The Split Personality of BPM

Derek Miers, Enix Consulting, United Kingdom

This paper explores the link between Business Strategy, BPM Technology and Process Architectures – specifically, highlighting the need for both procedural 'control' and evolving business practices. A key objective of this paper is to outline effective approaches for the design of process frameworks – especially where a BPMS implementation is intended target.

In Part I, we discuss the need for Procedures and evolving Business Practices. Part II will explore the need to understand and model 'what' you are doing before detailing how it is done. When it comes to modeling and representing that vision, flow diagrams just do not deliver. Part III discusses a few alternative modeling paradigms. We contrast the procedural BPMN modeling notation with approaches for modeling business Practices – one for capabilities and behaviors and another technique to capture the parallelism and synchronization. Part IV suggests an approach for designing flexible and adaptable BPMS process architectures - acting as a set of guiding principles for those involved in BPM modeling and deployment.

Procedures & Practices – Balancing Control & Agility

Introduction

Business Process Management (BPM) technologies promise a new era of operational excellence combined with business agility. Achieving that will entail adoption of effective standards that support service driven architectures and dynamic business models. These standards can then support interoperation up and down the value chain, while enabling users to drag and drop their process definitions, developed using one product, onto the BPM engine of another.

However, realizing that vision implies that BPM approaches and standards will need to support a very wide range of usage scenarios – from the dynamic supply chain examples cited in most discussions to date (where the focus is on physical goods and manufacturing), through to the supply of services (intangible goods) and the evolving interactions of knowledge workers. And it will not stop there – those involved in product development are looking for effective mechanisms to coordinate their work up and down the value chain – involving partners and suppliers as needed.

Following several years of jockeying, the heavyweights (IBM, Microsoft, SAP, et al) have 'decided' that the standard for business processes will be Business Process Definition Language For Web Services (BPEL4WS shortened to BPEL).[1] Meanwhile, the Business Process Management Initiative (www.bpmi.org) has released its proposed additions to support longer running processes (BPML) and a 'standard' process-modeling notation (BPMN). And, while all this technical work is going on, change agents inside our corporations are grappling with the human side of the equation, weighing up the methodologies and assessing their applicability.

At the same time, as all sorts of firms struggle to deliver faster time to market with best of breed services and products, managers must balance two sets of issues – efficiency and flexibility.[2] Behind this tension are two contrasting images of the organization and the environment within which it operates. Moreover, even within a single firm or department,

the interpretation of this dynamic is different. Developing an understanding of this conflict and its implications for process design is critical to future success with BPM.

Defining Processes

Before exploring BPM in detail, it is worthwhile pausing and considering what is meant by the term 'process'. In traditional, functionally oriented organizations, processes are often fragmented, invisible, unnamed and unmanaged. When asked to describe the true nature of a business process, most people have quite different perceptions. In our workshops, participants offer various definitions for a business process including:

- A sequence of activities performed on one or more inputs to deliver an output to a customer.

- A set of (partially) ordered steps intended to reach some goal.

- An organized collection of business behaviors that satisfies a defined business purpose and performs according to specified targets.

- A collection of business activities that create value for a customer.

- A systematic set of activities, which take a 'business event' to a successful outcome.

- A number of roles collaborating and interacting to achieve a goal.

- A way of linking people together.

- The way things get done around here. [3]

Most of those involved in the technology industry (vendors and IT professionals) tend to favor the earlier definitions in the list above. Of course what we call 'work' is vastly more complex than that represented by certain of these somewhat 'Newtonian' definitions.

To fully understand business processes, we have to be able to talk about a range of usages and sometimes, conflicting, perspectives.[4] We also have to bear in mind the subtly differing agendas of those involved. Some see business processes as the way of imposing control on their organization. Others see them as an opportunity to standardize their business offerings (reducing variation). As one engages more and more business leaders, one finds their vision is often more about enabling greater business agility.

Processes – Procedures, Practices or Both

On the one hand, we have an image of the organization as a machine, with its hierarchical structures and paths that run across them (i.e. one perspective of process). A process in this context is usually represented by a linear sequence of activities combined with functional decomposition. Historically, this is the result of the functional emphasis common in all but the smallest businesses. It is most effective in a relatively stable environment – both internally and in the firm's interactions with customers, suppliers and partners.

