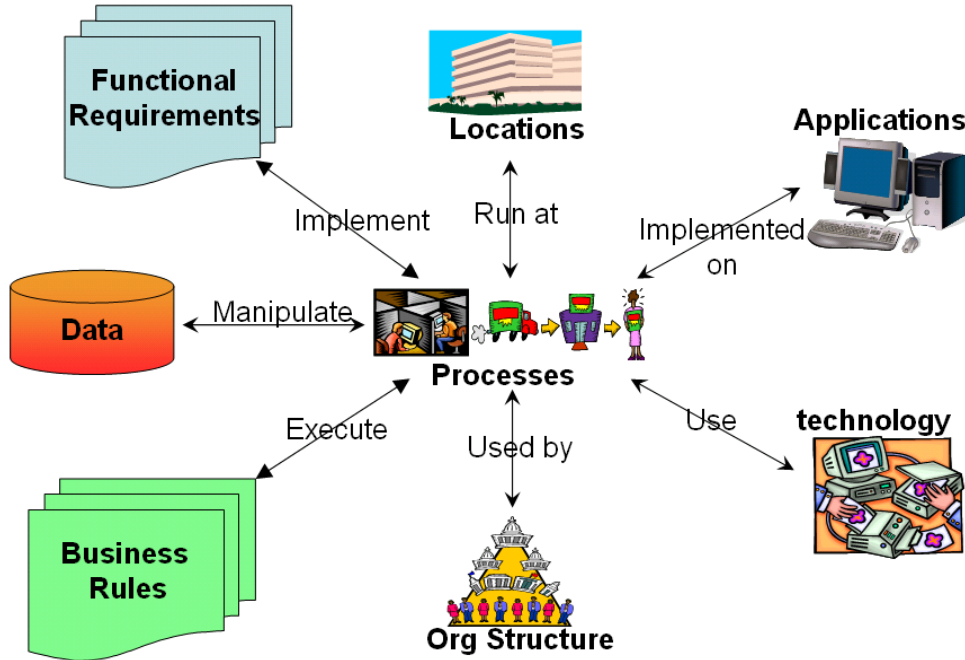


Why & How: Business Process Modelling

Introduction

The following diagram illustrates how process specifications are the pivot around which solutions are specified, designed and implemented:



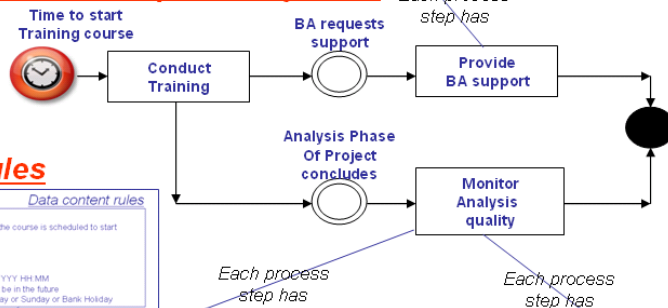
Processes are the user interface to a solution – as such they are highly visible and a lot of project effort is focussed on process modelling. In fact, there is a perception in some quarters that Business Analysts just draw process models and this is all they do. However, process models (the drawings of processes) are only one facet of the specification of a process. These are the other facets:

- A BA can request one of 4 types of support:
1. Phone or email based query about a specific point
 2. Informal review of a project deliverable
 3. Formal review of full set of project deliverables
 4. Facilitated workshop of how to apply analysis to a specific project

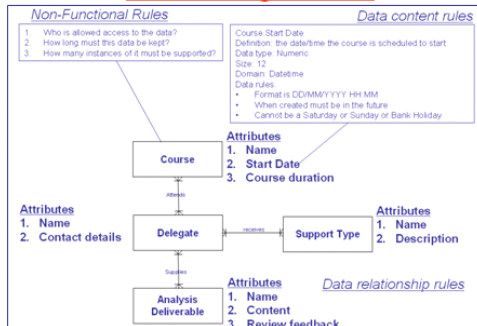
Process execution rules

1. In the case of phone or email query about a specific point the BA poses the question and the training provider will provide guidance for how the technicalities of Business Analysis apply to the problem
- ... etc

Process dependency rules



Data Usage rules



Non-Functional Rules

1. Who is interacts with process
2. Where they are
3. Availability of process
4. Volumetrics
5. Performance of process
6. Security & Authorisation levels

Why & How: Business Process Modelling

This article will look at the reasons why process modelling should be done (the benefits) and the techniques for

- drawing process models
- managing process decomposition
- documenting process execution logic
- documenting process non-functional requirements
- mapping processes to functional requirements
- mapping processes to data used (but not how to model data requirements – an article in it's own right!)

Finally – you will be offered the opportunity to have a go at process modelling yourself and get a review from an experienced Business Analyst.

Section 1: Why do process modelling?

