Business Process Modelling Toolkit

South East Training

228 Dyke Road
Brighton
BN1 5AE
Telephone: 01273 555567
Email: enquiry@southeasttraining.co.uk
Web: www.southeasttraining.co.uk
Toolkit: www.businessprocessmodelling.co.uk
Copyright

The Business Process Modelling Toolkit is owned by South East Training. Organisations and individuals are granted a free licence to use the toolkit on condition that they acknowledge both the ownership and the source if reproduced.

Ownership: © South East Training 2013

Source: www.businessprocessmodelling.co.uk

South East Training
228 Dyke Road
Brighton BN1 5AE

Telephone: 01273 555567
Email: enquiry@southeasttraining.co.uk
Web: www.southeasttraining.co.uk
Toolkit: www.businessprocessmodelling.co.uk

Contents

Contents ........................................................................................................................................... 4
About South East Training .................................................................................................................. 6
How We Can Help .............................................................................................................................. 6
The Business Process Modelling Toolkit .......................................................................................... 7
Business Process Improvement .......................................................................................................... 7
Business Process Modelling Notation (BPMN) .............................................................................. 7
Copyright ......................................................................................................................................... 7
Process Activities and Events .......................................................................................................... 9
   Events ........................................................................................................................................... 9
   Activities ...................................................................................................................................... 9
   Sequence Flow ............................................................................................................................. 9
Gateways and Exception Paths ......................................................................................................... 11
   Exception Paths .......................................................................................................................... 11
   Gateways ..................................................................................................................................... 11
Pools, Lanes and Sub-Processes ....................................................................................................... 13
   Lanes and Pools .......................................................................................................................... 13
   Activity Types ............................................................................................................................ 13
   Sub-Processes ............................................................................................................................ 13
Expanded or In-Line Sub-Processes ................................................................................................. 15
   Expanded or In-line Sub Processes ............................................................................................ 15
Hierarchical Expansion of Sub-Processes ....................................................................................... 17
   Hierarchical Expansion .............................................................................................................. 17
Loopback to Handle Exceptions ....................................................................................................... 19
   Loopback .................................................................................................................................... 19
   Infinite Loops? ............................................................................................................................ 19
   Sequence Flow ............................................................................................................................ 19
Parallel or AND Gateways ................................................................................................................ 21
   AND Gateways ........................................................................................................................ 21
   Splits and Joins ........................................................................................................................... 21
Collaboration Diagram ..................................................................................................................... 23
   The Black-Box Pool .................................................................................................................... 23
   Message Flow ............................................................................................................................ 23
Collaboration with Redefined Scope ............................................................................................. 25
   Defining the Scope of the Process ............................................................................................. 25
   Communicating with the Originator ........................................................................................... 25
Collaboration with Sub-Process ........................................................................................................ 27
   Collaboration with Sub-Process Elements ................................................................................ 27
Summary of Basic Notation .............................................................................................................. 29
   Start Events .............................................................................................................................. 29
   End Events ............................................................................................................................... 29
   Gateways .................................................................................................................................... 29
   Sequence and Message Flow ..................................................................................................... 29
   Link Events .................................................................................................................................. 29
Data Objects and Messages .............................................................................................................. 31
   Data Objects .............................................................................................................................. 31
   Messages .................................................................................................................................... 31
Splitting and Merging Recap ........................................................................................................... 33
   Exclusive Flow .......................................................................................................................... 33
   Parallel Flow ............................................................................................................................. 33
   Conditional Flow ....................................................................................................................... 33
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Splitting</td>
<td>35</td>
</tr>
<tr>
<td>Merging</td>
<td>35</td>
</tr>
<tr>
<td>More Start Events</td>
<td>37</td>
</tr>
<tr>
<td>More Than One Start Event</td>
<td>39</td>
</tr>
<tr>
<td>More End Events</td>
<td>41</td>
</tr>
<tr>
<td>End Events</td>
<td>41</td>
</tr>
<tr>
<td>More End Events - Examples with In-Line Sub-Proceses</td>
<td>43</td>
</tr>
<tr>
<td>More End Events - Examples with Collapsed Sub-Proceses</td>
<td>45</td>
</tr>
<tr>
<td>More Sub-Proceses</td>
<td>47</td>
</tr>
<tr>
<td>Sub-Proceses</td>
<td>47</td>
</tr>
<tr>
<td>Examples</td>
<td>47</td>
</tr>
<tr>
<td>Intermediate Events</td>
<td>49</td>
</tr>
<tr>
<td>Intermediate Events - Examples</td>
<td>51</td>
</tr>
<tr>
<td>Delay using an Intermediate Timer Event</td>
<td>51</td>
</tr>
<tr>
<td>Event Gateway</td>
<td>51</td>
</tr>
<tr>
<td>Boundary Events</td>
<td>53</td>
</tr>
<tr>
<td>Boundary Events - Examples</td>
<td>55</td>
</tr>
<tr>
<td>Repeating Activities</td>
<td>57</td>
</tr>
<tr>
<td>Repeating Activities - Basic Notation</td>
<td>57</td>
</tr>
<tr>
<td>Repeating Activities - Advanced Notation</td>
<td>57</td>
</tr>
<tr>
<td>Batch Processing</td>
<td>59</td>
</tr>
</tbody>
</table>
About South East Training

South East Training is a network of experienced business consultants who provide a wide range of consultancy services, focused on employee development, to help our customers enhance their business performance.

