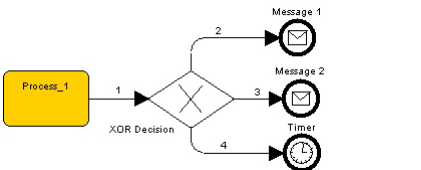
BPMN makes a second flow line – the Message Flow – available to model ordering of processes *between* organizations or departments (in other words, between pools). We will talk about the Message Flow later in this paper.

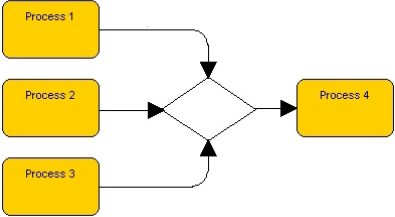
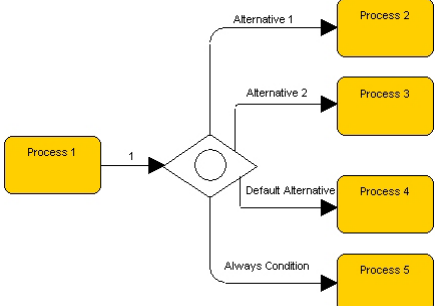
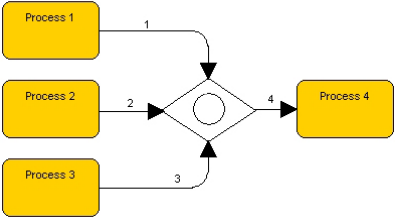
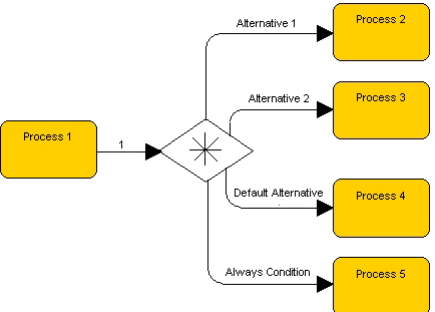
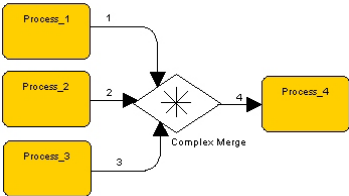
Modeling Decision Points with Gateways

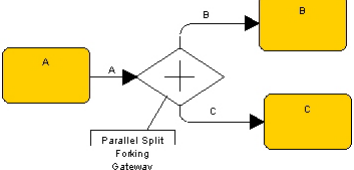
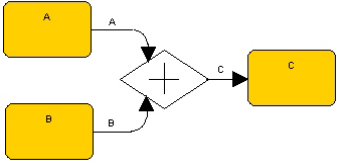
Decisions, merges, forks, and joins in the process flow are modeled with a **gateway** symbol. A gateway can be thought of as a question that is asked at a point in the process flow. The question has a defined set of alternative answers, which are in effect gates.

You may set the stereotype of a gateway, and thus change the logic specified by it, and the symbol representing it, as described in Table 3.

Table 3. Types of Gateways and Associated Symbols

Gateway Stereotype	Explanation
<p data-bbox="237 1167 613 1199">Exclusive Decision (XOR)</p> <p data-bbox="237 1230 591 1262">Data-Based XOR Decision:</p>  <p data-bbox="237 1482 604 1514">Event-Based XOR Decision:</p> 	<p data-bbox="724 1167 1386 1419">XOR gateways are used to model data-based or event-based decisions. Data-based XOR decisions are the most common XOR gateways used. A data token traverses the Process Flow and arrives at the XOR gateway. The path that it flows out on is chosen based on condition expressions for each gate of the gateway. It can only go out on one flow.</p> <p data-bbox="724 1455 1386 1707">Event-based gateways are a recent development in business process management (BPM). An event-based XOR gateway represents a branching point where the alternatives are based on an event that occurs at that point in the process flow. A specific event, usually the receipt of a message, determines which of the paths will be taken.</p> <p data-bbox="724 1743 1386 1898">For example, you can model a process flow wherein the system waits for a response from a customer. The customer’s response will either be a Yes message or a No message, and that determines which path is taken.</p>

