



Circles and Diamonds and Squares, Oh My!

Demystifying the BPMN Standard

BPMN standards can be confusing, but once you understand their purpose and how to use them, they can be lifesavers. This paper, based on a webinar by Justin Brunt, provides a walk-through of the BPMN 1.0 standard, shows how it is implemented within TIBCO Business Studio™ 2.0 (available as a free download) and how the combination of BPMN and Business Studio can help you be successful with BPM.



About the Presenter/ Author

Justin Brunt is Senior Product Manager for TIBCO's BPM product suite, TIBCO iProcess™ Suite. He's been involved with business process management for over thirteen years, initially as Director of Development for Staffware PLC and as Director of Research. Following the acquisition of Staffware by TIBCO, Justin took on his current product management role. Prior to joining Staffware, Justin held senior software development roles with Logica and Siemens. During his BPM and workflow career, Justin's been involved in the development of BPMN-related standards. Throughout this time he's been involved with the Workflow Management Coalition or WfMC and held the position of Vice Chair Europe of the WfMC's technical committee between 2002 and 2006, prior to his current position as Vice Chair Europe of the Steering Committee. He's also a fellow of the Workflow Management Coalition. Justin also represents TIBCO in the OMG's BPM-related initiatives.

What is a Process?

There are many descriptions of business processes, but I particularly like this one in the context of BPMN. The Workflow Management Coalition defines a process as the representation of a business process in a form that supports automated manipulation, such as modeling or enactment by a workflow management system. The process definition consists of a network of activities and their relationships.

What is BPMN?

Simply put, Business Process Management Modeling Notation (BPMN) is a graphical way of describing a business process. It's flowchart-based, so looking at the process definition you should immediately be able to see what the process is all about. It's basically an activity network combining various flow objects to describe the process. It's targeted at different sorts of people. It's aimed at the business analyst on one side and the technical person, who may have to implement a business process, on the other side. The process language was put together so that it would be applicable and understandable to people in different roles.

Origins of BPMN

BPMN came from an organization called Business Process Management Initiative (BPMI) that set about putting together specifications and standards for business process management. They initially put together a specification called Business Process Modeling Language (BPML). Associated with that, they wanted a visualization of a business process. That's where BPMN came in. It was originally put together by a number of people from different organizations including BPM vendors, academics, and end users. The language itself was derived from different modeling notations – UML, high def, activity decision flow. But the key thing about BPMN is that it really is targeted at describing a business process, whereas some of the other modeling notations have a wider scope.

BPMN is very specific to business process management. To date, we're aware of at least 44 implementations of BPMN, of which TIBCO Business Studio is one. The original specification came out in 2004 and the BPMI organization was eventually merged into the Object Management Group standards group. They adopted BPMN as one of their formal specifications last year.



Scope of BPMN

The goal of BPMN was to describe business processes and the full range of business process types – from workflow that may be very human-centric to the integration-centric processes you would see in EAI systems – and to unify the different people in an organization that were interested in the business process – from the high-level business user to the technical user who may be involved in implementing the process.

BPMN Limitations

In terms of the language itself, there are some limitations to it.

- It doesn't describe organizational structures and resources. That is covered by a separate specification. In fact, the Object Management Group (OMG) works on those kinds of specifications.
- It doesn't define the data model for the process.
- It doesn't describe business rules.
- And one of the key things the notation doesn't include is a specification for persisting the process definition. There are a number of initiatives that satisfy that requirement. One of them is the Workflow Management Coalition's XDPL specification. Another one is a new standard defined by the OMG: Business Process Definition Metamodel (BPDM).
- And, although it can show the flow of data, BPMN is not a data flow diagram.

Uses of BPMN

So what can you use BPMN for? There are three main types of business processes.

- private process: All of the activities within the process are completely within a department or an organization, with no communication from that process to the outside world. The process definition would include all of the activities that you see being performed within that process.
- public process: Represents an interaction between a private business process and another participant. In this case, only activities used to communicate outside the private business process are included.



- collaboration or global process: A process you would find in a business-to-business scenario. For example, in a supplier-customer relationship, you might only want to define the interactions between those organizations.