We provide management and interpersonal skills training at all levels and cover topics as diverse as business process auditing and business process improvement through to customer service excellence and communication skills.

South East Training also provides specialist consultancy services such as executive coaching of managers wanting to develop their project management, change management or internal consultancy skills. We also provide expert facilitation of bespoke events, for example, business planning and business process mapping workshops.

If you are looking for a learning and development provider that is committed to working in partnership with you to motivate your personnel and enhance performance or, for example, to develop your managers’ leadership and communication skills, contact us now to arrange a visit or request a proposal.

South East Training is registered with the Institute of Leadership and Management as an ‘Approved Provider for Endorsed Awards’. Our consultants are based throughout the UK and operate on a worldwide basis.

How We Can Help

South East Training provides learning and development opportunities in the following ways:

Management and Leadership Skills Programmes: We provide programmes aimed at all levels, from newly appointed to experienced managers, and covering topics as diverse as leadership and team building through to project management and business process mapping. We also provide programmes that enhance personal effectiveness such as time management and problem solving. More specialist programmes cover the consultancy skills required to be effective in the role of change agent within change management programmes.

Interpersonal Skills Programmes: These programmes cover the skills required to be effective at work in both formal and less formal situations. General topics cover the interrelated communication skills of being assertive, managing stress and working in teams. Specific topics include interviewing, negotiation, presentation skills and customer service excellence.

Business Coaching: We offer this service at all levels. Coaching is available as a standalone support for individuals who typically:

- wish to enhance their business skills
- want to understand and resolve specific professional issues that they feel are limiting their performance
- are experiencing a period of change.

We also offer coaching as a follow-on to learning events, to help embed best practice and provide confidence to learners that they are implementing their learning in the right way.

Services are available to be run at your premises or at any agreed external venue, anywhere in the world.
The Business Process Modelling Toolkit

This document is aimed at providing business analysts and others involved in process management and improvement activities with sufficient understanding of the notation and the associated rules to allow them to produce Business Process Diagrams that conform to a widely recognised standard. The benefit to your organisation will be the in-house capability to create Business Process maps that can be shared and readily understood by all stakeholders, both within and external to the organisation.

Business Process Improvement

As the recession deepens and managers at all levels are facing hard decisions about where to apply cuts in expenditure, South East Training is keen to help its customers make the right decisions for their businesses. Expenditure cuts applied in the wrong place can damage an organisation’s capability to continue meeting its customers’ expectations. A failure to provide the service that your customers have come to expect leads to poorer business results and possible long term decline. The goal, therefore, is to find a way to rationalise your business while continuing to provide excellent customer service, leading to future growth. The answer lies in improved Business Process Management.

Business Process Modelling

Before you can set out improving your business processes, you have to know how they are operating at the moment. Business Process Modelling allows you to illustrate how operations, processes and work activities within a business are sequenced to transform inputs into outputs that meet internal or external customer needs. By showing the interaction between operations, processes and tasks, they provide not only a detailed view of how an organisation works, but also provide the foundations for improving the economy, efficiency and effectiveness of business operations.

The business process diagram is fundamental to any improvement activity.

Business Process Modelling Notation (BPMN)

At South East Training we advocate the use of Business Process Modelling Notation (BPMN) for illustrating your processes.

Business Process Modelling Notation (BPMN) is a standard for process modelling that provides a graphical notation for illustrating business processes in a Business Process Diagram based on a flowcharting technique. The objective of BPMN is to support business process management for both business users and IT specialists by providing a notation that is intuitive to business users yet able to represent complex process semantics.

Copyright

Please feel free to distribute this document to anyone involved in business process auditing. Please remember, however, when reproducing any part of this document to include the ownership, © South East Training 2013, and the source, www.businessprocessmodelling.co.uk.
Activities and Start and End Events

Example – Raising a Purchase Requisition

© South East Training 2011
Process Activities and Events

Events

- Every diagram should have a start and end point.
- A thin circle at the start of a process is called a start event.
- A thick circle at the end of a process is called an end event.
- Events do not involve work.

