White Paper

As-Is and To-Be Process Modeling: a Flawed and Failed Paradigm

WP0065 | March 2013

John Owens

John Owens is a thought leader, consultant, mentor, practitioner, blogger and writer in the worlds of strategic requirements, business function, process and data modeling, data quality, enterprise architecture and master data management.

He has built an international reputation as a highly innovative specialist in all of these areas and has worked in and led multi-million dollar projects in a wide range of industries across the UK, Ireland, Europe and New Zealand.

Although the modeling of 'As-Is' and 'To-Be' Business Processes is one of the most widely practiced modeling exercises in enterprises around the world, paradoxically it is an essentially flawed and failed paradigm.

This white paper clearly explains the reasons why it is flawed and describes a tried and tested alternative approach that has none of its defects and can greatly accelerate bringing positive benefits to enterprises of any size.

The Origins of 'As-Is' Errors

Our story starts back in the heyday of business improvement projects that were variously termed as Business Process Re-Engineering (BPR), Business Process Management (BPM) or Business Process Improvement (BPI) projects. It was at this time the concept of first modeling the 'As-Is' and then modeling the 'To-Be' Business Processes was introduced by large BPR/BPM consultancies.

The reason they gave for first modeling one set of processes and then the other was along the lines of, 'before you can move forward, you need to know where you are.' This, which at first seems an eminently sensible statement, is a complete fallacy, as I'll explain later on.

However, this fallacy has served the purposes of large modeling consultancies very well as it has allowed them to charge for not one, not two, but up to three separate projects, when one would have done.

The As-Is To-Be modeling approach is based on a fallacy…

The first project modeled the 'As-Is' Business Processes and the second the 'To-Be' Business Processes. The third project was all about moving the enterprise from the 'As-Is' to the 'To-Be'.

The consultancies involved were very happy, as this tripling up generated three times more revenue for them on each BPI project. The client was not so fortunate, as they were penalized in two ways. Firstly, they had to pay three times more for the project than they ought to have done. Secondly, each BPI project took three times longer to implement than was necessary and caused three times the disruption in the enterprise throughout that time.

This triple charging cost each enterprise a huge amount of money at the time of the project. However, that cost was nothing when compared to the huge on-going costs to enterprises around the globe caused by misguided and ill-trained BAs persisting in using this totally flawed practice of modeling 'As-Is' Processes at the beginning of a BPI project.

The fact is, if an enterprise is going to change its business processes, then modeling the 'As-Is' business processes serves absolutely no value. On the contrary, it is a practice that has a severe negative impact on all improvement projects in terms of time, financial cost, stress, disruption to the business and, probably the biggest cost of all, the loss of goodwill in the enterprise operational teams.

Is It Really Flawed?

Modeling the 'As-Is' Business Processes in an enterprise is only of value if you are not going to change them. If you're not going to change them, then they are what they are meant 'to be'. This, obviously, makes them the 'To-Be' Business Processes! In which case, you can safely model them.

So why are the 'As-Is' processes of no value if you are going to change them? Surely, by knowing what they are, you can then easily convert them to what they ought to be? The answer to this is a very definite, 'No, you can't!'

This is a trap into which BAs who have not come from a programming background fall headfirst. They erroneously assume that by looking at an existing business process that you can a) understand what it is meant to be doing and b) tune it to do it better.

Both of these assumptions are totally wrong.

You cannot infer from an As-Is process what it OUGHT to be doing…

Processes Are Programs

All good programmers know that you cannot look at a piece of code and infer what it is meant to be doing. This may be frustrating, but it's as simple as that, it just can't be done! So, if you cannot infer what it is that the piece of code ought be doing, you cannot tune it to make it do it more efficiently. This is because, to make it more efficient, you would first need to know what it is MEANT to be doing. Only when you know this can you tune the code.

Business processes are really no more than high-level computer programs and, as such, are subject to the same fundamental logic rules. This means that you cannot look at an existing Business Process and infer what it is meant to be doing. Neither can you, for the same reasons given for programs, tune a Business Process to make it more efficient without first knowing what it is meant to do.

If these two things are true, and you can take it from me that they are, then why are BAs around the globe modeling 'As-Is' business processes as a precursor to modeling the 'To-Be' business processes?

Is There Nothing of Use in the 'As-Is'?

Another problem that besets both the process modeling and systems development communities is the use of the term 'As-Is' in a standalone context. BAs and developers will ask questions like, "What will we capture about the 'As-Is'?"

