Business Process Management
A Rigorous Approach

Martyn A Ould
Venice Consulting Ltd
The Old School, Hinton Charterhouse
Bath BA2 7TJ, UK

phone +44 (0)1225 723 822
fax +44 (0)870 056 7443
e-mail mao@veniceconsulting.co.uk

THIS is the text of the lecture I gave at the launch of my new book *Business Process Management – A Rigorous Approach* on 25 January 2005 in London.

The book describes a method, called *Riva*, for working with business processes. In my lecture I described the rationale for the method and its positioning in the Business Process Management field.

The book is published by the BCS and Meghan-Kiffer Press (North America). Details are available at www.bcs.org/books/bpm. ISBN 1-902505-60-3

I'd like to present three challenges that I believe face us today as system developers, and from those challenges I hope to demonstrate that this is a really useful book … or at least is addressing some important questions that are in the air right now.

I can perhaps summarise the challenges by saying that we are moving from the Information Age to the Process Age. Some might say this move started ten or twenty years ago. But for a number of reasons I think the move is accelerating.

CHALLENGE ONE: SYSTEMS MUST CHANGE AS PROCESSES CHANGE

Firstly, businesses are taking a more process-oriented view of themselves. If the titles of conferences, journals and websites are anything to go by, this is certainly true. and if there's smoke and mirrors then perhaps there is also fire. We have probably always had TQM, going back to the days of Deming and Juran and Crosby – and so at least for two or three decades. But the concern about process was largely limited to the small scale – perhaps it would be exaggerating to say it was about 'tweaking', but you know what I mean. With BPR we took a rather harder and disruptive view of things, deciding to axe whole areas of process and to question structure as well as detail.

This all went hand-in-hand with the ERP phenomenon where, allegedly, best-of-breed processes could be bought off the shelf, so you could have the same process as your competitors. Tie that up with Y2K as the opportunity to replace systems wholesale and we went in for disruption on a massive scale. We knew, didn't we, that because our systems had paved the cowpaths that were our processes, this was the opportunity to carry out wholesale change of our processes while we had the chance. We straightened out our processes and repaved them.

This momentum continues to build. As business gets more interested in its processes, so it gets more interested in the alignment of its computer systems with the processes they are supposed to support. The problem is of course that those systems are information systems and not process systems. The information technology world (of which I have been part and so I'm allowed to say this) has managed to sell the message that businesses are about information and so information systems are what you need. In reality the information systems world is only capable of imagining information systems and only capable of building information systems. And I guess this all works fine as long as we restrict ourselves to providing point solutions to point problems, satisfying the information needs of the individual.

It has also all worked fine as long as the processes the information is designed for remain the processes the business actually operates. And I think this is the crucial point: information is the oil that lubricates the process. It's as if we have run down a dead end where we have been totally focused on storing and retrieving data with ever greater efficiency, security, scalability and so on. And even though a business thrives by *doing* things we have ignored the doing and we've concentrated on what we've always done which is to look after people's data. I think the day of reckoning has arrived. Fortunately, so has the necessary technology.

Twenty years ago at Praxis, we developed a prototype process support system which allowed on-the-fly change. I guess it was twenty years ahead of its time, but it was also twenty years ahead of the technology necessary. At the time it was built using SmallTalk on Informix as I remember. Not exactly a scalable solution. But web technologies are changing all this.

It's easy to see that information must take second place to process. For two reasons. Firstly, if you decide that you need certain processes, for instance for handling an employment application, then you will need certain information to do that process; if you don't have that process, then you don't need that information. Secondly, having decided that you need that process, you must then decide how it will operate across your organisation, via the collaboration of the people in the organisation. And perhaps you will need also to make some changes to organisation structures, job descriptions and so on to fit the process into daily life. Having decided how you want everybody to collaborate, you can then decide what information everybody needs and how information must flow between them.

The moment we say all this, we see that information is subordinate to process. If we change the business we are in, and hence the processes we need, our information needs to change. If we change our organisational structures, we change the information flows and the information needs. Information simply oils the wheels of process. If you change the machine, different oils will be needed in different places.

So here we have a world where businesses are constantly reviewing and revising their processes. As a direct consequence, they are also implicitly constantly changing their information needs. Where does that leave us as far as our information systems are concerned? Especially our ERP systems.