Gateway Stereotype	Explanation
<p>Exclusive Merge (XOR)</p> 	<p>XOR gateways are used to model data-based or event-based merges. Exclusive means only one of many inputs is chosen to be output from the gate.</p>
<p>Inclusive OR Decision</p> 	<p>Inclusive (think 'including') means one or more of the outgoing Sequence Flows from the decision may be taken. There cannot be zero output flows -- you must specify a default flow.</p>
<p>Inclusive OR Merge</p> 	<p>Inclusive (think 'including') means that the process flow continues when the first input signal (a Token) arrives from any of the set of input Sequence Flows. If other signals subsequently arrive from the other input Sequence Flows, they are not used.</p>
<p>Complex Decision</p> 	<p>You specify a complex flow condition that references outgoing Sequence Flow names. The expression determines which output flow is taken.</p>
<p>Complex Merge</p> 	<p>You specify a complex flow condition which references incoming Sequence Flow names and/or process data that is coming into the gateway. The expression determines when the task starts.</p>

Gateway Stereotype	Explanation
<p>Parallel Forking (AND)</p> 	<p>A Parallel gateway is also called an AND gateway. All Sequence Flows drawn out of the AND gateway are taken.</p>
<p>Parallel Joining (AND)</p> 	<p>The AND gate must receive an input signal (a Token) from all input Sequence Flows for the output flow to be taken. The process flow waits for all signals to arrive at the AND gateway before it can continue.</p>

Who Does What – Pools and Lanes

As you progress in modeling business flows, you take the processes, events, and gateways of the business process diagram and place them in pools or lanes. A pool is drawn as a rectangular region drawn horizontally across the diagram or vertically down it. A lane is a sub-partition within a pool and extends the entire length of the pool. Typically, a pool represents an organization, and a lane represents a department within that organization. By taking processes and placing them in pools or lanes, you are specifying **who** does **what**, for events you specify **where** they occur, and for gateways you specify **where decisions** are made, or **who** makes them.

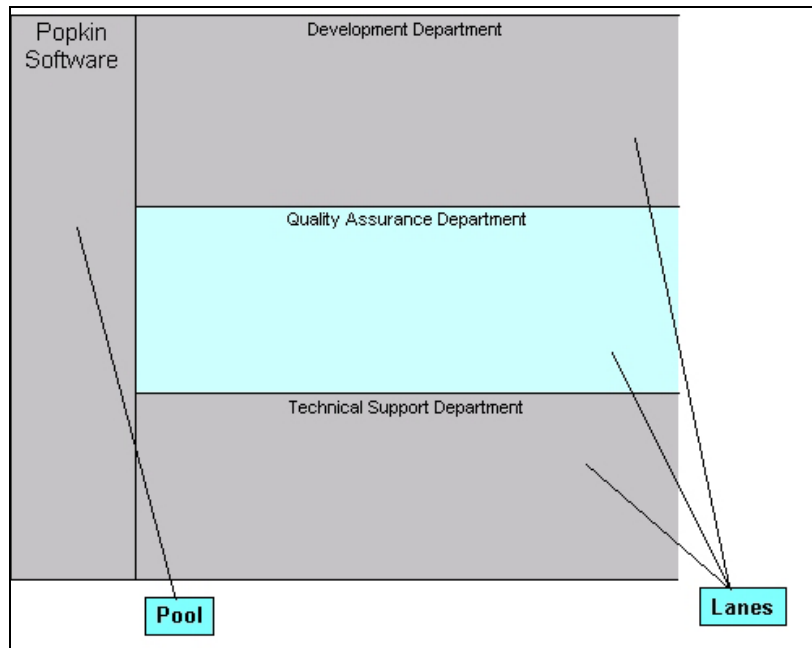


Figure 8. Pools and Lanes show who performs processes or makes decisions, or where events occur.

