



Bruce Silver Associates

Independent Expertise in BPM

BPMN and the Business Process Expert

Summary: BPMN has become the standard language of the Business Process Expert, usable for descriptive process modeling, simulation analysis, and even executable implementation design of end-to-end business processes. BPMN extends the familiar swimlane flowchart paradigm with events, the key to incorporating exceptions into process models and mapping to today's SOA middleware. First of six parts.

Author: Bruce Silver

Company: Bruce Silver Associates

Created on: 30 October 2007

Author Bio



Dr Bruce Silver is an independent industry analyst and consultant focused on business process management software. He provides training on process modeling with BPMN through <http://www.bpmessentials.com>, the BPM Institute, and Gartner conferences, and is the author of The BPMS Report series of product evaluations available from www.bpminstitute.org.

Business process management (BPM) is an emerging discipline that looks at the enterprise in a radically new way. Instead of trying to automate and optimize individual functional units, like sales, supply chain, and customer service, in isolation, BPM views your company from the perspective of end-to-end *cross-functional processes* – exactly the way your customers and trading partners see you. At the same time, new BPM tools have emerged that let you model, automate, measure, and optimize the business from such an end-to-end process perspective. These tools straddle the traditional business/IT divide, and are elevating the importance of a new role in the organization, the Business Process Expert.

The Business Process Expert is neither a traditional developer nor a traditional business analyst, but is able to apply the concepts, metrics, and performance objectives of business in the analysis, design, and optimization of IT implementations capable of executing and monitoring cross-functional processes. Such a role demands a new common language that bridges the worlds of business and IT as well, and now we have one: the *Business Process Modeling Notation* (BPMN), a standard from OMG.

BPMN is simple enough to be readily understandable by business, yet rich enough to support executable implementation – without changing the underlying metamodel! As a result, it has become the de facto standard or announced future direction of BPM Suite vendors ranging from Lombardi, Savvion, and Appian, to TIBCO, Oracle, IBM, and – yes – SAP. Thus, understanding how to model processes effectively using BPMN is becoming a must-have skill for the Business Process Expert.

Bruce Silver Associates

BPMS Watch www.brsilver.com/wordpress

BPMN Training <http://www.bpmessentials.com/>

500 Bear Valley Road, Aptos CA 95003

Tel: 831.685.8803 Fax: 831.603.3424 E-mail: bruce@brsilver.com

Over the next several weeks, this six-part series will explain to the BPX community the unique features and benefits of BPMN; the notation and its underlying semantics; best practices for effective modeling with BPMN; understanding BPMN events; useful patterns for modeling exceptions; and more. In this first installment, we'll look at exactly why BPMN is vital to the Business Process Expert.

What is BPMN?

BPMN is a graphical notation for modeling business processes. A BPMN model is essentially a diagram of the process flow, but a process model is more than a drawing. Each diagram element, called a *flow object*, carries a specific meaning defined in the BPMN specification and subject to its rules. That meaning of the process diagram is intelligible to the business user, but the semantics are rich and precise enough to serve as a foundation for an executable implementation of the model. That combination sounds simple, even obvious, but it's something that we've never had before.

It also allows the process model to be more than a graphical "business requirements" document. In an increasing number of BPM suites, the model – supported by IT-added implementation properties for each flow object – actually becomes executable! In the BPMN model, each activity in the process flow is defined "abstractly" by its name, its performer swimlane, and where it fits sequentially (and hierarchically) in the flow.

Technical details required for the implementation, such as business rules, user interfaces, or service call parameters, are typically specified *outside* of the BPMN standard but bound to diagram elements using tool-specific properties. (BPMN also provides a few standard attributes for this purpose, but most tools ignore them.) Thus, a BPMN model can serve as a *business view* of an end-to-end process solution that remains valid throughout the implementation lifecycle.

BPMN is a *vendor-independent standard*, maintained by the Object Management Group (OMG). That sets it apart from a long history of business process analysis (BPA) tools based on vendor-proprietary process modeling notations. Tool vendor independence is tremendously significant. In practice it means a wide choice of tools supporting the same notation and process semantics, lower in cost than traditional BPA tools. A number of BPM suite vendors offer their BPMN modeling tools as free downloads. That means you can afford to distribute the task of process modeling broadly throughout the organization. How many Business Process Experts would you like to have in your company? If the answer is 100 or more, BPMN is the only way to go.