Technology is introduced to reduce the (human) resources required to undertake a given amount of work while allowing the business to scale. The emphasis is on speed – where tasks or activities are served up to employees who service the beast (the BPM system). Employees are not required to know much about the overall end-to-end process – they are focused on handling those calls in double quick time, or dealing with X number of work items per hour.

They are usually governed (controlled) by 'Procedures' that are *imposed* on them to ensure control and compliance with the pre-ordained approach (the term 'Procedure' is more accurate than 'Process' in this context).

Technology solutions are usually designed to lower the number resources needed (the denominator side of the productivity equation).[5] They rely on standardization and predictability – i.e. the nature of work is assumed to change relatively slowly. Further, there

---

is a relatively low level of trust between management and employees – i.e. control is often a key driving force.

At the other end of the scale, the business is seen more holistically – in terms of its people (including knowledge workers, front office employees, etc.), processes and systems – all evolving together. Processes might be thought of as a set of role interactions, with the communication and synchronization of participants as important as the sequence of activities.

The emphasis here is on the goal, satisfying customers, quality and better long-term relationships. Although difficult to quantify, technology solutions are often targeted at increasing the value proposition (the numerator side of the productivity equation).

Empowered workers make decisions at the front line, doing what is right for the customer and the business, rather than being constrained (unnecessarily) by a procedure. Employees seek to understand the entire process (rather than just the scope of the current task or activity), growing their business acumen and learning. Indeed, as people develop new ways of doing things, the organization can also 'learn'.

This doesn't mean that workers operate without business rules - policies, procedures, standards, responsibilities and levels of authority exist at all levels of the organization and they are continually changing.

A natural result of this knowledge worker orientation is that business 'Practices' evolve, as the organization keeps in sync with its market. At this end, the term 'Practices' is perhaps a more accurate than the rather vague word 'Process'.

Another way of thinking about it is that Practices reflect some of the 'vision' of management, whereas Procedures are all we usually end up with (the interpretation by technologists). If you unpick the vision, it becomes possible to describe the capabilities and behaviors required (i.e. the Role behaviors along with their competencies and capabilities), from which, one can then describe the desired organization. In a sense, at this end, roles act as an abstract way of corralling responsibilities and capabilities – providing independence or allowing the encapsulation of specialist responsibilities.

The problem is that the normal start point for most process modeling exercises is the organization itself – with its legacy of functional and political baggage. Having drilled down from functional descriptions, activities are connected up to describe procedures. Indeed, if the only visible representations of processes are boxes and arrows, laid out in a linear sequence, then that underlying process paradigm often becomes the widely accepted reality.
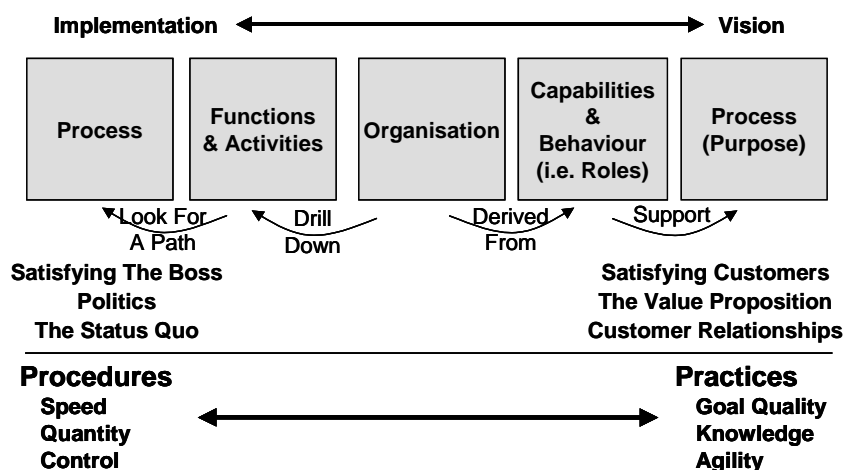

Figure 1 - The dialect of business processes is often confusing to those involved – Practices and Procedures offer more accurate descriptions

Procedures or Practices, compliance versus agility, control versus adaptability – it is not that one is superior to the other – the reality is that every company needs both. Every firm has

procedures and practices, but understanding which end of the spectrum is under consideration is essential for good process design.

The reality is that the vast majority of work as we know it is somewhere between – where procedural aspects are often entirely appropriate and, at the same time, degrees of flexibility apply.