The analogy between this representation and swimming pools is a useful one. You can imagine a process swimming down a lane, and changing lanes as need be to perform an activity, within a pool. The pool can be considered a 'pool' of resources. There are occasions when the process needs to jump to another pool, because it has different resources needed to complete the activity.

This is particularly apt where there is a need to describe B2B processes, where different organizations pass messages among one another to perform an activity. This is described in the next section, *Modeling B2B Message Flows*.

A Pool Can Represent Many Things

A pool can represent other things besides an **organization**, such as a **function** (something that the organization performs, like Marketing or Sales or Training), an **application** (or computer software program), a **location** (a physical location in the company), a **class** (a software module in an object-oriented computer software program), or an **entity** (representing a logical table in a database). It can only represent one thing, but that thing comes from this 'heterogeneous list' of different types of things.

Modeling B2B Message Flows

As mentioned previously, one goal of the BPMN business process diagram is to enable modeling of B2B messaging. To this end, the BPMN Business Process diagram offers the ability to model message flows. Traditional business process diagrams enable the modeling of sequential process flows -- from starting events to ending results. The BPMN business process diagram augments the Sequence Flow line with a Message Flow line, so that you can model people or machines sending messages to one another – an important part of depicting and understanding business-to-business and business-to-consumer processes.

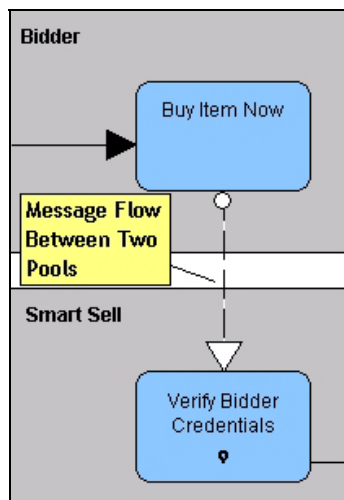


Figure 9. Messages Flows are used to model message passing between organizations or applications.

Enforcing B2B Rules

BPMN specifies certain rules for modeling message flows and sequence flows. Sequence Flows can only be drawn among events, processes, and gateways within the same pool. Message Flows can only be drawn between events, processes, or gateways that exist in different pools – since messages are only passed between different organizations or applications, and so forth.

BPMN suggests that these rules be enforced by the modeling tool providing BPMN support. System Architect enforces these drawing rules by presenting a ghostbuster symbol and disallowing connections between the wrong elements; it only allows connections to proper model elements. This helps you prevent introducing errors or logical inconsistencies into B2B systems as you model.

Black Boxes and White Boxes

There are times when you are modeling that you don't care how a process is performed in a company. It may be another company or a customer that is outside your scope; you have no control over it. You don't care how the company creates a message; you only care that the message has been delivered to you and contains information that you can use. Or you don't care what a company does with a message that you deliver to it – you trust that it does the right thing with it.

You can treat the company (or application, function, and so forth) as a 'black box' – only draw Message Flows to or from the pool representing it, and not show any details inside the pool. This is in contrast to the pools that you model processes in, which can be considered 'white boxes' – you can see into them and examine their details.

Figure 10 shows messages drawn from and to processes in a 'white box' pool (representing the Bidder in our example) to and from a 'black box' pool (representing the Seller).