The fact is there is no such thing as the 'As-Is'. The term 'As-Is' is an adjective not a noun. In order to have any meaning it needs to be used to qualify a noun or nouns, for example, "the 'As-Is' Business Processes", "the 'As-Is' Technology", etc.

Now, the questions that are asked can be meaningful and unambiguous.

- "What do we need to model about the 'As-Is' Business Processes?"
- "What do we need to record about the 'As-Is' technology?"
- "What do we need to record about existing applications?"

So, in any systems development project there may be many essential things to record about 'As-Is' facets of the enterprise.

However, the fact remains that, if you're going to change the Business Processes, then there is absolutely nothing to be gained from modeling the 'As-Is' Business Processes. Quite the contrary, it is actually a negative thing to do as it will bring no benefits, yet consume large amounts of effort, time and money.

There is only one model that can show what all of the enterprise OUGHT to be doing.

Is There a Solution?

Happily, there is a solution. Also, it is a very simple solution. As I stated above, in order to be able to build an effective computer program, you first need a definition (or specification) of what it ought to be doing. The same is true for a Business Process. Before you can build, or tune, any business process, you must first have a definition of what it OUGHT to be doing.

There is only one business model that unambiguously defines what all of an enterprise OUGHT to be doing; this is the Business Function Model (BFM), often referred to as the Function Hierarchy.

All Business Process Improvement (BPI) Projects should start by first making sure that the enterprise has a current Business Function Model or, if it does not, then the first task in the BPI is to build one.

The reason for this is that ALL business models are based on and derived from the Business Function Model (BFM). Without the BFM, all other models are just being 'plucked' from thin air and, as a result of this, lack all integrity.

The Business Function Model is a complete, structured definition of everything that an enterprise OUGHT to be doing.