Activities

- The rounded rectangles are called activities. They represent actions or work performed.
- Activities are named in the form: VERB-NOUN, e.g. Check Credit, Review Plan, Complete Form. If an activity cannot be described in this way, it is not an activity.

Sequence Flow

- The arrows, referred to as sequence flow, represent the logical flow.
- This diagram above illustrates the normal path if nothing goes wrong.
Example – Raising a Purchase Requisition With Exception Path

1. Identify Need
2. Check Budget
   - Budget OK?
     - Yes: Complete Requisition
     - No: Submit Requisition
       - No: Cancelled
       - Yes: Complete
Gateways and Exception Paths

Exception Paths

- Processes do not always operate as we would hope. The requisition process, for example, might fail because there is no budget for the products the originator wishes to purchase. This leads to the need for exception paths to be included in the diagram.

Gateways

- Exclusive or XOR gateways are used to separate exception paths from the normal path.
- The sequence takes one path or the other based on a logical condition. For example, is there budget provision - yes or no?
- This produces two end events. BPMN does not require multiple end events, but it is good practice to separate different end states.
Pools, Lanes and Sub-Processes

Lanes and Pools

- BPMN lets us identify who performs each activity by use of lanes. Lanes typically represent roles or organisational units such as departments.
- Lanes appear as subdivisions of pools. Pools typically represent processes. However, in some cases, they represent organisations.
- Labelling of lanes and pools is different. Pool labels appear in a box. Lane labels do not.

Activity Types

- It is also possible to distinguish between different types of activities by different labelling:
  - An activity with a human figure in the top left hand corner indicates this is performed by a person (a user task)
  - An activity with a cog in the top left hand corner is an automated task performed by a system (a service task)
  - Some people put automated or system related activities in separate lanes.

Sub-Processes

- An activity with a small cross in the middle of the bottom edge is a collapsed sub-process consisting of a flow of activities having a start event and one or more end events.
Example – Purchasing Process - Expanded or In-Line Sub-Process

© South East Training 2011
Expanded or In-Line Sub-Processes

Expanded or In-line Sub Processes

- One way to represent the ‘expanded’ sub-process is ‘in-line’, as shown.
- Note that the sub-process looks exactly like a process with a start event, a flow of activities, and end events.
- The start of the sub-process is triggered by the flow into it.
- The end of the sub-process depends on whether the normal or exception flow has been followed.
- The gateway appearing immediately after the ‘Raise Requisition’ sub-process directs the flow according to which end event has been reached within the sub-process.
- It is good practice to label the gateway with the same name as one of the end events; in this case, 'Proceed?'
Example - Purchasing Process – Hierarchical Expansion

Parent Diagram – Purchasing Process

Child Diagram – Raise Requisition Sub-Process

Child Diagram – Process Requisition Sub-Process

© South East Training 2011
Hierarchical Expansion of Sub-Processes

Hierarchical Expansion

- An alternative to in-line expansion is to use *hierarchical expansion*, with the expanded sub-process appearing on a different page (above they appear on the same page simply to save space).
- The diagram containing the collapsed sub-process is referred to as the *parent diagram* and the expanded sub-process is referred to as the *child diagram*.
- Note that the expanded sub-process omits the pool shape. Further, the sub-process does not have the large rectangle surrounding it as we see for the in-line expansions.
- Sub-processes can contain lanes.
Example – Process Requisition Sub-Process with Loopback to Handle Exceptions
Loopback to Handle Exceptions

Loopback

- BPMN allows the sequence flow to loop back to a previous step if an exception is encountered.
- In the example above, the process designer might have considered that it was unrealistic simply to end the process if Purchasing was unable to find a supplier able to meet the Originator's specification. A more realistic approach would be for Purchasing to go back to the Originator to discuss an alternative specification. No doubt Purchasing will have received feedback from potential suppliers, so will be in a better position to help the Originator refine their requirements.

Infinite Loops?

- Of course, using this approach, there is always the danger of ending up in an 'infinite loop' with the process never reaching an end point. This rarely happens in manual processes as humans usually intervene with an alternative solution. It could happen, though, if the process was automated. In this case, it would be pertinent to build in a counter or timer to tell the process when to stop looping.

Sequence Flow

- BPMN does not ascribe any significance to whether sequence flow enters an activity from the left, right, top or bottom. Nor does it specifically require lanes to run horizontally. What is important is the user can follow the sequence easily.
Example – Goods In Sub-Process with AND Gateways

- Quarantine Consignment
- Check Consignment
- Consignment Okay?
- Store Consignment
- Update Stock Record
- Process Faulty Consignment
- Notify Supplier
- Out of Stock
- Good In Stock

**AND Split** means both paths are enabled and the following activities can start.

**AND Join** means both preceding activities must be completed before the next can start.

© South East Training 2011
Parallel or AND Gateways

AND Gateways

- So far, we have seen the gateway used to direct flow along one path or another depending on a logical condition.
- BPMN, however, has a number of different gateways. In this example, we are seeing the use of the AND or Parallel Gateway.
- The process designer might decide that it does not matter whether the received goods are placed in store before or after the stock records are updated. Or indeed, if there is more than one person in the stores, the activities can be conducted simultaneously.