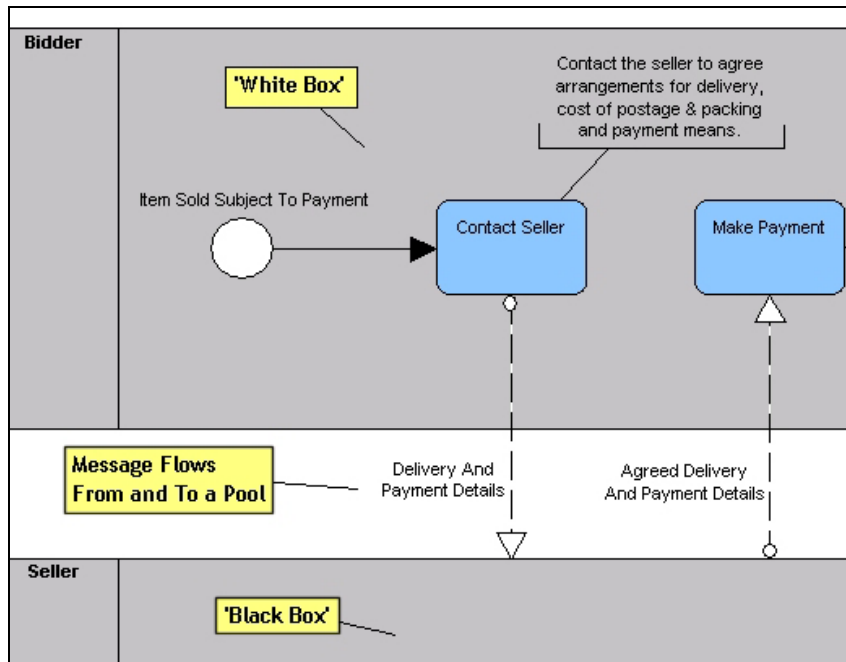


Figure 10. Message flows drawn between a 'white box' and a 'black box'.

You can also show messages passed between two 'black boxes', as shown in Figure 11.

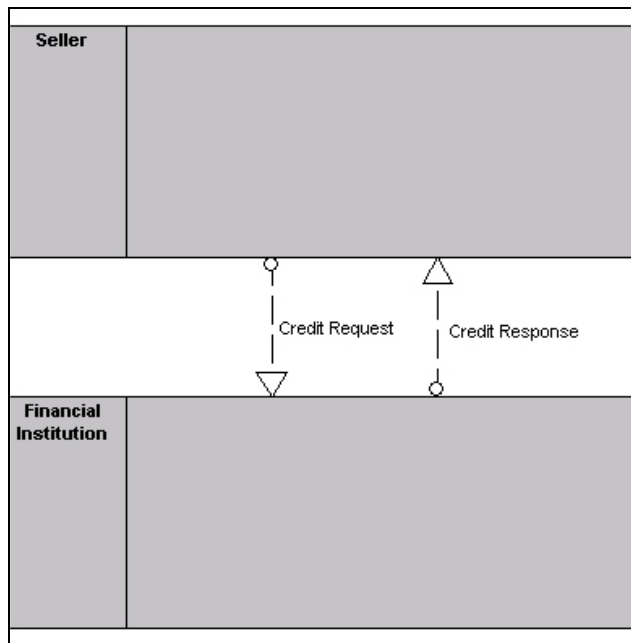


Figure 11. Message Flows drawn between two 'black boxes'.

Understanding How Data Is Transformed

Remember that processes transform data in your organization. An order request causes an order to get generated. When the product is shipped to the customer, the order is fulfilled. A bad credit card might cause the order to be cancelled. A customer might update their account information with a new credit card number or address.

You can model how data is transformed during a process flow by depicting **data objects** on the BPMN business process diagram. Data objects are artifacts that may represent many different types of electronic or physical items. Since they represent data, they are defined by a combination of one or more entities (corresponding to a database tables) or classes (corresponding to object-oriented software modules that contain data).

A data object is depicted by a rectangle with a folded-over upper-right corner, making it resemble a document.

Modeling data objects is optional – they do not have any direct affect on the process flow. They simply provide information about what the process flow does. You can either attach a data object to a Sequence or Message flow with a dashed line (shown in Figure 12), or draw association lines to and from data objects and processes (shown in Figure 13), creating data flows within the process flow.

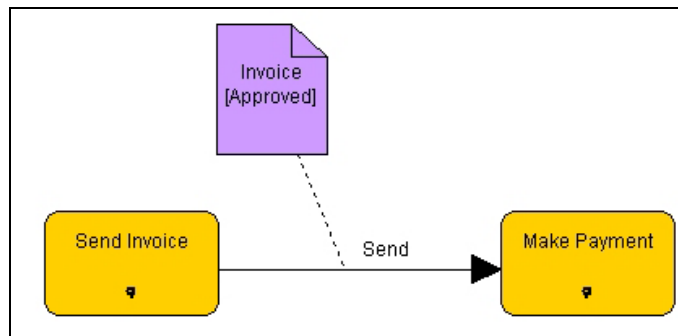


Figure 12. Attaching a data object to a Sequence Flow to show .