We often think of Procedures as applying to back office workers and Practices as the domain of knowledge workers – but that does not reflect the real world either.[6] Another grouping along this spectrum is what we call 'case workers' – where jobs do not quite fit either category. Case workers need to exercise their judgment, but usually within certain constraints on what can, or cannot be done at any given point.

As an example, consider an international bank – everyone would agree that the teller should not get creative with bank drafts. They have very strict procedures (and systems) for handling money directly. On the other hand, try applying a rigorous procedural approach to those developing new trading instruments or those managing the banks largest corporate customers (or even its executives). They would rebel if everything they did had to fit within rigid computerized procedures developed beforehand. But, if those users were provided with accessible and easy to use mechanisms (for them) that allowed the dynamic recombination of procedural fragments that helped automate the repetitive aspects of their work (much like combining a series of word processing macros), then they would probably embrace the technology. They would probably start to regard those facilities as an essential part of their work environment, enabling them to achieve more in the time available.

Conclusion to Part I

There are aspects of every business that are, and always will be, governed by procedures. Yet, even within the same job, there are varying shades of empowerment depending on how long one has been doing the work.[7] At the same time, in the same department even, there are rules – policies, goals and principles that continually evolve – business practices.

Addendum to Part I

Following the completion of this segment of the paper (in early September 2003), we have since discovered an extremely relevant article in the latest issue of MIT Sloan Management Review (Fall 2003). "The Performance Variability Dilemma" by Eric Matson and Laurence Prusak discusses the need for evolving business practices from the perspective of Business Performance Management. The paper takes an ethnographic approach, citing several case study examples in major corporations and discussing why it is vitally important to understand and treat processes (procedures) differently from practices. Moreover, it offers guidance on assessing the predictability and frequency of knowledge centric business practices that do not easily translate into procedural replication.

Part II - Don't Put The Cart Before The Horse

Introduction

It is first necessary to develop a clarity about 'what' you are doing (developing the vision) before deciding precisely 'how' you are going to do it. Understanding the 'what' implies the development of models that help people see where the business fits into the various value chains involved. Here the emphasis should be on the capabilities required and the interactions between the parties, rather than the minutiae of the implementation.

Even if you think you already understand the 'what', it is still well worth modeling the role interactions and behavioral capabilities required within the organization independently of the procedures. Once that vision is clarified, then it becomes possible to define the degrees of flexibility required in the 'how'.

"All Models Are Wrong - Some Are Useful"

This quote, variously attributed to W. Edwards Deming and the economist George Box, could never have been more relevant. When we talk about process models we need to remember that they are often needed for very different purposes – from providing people with a basis for discussion about how a business relationship might be constructed, right through providing a set of executable descriptions that could drive the work.

The truth is that most firms struggle to understand their business processes – no matter what the notation. Perhaps one of the reasons is that people are usually only looking at one side of the equation – the procedural end of activity sequences. Procedures struggle to convey the richness required to accurately capture a business capability or the parallelism/synchronization required in many industries.

When constructing process models of work in the real world, one must remember that they are just that – models – i.e. there is some sort of abstraction of the real world. We are, in effect, making assumptions about (removing some aspect of) the real world from our representation in order to convey something – to communicate.

In our workshops with end-users (where we teach several approaches to process modeling), we ask participants what they want to capture with their process models. It is quite easy to fill a couple of whiteboards with their responses. And then, when you stop and ask the question "Do you want all of that to appear on one process model?" the penny drops. All of those different dimensions are important, yet we only tend to think of sequences of activities as the 'process'.

Clearly, it is necessary to first understand a process fully before redesigning or automating it. But developing a deep understanding (of the business, and its interactions in the value chain) follows from contrasting alternative perspectives, rather than slavishly following one 'true' method. People drive innovation though visionary insights into what is possible, and no tool or method can automate the generation of ideas.

In a changing world, it is more important to do the right thing in a way that is timely and good enough, than to do the wrong thing well, or the right thing too late. Sterile efficiency is less important (providing there is a reasonable level of management) than developing the right strategy. "Do the right thing" rather than "Do things right". This needs to be understood at all levels of the firm.

When trying to establish an appropriate vision, what is needed is a way of understanding (modeling) the capabilities and behaviors required while also exploring the concurrency of a process. We need a better way of capturing the interactions between the business entities involved – the internal actors, customers, partners, suppliers and systems. Technology systems are then placed in the right context – they have a role and can facilitate the process, but should not necessarily be regarded as the system. In the end, technology systems facilitate the business and its processes – not the other way around.[8]

But the aim of Business Process Modeling Notation (BPMN)[9] is to provide an effective standard for the capture of process models and then use these same models to drive the business environment. The theory is that business and technology users can share the same modeling notation while ensuring compliance with the latest BPM interoperability standards (BPEL4WS).