1. Why do process modelling? ...Because of the benefits of process modelling

a. Definition of what is process modelling

For the purposes of this article, a process is a connected series of business activities that result in a significant business outcome, and a process model is the specification of a process sufficient to allow a solution to be designed, developed and implemented (computerised or not). From this it should be obvious that process modelling is all the activities that need to be undertaken to document a process.

b. The uses of process models and the risks of poor/missing process models and specifications

Use	Description	Risks if process modelling not used for this
Defining the as-is process.	This is the starting point for a change to processes	<ul style="list-style-type: none"> • Project overrun due to scope of process change not being understood – this makes planning that much harder. • Reduced project benefits due to missing process requirements (a lot of existing requirements typically carry forward to the new solution).
Defining the to-be process.	The end point for a change to processes	<ul style="list-style-type: none"> • Project overrun due to scope of process change not being understood – this makes planning that much harder. • Reduced project benefits due to missing process requirements.
Defining the scope of the data requirements.	processes manipulate data (a process that is not manipulating data is not doing anything) and therefore there will be a set of data that each process needs in order to be able to execute. The scope of the project data requirements is the set of data needed by all the processes in order to be able to execute.	<ul style="list-style-type: none"> • Project overrun due to scope of data change not being understood – this makes planning that much harder. • Reduced project benefits due to missing data requirements.
Designing the solution.	the process model defines requirements from a process perspective and the solution must propose the actual solution that will achieve these process requirements. Some parts of the processes will be automated and some will not. Most will provide automated support for user activities.	<ul style="list-style-type: none"> • Project overrun due to rework required if testing shows that the solution is not satisfying all the requirements. • Reduced project benefits if the solution does not satisfy all requirements.
Testing the solution including User Acceptance Testing.	Regardless of the technicalities of the design, the solution must achieve the requirements that will in turn yield the project benefits. Testing that the solution achieves the process requirements validates that the process requirements and related benefits are being delivered.	<ul style="list-style-type: none"> • Project overrun due to rework required if errors are not found until post testing. • Reduced project benefits if the decision is taken not to fix the errors that are not found until post testing.

Why & How: Business Process Modelling

Use	Description	Risks if process modelling not used for this
Procedure design.	The instructions for operating the process must cover as a minimum the process requirements (there will in all likelihood be more than this required: additional instructions to operate the physical solution that are concerned with the technicalities of the solution and not the process requirements).	<ul style="list-style-type: none"> Reduced project benefits if the procedures result in the processes not being operated as efficiently as they could be.
Training design.	As a minimum the training must cover all the activities defined in the process models (there will in all likelihood be more than this required: additional training to operate the physical solution that are concerned with the technicalities of the solution and not the process requirements).	<ul style="list-style-type: none"> Reduced project benefits if the training does not instruct delegates on all the processes and/or on the best way to operate all the processes.

2. Why do process modelling?...Because process should be the focus of design and development

a. Functional requirements result in processes

Every functional requirement will have *at least* one process that enables that functional requirement to be delivered as part of the solution. This is because functional requirements are statements about what the solution needs to be able to *do* – and doing something is a process. Example functional requirement for a double glazing company: “Be able to record sales”. The processes that deliver this functionality may be “Record Sale” and also (in the case of a new customer) “Create Customer”. This example shows that one functional requirement might need more than one process to satisfy it. The reverse is also true: perhaps “Create Customer” is also required for the functional requirement “Be able to provide quote”.

Development should focus on ensuring that processes are delivered because when all processes have been delivered, so have all functional requirements.

b. Processes specify business rules

Process specifications are concerned with specifying different types of rules (that is process dependency rules, process execution rules, process non-functional rules and process data usage rules). All these rules must be enforced in order that the business can deliver the project benefits. These rules have nothing to do with systems or operational design, only what is *required* in order to deliver benefits.

Development should focus on enforcing these rules because when all the rules are delivered, the processes are all delivered.

c. Process that don't manipulate data don't do anything

As has been mentioned, processes create, read, update and delete data. A process that is not doing that is not doing anything at all (unless there is some functional requirement to do nothing!!!). In addition, the only way to manipulate data is via a process. It follows then that the data required by processes in order to be able to operate is the scope of the data required by the solution. Database and forms (computerised or not) designers therefore need to focus on data required by processes. Once all data required by all processes has been designed then all data requirements have been satisfied.

Section 2: How to do process modelling

1. Modelling process dependency rules using Business Process Modelling Notation (BPMN)

Note: this part of the article concerned only with how to do process modelling. If you are already familiar with how to do process modelling and/or are not interested in learning about BPMN then please go straight to the conclusion.

This article uses Business Process Modelling Notation (BPMN) notation as it is as good as any other and is meant to be the standard notation. Having said that, the standard is not (and can not) be enforced. In addition, ‘standards’ in Business Analysis seem to have habit of coming and going – only time will tell if this standard will persist or not. It is worth stressing though that all process model notations model processes and therefore cover the same ground and therefore roughly do the same thing using different symbols and terms. That means the concepts that each process model notation consists of will have equivalents in the other notations even if the symbols and names of the symbols vary.

This article is not an exhaustive training course in BPMN and covers only the main symbols and concepts.