One other factor makes this challenge greater. The talk is increasingly about end-to-end processes, about business-to-business processes, about supply chains. In summary, about very long, very complex and distributed processes.

So the first challenge comes from the fact that business processes are increasingly under change, and that because information is the oil that lubricates processes, it becomes ever more important for the information systems developers to understand those business processes and the changes they are undergoing.

CHALLENGE TWO: THE NEW TECHNOLOGY OF BPMS

Let's turn now to what I see as the second challenge. The growing technology labelled as 'business process management systems'. Once again we are entering smoke and mirrors territory. Old information products are being rebadged as process products. Old methods for building information systems are being rebadged as methods for building process support systems. It all sounds terribly familiar. I guess this is all a natural reaction. Product developers see a new marketplace for their existing products. System developers have always used hammers and reckoned those hammers will work on screws.

However, if the whole idea doesn't get watered down to just web-enabled workflow, there are some genuinely new features of what I might call 'true' business process management systems for which new methods will be necessary.

One of the terms being bandied around is 'pi-calculus'. This is a mathematical theory about concurrent systems with certain properties, namely the dynamic creation and communication of channel names … don't ask. Proponents assert that pi-calculus should be the basis of true business process management systems. This is a debate that I don't want to get into, especially here. But I think it is worth looking for a moment at some of the things that happen in complex real-world processes that we shall need to be able to deal with and talk about with our business process management methods. Some of these things are dealt with, in a mathematical way, in pi-calculus, so although I would not number myself amongst the fundamentalists I can see where they're coming from.

A problem with traditional approaches to thinking about business processes is that, ironically, they take a rather static view. Clearly, in a workflow things happen, and to that extent we capture some dynamics. But those dynamics tend to be rather fixed and not at all representative of the real-world except very simple bits of it.

Howard Smith points to e-mail as a process that would be impossible to capture as a workflow, simply because of the dynamic way in which e-mail conversations blossom and die away. The activity of sending an e-mail is itself a very simple one, but the dynamics arise from the possibility that new people are drawn into the conversation, new connections between people are made, new threads and conversations are started, and there is a flux of activity that cannot be predicted at the outset, in other words that cannot be described in full at the outset. What is necessary is a way of defining all of the potential behaviours.

This concept of process 'mobility' is not recognised by traditional approaches. And yet in real life we see it all the time. We see the dynamic generation of responsibilities: the responsibility to handle a new application, the responsibility to manage a new project, the responsibility to answer a customer query. These are all created – I might say 'instantiated' – on-the-fly, dynamically, as-and-when necessary. And to carry out their work, these responsibilities must all be tied together, or, to put it more crudely, they must be introduced to each other, and all of their interactions must be choreographed. Once again, information-oriented approaches simply don't have the language we need to talk about such ideas.

Another feature of organisational activity is that of collaboration. I'm not very much in favour of trying to provide one-liners for defining the word *process*, but when I do try to do that the phrase 'the activity of a group of people collaborating to achieve a business goal' comes close to what I want. Processes are *group* activities and when people work in the group they sometimes take actions on their own as individuals and sometimes have interactions with other individuals. When we describe processes we should expect therefore to define actions and – I want to concentrate on this here – interactions. Sometimes those interactions are pre-ordained: role A must interact with role B. But sometimes those interactions are defined dynamically – at run-time. New roles and new responsibilities are created on-the-fly, and hence their interactions are also created on-the-fly. What we see happening here is the instantiation of roles, responsibilities, actions, and interactions dynamically. Things are very definitely not static. An organisation is just a flux of instances. Though I possibly wouldn't say that to the average CEO.

Now, an information system helps us work in our process by providing the lubricating information at the right moment. Additionally, a business process management system (BPMS) helps us work with our processes by making those processes *the subject matter of the system*. The process is not something implied. It is not something separate from the system. The system is not just plumbing people into the information they need to carry on the process that is happening elsewhere. An information system is to all intents and purposes a bucket with information in it. It is essentially ignorant of the process being operated, simply allowing appropriate people appropriate access to the information in it, perhaps with a some simple business rules attached. The BPMS on the other hand is where the process is mediated.