However, BPMN has only really considered the 'processes as activity sequence' perspectives drawing on approaches such as UML, IDEF, ebXML BPSS, Activity-Decision Flow (ADF) Diagrams, RosettaNet, LOVeM and Event-Process Chains (EPCs). All of these notations share one common assumption – that goals and purposeful behavior is reducible to a set of procedures – a perspective that is rooted in the work of Newton and Descartes.

BPMN does provide a rigorous and effective modeling notation for describing detailed implementation procedures. But presupposing that a single abstraction paradigm is suitable in all situations where people wish to model processes is sheer folly. A modeling approach used for process enactment is almost certainly inappropriate for exploring the ways in which a business relationship operates or the value chain functions.

There is another issue here - if your aim is to fundamentally understand a process (procedures and practices) then techniques that limit your scope to just one end of the scale will inevitably skew your perspective. At the heart of the procedural viewpoint are the artifacts that run through the process. Data and documents are 'implementation details' – they are the mechanisms of coordination of business processes. The need for, and use of, data to co-ordinate a process should follow analysis – i.e. "process precedes data".

A Case In Point

While all of this sounds a little academic and vague, let us consider an oft quoted but real life example. In the frothy days of Business Process Re-engineering, we heard all about Ford and their $100m invoicing system. It was planned to support the reduction in staff of the Accounts office from 500 to 400 people. Then some bright spark MBA noticed that Mazda had only five people in their Accounts function (Ford had just invested in Mazda). Taking into account the economies of scale and relative size of Mazda, they should have had around 75 people. The difference was that instead of paying on invoice, the Receiving Department at Mazda checked the delivery against the original purchase order – if all was well, then the payment was processed (i.e. without an invoice). Ford applied the same process to their situation, refined their purchasing processes and by the early 1990s had reduced their Accounts staff to just 125. And while they were at it, they saved the $100m earmarked for the new Invoicing System.

But the story goes on – although it becomes more anecdotal. Egged on by an army of consultants and numerous HBR/Sloan articles, the rest of the US auto industry followed suit – they realized they could largely do without invoices. So, when one of the major US auto manufacturers went to visit Toyota, the benchmarking team was amazed to find that they only had six people in their accounts department. Somewhat befuddled, they asked how it was possible that such a major business could have so few people in their accounts function. The answer – even simpler still. Instead of getting very good at purchasing (i.e. having to create more effective purchase orders that were matched with goods received), they paid their suppliers when they made cars! Every car had one steering wheel, a given number of doors, a sunroof, etc. It was up to the supplier to ensure that suitable supplies of quality goods were present at the plant. Sure, they now had to manage their suppliers more effectively and provide them with information access (stocks available, forward orders, etc.), but they had found that they did not need to manage specific orders or invoices.

The moral of this story - to achieve quantum leaps in business performance, concentrate on finding ways of doing what is needed, but without the traditional mechanisms of coordination. Do not follow the paper trail and then automate it. Think carefully about why those coordination mechanisms are needed in the first place and what would happen if they were removed.

The approaches that were considered when defining BPMN all have a strong, procedural/control flavor. As a result, the proposed standard concerns itself with the definition of procedures. And, without extremely thoughtful design, representing the world using only procedures can often provide a faster route to a bigger mess. In a sense, the only thing we have learned from the mistakes of the past is how to make them all over again.

One of the key problems with this procedural approach is that it does not help identify bad processes. Sure, a few activities can be combined and responsibilities moved around, but with a flow diagram it is possible to prove just about anything. You can even travel in time - just put in an arrow and connect back to the beginning.

Perhaps it is best to demonstrate this point with another anecdotal story from the days of BPR. The story comes from the world's largest supplier of chemicals. After employing an army of consultants to model and map the 'North American Sales Process', a presentation was planned to the board in the UK. The resulting process model had its own seat on the jumbo jet across the Atlantic (in a 5 foot high tube) and was set up in a large conference room, covering 60 feet of wall space. After presenting their findings, the chairman of the board asked a simple question that stopped the consulting partner in his tracks – "Is it a good process?" the point being, it was impossible to tell.