Splits and Joins

- The first AND Gateway, sometimes referred to as an AND-split or parallel-split, enables both Store Consignment and Update Stock Record activities to start at the same time.
- The second AND Gateway, sometimes referred to as an AND-join or synchronising-join, indicated that both sequence flows must complete before the next activity can start. In this case, there is no activity but the 'Goods in Stock' end state.
- BPMN does not require all parallel-splits to be merged downstream at synchronising-joins; they can lead to different end events but both end events must be reached to complete the process.
Example – Purchasing Process with Collaboration Diagram

- Identify Need
- Raise Requisition
- Process Requisition
- Inform Originator
- Complete

A Black-Box Pool is empty
Requests for Quotes
Quotes Purchase Order
Message Flows can only be drawn between pools
Message Flows attaches to the boundary of the Black-Box Pool

© South East Training 2011
Collaboration Diagram

The Black-Box Pool

- From time to time, we need to communicate with parties outside our processes. Typically, these will be customers, suppliers and regulators, but could be internal functions that we do not want to include in the model.
- We represent each of these parties by a Black-Box Pool. The Black-Box Pool is empty because we do not know (and, possibly, don’t care) what process the third party follows.

Message Flow

- In BPMN, the representation of communication between pools is called a collaboration diagram.
- Communication between our process and the third party is by means of message flow, represented by a dashed line with an unfilled arrowhead.
- Message flow can only be drawn between pools.
- Sequence flow, represented by a solid line and a filled arrowhead, can only be drawn within a pool.
- Message flow connects the boundary of the Black-Box Pool directly with the sub-processes, activities or events within the White-Box Pool.
- Note, you cannot connect message flow to a gateway.
Example – Purchasing Process with Originator Out of Scope

© South East Training 2011
Collaboration with Redefined Scope

Defining the Scope of the Process

- It makes sense to regard customers, suppliers and regulators outside the scope of any process as, generally, we have no control over their internal processes.
- However, sometimes it is also convenient to regard 'in-house' functions as outside the scope of the processes. This might be the case if, for example, we have not yet got round to mapping their part of the process or if their internal activities are irrelevant to the current improvement project.

Communicating with the Originator

- If the process is initiated by a message from a Black-Box Pool, the start event is shown as a message start event. This is represented by a thin circle with a clear envelope in the middle.
- The final event is a message end event, informing the originator that the purchase order has been submitted. This is represented by a thick circle with a filled-in envelope in the middle. Note, this replaces the 'Inform Originator' activity box that we used when initially describing the Purchasing Process.
- A message start event is one of a number of catching events used in BPMN. The event is 'catching' the request from the originator.
- A message end event is one of a number of throwing events used in BPMN. The event is 'throwing' a confirmation to the originator.
Example – Process Requisition Sub-Process Collaboration Diagram
Collaboration with Sub-Process

Collaboration with Sub-Process Elements

- The diagram above shows how we can expand the sub-process seen in the previous diagram to explore how the individual activities and events interact with the Black-Box Pools.
- It is important to note that the sub-process begins with a *none start event*, that is a plain circle, as the sub-process is triggered by the sequence flow coming into it. Sub-processes are never triggered by a catching event.
- Sub-processes, on the other hand, can include throwing events, and we will see examples of this later.
First Review – Basic Symbols

None Start Event – Means the trigger is unspecified. Sub-Process must have a None Start event.

Message Start Event – Means the process is triggered by an outside event. Cannot be used in Sub-Processes.

Timer Start Event – Means the process is triggered by a scheduled event.

None End Event – There may be more than one. No result signal is thrown when the event is reached.

Message End Event – Signifies a message is sent when the event is reached.

Terminal End Event – Immediately ends the process or sub-process even if other paths are still running. Used in a sub-process, it would end the sub-process but not the main process. No results signal is thrown.

Exclusive (XOR) Gateway – Has one sequence flow in and more than one out. Only one flow out can be followed.

Parallel (AND) Gateway – May represent a parallel split, with one sequence flow in and more than one out, or a parallel join with more than one sequence flow in but only one flow out. Parallel flows may join downstream or may lead to separate end events. Each parallel path must end for the process or sub-process to complete.

Sequence Flow – Links activities, gateways and events within a single pool. Orchestrates the sequence, meaning at when the node at the tail is complete, the node at the head is enabled.

Message Flow – Represents a signal sent between pools. It may not be used to connect nodes within a pool.