BPMN Core Elements

Core to the BPMN specification are a number of elements that fall into two main areas:

- flow objects – things that happen during the process
- connectors – things that describe how the different flow objects relate to each other

Under flow objects you have events, activities, and gateways. Under connectors you have sequence flows, message flows, and associations.

Figure 1. Core set of BPMN elements

Events



Activities



Gateways



Sequence Flow



Message Flow



Association



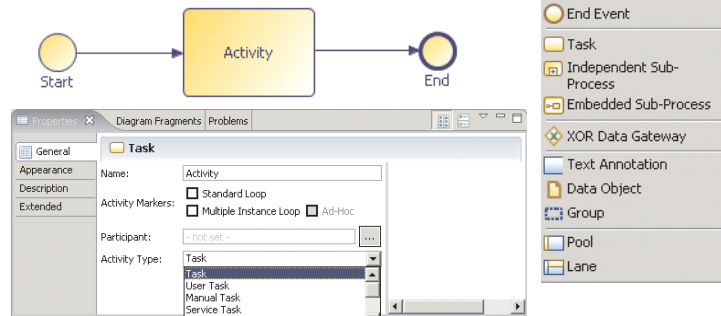
The core set of modeling elements enable the easy development of simple business process diagrams that will look familiar to most business analysts (a flowchart diagram)

BPMN elements look very familiar because they are drawn from the flowcharting paradigm. The TIBCO Business Studio palette, which helps you draw the business process, includes all of these BPMN shapes. For instance, under events you'll see that there are start events, intermediate events, and end events. Using those shapes and others, you can build your business process. Figure 2 shows what is perhaps the simplest of all business processes: start a process, do something, and complete the process.



Figure 2. BPMN core elements

- Core elements as they appear in Business Studio
- The world's simplest business process



When you look at a particular shape in TIBCO Business Studio, you are able to see a properties view for each shape.

ACTIVITIES

In the simple process described above – start, do something, end – we just had a task; there was no specialization at all. BPMN allows us to have much more specialization than that. For example, you can have a sub-process. An activity could be looped so it occurs continuously. An activity could be performed in multiple instances; perhaps an activity is delivered to more than one person or performed by more than one person. And you can combine these different activities; so a sub-process might also behave in a looped manner or in a multiple instance manner, going out to different instantiations.



Figure 3. Activities

- **An activity is work that is performed within a business process**
- **Activities are defined as being atomic or non-atomic (compound)**
 - Task – the work is not broken down into finer detail
 - Sub-process – the work can be broken down into finer detail
 - Sub-processes contain sub-activities



Task

Performed once



Sub-Process



Looped

Includes an internally defined loop – can be applied to sub-processes



Multiple Instance

Like a broadcast to multiple participants – can be applied to sub-processes



TASKS

One interesting thing about BPMN is that the specification allows you to embellish the shapes themselves. When you look at the shape for an activity, it's fairly bland. It's just a round-cornered oblong. But to convey more meaning, you can embellish it with colors or icons. You can change the size of the icon.

In Business Studio, we allow you to define what kind of activity you want to represent. You might have a User Task, where somebody is sitting at a computer and an activity is delivered to them through a workflow or BPMN system. You might have a Manual Task, that represents a point in a process where somebody needs to do something with some paper – a manual task, but you still want to represent that within the business process. You might have a Service Task, where some kind of automation occurs. For instance, a system may go and do a credit check. Or you might have a Script Task, where an automated script is run to perform some logic that helps with the execution of the process.



Figure 4.Tasks

- **BPMN permits the addition of markers to indicate the type of activity that is being performed**
 - Markers must not change the semantics of the task or conflict with other standard BPMN constructs
 - Examples of Business Studio's markers:



User Task

Performed by a user within a BPM system



Manual Task

Performed manually by a user – no automation



Service Task

Performed by an automated system – web service



Script Task

Executes an automated script

So you're allowed to extend BPMN by having these colors and definitions, the only constraint is that you not change the meaning or the semantics of the process specification.