You may specify the state of the data object in brackets below the name of the data object. This provides information on how it is being transformed during the process. In Figure 12, you can see that Invoice is being approved when it is sent from the Send Invoice process to the Make payment process. In Figure 13, you can see that the process Approve Purchase Order changes the state of a Purchase Order data object from Complete to Approved.

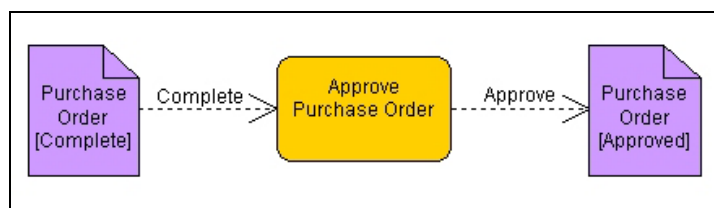


Figure 13. Drawing association lines between data objects and a process.

Annotating the Models with Text

A picture is worth a thousand words, so the saying goes. Conversely, sometimes a picture isn't enough – you need words to describe the nuances of something that a picture cannot do justice to. Thus BPMN provides you with a textual annotation that can be affixed to any model element, so that you may describe extra details about the element in good old-fashioned words.

You may use Text Annotations on all model elements of the BPMN Business process diagram. Text Annotations are displayed within an open rectangle, attached to the symbol by a straight line, as shown in Figure 14.

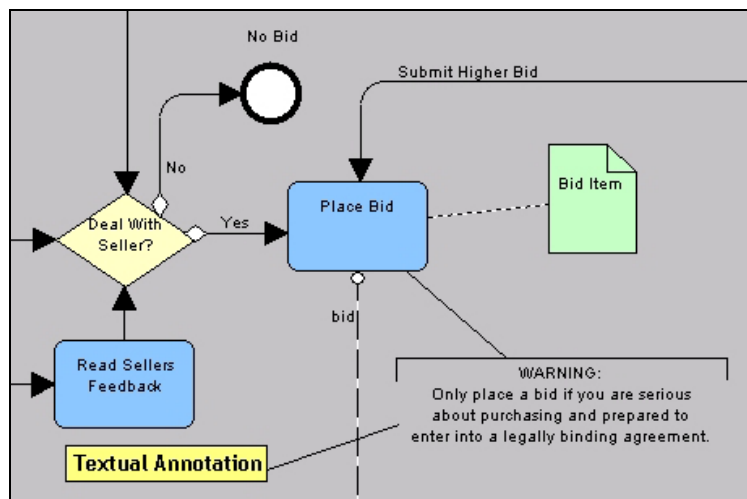


Figure 14. Textual Annotation; for circumstances where pictures aren't enough.

BPMN Conclusion

In conclusion, BPMN is poised to become the new standard to model business processes and web services. It is designed to enable you to easily model typical business processes, and offers the capability to model complex business processes, including the message passing of web services.

Simulating Business Processes

A model described using BPMN is a logical description of how the business operates, from which business process languages can be generated. However for optimal results this approach should be used hand-in-hand with business process simulation.

Simulation is a powerful technique available to business analysts to analyze their models prior to their realization. A model, when simulated, mimics the operations of the business, by stepping through the events in compressed time while displaying an animated picture of the flow.

Because simulation software keeps track of statistics about model elements, performance metrics can be evaluated by analyzing the model output data. This enables you to avoid expensive mistakes by thoroughly reviewing the efficiency of a business model before actually implementing it.