Conclusion

The capability to grasp a new way of working is related to how the problem is recognized and explained. Finding process breakthroughs and how they relate to business, does not come from endlessly executing each step in the sequence in the hope that real alternatives will present themselves. Breakthroughs normally come about through the development of fresh perspectives - i.e. by setting up the problem differently.

The way you set up the problem is in turn influenced by how you see the world. If all one has in one's vocabulary are sequential activities, then the whole world can appear as though it is made up of linear flows. It is simply not possible to model all purposeful work activity as a set of neat procedures. Moreover, we doubt that, in the end, the current conception of BPMN will satisfy all parties.

Paraphrasing Aaron Levenstein (talking about statistics), "Process Models (Statistics) are like a bikini. What they reveal is suggestive, but what they conceal is vital".

Part III – Problems Modeling Processes

Modeling business processes is fraught with problems. In Part II, we closed with a paraphrased quote from Aaron Levenstein (talking about statistics) - "Process Models (Statistics) are like a bikini. What they reveal is suggestive, but what they conceal is vital". At the heart of this statement is the fact that when we think of process models, we must remember that we are building models – mere representations of the real world. And given that the term 'process' is poorly defined at best, everyone has a subtly different perceptions of what constitutes business processes and, more importantly, they have many different reasons for wanting to construct process models.

Despite the claims and aspirations of some in the BPM industry, the majority of process modeling is still used to support people as they communicate with their fellow workers, usually as they try to build a shared understanding. Sometimes these models will form the basis of a business reference or support a proactive training environment. At other times, the aim is oriented toward helping improve efficiency, responsiveness or customer service. Models are often used to support discussions between potential business partners on how the two organizations could work together.

Process models are also used to support discussions around 'best practice' between different groups inside the business and with benchmarking partners. Further, whenever we start talking about business measures, they usually require a process model as a framework against which analysis is undertaken. Either way, the primary need is to support communication amongst people.

Of course, process models can also support the implementation of a BPMS or workflow environment. Here models are constructed to provide an unambiguous set of instructions to drive work through the system, integrating people and business applications while ensuring all appropriate business rules are observed. We also need models that define how a series of individual processes are orchestrated together to create another process or service and, in the world of Web Services, the source of these constituent services could easily be external to the firm.

But when we construct models, we need to remember that, in the course of modeling, we make some assumptions about how the real world is structured. When drawing a graphical model, we use icons (and lines) to represent the semantics of our assumptions. Some approaches are rigorous, yet difficult to understand. Others seem easy to understand, but lack the rigor required to communicate exactly as intended. It is these semantics and the icons used to represent them that are at the heart of the religious wars that have raged around modeling methodologies and notations. Some people become fixated with one approach to the exclusion of all else, seeing only the benefits of their favored methodology.[10]

In parallel, we have seen a plethora of tools and bespoke process modeling approaches, with each vendor defining their own semantic constructs and symbols. As customers ask for subtly differing concepts, the number of symbols usually grows. In some cases, this 'scope creep' has made the products almost unusable. One of the products we looked at in the late 90's had grown to include as many as 24 different modeling notations and over 80 different types of icon available on just one of those diagram types.

In parallel with this, those looking to actively support the process (i.e. workflow and BPMS vendors) have developed their own proprietary modeling environments, full of constructs that reflect their own particular view of organizational behavior.

Of course, with such fragmentation, it is hardly surprising that users are frustrated. Without training for both the modeler and the reader, the iconography is often difficult to understand and hence, the definitive meaning impossible to understand. Too often, the true intention of the modeler is lost as the reader misinterprets the approach or the 'enhanced' semantics used by the modeler to suit his or her own short-term requirements. Very often, even with a rigorous observation of the modeling approach, it is still only understood fully by another expert of that modeling technique.

Modeling Procedures

Virtually all widely known methodologies of process modeling are based on procedural concepts. Generally, when people talk about modeling a process, most people tend to think of 'Flow Diagrams' incorporating concepts such as sequence, activities, tasks, etc.

On the one hand, flow diagrams can provide some useful insights (if never explored before hand) and understanding them does not require any real shift in understanding (they are based on our neatly ordered Newtonian view world view). On the other hand, without a great deal of enhancement, they are completely lacking in rigor and only provide a crude sign language with an extremely limited vocabulary.