SUB-PROCESSES

The sub-process activity provides a means to define logic that you wish to separate from the main body of the process. For example, you may want to encapsulate this process logic to enable reuse, so this piece of logic can be used by different processes or within different parts of the same process. Or you want to hide the detail of one part of the process because it detracts from the way the process can be understood by people reading it.

There are two different types of sub-processes. One is a separate sub-process where it looks like a complete process definition. (In Business Studio, when you want to investigate the process definition for the sub-process you can click on the object that you see on the screen and the tool would open up a new window where you could see the detail of the sub-process.) The other type is an embedded sub-process, where it's not separate from the process definition that you're working with, but surrounds all of the logic you're interested in within a boundary and it appears as a single task within the process. Within the sub-process there is some more detail, but it isn't held in a separate place like the other type of sub-processes.



Figure 5. Sub-process

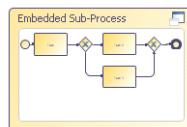
- **Enable hierarchical design**
- **Promote reuse of common process logic**
- **Aid understanding by simplifying higher levels**
 - Hide unnecessary detail



Collapsed

- Some tools will allow sub-process to be displayed or expanded
- Business Studio opens sub-process in a separate window

- **Embedded**
 - Part of the parent process
Expand or collapse
- Collect activities and re-factor as sub-process



EVENTS

Events define something that happens during the life of a process.








- Start Events kick off the process.
- End Events define where a process or a branch of a process ends.
- Intermediate Events define things that happen during the process. These can be part of the flow or associated with different activities.

START EVENTS

For each of the event types there are further specializations. For instance, a process could be started by the receipt of a message from someone. A process could be started by the receipt or triggering of a timer. For example, a process might start every Monday or once a month. So there are different ways of starting processes, and you can use different event types to specify how those processes will start.













Figure 6. Start events

-  **None** 
-  **Message**
 - Process starts as the result of receiving a message
-  **Timer**
 - Process starts as the result of a timer-based trigger
-  **Rule**
 - Process starts as the result of a conditional event
-  **Link**
 - Process starts as a result of control jumping to this point
-  **Multiple**

INTERMEDIATE EVENTS

As with Start Events, Intermediate Events can be started by a message. (In fact, a process could send a message to another process, so you could get this interaction between different processes.) A process could include some timer-related information. You may need to wait until a particular time before something else needs to happen. Or you may use it as part of an escalation capability, where if a particular action doesn't happen by a particular time; you can specify some behavior that will occur on the expiry of a timer. An event could be initiated as a result of a rule or condition. Figure 7 shows how that would appear in the Business Studio palette. So you can click on Intermediate Event and select the particular Intermediate Event type you want to use as part of your process.

Figure 7. Intermediate events

-  **None** 
-  **Message**
 - Receiving or sending a message
-  **Timer**
 - Receiving a timer based trigger
-  **Error/Exception**
 - Receiving an exception-based trigger
-  **Cancel**
 - Receive a cancellation
-  **Compensation**
 - Undo as a result of failed transaction
-  **Rule**
 - As the result of a condition
-  **Link**
 - Off page connector, GOTO
-  **Multiple**



END EVENTS

You have similar options with End Events. A process could end by sending a message to something or someone. If you get to a particular point in a process, it may terminate the process earlier than it would have done if it had gone down some other route. These are some of the different ways you can end the process.

Figure 8. End events

 **None**



 **Message**

- Process ends by sending a message

 **Error/Exception**

- Process ends by sending an exception

 **Compensation**

- Process ends by sending a compensation

 **Cancel**

- Process ends by sending a cancellat

 **Link**

- Process ends by jumping to another location

 **Terminate**

- Process terminates when it reaches point

 **Multiple**

EXAMPLES OF EVENTS

Figure 9 shows some examples of how events could be used. For the first one, we've got a process that is started by the receipt of a message. For example, you receive a purchase order and that instantiates a process that manages the fulfillment of the order.

In the second one we've got a timer within the flow. Once an activity is performed, the process needs to wait for a message to be received before it proceeds to the next activity within the process. Perhaps it's waiting for the receipt of a scanned contract.