Link Event – Throwing – Link to next page.

Link Event – Catching – Link from previous page.
Summary of Basic Notation

Start Events

- Start events include: none start events, message start events and timer start events.
  - A none start event means the trigger is unspecified. Sub-processes must have a none start event.
  - A message start event means the process is triggered by an outside event. These cannot be used within a sub-process.
  - A timer start event means the trigger is a scheduled event. Again, these cannot be used within a sub-process.

End Events

- End events include: none end events, message end events and terminal events.
  - A none end event ends the process and no results signal is thrown.
  - A message end event signifies a message is sent when the event is reached.
  - A terminal end event immediately ends a process, even if other paths are still running. Used within a sub-process, it stops the sub-process but not the main process. No message is sent.

Gateways

- Exclusive or XOR gateways have one sequence flow in and more than one out. Only one sequence flow can be followed.
- Parallel or AND gateways can be used to represent a parallel split or a parallel join.
  - Parallel splits have one sequence flow in and more than one out; all paths are followed.
  - Parallel joins have more than one path in but only one flow out. All incoming paths must arrive before flow out.
- Parallel paths do not need to join downstream but all paths must complete for the process or sub-process to complete.

Sequence and Message Flow

- Sequence flow links activities, gateways and events in a single pool. You cannot have sequence flow between pools.
- Message flow represents a signal flowing between pools. You cannot have message flow within a pool.

Link Events

- Link events are used to link diagrams that spread over more than one page. There are two sorts:
  - Throwing link event - this throws the sequence to the next page
  - Catching link event - this catches the sequence flow from the previous page.
South East Training - Business Process Modelling Toolkit

Data Objects and Messages

South East Training 2011
Data Objects and Messages

Data Objects

- Sequence flow simply indicates the sequence in which events and activities occur. It does not infer the movement or transmission of data.
- To represent data or documents, we need to use data objects.
- Data objects can be linked to a sequence flow using a dotted line showing the association. In this format, the association is non-directional, with no arrow. The sequence flow indicates the direction. Care needs to be taken to ensure there is no ambiguity as to the movement of data if this method is used.
- Alternatively, data objects can be linked to activities or events using a dotted line with an arrow. In this case, the association is directional, as shown in the diagram above.
- The status of a data object is included in square brackets following the object name. In the example above, we have Goods Receipt Note [Rejected] and Goods Receipt Note [Accepted] following the normal flow and exception flows respectively.

Messages

- Message flow, as we have seen earlier, can only flow between pools.
- Rather than using data objects to indicate the flow of data or documents, we use an envelope, referred to as a message.
- The message flow represents the fact that a message is being sent, the message symbol (or envelope) represents the content.
South East Training - Business Process Modelling Toolkit

Splitting and Merging Recap

Splitting Exclusive Flow

Exclusive or XOR Gateway

Splitting Parallel Flow

Parallel or AND Gateway

Parallel paths emerging from an activity is the same as emerging from a Parallel or AND Gateway

© South East Training 2011
Splitting and Merging Recap

Before we can move on to looking at more complex splitting and merging patterns, it is worthwhile recapping what we have seen so far.

**Exclusive Flow**

The *Exclusive* or *XOR Gateway* is used to send the flow along one path as the result of an exclusive decision, typically the answer to a yes/no or closed question.

Downstream, paths may be merged using an *XOR Gateway* or by exclusive paths flowing directly into an activity. The following activity is initiated whichever path is enabled.

**Parallel Flow**

The *Parallel* or *AND gateway* (*AND split*) is used to send the flow along two or more paths simultaneously. An alternative is to have parallel paths emerging from an activity. In this case, the *AND Gateway* is not required.

Parallel paths may merge downstream using an *AND Gateway* (*AND join*). All incoming flows must reach the *AND Gateway* before it will permit onward flow. Alternatively, parallel flows can lead to separate end events. However, as with the *AND join*, both flows must reach their end events for the process to complete.

**Conditional Flow**

This recap now allows us to consider Conditional Flow.
**Splitting and Merging – Conditional Flow**

**Conditions are independent**

- **Default or 'otherwise' path**

  - OR Gateway is used to direct flow along all paths where the conditions are met. In the example above, flow will always pass to B but only to A and C if the relevant conditions are met. Flow is parallel for all true conditions.

  - This example differs from the one on the left in that the 'always' or 'unconditional' path has been replaced by the 'default' or 'otherwise' path indicated by the tick. Flow to B will only occur if neither of the other two conditions is met.

**Conditional Sequence Flow** offers an alternative way to model the situation shown in the diagram on the top left. Conditional sequence flow shown; by the diamond on the tail of the arrow, can only flow out of an activity. The diagram only the top right, illustrating the default, or otherwise path can be drawn in a similar way.