The BPMN modeling notation is designed to bring some order to this continued fragmentation while providing a rigorous, yet understandable approach to modeling procedures.[11] Moreover, BPMN aims to bridge the "technical gap that exists between the format of the initial design of business processes and the format of the languages, such as BPEL4WS, that will execute these business processes".[12] It is likely that Business Analysts and technologists will prefer to work at the abstract level of BPMN and have the appropriate BPEL4WS generated by whatever toolset they are using.

Overview of BPMN

BPMN is intended as an extensible notation, allowing modelers and modeling tools to add non-standard elements or artifacts that satisfy particular need such as the unique requirements of a vertical domain. However, the core set of boxes and arrows used in flow diagrams have been enhanced in several crucial ways.

'Activities' are separated from 'Events', 'Gateways' and the Flows between them (there are various types of each). A swimlane metaphor is used to group graphical elements - 'Pools' usually represent organizational boundaries and 'Lanes' the internal functions.[13]

BPMN consists of three main diagram (sub-model) types:

- Private (Internal) Business Processes are internal to an organization. A single private business process will map to a single BPEL4WS document. If swimlanes are used, then the Private Business Process is contained within a single Pool. However, multiple Private Business Processes might be shown a single BPMN Diagram, with Message Flows used to show the interactions between Pools. Thus, a single diagram may show multiple private business processes, each mapping to a separate BPEL4WS document.

- Abstract (Public) Processes are used to represent the interactions between a Private Business Process and another process or participant. Only those activities that are used to communicate outside the private business process are included. The Abstract Process shows to the outside world the sequence of messages required to interact with that business process. An Abstract Process is contained within a single Pool and may be shown within a larger BPMN Diagram to show the Message Flow between the Abstract Process activities and other entities. If the Abstract Process is in the same Diagram as its corresponding Private Business Process, then it is possible to associate the activities that are common to both processes.

- Collaboration (Global) Processes are designed to show interactions between two or more business entities – think of it as a set of collaborating procedures (one in each pool) with the message exchange patterns between the entities. This is one of the most powerful ways of modeling a process.

These diagram types are mixed and matched, allowing the notation to support a wide variety of potential uses. Depending on the purpose and the intended audience, one could model a detailed Private Process with its interactions to one or more external entities. The modeler could show two or more detailed Private Processes interacting with each other. Other diagrams could show how a detailed Private Process relates to an Abstract Process or how a detailed Private Process relates to a Collaboration Process. (The list goes on - see the BPMN spec).

However, if too many types of sub-model are combined, such as three or more private processes with message flow between each of them, then the diagram could easily become too hard for someone to understand. As with all modeling exercises, the modeler should clarify the purpose of the diagram and ensure that the scope is not too great.

While the BPMN approach to modeling procedures is likely to eventually become common in BPMS environments (both process modeling repositories and process support environments), there are still issues to be addressed.
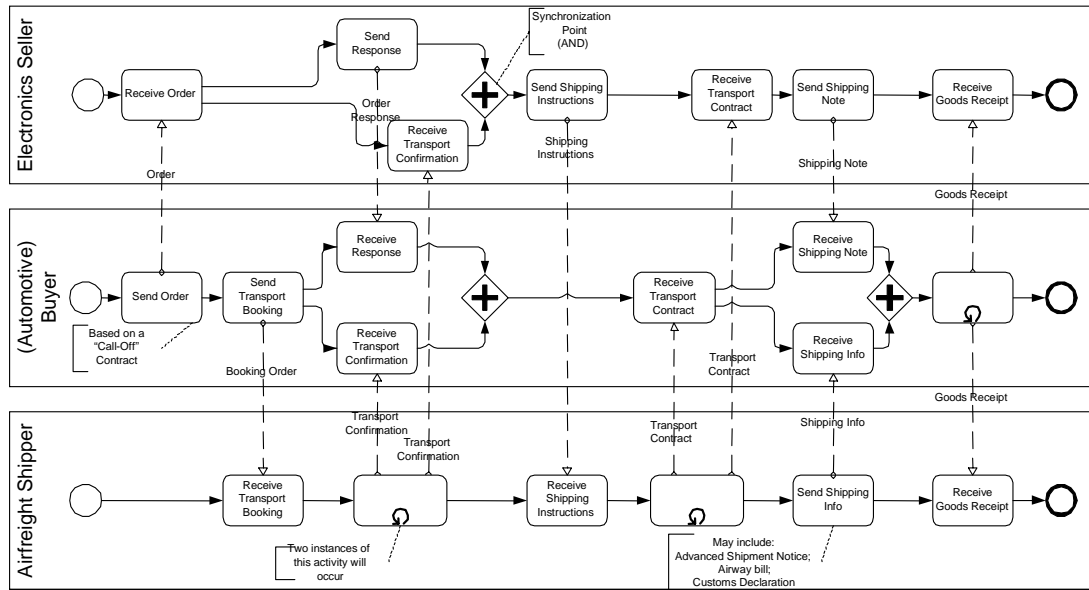