In the next one we have an event on the boundary of an activity. We're processing an order, but we've got a timer trigger such that after 48 hours if something hasn't happened, we can escalate to the manager. The manager gets a notification that something is not happening as it should.



Figure 9. Example of events

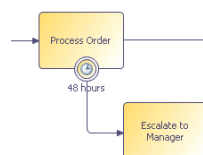
- **Start process on receipt of message**



- **Suspend process waiting for an event**



- **Timer event on an activity**



GATEWAYS

Gateways help us specify the flow of the process. There may be points where you need to decide which path to go down through the process. There may be other points where you need to have several things happen in parallel. Different types of gateways allow you to change the way the process flows.

The first of these is an exclusive gateway. And there are two different types of these. One is data based, meaning that it makes a decision based on data. For example, if the value of a purchase order is greater than \$10,000, proceed down this path, otherwise, proceed down that path. There are two different representations of this gateway in the specification. One is shown here with the cross in the middle. (An empty diamond means exactly the same thing, but in Business Studio we've chosen to use the cross symbol for added visualization.) The other type of exclusive gateway is event based, so the determination of which path to go down is determined by the receipt of one type of event or another.

Another type of gateway you'll see and may want to use yourselves is a parallel gateway. Several paths go out from this type of gateway and all will be performed in parallel.

The other gateway is an inclusive decision/merge. In this case, you may have conditions associated with the different routes. You may then end up with one or more paths being followed depending on which conditions are met.



The final type of gateway is a multiple gateway. This is less defined by the BPMN specification. The behavior can be defined by the people developing the tools, so it's completely up to the tool vendor as to how they represent that.

If none of the above mechanisms satisfies the requirement, tool vendors have the opportunity to use the multiple type. Gateways can be used for splitting into several paths and for joining multiple paths back together. If you have an exclusive decision gateway that then splits into multiple paths, you will see a gateway that joins them back together again.

Figure 10. Gateways

▪ **Control the flow within a process – diverge and converge**

- All represented by diamonds – familiar flowchart symbol



XOR – Exclusive Decision/Merge (data based)

- Decision point – only one outgoing path can be taken
- Empty diamond is an alternative to this symbol



XOR – Exclusive Decision/Merge (event based)

- Triggered by receipt of alternative messages



Parallel – Fork/Join

- Unconditional splits into multiple paths
- Also used for synchronization (wait) of multiple convergent paths
- Parallel gateways are not compulsory



OR – Inclusive Decision/Merge

- More than one possible outcome



Multiple – Complex Decision/Merge

- Advance split and merge behavior

EXCLUSIVE GATEWAY

In Figure 11 we have an example of an exclusive gateway where we've got conditions: If a loan is greater than this value, go down this path to have the loan approved. If it's a lower value, we can have a different person approve it. Later you can see that the paths merge. In this particular instance you will only see one of the paths followed through the process.

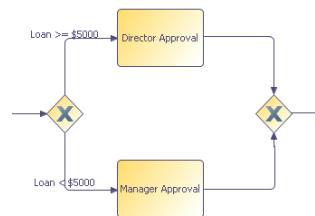
We also have an example of the event-based gateway, where we may be waiting for different things to happen. One might be an acceptance message. The other might be a cancel message. So we would follow a different path depending on the event that was received by the process. Again, we see the joining of the different paths at the end and we see that only one of these paths would be followed through the process.



Figure 11. Exclusive Gateways

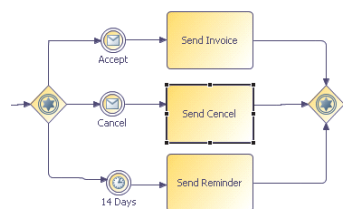
▪ **Data based gateway**

- Evaluation made on data



▪ **Event based gateway**

- Triggered by one of the events following the gateway

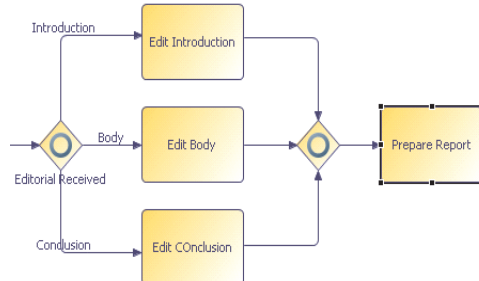


INCLUSIVE GATEWAY

The inclusive gateway works the other way around, where multiple paths may be followed. In this example, we have several chapters of a book to edit, and depending on how many of those chapters are available, we'll define how many of those paths we go down.