BPMN consist of symbols for different types of


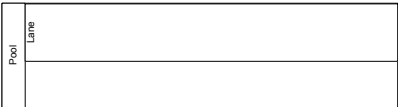
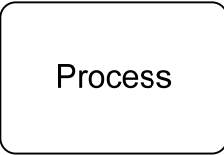
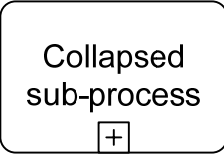
- swimlane
- activities
- events
- gateways
- flows

The following table shows the different symbol types and their meaning

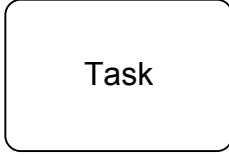
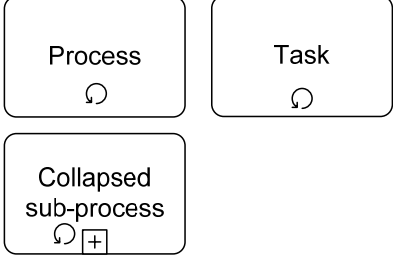
(most definitions taken from <http://www.bpmn.org/Documents/BPMN%201-1%20Specification.pdf> and <http://www.modernanalyst.com/Resources/Templates/tabid/146/articleType/ArticleView/articleId/487/Default.aspx>)

Type of symbol	Symbol name	Symbol	Description
Swimlane			A Swimlane is a graphical container for partitioning a set of activities from other activities. BPMN has two different types of Swimlanes.









Why & How: Business Process Modelling

Type of symbol	Symbol name	Symbol	Description
	Pool		<p>A Pool represents a Participant in a Process. It also acts as a “swimlane” and a graphical container for partitioning a set of activities from other Pools, usually in the context of B2B situations. It is a square-cornered rectangle that is drawn with a solid single line. A Pool acts as the container for the Sequence Flow between activities. The Sequence Flow can cross the boundaries between Lanes of a Pool, but cannot cross the boundaries of a Pool. The interaction between Pools, e.g., in a B2B context, is shown through Message Flow.</p>
	Lane		<p>A Lane is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally. Lanes are used to organize and categorize activities within a Pool. The meaning of the Lanes is up to the modeller.</p>
Activity			<p>Always named Verb phrase and a Noun phrase: do something to something. An activity is a generic term for work that a company or organization performs via business processes. An activity can be atomic or non-atomic (compound). The types of activities that are a part of a Process Model are: Process, Sub-Process, and Task.</p>
	Process		<p>A Process is any activity performed within a company or organization. In BPMN a Process is depicted as a network of Flow Objects, which are a set of other activities and the controls that sequence them.</p>
	Sub-process		<p>A Sub-Process is Process that is included within another Process. The Sub-Process can be in a collapsed view that hides its details. A Sub-Process can be in an expanded view that shows its details within the view of the Process in which it is contained. A Sub-Process shares the same shape as the Task, which is a rectangle that has rounded corners.</p>

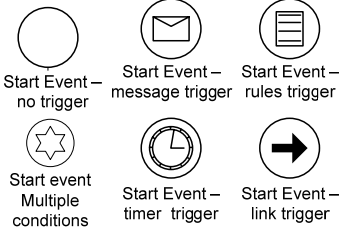
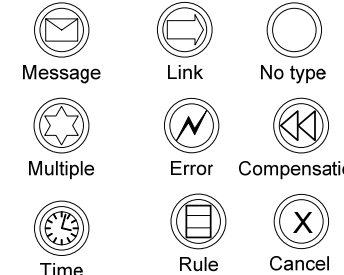

Why & How: Business Process Modelling

Type of symbol	Symbol name	Symbol	Description
	Task		<p>A Task is an atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail. Generally, an end-user and/or an application are used to perform the Task when it is executed. A Task object shares the same shape as the Sub-Process, which is a rectangle that has rounded corners.</p>
	Iteration		<p>To show that a process, sub-process or task can be performed 1 thru 'n' times defined by the processing logic in the iteration.</p>
Event			<p>An event is one of:</p> <ul style="list-style-type: none"> • something that occurs externally to the solution that the solution has to respond to • a break in the process that means the process will not resume until another external event occurs • a product the solution produces as output and that product leaves the scope of the solution • the solution terminating. <p>The event may have a symbol inside that indicates the type of event. Not all symbols are relevant to all types of events.</p>

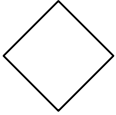






Why & How: Business Process Modelling

Type of symbol	Symbol name	Symbol	Description
	Symbols	 Message  Link  Multiple  Error  Compensation  Time  Rule  Cancel	<ul style="list-style-type: none"> • Message: data that arrives to or is produced by the solution. • Link: A Link is a mechanism for connecting two sections of a Process. Link Events can be used to create looping situations or to avoid long Sequence Flow lines. Link Event uses are limited to a single Process level (i.e., they cannot link a parent Process with a Sub-Process). Paired Intermediate Events can also be used as “Off-Page Connectors” for printing a Process across multiple pages. They can also be used as generic “Go To” objects within the Process level. There can be multiple Source Link Events, but there can only be one Target Link Event. • Multiple: any combination of the types of event. • Error: an error has occurred and this symbol will be in an event physically on the border of the process that can generate it. • Compensation: undoing activities that have taken place. • Time: A specific time-date or a specific cycle (e.g., every Monday at 9am) can be set. • Rule: a Condition such as “S&P 500 changes by more than 10% since opening,” or “Temperature above 300C” become true. • Cancel: used within a Transaction Sub-Process to cancel their execution