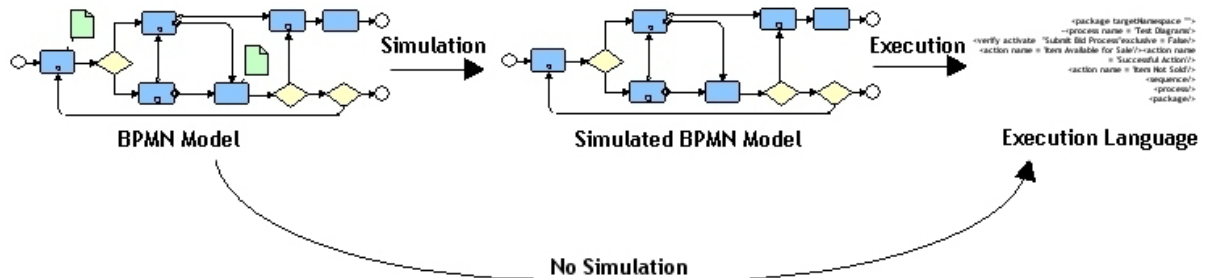


Figure 15. Simulating and executing a BPMN business process diagram.

Mapping to Business Execution Languages

Business Process Modeling Languages are XML-based meta-languages used as a means of modeling business processes in an XML format. Many different Business Process Modeling languages have been proposed. Most of them use XML and build on top of the Web Service Description Language (WSDL) by the W3C standards body. A major flaw of WSDL is that the language mixes static interface description and binding information to certain communication protocols.

The new breed of Business Process Modeling Languages include the Business Process Execution Language for Web Services (BPEL4WS), created through a joint effort of BEA, IBM, and Microsoft, and the Business Process Modeling Language (BPML) created by BPMI.ORG.

Both of the standards provide specifications for:

- Dataflow
- Messages
- Events
- Business Rules
- Exceptions
- Transactions (Distributed, Compensating, Synchronous, Asynchronous)

BPMN Maps Directly to Execution Languages

As mentioned in the introduction, BPMI has developed BPMN with a solid mathematical foundation – the Pi-Calculus branch of Process Calculi has been used. This is a formal method of computation that forms the foundation for dynamic and mobile processes. It means that business processes designed using the BPMN standard can be directly mapped to any business modeling executable languages for immediate execution. Again, this is analogous to the functionality of relational data models and the generation of SQL/DDDL statements.

BPMS's Orchestrate Web Services

The internet is a heterogeneous environment of many different platforms and applications. In an end-to-end value chain, organizations and individuals want to pick best-of-breed components that provide the value chain with best value. The applications and services need to work together harmoniously. This is one of the driving forces for the standardization of web services.

As we mentioned at the beginning of this paper, making web services work is a four stage process – designing the processes with BPMN, verifying them for efficiency with simulation, making them available by publishing them using a business process execution language, and orchestrating and coordinating them using a Business Process Management System (BPMS).

BPMS's offer the ability to transform the distinct disciplines of workflow, EAI, and B2B from a complex high-end solution practiced by a few highly skilled practitioner consultants into an open solution accessible to the masses of developers producing new types of agile, loosely coupled applications. BPMS's orchestrate participants (applications, people, partners) into executable, end-to-end processes and close the gap between strategy and business execution.

Key companies developing BPMS's include IBM, BEA Systems, Vitria, Intalio, FileNet, Fuego, and Collaxa.

How BPMN Fits In with UML

The advent of BPMN, BPML's, and BPMS's does not render obsolete the need for systems development, such as that performed using the Unified Modeling Language (UML). Systems development still has an important role to play in the overall enterprise architecture process.

UML is a language that helps developers specify, visualize, and document models of software systems. It is very much targeted at system architects and software engineers. It has been developed as a means to streamline the software development process, from architecture design to application implementation for use by a technical audience.

BPMN is targeted at business analysts, system architects, and software engineers. It has been developed as a way to streamline the overall business lifecycle development process from process design – performed by a business audience.

UML Is Alien to Most Business Analysts

UML defines a number of diagrams that fall into one of three categories that describe:

1. Static application structure
2. Dynamic behavior
3. Management and organization of software solutions

Of these categories it is the dynamic behavior diagrams that are often used for modeling business processes, such as the UML Activity diagram and Use Case diagram. BPMN is related to UML in the sense that it defines a graphical notation for business processes that is similar to UML behavior diagrams. However, BPMN and UML have very different approaches to business process modeling.