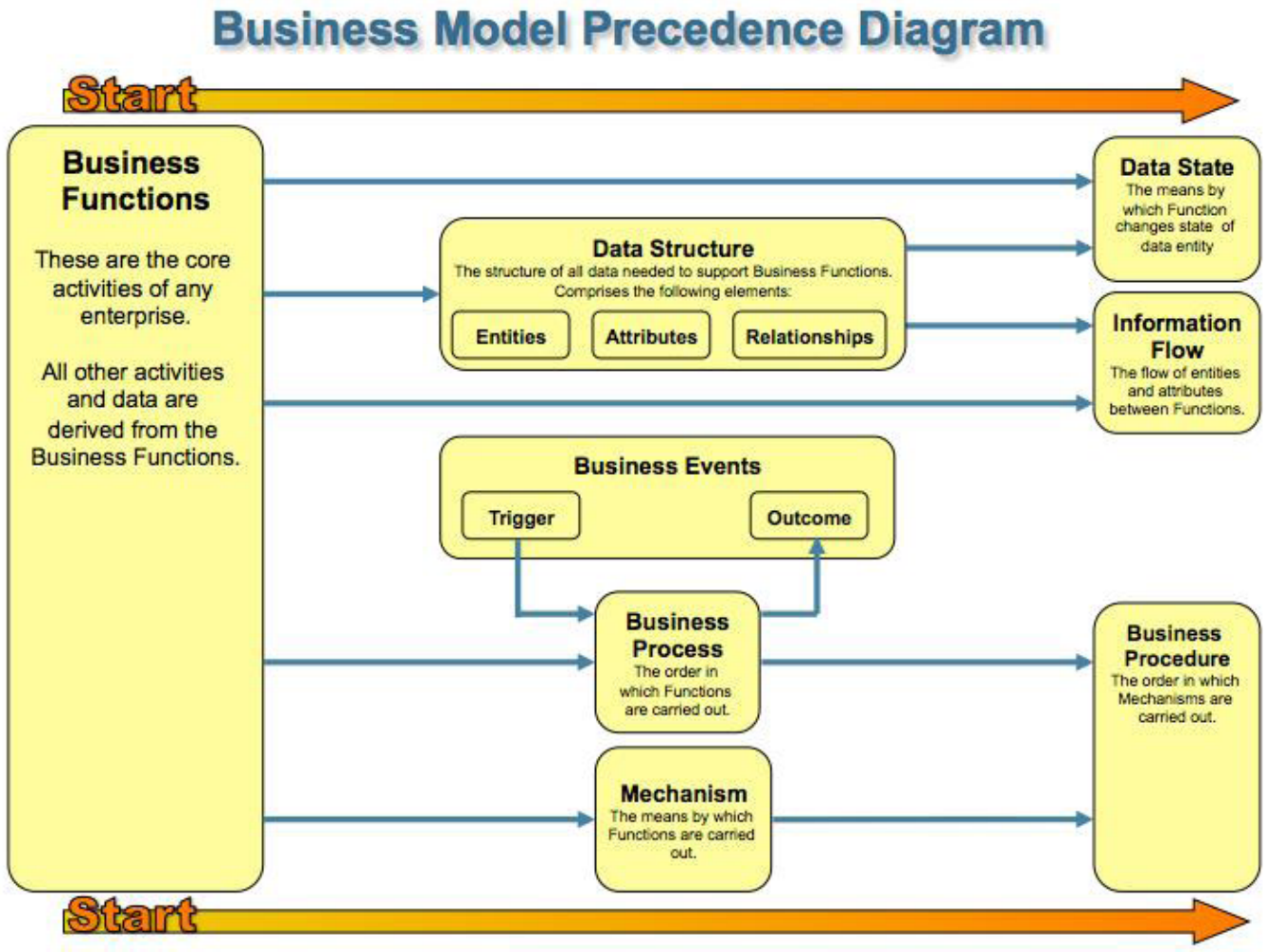

Figure 1

Figure 1 shows the order in which business models need to be built if they are to maintain their integrity.

Function vs. Process

Many business analysts make the mistake of believing that Business Functions and Business Processes are synonymous. They are not.

A Business Process is merely the definition of the order of execution of selected Business Functions in response to a specific Business Trigger, in order to arrive at the specified Preferred Outcome for that Trigger.

This definition tells us that, before we can build a Business Process we need to have five essential elements, 1) a valid Business Trigger, 2) the Preferred Outcome associated with that Trigger, 3) the Business Functions that are required to be executed in order to get from the Trigger to the Preferred Outcome, 4) the order in which these Business Functions need to be executed and 5) a list of valid Non-Preferred Outcomes in case the Preferred Outcome cannot be attained.

Every step in a Business Process is a Business Function and, as such must come from the Business Function Model. By ensuring that all of your Business Functions come from the BFM, all of your Business Processes will represent WHAT the enterprise OUGHT to be doing.

Elementary Business Function (EBF)

To be precise, every Process step will be an Elementary Business Function (EBF) from the Business Function Model.

An Elementary Business Function is a Business Function, which once begun, must be completed or, if not completed, must be wholly undone.

If there is a valid intermediate state for the Business Function it is not elementary.

Triggers and Outcomes

How do you tell what starts, or ‘Triggers’, a Business Process and what ends it? This is where the Business Analyst needs to know how to model Business Events.

There are two major types of Business Event, Triggers and Outcomes. As their name suggests, Triggers initiate a Business Process. Outcomes represent the result that the Business Process ought to bring about.

There are also two types of Outcome; Preferred Outcome and Non-preferred Outcome. A Preferred Outcome is the preferred state that

the enterprise wants to achieve in response to a specific Trigger. The Business Process that is initiated by the Trigger is merely the means to bring about this preferred state.

A Non-Preferred outcome is a valid alternative state that the Business Process ought to bring about, if the Preferred Outcome cannot be achieved.

In order to do effective Business Process Modeling, BAs must, in partnership with key managers in the enterprise, identify and document all of the Business Triggers for the enterprise and all of the Preferred and Non-Preferred Outcomes for each Trigger.

Although a single Trigger will have only one Preferred Outcome, it might well have several Non-Preferred Outcomes.

Overall Approach

Adopting the following approach will enable you to successfully build effective and robust Business Processes that will enable the enterprise to do exactly WHAT it OUGHT to be doing.

Effective Process Modeling starts with Triggers and Outcomes. It starts with a question like, “Given the Trigger X, how do we get to the Preferred Outcome for Trigger X? What Business Functions do we have to execute and in what order?”

The BA then has two options, 1) work from Trigger to Outcome or 2) work from Outcome to Trigger.

Trigger to Outcome

Taking the Trigger to Outcome approach, the BA will look at the Trigger and ask, “What function does this trigger initiate?” When they have identified this, they add it to the process immediately after the Trigger.

They then ask, “What happens next? After we have executed that function, what then happens?”

With this approach they keep adding appropriate Business Functions to the Process until they arrive at the Preferred Outcome.

Each Business Function that they add to the Business Process being built must come from the Business Function Model. If no appropriate Business Function currently exists in the BFM, then the BA must add one to it, so that it can be used in the Process being built and in every other Process where it is required.