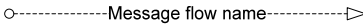

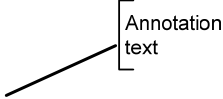
Why & How: Business Process Modelling

Type of symbol	Symbol name	Symbol	Description
	Start event	 <p>Start Event – no trigger Start Event – message trigger Start Event – rules trigger</p> <p>Start event Multiple conditions Start Event – timer trigger Start Event – link trigger</p>	<p>A Start Event indicates where a particular Process will start. In terms of Sequence Flow, the Start Event starts the flow of the Process, and thus, will not have any incoming Sequence Flow. A Start Event can have a Trigger that indicates how the Process starts: Message, Timer, Rule, Link or multiple (any combination). The Start Event shares the same basic shape of the Intermediate Event and End Event, a circle, but is drawn with a single thin line.</p>
	Intermediate event	 <p>Message Link No type</p> <p>Multiple Error Compensation</p> <p>Time Rule Cancel</p>	<p>An Intermediate Event is an event that occurs after a Process has been started. It will affect the flow of the process, but will not start or (directly) terminate the process. An Intermediate Event will show where messages or delays are expected within the Process, disrupt the Normal Flow through exception handling, or show the extra flow required for compensating a transaction. The Intermediate Event shares the same basic shape of the Start Event and End Event, a circle, but is drawn with a thin double line.</p>
	End event	 <p>No type Message Compensation</p> <p>Error Terminate Link</p> <p>Multiple Cancel</p>	<p>As the name implies, the End Event indicates where a process will end. In terms of Sequence Flow, the End Event ends the flow of the Process, and thus, will not have any outgoing Sequence Flow. An End Event can have a specific Result that will appear as a marker within the center of the End Event shape. End Event Results are Message and Link. The End Event shares the same basic shape of the Start Event and Intermediate Event, a circle, but is drawn with a thick single line</p>
Gateway			<p>A Gateway is used to control the divergence and convergence of Sequence Flow. Thus, it will determine branching, forking, merging, and joining of paths. Internal markers will indicate the type of behavior control. There are five types of gateways.</p>

Why & How: Business Process Modelling

Type of symbol	Symbol name	Symbol	Description
	Exclusive or	 Xor condition  Xor condition	Data-Based exclusive decision and merging. Shown with or without the “X” marker – i.e. the default is Xor.
	Inclusive or	 Inclusive or condition	Inclusive decision and merging
	And	 And condition	Simultaneous forking or joining
	Complex	 Complex condition	Complex conditions and situations (e.g., 3 out of 5 times go one way, 2 times out of 5 go the other)
	Event based gateway	 <i>(this symbol is not named)</i>	An “exclusive or” decision made on the basis of whichever associated intermediate event occurs first.
Flow			A Flow is a graphical line connecting two objects in a BPD. There are two types of Flow: Sequence Flow and Message Flow, each with their own line style. Flow is also used in a generic sense (and lowercase) to describe how Tokens will traverse Sequence Flow from the Start Event to an End Event.
	Sequence flow		Flow that proceeds, unrestricted, from one Flow Object to another, via a Sequence Flow link, without any dependencies on another flow or any conditional expressions. Typically, this is seen as a Sequence flow between two activities, without a conditional indicator (mini-diamond) or any intervening Gateway.

Why & How: Business Process Modelling

Type of symbol	Symbol name	Symbol	Description
	Sequence flow with condition		Flow that proceeds from one Flow Object to another, via a Sequence Flow link, but is subject to either conditions or dependencies from other flow as defined by a Gateway. Typically, this is seen as a Sequence flow between two activities, with a conditional indicator (mini-diamond) or a Sequence Flow connected to a Gateway.
	Message flow		A Message Flow is a dashed line that is used to show the flow of messages between two entities that are prepared to send and receive them. In BPMN, two separate Pools in the Diagram will represent the two entities.
	Data Object		In BPMN, a Data Object is considered an Artefact and not a Flow Object. They are considered an Artefact because they do not have any direct affect on the Sequence Flow or Message Flow of the Process, but they do provide information about what the Process does. That is, how documents, data, and other objects are used and updated during the Process. While the name “Data Object” may imply an electronic document, they can be used to represent many different types of objects, both electronic and physical.
Text annotation			Can be applied to any object on a BPMN diagram to add clarity. Is for communication purposes only and does not form part of the specification.