A true BPMS naturally assumes that we shall want to change the processes we are using. Again putting it in crude terms, we might say that we know we shall want to change the model of the process being held by the BPMS. Of course, this is not a model in the sense that a model ship is a model of a ship, something separate from the ship. In a BPMS the model is intimately connected to the real thing. By changing the model we change the way the world works. This is the view of processes as 'actionable business assets'.

So we can summarise the second challenge by saying that the new technology of BPMS demands that we are able to talk about processes with both types of dynamism: Dynamism in the sense that processes can be mobile. And dynamism in the sense that processes can be changed. Flow charts and information-based thinking will not be enough to deal with this dynamism.

CHALLENGE THREE: JUST WHAT PROCESSES DO WE HAVE?

The third feature that I want to describe touches on the basic notion of 'process' itself.

I guess we could think of an organisation such as, say, the British Computer Society operating just one process: the 'British Computer Society process'. We might quickly decide that this is not a useful or even truthful approach. Something tells us that at any one moment the British Computer Society is operating many processes in pursuit of its goals as a professional body. We might go further and guess that it would be more accurate to say that at any one moment the British Computer Society is operating *many instances* of many processes in pursuit of its goals.

When I walk into the building I sense that many processes are in action at the same moment. I also suspect that there are some processes that are being carried out in many instances at the same moment. 124 membership applications are being dealt with at this moment, say. Presumably 124 instances of the process for handling a single membership application are in progress. And that set of instances is changing from moment to moment as applications are dealt with and closed and new ones arrive. There is a flux of process instances. And presumably there are also processes that manage that flux, and yet other processes that take a strategic view of how we handle membership applications.

The question arises: how do we decide what processes an organisation has and what the dynamics of that collection of processes are? How and when are the processes instantiated and how do the instances interact?

We can rephrase this question rather more crudely: how do we chunk all that organisational activity into meaningful chunks?

Suppose I walk into the dissecting room at the teaching hospital to lecture to medical students on how the human body is constructed. Awaiting my arrival is my assistant, ready with a thick marker pen. There, on the table, is the cadaver. I have brought with me an axe. With a deft overhead blow I lop off the lower part of one leg. My assistant labels it 'The A bit'. The students dutifully record the name against their sketches of the body. Some aren't sure quite where on the leg the axe fell, but choose a point anyway. With no more ado, another blow removes the top of the skull, which my assistant labels 'The B bit'. More scratching in notebooks. Further blows yield bits C, D, E, the remainder on the table being labelled 'The F bit' with the marker pen. 'The F bit' is still quite large, so four swipes render it into five pieces, which my assistant labels 'The F1 bit', 'The F2 bit', and so on. The students take notes, increasingly unsure about exactly how much corpse each bit is made up of. Never mind, I now take bit F2 and, putting down the axe in favour of a small meat cleaver, I cleave it into bits whose names, I tell the students, are F2a, F2b, and F2c.

What understanding do the students now have about the way the body is constructed and how it works? Has the chunking been guided by an understanding of what a human body is all about? Would each student have the same understanding of exactly what constituted each bit? If I gave the same lecture next week, would the student's drawings be anything like those of this week's?

You would, I'm sure, prefer that I had taken a scalpel with me into the dissecting room, together with an understanding of what a human body is all about – the fact that there are 'natural cleavage lines' that separate the central nervous system, the gastro-intestinal system, the skeleton, the musculature – and of how those systems are connected. We look for these things because we know that a human body is 'in the business of' feeling and sensing, nourishing itself, standing and moving.

When we chunk organisational activity we need a similar scalpel that will allow us to cut along the natural cleavage lines of that activity, to separate out the processes using an understanding of what the organisation is all about, an understanding in particular of what business it is in. We want to be able to chunk all the organisational activity into processes, to understand which of those processes can be instantiated and when that happens. We want what I call a *process architecture*.

Now, in a way I have already begged another question: 'what is this *organisation* we are concerned with?' As I suggested earlier, it is no longer enough to think about how we in our little organisation operate. We have customers and our processes extend into those of our customers. We have suppliers and we would like our processes to link back into those of our suppliers. Before we know where we are, we're constructing what we like to call *supply chains*.

Moreover, there are times when we shall decide to outsource part of our business. There will be other times when we shall decide to bring something back in under our own wing. On other occasions we shall merge two organisations that are in the same business and hence might be suspected of having the same processes. And so on. Can our method help us work through the process implications of these strategic changes?