Figure 2 – Here the modeler uses a Collaboration Process model to show how the entities interact in an e-commerce scenario (source BPMN Working Group)

It will be interesting to see how many vendors actively support this notation in their products. Privately, some of the vendors have pointed out that there is little in terms of competitive advantage (for them) in supporting this standard.[14] It will take pressure from user organizations to ensure that vendors quickly incorporate BPMN support into their products.

Modeling Practices

Most readers will quickly identify the notion of process as procedures – where the core definition is constrained to activity sequences connected by arrows. Techniques such as BPMN add the graphical capability to show which organizational entity is responsible for undertaking activities.

While BPMN is fine for the mechanistic procedures of the workplace, they do not make easy to envision the more holistic world where most knowledge workers and executives work. Imagine trying to encapsulate the work of a busy advertising account manager or PR executive with such procedures. It just would not work – all cases appear to require unique solutions. But at the same time, these cases usually follow a number of generic patterns and the employees themselves are often constrained by contextual business rules and policies that influence their flexibility. We have described these types of processes as 'Practices'.

When thinking about organizational design, we need the capability to model practices and describe the sorts of competencies and capabilities required. Moreover, we need mechanisms that can more easily describe the parallelism and synchronization required in many of these knowledge worker centric processes.

During the 80's and 90's, several alternative approaches to process representation were developed that sought to provide an effective solution in these areas. In our opinion, the best of these is probably Role Activity Diagrams (RADs)[15] [16] and the Capability and Context models of the MooD methodology. Neither of these approaches is perfect, but they do have value and should feature in a methodology toolbox.
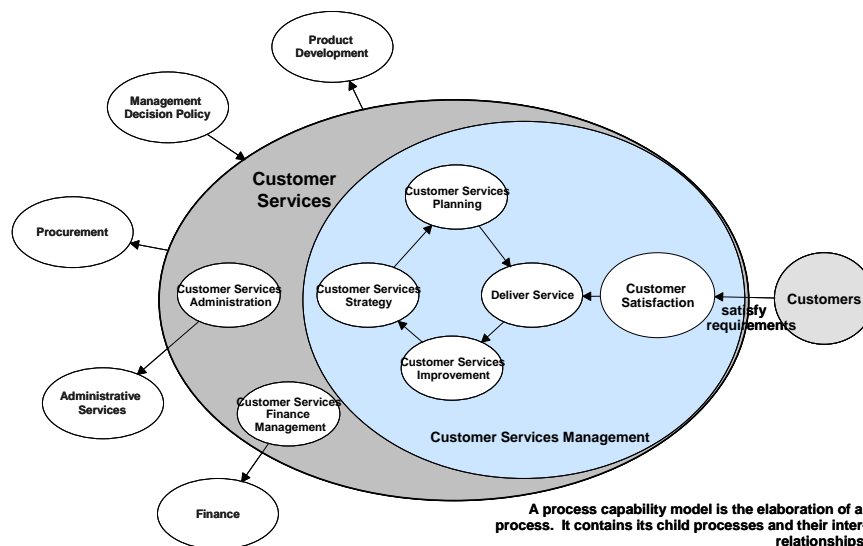
Business Context and Capability Modeling

Figure 3 - A Business Capability Model (source MooD sample repository) [17]

A Business Context model describes the boundary of an organization or a part thereof – it depicts the strategic context within which the firm operates. The boundary is defined in terms of the strategically significant influences that the organization must monitor (and respond to). Clearly, most commercial organizations would recognize customers as a

strategic influence, along with other external entities such as suppliers, technology, regulation and so on.

With a strategic context, it is possible to then represent a set of constituent capabilities that are required to support the organization in its strategic context. Capabilities describe organizational proficiencies or competencies – things the firm needs to be able to do on an on-going basis. They might be defined in terms of the relationships they have with each other and the outside world, objectives, KPIs and issues or required improvements. From a modeling point of view, a defined business capability can then be re-used from one model to another.