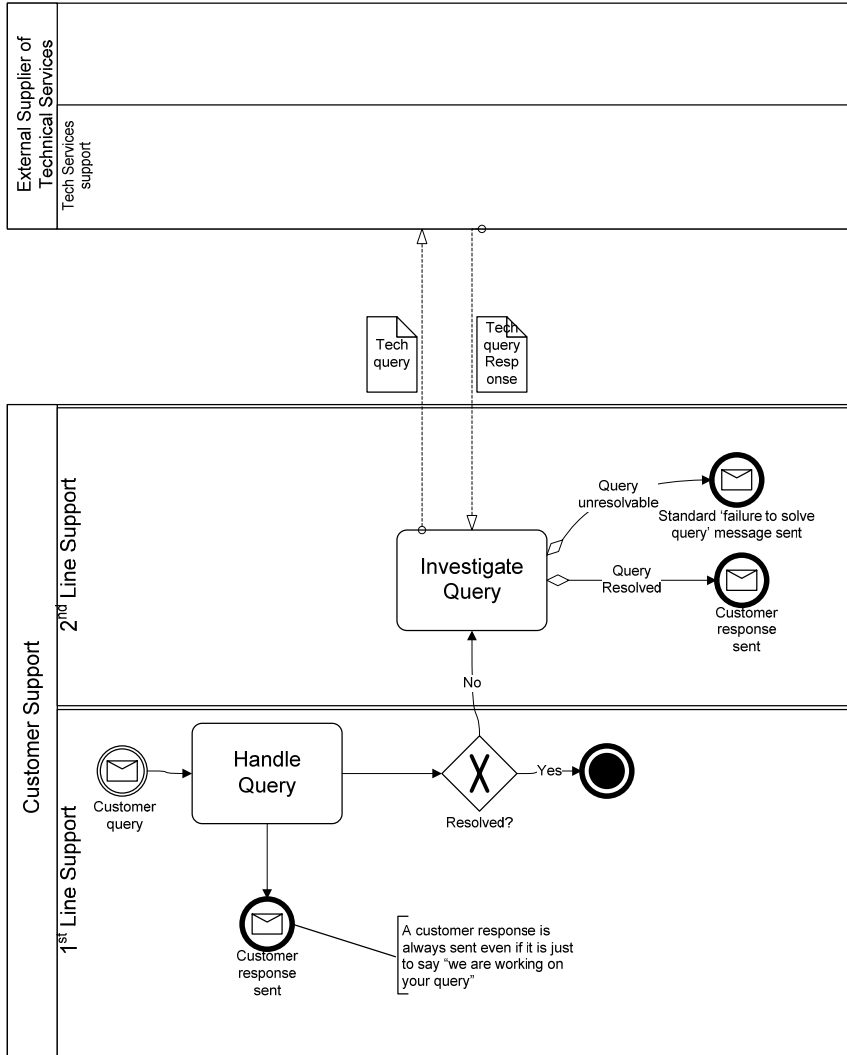
Why & How: Business Process Modelling

A process model defines dependencies between process steps.

Have a look at the following example process model. You should be able to read and understand how the process

- starts
- who is involved with the process
- what processes it goes through
- under what conditions which flows will be followed
- how it terminates

The business process being modeled is customer query handling by 1st and 2nd line support teams, who use an external agency to solve technical queries.



Notes:

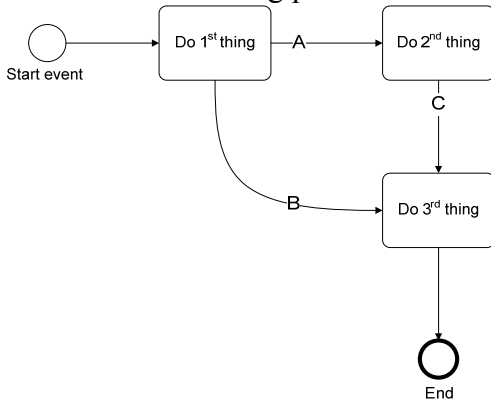
1. There is an xor gateway called 'resolved' which does essentially the same job as the two conditional flows coming from "investigate query". The only difference is that although you would expect the two conditional flows to be mutually exclusive the diagram does not enforce the rule – conditional flows will be followed when the conditions are satisfied and two conditions could be satisfied at once (see next example).

Why & How: Business Process Modelling

- The default for BPMN for flows leaving a process, sub-process or task is that they all leave at the same time unless there is some rule in the diagram that says they cannot – in the example above the process flow from “handle query” to “customer response sent” and to the gateway “resolved?” both occur as that process step concludes.

Some more notes about BPMN usage:

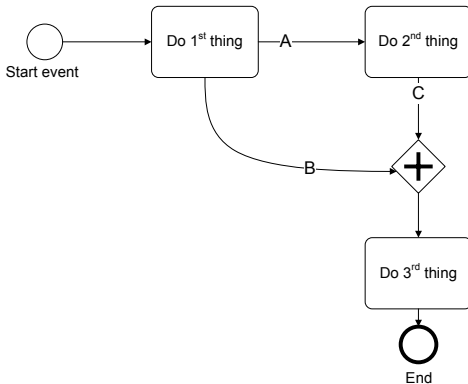
Consider the following process



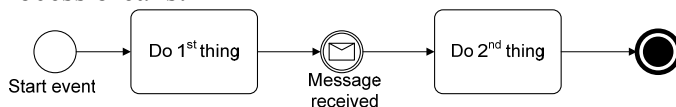
The task “Do 1st Thing” has 2 flows (A and B) leaving it. As both flows are unconditional flows, these would be followed unconditionally *and simultaneously* every time.

The task “Do 3rd Thing” has 2 flows (A and C) coming in to it. The default notation for BPMN is that **either** flow A **or** flow C can trigger the task.