Standardization of the diagram semantics also dramatically enhances *shared understanding*, which is the first and perhaps most important goal of process modeling. A properly constructed BPMN diagram has the same meaning to anyone looking at it, and reducing your current processes to BPMN has a quick payoff. The low-hanging fruit of process improvement often requires no IT implementation at all. Instead, process participants gathered around the diagram will readily identify potential improvements simply by inspection. In fact, this may be the first time any of them has seen or thought about the entire process end-to-end. BPMN provides the common visual language for that understanding in a way that a 300-page text document never can.

BPMN can be used for descriptive modeling and analysis, without the benefit of implementation detail, or as part of a full IT process solution. OMG says that BPMN is *methodology-neutral*, meaning it is intended for a wide range of use, from simple description to executable design, and you can use just as much of it as you need. This flexibility has

helped jump-start the BPMN bandwagon, although it has created some obstacles to true model portability.

A key component of BPMN's appeal to business users is its *familiarity*. In its simplest form, a BPMN diagram is just a flowchart, with swimlanes representing participant roles or organizational boundaries. It is not block-oriented like BPEL, which requires every conditional branch or parallel split in the flow to rejoin downstream with no dangling strands. Instead, BPMN is graph-oriented, meaning any activity can be routed anywhere else in the process, even back to a previous step. Business users like that freedom (although it occasionally makes mapping to BPEL execution engines difficult).

Another component of that appeal is its inherent *simplicity*. There are only three basic shapes, or flow objects, in the diagram:

- *Activity*, denoted by a rounded rectangle, represents an action or work done in the process. It is the only flow object that is performed by a resource.
- *Gateway*, denoted by a diamond, represents flow control logic, such as conditional branching, splitting, or merging process paths. It is pure logic, not work done in the process.
- *Event*, denoted by a circle, represents a signal that something has happened. The event flow object in BPMN describes how a process can wait for such a signal, react to it if and when it occurs, or send such a signal either to an external entity or to another part of itself.

BPMN goes on to define various subtypes of each of these, distinguished in the diagram by their icons or border styles, but the essential nature of each of the three base types is maintained throughout. More important is what is *not* included in BPMN: data flow, organizational roles, service components, physical systems, strategies and goals. Unlike the EPC diagrams of IDS Scheer ARIS, for example, which links each process activity to these various elements defined in other models, BPMN describes only the activity flow.

Also, while in EPC the arrow that connects one diagram element to another can specify one of many user-defined relationships between the elements, in BPMN such an arrow, called a *sequence flow*, means only one thing, *flow of control or enablement*: after the previous activity is complete, go to the next one. Thus while the ARIS metamodel is undoubtedly richer, the simplicity of BPMN is precisely what allows it to be supported by such a wide variety of tool vendors. Elements such as data, roles, and service components are supported in those tools, but in a manner specific to each, and external to the process diagram.

Activities and gateways are familiar flowcharting concepts, but events are not. In fact, BPMN's use of events is its singular distinguishing feature and the source of its remarkable expressive power. BPMN allows you to specify how a process can be started by an event, can wait for an event, or can be interrupted by an event and diverted onto an exception flow. In addition, in BPMN you can specify how a process can generate events that trigger other processes, respond to external requests, propagate errors, and even recover from failed business transactions.

By making events first-class modeling objects, BPMN stands out by making exception-handling behavior a primary concern of the Business Process Expert. Traditionally, exception handling was the part of process implementation tossed over the wall to developers. But if you believe in the notion of business empowerment in IT, this makes no sense. The familiar 80-20 rule says that 80% of the costs, delays, and errors come from 20% of the instances – the exceptions. So it should be up to the business to specify – in a non-

technical way – how any type of exception should be handled: a customer cancels or changes the order, an item is out of stock, or timeout occurs. If the event occurs, what should happen? The BPMN process model lets you specify that precisely in terms of the *what*; the implementation of model activities – specified outside of BPMN – defines the *how*.