- The OR Gateway can also be used to merge flows that are neither alternative (from an XOR gateway) nor unconditional parallel (from an AND split). Neither the XOR nor the AND gateway would do the job. The OR Gateway waits for all enabled paths to complete.

© South East Training 2011
Splitting and Merging Conditional Flow

Splitting

The *OR Gateway* is used to direct flow along all paths where the conditions are met. In the example on the top left, flow will ‘always’ pass to B but only to A and C if the relevant conditions are met. A and C could represent monetary limits or levels or risk, for example.

The example on the top right is similar, but the ‘always’ or ‘unconditional’ path has been replaced by the ‘default’ or ‘otherwise’ path, indicated by the tick. Flow to B will only occur if neither of the other two conditions, A and C, is met.

*Conditional Sequence Flow* offers an alternative way to model the two examples given above. Instead of the *OR Gateway*, a diamond on the tail of the sequence flow arrow is used to indicate *Conditional Sequence Flow*. This can only flow from an activity, not out of an event or a gateway. The diagram on the bottom left illustrates how the situation on the top left can be modelled. The diagram on the top right, illustrating the use of the ‘default’ or ‘otherwise’ path, can be drawn in a similar way.

Merging

The *OR Gateway* can be used to merge flows that are neither alternative (from an *XOR Gateway*) nor unconditionally parallel (from an *AND split*). Neither the *XOR* nor the *AND Gateway* would do the job. The *OR Gateway* waits for all enabled paths to complete.
More Start Events

- None Start Event – Means the trigger is unspecified. Sub-Process must have a None Start Event.

- Message Start Event – Process is triggered by an external signal, typically a customer request that is directed at the process.

- Timer Start Event – This signifies a scheduled event. In this case every Monday at 8am.

- Conditional Start Event – This signifies triggering by a ‘watched’ data condition, say inventory levels.

- Signal Start Event – This is triggered by ‘listening’ for a signal that is broadcast to any listening process. For example, the addition of a new customer onto the company database may trigger a number of processes, such as marketing or supply of customer information to a partner organisation.

- Multiple Start Event – This signifies triggering by any one of multiple signals, such as a message flow from two or more pools or, as in this case, an incoming message or a scheduled event.
More Start Events

- We have already seen the none start event and the message start event used in previous examples. The diagram above explains the use of four more start events, the timer start event, the conditional start event, the signals start event and the multiple start event.
- The second diagram illustrates how we can deal with more than one start event.
More than One Start Event

Channel Dependent Start – Different triggers initiate different sequence flows. Once triggered, the process ignores any subsequent signals received by other start events. Any new signal is considered as a new occurrence of the process.

Multiple Start Event – Different triggers initiate the same sequence flow.
More Than One Start Event

- The first example above shows a *channel dependent start*. This is where different triggers initiate different sequence flows. In this case, orders received by phone will initially be treated differently from those received over the internet.
- The second example shows a *multiple start event* where triggers from different sources initiate the same sequence flow. In this case, we have customers and stock-brokers contacting the organisation to purchase shares.
More End Events

- None End Event – There may be more than one. No result signal is thrown when the event is reached.

- Message End Event – Signifies a message is sent when the event is reached.

- Signal End Event – This broadcasts a message to any listening process. As it is not directed to any particular process, there is no message flow emitted.

- Error End Event – This immediately ends the process level, even if parallel paths are still active. It sends a signal to the parent level to initiate an exception path.

- Terminate End Event – This immediately ends the process level, even if parallel paths are still active. It does NOT send a signal to the parent level. Flow continues out of the sub-process on the normal flow – see example.

- Cancel End Event – This is a special form of Error Event. It is only used in Transactional Sub-Processes, otherwise it operates in exactly the same way as the Error End Event.
More End Events

End Events

- We have already seen the *none end event* and the *message end event* used in previous examples. The diagram above explains the use of four more end events, the *signal end event*, the *error end event*, the *terminal end event* and the *cancel end event*.
- The second diagram illustrates the use of the terminal and error end events to end a sub-process immediately when an exception is encountered.
- The diagrams on the next page illustrate how the in-line sub-processes shown in this example will appear in their collapsed format.
South East Training - Business Process Modelling Toolkit

End Events – Some Examples – In-Line Sub-Processes

© South East Training 2011
More End Events - Examples with In-Line Sub-Processes

- The first diagram above shows the use of the *terminal end event*.
  - Reaching the terminal end event immediately ends the process level, even if parallel paths are still active.
  - It does not send a signal to the parent level.
  - Flow continues out of the sub-process in the normal way.
  - An XOR gateway is needed after the sub-process to direct flow along the appropriate path depending on whether or not the sub-process completed normally.
  - In the example, a failure during either the technical or financial review to show that the business case met the specification would cause the sub-process to stop immediately.