If you wanted to show the business rule that flows A **and** C are required before the task can start you would show it like this:



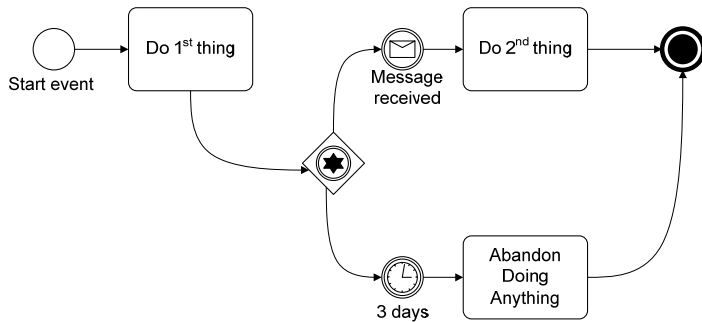
Process breaks:



There is a process break between “Do 1st Thing” and “Do 2nd Thing”. That break comes to an end when the intermediate event “message received” occurs.

Suppose the business rule is that the message must be received within 3 days or the process stops:

Why & How: Business Process Modelling



Note the use of the event based gateway. It shows that there is a process break here which will end when one of any number of events occurs.

Further reading:

BPMN1.1. <http://www.bpmn.org/Documents/BPMN%201-1%20Specification.pdf> Section 11 – BPMN by example for extended model – note that this contains some notation we have not covered.

<http://www.bpmn.org/exampleIndex.htm> - a set of example BPMN diagrams – note that these contain some notation we have not covered.

<http://www.businessprocesstrends.com/publicationfiles/07-04%20WP%20Intro%20to%20BPMN%20-%20White.pdf> a good introduction from Stephen White who is on the OMG that controls BPMN with a discussion at the end about where BPMN may evolve to.

2. Process execution logic rules

The process/task specification defines the execution logic that the process implements.

Processes (by BPMN definition) are not the bottom or atomic layer in a process model – they will always be decomposed: sometimes to sub-processes, sometimes to tasks (the bottom layer).

Definition of the process is therefore going to be a summary definition of the sub-processes and tasks within them.

It is not feasible (or desirable) to specify precise execution logic for a process as it would be impossible to ascertain which task each statement belonged to.

What is useful to define at the process level is

1. Description of the process at the level shown on the diagram.
2. Task Metrics (see next part: **3. Process non-functional rules**) provided that they apply only to the process at the level it is shown on the diagram.

Tasks (by BPMN definition) are the bottom or atomic layer of a process model.

Therefore it will be necessary to specify

1. a description of the task
2. the precise logic that must be executed by the task.

In addition, the following should be specified *where not present in the processes that the task is a part of*:

Task Metrics (see next part: **3. Process non-functional rules**)

Description of a process/task: a natural English description of what the process is for and an overview of how it does it.

Example description for Take Order process:

“The process Take Order captures the details of a customer instruction for a sale.

First the customer is found by using certain search criteria (if not found a customer can be created) and then the product selected and a sale price is agreed and recorded.”

Why & How: Business Process Modelling

Execution Logic: there are a variety of ways of doing this ranging from natural English to highly specialised and formalised languages such as Zed. The most common methods that strike a balance between these two extremes are Structured English and pseudo-code.

Structured English uses English to describe the bulk of the specification with certain reserved words/phrases that need to be agreed upon by all who are going to create and use the logic.

The reserved words and phrases typically include

- Create
- Read
- Update
- Delete
- For each ... end for each
- If ... else ... end-if
- Go to
- Display
- Input
- Prompt

Example:

Prompt the user to input a Customer Name

Read Customers

For each Customer where the Customer Name = input Customer Name

 Display Customer Name
 Customer Address
 Customer Number

End for each

If no Customers were found

 Then invoke the task Create Customer

End if

Pseudo-code takes this one stage further and represents the logic in the programming style of the language to be used to code the solution. As such it will where feasible use the syntax of that language.

Example:

Procedure Find_Customer

Declare Input_Customer_Name Char(50) init(“”)

Declare Found Boolean init False

Display “Please enter the customer name: ” & Input_Customer_Name

Do while ¬EOF Customer and ¬Found

 Read Customer

 If Customer.Name = Input_Customer_Name then

 Display Customer_Name

 Display Customer_Address

 Display Customer_Number

 Found=True

 End-if

End Do-while

If ¬Found then go to procedure Create_Customer

End-if

Why & How: Business Process Modelling

In either case or any other technique used, it is the business rules that must be defined and nothing is defined that is only to do with making code work. Example, setting pointers or maintaining referential integrity in a relational database.

3. Process non-functional rules

Every process/task will have a set of non-functional requirements defined as well.

Non-functional requirements: There is no hard and fast definition of a non-functional requirement. In fact, even it's name only states what it is *not* rather than what it *is*.

Every project has non-functional requirements and they can relate to the whole solution (i.e. all processes and data within the solution) and/or to processes (i.e. all sub-processes and tasks within the process), tasks or data.

The *usual* ones are

- Who can use it (names, job roles, system roles, etc). Example: Sales Order Clerk
- How many users can use it concurrently. Example: up to 100 concurrent users.
- Where (physically) it can be run. Example: The Call Centre at No 1 The High Street, Anytown.
- When it is available for use. Example: 08:00 to 18:00 Monday to Saturday excluding Bank Holidays.
- How often it is run. Example: up to 1,500 transactions per hour.
- How quickly it should execute. Example: it should take no longer than 5 seconds to find a customer.
- How reliable it should be. Example: no more than 2 un-planned system unavailable events per year.
- Any usability requirements. Example: allow the order in which data items are supplied to be customised.
- Anything else which is not a functional requirement.