Figure 12. Inclusive Gateways

- **One or more of the paths can be followed depending on circumstances when gateway is reached**



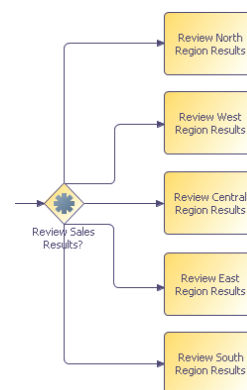


COMPLEX GATEWAY

Here is an example of the complex gateway, where you might have more choices to make and different behavior following from that, which can be defined by the tool vendor as to how that is implemented and appears within a product.

Figure 13. Complex gateways

- All other gateways have specific behavior defined for them
- Complex gateways can be tool specific



PARALLEL GATEWAY

Going on to parallel gateways, in Figure 14 we have an example where a process can split into more than one path. In this particular case, we have two different parallel paths that will be followed together. The process will then wait until both of those activities have been performed before proceeding to whatever activities appear after the joining gateway.

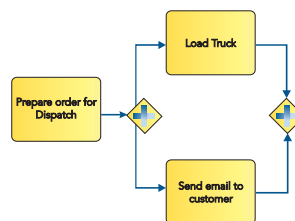
One of the other things BPMN allows you to do is have very similar logic, but for it to be defined in different ways. So here we have two paths within a process that have exactly the same meaning but different ways of representing them. In the top one we're using a gateway to say this is how the process splits into two parallel paths. The bottom one allows you to have multiple paths out of the first activity into the secondary activities, and then they join back into the next activity via a parallel gateway. The behavior of those two process fragments is exactly the same, but you can represent them in different ways. It



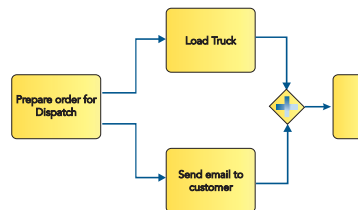
may be that an organization decides that they have a particular methodology for defining a process and one may decide that they use gateways to define the splitting. Another organization may decide they're quite happy to just have multiple paths coming out of activities to define the same thing.

Figure 14. Parallel gateways

- Explicit use of Gateway to cause split and rejoining of paths



- Can also perform without explicit use of Gateway
- Methodology could define use



CONNECTORS


The last set of elements within the specification are connectors – the things that join the different parts of the process together. Connectors define how the flow objects – the events, gateways and activities – are joined together, what sequence they appear in. And again, there are different specializations of those elements.



In Figure 15, we have a normal sequence flow that says that this follows this and this follows this. The next is a conditional connector and this has the same kind of idea that we saw in Figure 14 whereby you can choose to use a gateway for splitting into separate paths, or you can have multiple paths coming out of an activity. The conditional connector provides you with much the same idea in that you can have an activity, which has multiple paths coming out of it, and each of them has this connector which has the diamond on which defines that there is a condition associated with each of these paths. And only if that condition is met will it go down that particular path onto the next activity. The next kind of connector is the association. We have a number of different elements within BPMN that allow us to associate information with the process itself. This dotted line allows us





to associate things like data and information with the process itself and we'll see some examples of that soon. The next one is a message flow and this shows how messages go between one part of the process and another or, indeed, between different processes themselves. We'll see some examples of that, too.