Outcome to Trigger

With this approach, the BA starts at the Preferred Outcome and asks the question, “What must occur immediately before this Preferred Outcome can be deemed to have been reached? What Business Function must have been executed in order to bring about this state?”

They would then add this Business Function to the Business Process, immediately before the Preferred Outcome and connect it to the Preferred Outcome. They would then look at the Business Function they have just added and ask, “What other Business Function must have been executed to allow this Business Function to bring about the Preferred Outcome?”

The BA would continue working backwards from the Preferred Outcome, adding Business Functions, until they have reached the Trigger for the Process.

Cross Checking

The Trigger to Outcome and the Outcome to Trigger approaches have only been sketched in the outline above. A full description would require far more detail to take account of such things as conditional and unconditional branching, etc.

Both approaches are valid and can be used both individually and as a means of cross-checking the other approach.

However, whichever approach the BA takes, the object of the exercise is to build a valid Business Process that will ensure that the Preferred Outcome for the initiating Trigger is reached in the fewest possible steps and in the most effective and robust manner.

Summary

The traditional ‘As-Is/To-Be’ approach to Business Process Improvement is essentially flawed as it completely ignores some fundamental logic rules of programming. Because Business Processes are, in essence, high-level programs, this makes any Business Process based on this approach suspect.

Contrary to what so many BAs believe, there is a perfectly safe way to develop Business Processes that accurately reflect WHAT it is that the enterprise OUGHT to be doing. In a nutshell, this is:

- Start by building a Business Function Model for all or part of the enterprise.
- Identify all significant business events, i.e. Triggers and Outcomes.
- Use Business Functions from the Business Function Model to plot a path from Trigger to Outcome, or vice versa.

This simple approach will produce high quality Business Processes first time, every time and at an accelerated rate.

Definitions

A major cause of confusion amongst Process Modelers is that this community lacks formal definitions for all of the elements of Process Modeling.

It is sad, but true, that most Process Modelers cannot tell the difference between a Function, a Process or a Procedure. For that reason, I provided a set of clear definitions for all of these elements, and more, in the table on the following page.

Business Element	Description
Business Function	A Business Function is a discrete activity or a coherent set of activities that an enterprise must perform in order to meet its objectives and continue in existence. ALL DATA in an enterprise is created and transformed by the Business Functions.
Business Process	A Business Process is the definition of the order of execution of selected Business Functions in response to a specific Business Trigger, in order to arrive at the specified Preferred Outcome for that Trigger. A Business Process comprises five essential elements, 1) a valid Business Trigger, 2) the Preferred Outcome associated with that Trigger, 3) the Business Functions that are required to be executed in order to get from the Trigger to the Preferred Outcome, 4) the order in which these Business Functions need to be executed and 5) a list of valid Non-Preferred Outcomes in case the Preferred Outcome cannot be reached.
Atomic Business Function	This is a Business Function at the bottom of a Business Function Hierarchy; it is a Business Function that has no child functions attached to it.
Elementary Business Function	An Elementary Business Function is a Business Function which, once begun, must be completed or, if not completed, must be wholly undone. An Elementary Business Function may take the business from one valid state to another or may leave the state unchanged. If there is a valid intermediate stage for the Business Function it is not elementary.
Business Mechanism	This is the means by which a Business Function is executed within an enterprise. One Business Function might be executed in several different ways; that is, using several different mechanisms. The specific mechanism would be dependent on the resources and technology available to those carrying out the Business Function.
Business Procedure	Whereas a Business Process defines the order of execution of Business Functions, a Business Procedure defines the order of execution of Business Mechanisms in order to arrive at the Preferred Outcome associated with a specific Trigger.
Business Event	This is any occurrence of significance to the enterprise. Business Events fall into two main categories - Triggers and Outcomes, and can arise due to occurrences both within and without the enterprise.
Business Trigger	This is an occurrence of significance to the enterprise to which a response must be initiated. This response could either be the execution of a single Business Function or the initiation of a Business Process.
Preferred Outcome	This is the outcome that the enterprise would prefer to have achieved by the execution of a Business Function or of a Business Process initiated by a Business Trigger.
Non-Preferred Outcome	If a Preferred Outcome cannot be achieved by the execution of a Business Function or a Business Process, then this is a valid, acceptable, if non-preferred, alternative outcome.

Orbus Software
3rd Floor
111 Buckingham Palace Road
London
SW1W 0SR
United Kingdom

+44 (0) 870 991 1851
enquiries@orbussoftware.com
www.orbussoftware.com

orbus software