Finally, I can see no reason why that process architecture should change just because we change our culture. Nor should it change because we adopt new technologies. Now should it change if we tweak our organisational structure. Of course, the way we *do* our processes will change, but the list of processes and their dynamic relationships must remain invariant if we stay in the same business. On the other side of the coin, if we change the business we are in, then we can expect the architecture to change.

So we can summarise the third challenge for system developers by saying that we need an approach to process thinking, across whatever organisation we are interested in, that provides us with a process architecture, a solid foundation that we can build all our work on, an architecture that will not drift as technology drifts, or as culture drifts, or as organisational structures drift. An architecture that truly reflects the business we are in. We want good chunking!

So these are the three challenges for us as people who develop systems that support businesses:

1. Our systems must change as our business and its processes change.
2. The new BPMS technology offers new ways of working with processes that information systems do not have.
3. Now that we increasingly want to deal with the big processes in life, we need to achieve a robust chunking.

SATISFYING THE REQUIREMENTS

You can think of what I have said as a sort of requirements specification for any method designed for talking about business processes in this new age.

I think if I had to choose one concept to characterise it an appropriate method, it would be *flux*. When we walk into a building there is a massive amount happening. And what is happening changes from moment to moment. Static models are not enough. And to capture that flux I believe we shall need some key concepts underpinning the languages our development methods use.

Firstly, I'm convinced that we are lost if we do not recognise that processes are dynamic objects – things come and go. Without instantiation we shall only be able to talk about statically defined processes, about fixed behaviours. We could not even describe an e-mail conversation. At the heart of *Riva* you will find instantiation: the instantiation of processes, of responsibilities, of actions and of interactions. If we use *Riva* to examine an organisation at a particular moment in time, we shall see a network of interacting process instances. If we examine a single process instance we shall see a network of interacting responsibility instances. If we examine a single responsibility instance we shall see a network of action and interaction instances. These networks are under constant change – this is the flux I have talked about – things come and go.

Secondly, I'm convinced we are lost if we do not recognise that processes are about interaction, about collaboration. The British Computer Society's process of launching a book requires this interaction that we are participating in here and now: the launch event where the author lectures an audience who then ask questions. Before that there have been many more interactions between the various responsibilities that have been dynamically created to get this book to market successfully. Other processes have marshalled the available resources between the process instances operating for this book and the process instances operating for other books being prepared for publication.

Thirdly, I'm convinced we are lost if we cannot reliably chunk all the organisational behaviour into meaningful processes – processes that are instantiated and that interact. It would be no use if our method produced quite different results depending on who used it. And I have already made the case that I want a method that determines the process architecture solely from a characterisation of the business the organisation is in.

These three concepts – instantiation, interaction, and process architecture – lie at the heart of the method described in the book.

I'm a process freak. I see the world in terms of processes. I can't go to the theatre without watching the acting process, rather than the thing that is being acted. I have to love a piece of music a great deal not to find myself watching the quartet process, rather than listening to the quartet. So when I think about process I'm naturally very interested in *how* we think about process. I want to know what *methods* we are going to use to think about process, methods based around appropriate notations.

This is one reason why I get annoyed when I see data-oriented languages and data-oriented methods being dragged, silently kicking and screaming, out of the world of information system development into forced labour to describe – or attempt to describe – business processes. What reason could we have for believing that this will work? Check the literature.

I want my language and method to use organisational concepts, not IT concepts. I don't just want a bunch of blobs and arrows, even though I have formal semantics for the blobs and for the arrows. I want a way of saying meaningful and useful things with that language of blobs and arrows. I want a way of working with real situations and using my very special language to say important things about those situations. I want a way of deciding what processes there must be in an organisation in a given line of business. I want to be able to sensibly chunk organisational activity into processes, processes that have reality, not processes that are arbitrary chunks.

I want to see if there are other ways of doing business that radically simplify the collection of processes I need. Do I really want invoices? Are timesheets really necessary? Why do we have project plans? What happens if we outsource deliveries? What happens if we bring clinical trials back in house?