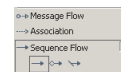
Figure 15. Connectors

- **Sequence Flow** 
 - Connects activities, gateways, and events
 - Shows the order in which activities are processed
 - Cannot cross sub-process or pool boundaries

- **Conditional connector** 
 - Use instead of gateway
- **Default connector** 
 - Indicates path to be taken if not other conditions are satisfied

- **Association** 
 - Associates, data, information, and artifacts with flow objects

- **Message Flow** 
 - Shows flow of messages between entities
 - Across pool boundaries



SEQUENCE FLOW

Figure 16 shows some examples of the use of the connectors. In the first example, we've got one activity following another, using the normal sequence flow connector. In the second example we've got conditions and a default path where the flow will go if none of the conditions are met.



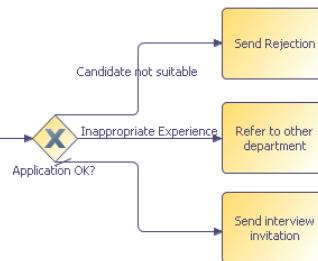
Figure 16. Sequence flow

Normal Sequence Flow



Default Sequence Flow

- Either evaluation dictates path to be followed or default path is followed

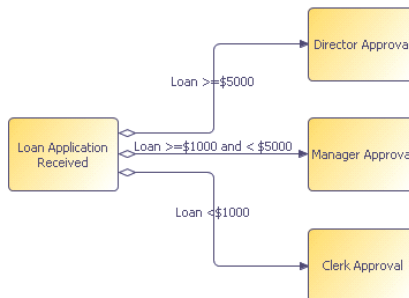


CONDITIONAL SEQUENCE FLOW

The conditional sequence flow is an alternative to using a gateway. In Figure 17 we have three paths coming out of an activity, each with a condition on. Depending on the evaluation of the condition, one of those paths will be followed. Indeed, you could have a further path, which is that none of those conditions are met. In that case, you would have a default flow.

Figure 17. Conditional sequence flow

Can be used as alternative to exclusive gateways





ASSOCIATIONS

We talked earlier about the associations connector. There are a number of things you can associate with a process. The first one is an annotation, that's basically a way that you can comment the process. It adds no logic to the process and it doesn't affect the way the process behaves, but it does give you the opportunity to add information to help the reader make sense of the process you're defining.

You can also add a data object. For a purchase order, you could use data objects to indicate that a purchase order has been received, approved, or completely fulfilled. For a job application process, you may want to indicate that the state of the job application changes as it goes through the process. It helps the readability of the process by letting you know what kinds of thing the process is dealing with.

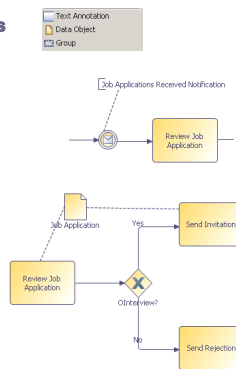
Figure 18. Associations

- **Used to add information to the process**

- **Text annotations**

- Comments

- **Data objects**



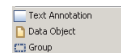
GROUPS

Groups is a way of surrounding a number of activities within a process or across multiple processes. If you're dealing with multiple processes within the same diagram, you can associate them by lassoing them with the group element. This has no bearing on the logic of the flow. It simply helps the understanding of the process, helps you categorize and group activities that can be considered part of the same phase of a project or are otherwise related to each other.



Figure 19. Artifacts

▪ **Provide a mechanism to show additional information in addition to flow of process**



▪ **Annotations**

- Provide additional information about process (comment)



▪ **Data objects**

- Used to show how data and documents relate to the process
- Can define inputs and outputs to activities
- Can be given state



▪ **Groups**

- Used to highlight parts of process
- Not constrained by pools and lanes



▪ **Can extend artifacts**

The specification also allows you to add your own artifacts to the process modeling language, while making sure that you're not clashing with anything that is specified as part of the core standard. So you could add artifacts to represent a database, an organizational modeling system, things that add to the information about the process.

POOLS AND LANES

Pools and lanes (or swim lanes, as they are sometimes called) are ways of describing how a process or processes are organized. They define how different organizations or different people get involved with the process. By default, each process lives within a pool, so a number of activities will live in a pool.