UML offers an object-oriented approach to the modeling of applications, while BPMN takes a process-centric approach. Most UML methods ask you to find the objects first using static structure diagrams, and then ask you to build dynamic behavior diagrams to show how objects interact. As a way to model, this method is alien to most business analysts.

BPMN offers a process-centric approach that is more natural and intuitive for the business analyst to use. With BPMN, control and message flows of processes are modeled first. An object model for the process is defined implicitly rather than explicitly. BPMN also offers you the option of explicitly modeling business objects that may be exposed through business services in your process flows.

UML Lacks an Implementation View of Business Models

UML is an assemblage of diagrams that are the results of the collective best practices of the various founding practitioners. Unfortunately, what this means is that the diagrams are an aggregation that have not been specifically designed to work with each other. As a consequence, developers can only model part of their applications with UML; the detailed implementation level is not covered.

In contrast, BPMN defines a single type of diagram that has multiple views derived from the same underlying process execution meta-model. The natural result of this is that implementation in a business process execution language merely becomes another logical view of the process.

UML Lacks Mathematical Foundation to Map to BPEL's

Finally, UML does not define any execution meta-model for business processes modeled with it. Instead, any execution meta-model must be defined using Model Driven Architecture (MDA). BPMN is based on BPML's process execution meta-model and so does not require any additional steps for the modeling fully executable processes.

BPMN and UML Play Together

It is anticipated that BPMN and UML will co-exist. There will be technical users that do not intend to use BPML as the final means of deployment, who will continue to use UML. Figure 16 shows that BPMN can be used to drive solutions that will run directly on a BPMS or be used as a business analysis front-end for subsequent systems development using UML. In this scenario UML users would regard business processes merely as another type of component.

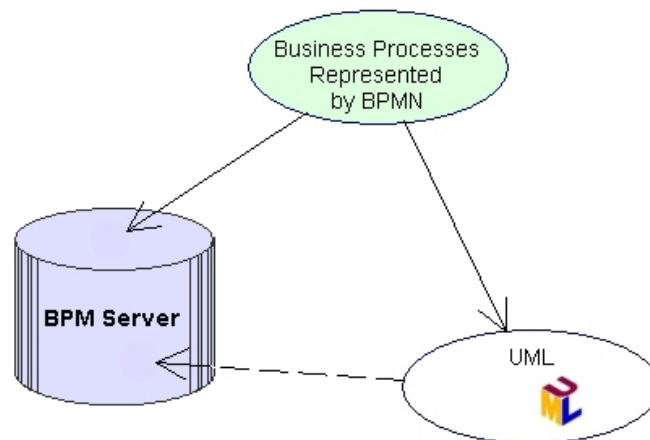


Figure 16. BPMN and UML are both used to design the business processes and applications that run on a Business Process Management Server (BPMS).

Conclusion

Popkin Software believes that a standard modeling notation used amongst modeling vendors, business analysts, and the IT community is fundamental to the management of business processes and the alignment of business with information technology (IT) architectures.

Business process execution is an alternative development paradigm to traditional development techniques. Traditional development will not disappear and in fact, is fundamental to supporting the implementation of business process management servers (BPMS's).

Modeling with BPMN is essential to understanding and communicating business processes across the enterprise. BPMN provides a powerful augmentation to other modeling techniques such as relational data modeling, application and system design with UML, XML schema design, and network architecture design. These modeling techniques enable a firm to understand and design its enterprise architecture, which enables it to react to change quicker, and in a safer manner.

Popkin Software is continuing to lead and embrace industry standards in this field. The simplification and demystification of web services and their usage in the enterprise is key to allowing our customers to succeed in their market places.

For more information on BPMN, or to try using BPMN, visit our website (www.popkin.com) for a free trial version of System Architect.

References

BPMN vs. UML, By Ismaël Ghalimi, Chief Strategy Officer, Intalio (www.intalio.com)
— September 2002