Now I'd like to take this thought one step further. I often see descriptions of 'a process'. I was looking at one last night that was an administrative process in a university, running from the point where someone enquires about undertaking a research post at the university to the point where a formal offer is made to them. It was presented as one single process. It was prone, apparently, to delays, with the possibility that things would get stuck in the works. This is nothing unusual.

The problem is that when we examine such a process – thinking it is indeed just one – we start at the/some beginning and trace the sequential flow through to the/some end. In reality three different things are being dealt with (probably more if one were to examine it in more detail): the enquiry, the application, and the offer. In reality there is a process for handling each of those: a process for dealing with an enquiry, a process for dealing with an application, and a process for dealing with a formal offer. And because there is a flow of enquiries, and a flow of applications, and a flow of formal offers, we need processes to handle each of those flows. Those three processes are management processes. How often do we completely ignore management processes, the processes that schedule and resource and prioritise things? The very processes that can be the cause of delays. It is at these processes that queues build up, in the form of in-trays, or lists or post-it notes even. So far then, I have six processes. And what looked like a simple sequential process is in fact a combination of six interacting processes. In fact it's worse than that because of course the processes for handling things like an application are instantiated many times. So, a true picture of what is happening would talk in terms of six processes, three of which would exist in many concurrent instances and three in a single instance. That true picture of a dynamic network of process instances is what I mean by a process architecture.

I don't want to make this talk into a course on *Riva* but I hope you can see from this simple example how our traditional perspective on our processes is just too simplistic. And over-simplification is the curse of good system development. And it certainly scotches any idea of getting to the bottom of problems with an organisation, even if we are not concerned with computer support. So step one in any work using *Riva* is to construct that process architecture, even if – especially if – it appears at first that there is only one process.

Having decided what processes there must be, I want to be able to work with individual processes.

I want to capture and describe an existing process – perhaps just to understand, perhaps to define it so we can share that understanding and all follow the same process, or perhaps so that we can diagnose the process and see why bad or undesirable things are happening in it or because of it.

I want to design a new process for a new business need.

I want to take an existing process and change the organisational structures or the culture or the technology for it to make a step change in its effectiveness or efficiency or whatever.

These three needs exist whether or not I plan to use computers to support the processes concerned. They correspond to the traditional TQM and BPR questions.

Finally, I want to be able to express the dynamics of a process in sufficient detail to enact it on a BPMS.

I want ways of doing all these things, ways that exploit the power of the language that I have chosen.

One question that might just be on your lips is 'how does all this fit with all the standards stuff going on in the BPMS world? In particular, how does it tie in with BPMN? Currently, the industry is working on the development of a standards stack which has a web services stack at the base, with – successively – BPEL and BPQL plus service choreography, then BPXL that extends BPEL for transactions, business rules etc, the BPSM (Business Process Semantic Model), and finally the BPMN at the top. But BPMN is now the machine code of the BPMS. It is executed directly.

Two things are worth noting. One is that these are only notations – they're not methods. Another is that they focus on the machine.

When I started out on this book I was concerned about ending up writing a competitor to BPMN, an exercise that would be rather futile. I approached a number of people in the BPMS area and asked for an opinion. The message I got was along the lines of 'at last someone has come up with a *method* that we can use before we start slamming out volumes of BPMN.'

Many years ago – 49 to be precise – a man named Herbert Benington had the brilliant idea that before you write code it would be a good idea to design it, and before you design it, it would be a great idea to specify what the customer wanted. I suspect we are at that stage today with writing business process management systems. We have invented the machine code and have started coding. And that machine code is at the top of the stack. Mr Benington, we need to heed your advice again today.

So that is where I am positioning the method, and this book. In a couple of months' time I'm speaking at a BCS branch meeting and my chosen subject is 'So you think you understand your processes'. Before we get anywhere near BPMN, let's – with method – understand our processes, understand why they are the way they are, and let's do all that in business terms, not software terms.

CLOSE

In summary, I want a language and a method with solid underpinnings, that talk in terms familiar to business people, that allow me to really describe the dynamics of organisational activity in all its depth and complexity.

I'm not content with simplistic schemes that ignore so much that they might be OK for a simple view of a simple process but they get nowhere near helping us get our heads around that complexity of concurrency – that swirling flux – which is the real world of organisational activity.

We are moving from the Information Age to the Process Age. We need some purpose-built process methods to replace our data methods.

My new book describes such a method.