Figure 20. Pools and lanes

▪ **Used to add clarity to process**

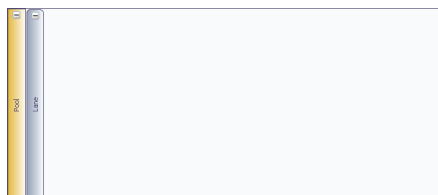


- Different organizational entities
- Different companies in a B2B scenario



▪ **Subdivide pools into lanes**

- Denote different participants (resources, departments, etc.)

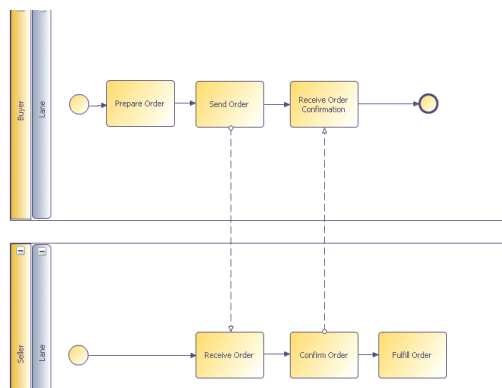




A pool can be split into lanes. You might have a pool that represents the organization and within that pool you might want to subdivide it into different departments that are involved within a particular process and show how the different departments interact with each other. Or you might want to have a swim lane for different individuals that participate in the process.

In Figure 21 we have an example where a process in one pool is interacting with a process in another pool. You'll notice that there are no sequence flows between the pools. The interaction between processes in different pools is facilitated by the flow of messages between the different activities in the processes. We've got this synchronization between the two processes where one sends out a message to say I've sent this order (and you may want to annotate that with a data object that says this is an order), and the other process receives the order and then confirms receipt of the order by sending a message back to the initiator of the process to say we've received the order, we'll now confirm it, and go on to fulfill the process.

Figure 21. Interaction between pools



So it's a powerful way of describing the interactions between different parties that may be involved in a global or collaborative process.

LANES

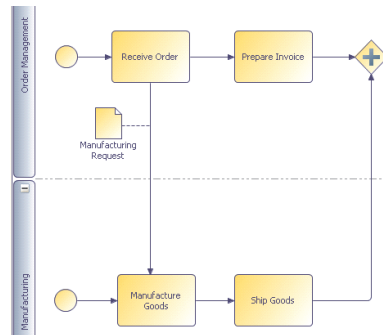
In Figure 22 we've got a view of different lanes within the same processor. Here we've got a process that's contained within one pool and that pool is broken up into different lanes. In this example we've got different groups in the organization: we've got the order management organization performing their tasks and interacting with the manufacturing



group. So we've received an order, we notify manufacturing that we've received this order, and we can then go and manufacture the goods, and so on. What you'll notice here is that the connector between the different pools is a sequence flow, a solid line. We're allowed to use that between different lanes within the same pool. The way that lanes are defined is an aid to understanding the process – how the organization works within the process, who does what – rather than it affecting the flow itself. That's why we're able to use the sequence flow rather than being restricted to using the message flow.

Figure 22. Lanes

■ Different roles/groups within organization



TRANSACTIONS AND COMPENSATING BEHAVIOR

A transaction is a specialization of an embedded sub-process. In this case, there may be a number of activities we wish to have performed, but only if all of them are performed and succeed do we want the process to go to the next step. For example, in a travel planning process we want to book a flight, but in association with that we also want to book a hotel. We don't really want to continue if one of them fails. So, if we can book a hotel, but we can't book a flight, we don't want to continue. In fact, we want to unbook the other activity. In some cases, the activities themselves will roll back. (If one fails, we will roll back to the point where the transaction started.) In this case, you can't undo them automatically, because they involve manual activities; you have to perform some compensating activity to undo what was done. If we book a flight, we can't simply forget it, we have to go and physically undo the booking of the flight.