- The second diagram shows the use of the *error end event*.
  - The failure of either the technical or financial review causes the sub-process to stop immediately, as with the terminal end event.
  - The difference is that the error end event sends a signal to the parent level via an *intermediate error event* on the boundary of the sub-process.
  - The intermediate error event initiates the exception flow, therefore, there is no need for an XOR gateway following the sub-process.
End Events – Some Examples - Collapsed Sub-Processes
More End Events - Examples with Collapsed Sub-Processes

- The diagram above simply illustrates how the process diagrams would look if the in-line sub-processes were collapsed.
- Use of the *cancel end event* is shown in a later example.
South East Training - Business Process Modelling Toolkit

Sub-Processes – Some Examples

Ad-Hoc Sub-Processes - Example

Transactional Sub-Processes - Example

© South East Training 2011
More Sub-Processes

Sub-Processes

Sub-processes are useful for numerous reasons. For example:

- They allow you to model end-to-end process without having to illustrate the detail (which can make the model too large to handle).
- The collapsed sub-process allows you to include in the end-to-end process those sub-processes that you have yet to model, without damaging the integrity of the overall structure.
- They help define boundaries so that process ownership can be defined at different levels.

Examples

Two useful sub-process types, that we have not so far seen, include:

- **The Ad-Hoc Sub-Process** - The expansion of the ad-hoc sub-process does not include sequence flow but instead a list of activities that could be performed. The order of execution is not specified and it is not necessary to complete all the activities to complete the sub-process.
  - The example illustrates what might happen when someone takes out a new gym membership. Various members of staff may offer a variety of services including: ordering a new welcome pack; setting-up an induction; signing you up for classes; and introducing you to a personal trainer. It does not matter in which order they are conducted.

- **The Transactional Sub-Process** - Denoted by a double boarder, this is a special type of sub-process that illustrates a single business transaction. If the transaction sub-process does not execute successfully, the system must be returned to the state it was in before the transaction took place.
  - The example illustrates the buying of tickets for the 2012 Olympics. Following the ballot, tickets were allocated and the payment was taken some days or weeks later. However, if the credit card payment failed, as it did in some cases, the allocation had to be cancelled so the tickets could be offered for resale. The transactional sub-process offers a convenient way of modelling this.
Intermediate Events

- **Message Intermediate Event (Catching)** – Process is triggered by a signal, typically a customer request that is directed at the process.

- **Message Intermediate Event (Throwing)** – Process sends a signal to another pool, typically an order acknowledgement.

- **Timer Start Event** – This signifies a scheduled event. In this case every Monday at 8am.
Intermediate Events

- So far, we have seen the use of start and end events to define the boundaries of processes and sub-processes. Start events act as triggers to initiate sequence flow while end events define the end state of a path.
- BPMN, however, also includes a variety of intermediate events that can be used in sequence flow or attached to the boundaries of activities or sub-processes.
- The diagram above shows two versions of the *message intermediate event*, one catching (similar to the message start event) and one throwing (similar to the message end event).
- The other is a *timer intermediate event*. The timer intermediate event represents a delay in the flow. It is often used in *event gateways* where it initiates an exception path after a given period if the normal flow is delayed waiting for a different event to occur, e.g. the receipt of a message.
Intermediate Events - Examples

Delay Using a Timer Event - In this example, HR is waiting to see if it has received enough nominations for a training event before scheduling. The timer event introduces a delay of 3 days between each check. Once 4 or more nominations have been received, HR schedules the event.

Exclusive Event Gateway - The gateway is initiated by whichever following intermediate event occurs first. In this case, the bidder is expecting a response, either positive or negative, within a week. If no response is received within that time, the bidder contacts the customer for an update.
Intermediate Events - Examples

Delay using an Intermediate Timer Event
The top diagram shows how the intermediate timer event might be used to introduce a delay. In the example, HR is waiting to see if they have enough interest in a training event before scheduling it. Rather than doing this continuously, they have decided to check numbers at three-day intervals. The XOR Gateway will only allow scheduling once four or more nominations have been received, otherwise, another three-day delay is introduced.

Event Gateway
The diagram on the bottom illustrates the use of an Event Gateway. The gateway is initiated by which-ever following intermediate event occurs first. In this case, the bidder is expecting a positive or negative response within a week. Receipt of a response will trigger the appropriate intermediate message event to initiate the following activity. If no response is received within that time, the intermediate timer event will open the path to allow the bidder to contacts the customer for an update.
South East Training - Business Process Modelling Toolkit

**Boundary Events**

- **Boundary Message Event** – A message is received from another pool interrupting the sub-process. This produces an exception flow.

- **Boundary Timer Event** – The sub-process is interrupted if the sub-process does not complete within the specified time.