One of BPMN's most valuable features is also its least appreciated and most subtle: *subprocesses*. In BPMN, any activity that has component parts is by definition a subprocess. In the diagram you can represent a subprocess either *collapsed*, as an opaque simple activity shape, or *expanded* to reveal its internal structure. Moreover, the expanded view can be shown either in line in the diagram or in a separate diagram hyperlinked to the parent and logically part of it.

While some process languages, such as BPEL, do not support subprocesses, subprocesses are central to the philosophy of BPM. For one thing, they are critical to modeling processes as end-to-end constructs. Subprocesses allow an end-to-end process to be described as a single hierarchical entity with a well-defined beginning and end, viewable at multiple levels of detail. The ability to render a BPMN subprocess as either collapsed or expanded lets you zoom in from the end-to-end view to any level of granularity, while maintaining the integrity of a single model at any level.

In addition, subprocesses allow the modeler to abstract unknown fragments of the end-to-end process. They also support distributed ownership of process fragments, since most organizations today are not organized and managed from a cross-functional process perspective. On the implementation side, subprocesses are perfect design-level containers for reusable business services, a central concept of SOA. Moreover, subprocesses allow those services to be stateful and be consumed in a rich self-describing way.

Finally, subprocesses provide a way to specify the scope of an event, meaning the boundaries of a process fragment where a particular business event, if it occurs, has a particular handler flow. For example, if between point A and point B a customer can cancel an order without penalty or special handling, you can enclose the fragment bounded by A and B in a subprocess, attach a customer cancel event to it, and add the handler flow to the event. In fact, you can even declare that subprocess to be a business transaction, and describe in the BPMN the required transaction recovery behavior using compensation. The combination of subprocesses and events gives BPMN remarkable expressive power in the hands of a Business Process Expert.

The preceding example hints at the last essential difference between BPMN and past modeling notations: BPMN was created with SOA in mind. That makes it a perfect fit with today's process implementations. Traditional flowcharting is based on a strict control flow paradigm with no real-time interaction with the outside world: after activity A do activity B, period. But in the age of SOA, the process is continuously interacting with the external environment, responding to service requests, requesting services from others, waiting for events, etc. BPMN models can describe that behavior explicitly with events and message flows, and can even describe the message exchanges (called choreography) between your process and an external process.

This is no accident, since BPMN was originally conceived as the graphical notation for a web service orchestration language called BPML. Thus the ability to send messages, wait for messages, or be interrupted by messages was an essential feature from the outset. In the end, BPML was trumped in the marketplace by BPEL, a similar language, and BPMN morphed into a more general process modeling notation, supporting human tasks and

subprocesses, for example. In 2005, the organization behind BPMN merged with OMG, which formally adopted BPMN 1.0 in February 2006.

This history may account for one of the biggest disappointments of BPMN, which is the lack of a standard for process model storage and interchange in XML. While the BPMN spec defines the shapes and their associated process semantics, the only sure-fire way today to import a model from BPMN tool A into BPMN tool B is to redraw the diagram. Even then, you can't be certain tool B supports all of the standard constructs you can model in tool A, since BPMN does not define a minimum set of supported elements to qualify as "compliant."

In the absence of a serialization standard from OMG, the Workflow Management Coalition extended its XPDL standard to capture all the elements of BPMN 1.0. XPDL 2.0 support was introduced in 2007 by a number of BPMN tool vendors, but interoperability between tools is still limited. OMG is in the process of releasing its own serialization of BPMN, called the Business Process Definition Metamodel (BPDM). BPDM is a formal metamodel based on OMG's Meta Object Facility (MOF), and in principle will allow mapping between BPMN and other notations such as UML. However, the draft BPDM specification diverges considerably from today's BPMN, so its applicability for interchange of existing models is questionable. Adoption of BPDM is expected by the end of 2007.

BPMN 1.1, a minor change to the BPMN 1.0 specification, reached final draft in July 2007, and is supported by a small number of tools. BPMN 2.0, a significant change that will merge the notation with the formal metamodel, is not expected until late 2008 or 2009. Thus BPMN 1.0/1.1, despite limited portability between tools, represents an excellent standard for the Business Process Expert that should remain stable for at least another year, and supported by an ever-increasing array of BPM vendors in both standalone modeling tools and full BPM Suites.

Bruce Silver