Non-functional requirements will need to be specified for all tasks and processes either directly or indirectly.

Processes: Non-functional requirements that **must** apply to all the sub-processes and tasks contained within a process can be specified at the process level.

Task: Non-functional requirements that do **not** apply to all tasks at that level must be specified for each task they do apply to.

4. Process data requirements

Essentially this is a CRUD (create, read, update, delete) matrix of tasks to data entities. The same argument about applying non-functional requirements to any level higher than the atomic applies to CRUDing.

The CRUD matrix specifies which data entities the task creates, reads, updates and/or deletes.

Typically a task will CRUD many data entities. For example, "Record Sale" may read the Product entity, create the Sale entity and update the Customer entity.

All entities should as a minimum be created and read by one or more tasks.

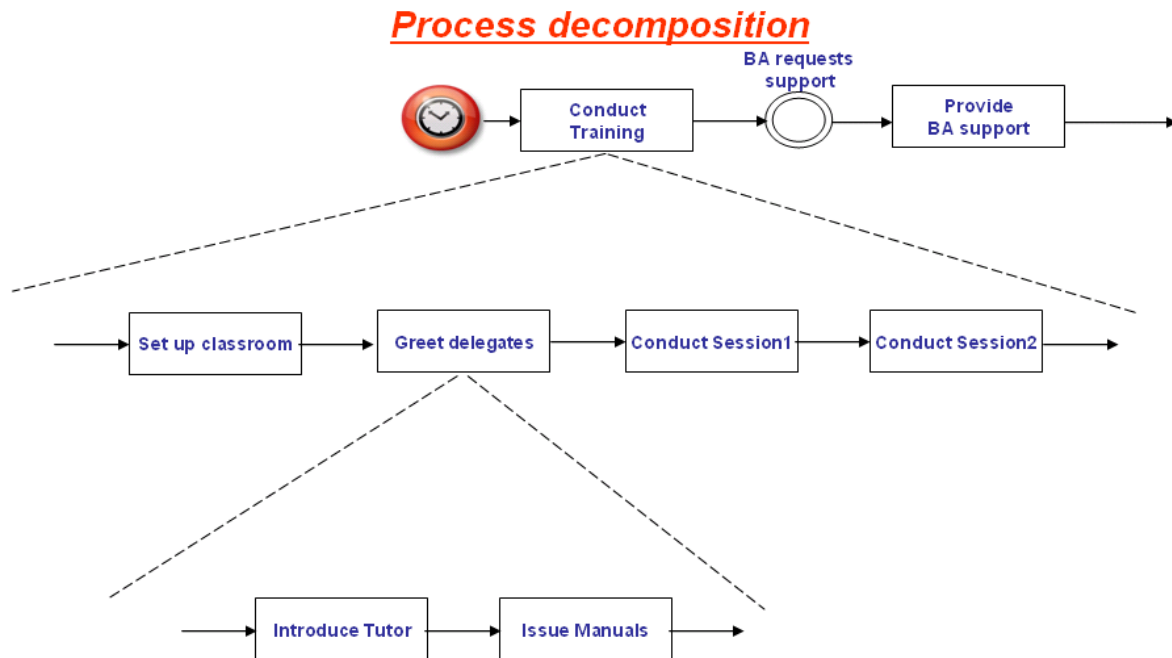
How data entities relate to each other and the other business rules defined for entities is not covered within this article.

Why & How: Business Process Modelling

5. Process model management

a. Process decomposition

Processes can be split in to sub-processes (also known as process decomposition) and so on until there is a level at which no further splitting is required (what BPMN calls a 'task').



Note that each level of these processes/tasks must 'balance' with the level it is a decomposition of: if a process has one input and that process is decomposed, then the input must also be input to at least one sub-process/task on the decomposition *and there can be no other inputs* although the single input can trigger more than one sub-process/task.

One question that arises is when to split or merge processes. It is not enough to say that a process is a connected series of business activities intended to achieve a significant business outcome: how do you know when and why to split or merge processes/tasks?

b. Identifying Business Processes

Top level processes: Arguably the easiest level to identify as there a good set of guidelines for identifying a process:

- i. Guideline of mutual dependency: if you conclude that Process A is always performed when Process B is performed, and that the only time Process A is performed is when Process B is performed then these processes should be combined in to one process. Example: Whenever the Process Create Customer is performed, the process Check Credit Rating is performed and the only time that Check Credit Rating is performed is when Create Customer is performed. Therefore, combine these processes.
- ii. Guideline of initiation and outcome linkage: if you conclude that whenever Process A is invoked *under certain circumstances* it will always result in initiating Process B (and/or outputs) and when Process A is invoked *under different circumstances* it always invokes Process C (and/or outputs) then it is *likely* that Process A has 2 or more processes combined in it and should be split out. Example: When Create Customer is invoked for a normal customer then the business rule is that the next process is always Place Order. When Create Customer is invoked for a customer who is also a company employee, then the business rule is that next process is always

Why & How: Business Process Modelling

Authorise Employee Customer. In this case it is likely that Create Customer has another process in it (Create Employee Customer) that in turn would include Authorise Employee Customer under the rule of mutual dependency.