- **Boundary Error Event** – The sub-process is interrupted by an error occurring within the sub-process, such as a bad credit notification.

- **Transactional Sub-Process Cancel Event** – The sub-process is interrupted by an exception occurring within the sub-process that requires the sub-process to be cancelled and reset to its starting position.

© South East Training 2011
Boundary Events

- A common use of intermediate events is to channel flow onto the exception path from the boundary of an activity or sub-process. We have seen the use of the boundary error event and the cancel event when we looked at transactional sub-processes.
- Two additional boundary events are shown: the boundary message event that is listening for a signal from within the sub-process and the boundary timer event that will channel flow along the exception path if the sub-process does not complete within a given period of time.
- In this diagram they are shown as interrupting boundary events, meaning the sub-process will be interrupted if the boundary event receives a signal.
- BPMN also offers designers the use of non-interrupting boundary events that allow the sub-process to complete normally but initiate a parallel exception path. A typical use of this facility would be to inform a supervisor or customer that a process was running late. You would not want to interrupt the process but you would need to let others know what was happening.
Boundary Events - Examples

Interrupting message boundary event

Interrupting timer boundary event

© South East Training 2011
Boundary Events - Examples

- The first examples shows the use of the interrupting message boundary event.
  - Here the process of preparing an order for shipment would be cancelled immediately if the customer chooses to cancel the order while the sub-process was active.
  - An alternative would be to use a transactional sub-process with a cancel event at the boundary. Resetting the transaction would mean putting goods for shipment back into the stock.
  - If the customer, however, simply wanted to amend the delivery address, you would use a non-interrupting message boundary event that would allow the sub-process to complete normally while updating the delivery details

- The second examples shows an interrupting timer boundary event.
  - Here, the HR receives a request for some training. The first option is to explore their internal capacity to meet this need. If they cannot do so within a week (one of their KPIs perhaps), they need to look elsewhere and commission an external trainer.
Repeating Activities

Looping Activity – This structure will continue to loop until the condition specified by the second XOR gateway is no longer met.

Collect 10 Nominations

Looping Activity – BPMN offers an alternative notation. The circular arrow at the bottom of the activity box indicates that, as above, the activity will loop until the condition is no longer met. This will need to be specified in the diagram.

Evaluate 10 Nominations

Looping Activity – The same notation can be applied to sub-processes but not to events.

Interview Candidates

Multi-Instance Activity – This is similar to a looping activity. Two key differences are that the number of iterations is known and the activities can be performed in parallel. So in this case, we could have two or more interviews taking place at the same time.
Repeating Activities

Repeating Activities - Basic Notation

- Basic notation allows us to model simple repeating activities by using a series of gateways.
  - In the first example above, the intermediate message event will pass the flow immediately to the following gateway as soon as it receives one nomination.
  - Without the gateway, the sequence flow would continue to process this single nomination.
  - In practice, it is likely that we will want to batch-process all nominations received. To achieve this we add an XOR gateway after the intermediate message event that redirects the flow to an earlier point in the sequence if the condition specified continues to be met. In this case, flow will only continue along the normal path when 10 nominations have been received or a deadline has been reached.
  - Note, the first gateway is a convenient device to join the sequence flow upstream of the intermediate message event.

Repeating Activities - Advanced Notation

- BPMN offers a simpler way to represent this same sequence.
  - In the second example above, we have used a 'looping activity', denoted by the circular arrow at the bottom of the box, to indicate that the activity will keep looping until the specified condition is met. In this case, normal flow will continue only once 10 nominations have been received.
  - The third example shows that we can apply the same notation to a sub-process. We cannot use it, however, with events.
  - The fourth example represents a different condition. The three vertical lines at the bottom of the sub-process box indicate that this is a multi-instance activity (or sub-process). It is similar to a looping activity but there are two distinct differences:
    - For a looping activity, we do not know the number of instances and have to set a condition to terminate the activity. With a multi-instance activity, we do know the number of instances.
    - For a looping activity, instances are dealt with sequentially. With a multi-instance activity, instances can be dealt with concurrently. In the example above, you could have two or more interviews being conducted at the same time.
South East Training - Business Process Modelling Toolkit

Batch Processing

© South East Training 2011
Batch Processing

A good use of repeating activities is in batch processing.

In this example, the real time ‘Dealing Process’ is initiated every time an instruction is received from a client or an advisor. However, this does not immediately initiate a purchase. Instructions are batched in order to create a bigger order, which in turn helps reduce the dealing costs per instruction.

The looping receiving task captures the instructions each time the Dealing Process runs. There are no conditions on the looping activity as we do not know how many instructions will be received before 14.00 each day.

The batch process runs once per day at 14.00. All instructions received from the real time process up to 14.00 are then batched and presented as a single order to the stock exchange. Dealing advice is then sent to the various advisors or clients.