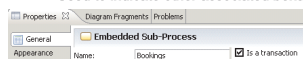
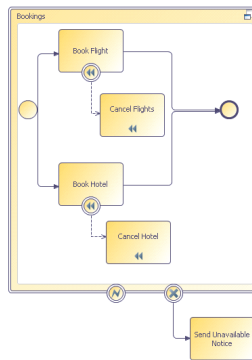
In this example we've got one of these events that appears on the edge of an activity that says, if I need to roll back, this is what I do. I associate an activity with that event. The



associated activity, in this case, is book a flight. If we need to roll back, because the other activity failed, I need to cancel the flight. So that gives us a way of undoing things.

Figure 23. Transactions and compensating behavior

- **Transaction boundary has double border**
 - Property of embedded sub process
 - All activities contained within it completed successfully or all rolled back
- **Some activities may need compensating actions to undo**
 - Compensating action "associated" with original activity
 - Cancel and Exception (Hazards)
- **Used to indicate other associated behavior**



Future of BPMN

What I want to do now is to describe where BPMN is now and what's going to happen in the future. The specification that TIBCO Business Studio satisfies is version 1.0. The Object Management Group has just recently completed the 1.1 version of the specification, which clarifies some of the ambiguities that may have existed in version 1.0 and adds a few features I'll discuss shortly.

The next major step is BPMN 2.0, which will be a combination of BPMN and BPDM, another Object Management Group specification. As I mentioned earlier, BPMN 1.0 doesn't have a description of how you would save a process definition. An XPD 2 from the WfMC allows you to do that now, but BPDM is an initiative from the Object Management Group to provide a mechanism whereby you could persist the process definition that you've drawn within BPMN and exchange it between different product tools, so between maybe a modeling tool and a process execution engine.

One of the other changes is to the BPMN acronym itself. Although it's got the same letters, the words will change to reflect the merging of these two processes. Instead of Business Process Modeling Notation, it's Business Processing Model and Notification.



This specification is at the beginning of its life and will probably take 18 to 24 months before it comes out. In that intervening period, we will continue to use BPMN 1.0 and you'll see process modeling tools picking up the changes that will be published in 1.1 and extending their capability.

BPMN 1.1

What have we got coming in BPMN 1.1? Some changes are purely descriptive. Earlier I talked about the XOR Exclusive Gateway and the All Inclusive Gateway. We've dropped the "XOR" and "All" to make it less of a technical description of the shapes and things that appear within the process. We've also changed the way one of the shapes is represented. The event-based gateway symbol had a star. Now it has a pentagon.

When we talked about the different events, I mentioned that you could either send a message or receive a message as part of the process. In BPMN 1.1 we're able to visually distinguish between the types of events. If it's a receive, it's an empty envelope. If it's a send, it's a full envelope.

We also have a new event type that appears within the specification itself called a signal. It's similar to the message event, but the message event, needed something to send to. A sending message needed to send to something that was going to pick up that message. A signal is like a broadcast. The process can send out the signal and it doesn't know who's listening to it. If there is something listening to it, they will pick up the signal and perform whatever needs to be performed following the receipt of that signal. But the sender doesn't rely on having a receiver ready and waiting for the signal. Another change relating to events is that, similar to the gateway, there's a change of shape from a star to a pentagon.



Summary

BPMN has become almost the de facto standard for representing business processes. We know of 44 implementations so far, and it's likely there will be more yet.

BPMN is aimed at both technical and business users. If you cast your mind back to some of the examples, you could draw a process that didn't need to have all of the detail about all the different types of events and activities. You could quite easily have a very good representation of a process. But if you need the detail, all the different elements within the language are there for you to add more meaning and information to it.

Business Studio provides quite a few examples. There are sample processes you can use as models. You can see how the different process elements are used and how they relate to each other to give you a good start in defining processes. There are also process fragments you can use to easily put together a particular construct, such as a split and a join with some activities in between. You can just drag those onto the pallet and get your process put together very quickly.

Here are some resources for further information about the BPMN specification.

<http://www.omg.org> <http://bpmn.org> <http://www.wfmc.org>

<http://developer.tibco.com> http://www.tibco.com/devnet/business_studio/default.jsp



Global Headquarters
3303 Hillview Avenue
Palo Alto, CA 94304

Tel: +1 650-846-1000
Toll Free: 1 800-420-8450
Fax: +1 650-846-1005

www.tibco.com