- iii. Guideline of user concurrency: If a process needs 2 or more different Job Roles in order to execute then it is highly likely there are 2 or more different processes in it unless there is a business rule that says this process must be executed by these job roles at the same time.
Example: Suppose a company allows its employees to be customers of that company's services, but to prevent fraud more stringent checks are made for so-called Employee Customers. If the process Authorise Employee Customer needs authorisation by Employee Line Manager role and Sales & Marketing Director role then consider having 2 processes because without doing this once the Employee Line Manager role has authorised the Employee Customer the process will wait until the Sales & Marketing Manager has also authorised it!
- iv. Guideline: If Process A does not produce any meaningful (to the Business) outcome or output then consider merging with the preceding or following process.. Which process to merge with should be assessed using the other guidelines. Example: the process Record Customer Date Of Birth does not produce a meaningful outcome and can be combined under the guideline of mutual dependency with Create Customer

Any level processes and tasks

- v. Guideline of unit of work: if it is conceivable that the process could or should be halted so that other processes can be started, then split the process. Example: The process Create Customer might contain the logic to record marketing preferences. This information is not required to Place Order so consider splitting Create Customer in to 2 processes: Create Customer and Record Marketing Preferences (which could be done at a more convenient point).
- vi. Guideline of conciseness of specification: if it is not possible to specify the execution logic on one page then consider splitting the process. This guideline is weaker than most.
- vii. Guideline of transaction steps: if a process performs only a few or one action then consider merging with other processes using the other guidelines. Example: Record Customer First Name and Record Customer Surname are probably part of the same process!

c. Process specification management

This can be done manually but has a significant overhead in maintaining the coherence of

- mapping functional requirements to the process(es)/task(s) that implement them
- mapping the data used by each task and how it is used
- balancing the inputs and outputs for process decompositions

In addition the following information needs to be captured potentially for all processes

- name
- description
- non-functional requirements

And for each task

- execution logic

Computer Aided Software Engineering (CASE) tools are very useful for managing this information and can also implement certain checks for you such as process decomposition balancing. They can also generate reports from the information for you.

Why & How: Business Process Modelling

Conclusion

1. Process models are the pivot around which projects develop solutions.
They are the visible manifestation of the way that users interact with the solution and they *drive* a lot of project activity.
2. The risks of not doing process models vastly outweigh the effort of doing them.
The highly likely and high impact risks of not defining process specifications are
 - Project overrun as the problems created by not doing the process specifications will in all likelihood have to be fixed.
 - Decreased project benefits if process specifications are incomplete or wrong
3. There is a set of information needed to specify processes – not just diagrams.
 - process dependency models
 - process execution logic
 - process non-functional requirements
 - mapping processes/tasks to functional requirements
 - mapping processes/tasks to data used
4. CASE tools can reduce the administrative workload significantly of maintain process specification coherence (if used with rigour).

This article has sought to achieve 2 goals:

1. make the case that process models are the central pivot around which most Business Analysis revolves
2. demonstrate a practical way of doing process modelling.

The author would welcome comments on whether these goals have been achieved!

And finally... fancy having a go yourself? Model the following business scenario and send me your BPMN models for a review and feedback – I might even put the best ones up on my site! You can model any or all aspects of the scenario – I would strongly recommend that you start your model at the highest level of process modelling using the guidelines in the article to decide when to merge processes. Extra kudos will be earned for decomposing any processes that warrant it. Mega kudos will be awarded for process specifications including non-functional requirements.

You will need to make many assumptions to complete this model – document them and annotate your model and/or process specifications where they have any impact.

Feel free to suggest (or even model!) process improvements, but be sure you can justify them in terms of either process efficiency (so reference back to your process models) and/or benefits to any of the participants in the scenario.

Scenario:

smart-BA provides an on-line mentoring service for Business Analysts. The idea is that instead of doing classroom based training Business Analysts use their current project as a ‘case study’ upon which to learn Business Analysis in the real world via a programme of modules covering the scope of business analysis.

The overall process is that a Business Analyst visits the site to find out about the programme and enrolls. Having enrolled they are given the option of downloading module 1 to start work. Meanwhile an email is sent to smart-BA containing the enrolment details. If no Business Analyst Mentors are available to take on the new Business Analyst, a ‘thank-you’ email is sent advising them that as soon as a mentor becomes free they will be assigned.

Why & How: Business Process Modelling

When a Business Analyst Mentor in smart-BA decides they can take on more work, they look in the queue of enrolment forms and emails the Business Analyst to let them know who their mentor is and to find out more about them.

At some point the Business Analyst will send in their assignments for module 1 and the relevant Mentor will review and assess the work, and respond accordingly. If the Mentor judges that the assignment is complete they send the next module, or if it is the last module they send the certificate of completion. If the mentor judges that the assignment needs more work, they advise the Business Analyst of that.

The Business Analyst only pays a single fee for each module which is not payable until the first review is requested. The mentor needs to ensure that before they review assignments that the relevant fees have been paid.

If at any point a Business Analyst becomes dissatisfied with the course and cannot resolve it through emails with their Mentor, they can request a refund of any fees paid for the module they are working on and have nothing further to do with the programme.

Good luck and have fun!