This approach allows a qualitative approach to capturing what the organization needs to achieve at a higher level, rather than attempting to specify process sequences or functions. Such an assessment of a firms capabilities facilitates a richer discussion on business needs, around which stakeholders can more easily reach consensus without getting bogged down in to the detail of activity sequences or functional decomposition.

Role Activity Diagrams

RADs were first developed by researchers at ITT working on concurrency problems in real-time operating systems used to control telephone switches. The technique makes use of a theoretical approach called 'Role/Activity Theory', which is a theory of organizational behavior – it grew out of computer-related research into communication, concurrency, and cooperation between independent entities, whether people or microprocessors.

In their simplest form, RADs focus on the Roles involved in a process, their behaviors (the actions they undertake) and their interactions. RADs are a formal graphical language for describing the behavior of organizations. They model "**how a Role changes state as a result of the actions and interactions that occur**". The two central constructs are concurrently cooperating Roles and Interactions. RADs specifically exclude the modeling of data and documents (although these can be shown on the diagram for completeness).[18]

**Figure 4 - A sample Role Activity Diagram to describe a scenario where a customer**
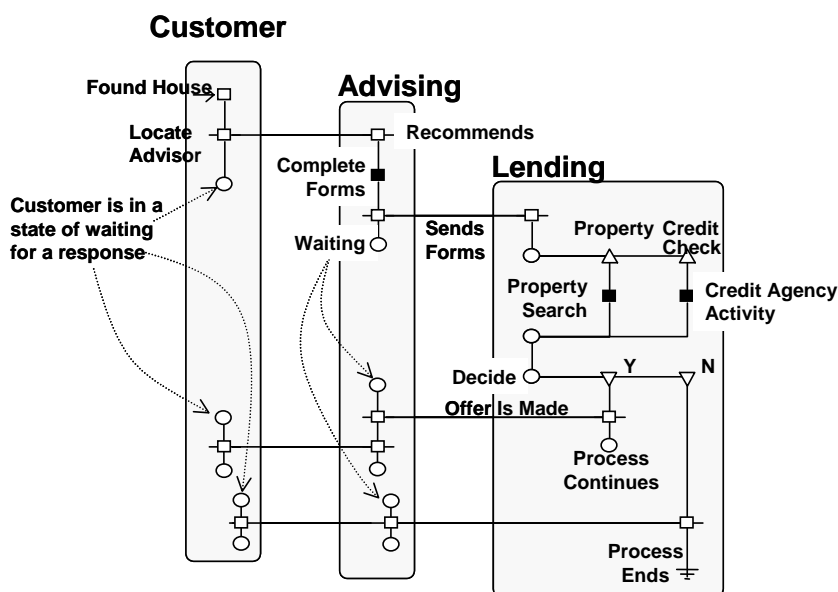
decides to buy a house

Understanding RADs – A RAD diagram is a representation of a 'state machine' – i.e. they can be read from any point. However, it is likely that the modeler drew the diagram starting with a trigger that often appears at the top left (the arrow) where the Customer finds a house. In RADs, the vertical lines explicitly represent a 'state' of the Role, while Interactions (horizontal lines joining Role states) may in turn be highly complex interactions. Activities (Actions) represent work effort and are shown with a black box – i.e. something that the Role does to change state

---

that does not involve interactions with other Roles. Meanwhile 'interactions' are work efforts performed by many actors together and synchronized. In the model above, the Advising role has only one Action (Complete Forms – perhaps incorrect in this model since the Customer would probably complete the forms). The Lending role undertakes a Property Search and Credit Agency Check in parallel (notice the arrowheads) before deciding whether to offer the mortgage or not. A decision is shown with the arrowheads inverted, implying the case would travel down one path or the other. In practice, another more detailed model might be used to show the way work is handled inside the Lending role. In this model, the Advising and Lending roles are deliberately loose – the Advising role might represent the customer's normal financial advisor, the local bank manager, or a mortgage company representative. It makes no difference to the model and the behavior of the Lending Role is still the same. How a Property Search is undertaken is not defined. If one were to 'open-up' that black box, one would find another process involving a different set of roles (local government, land registry, etc.). The waiting time of the Customer would become an attribute of the state line running below the Locate Advisor interaction. The state of a role might be explicitly labeled (as in the Customer role), but are often implied without specific labels. In this model, the pre and post state descriptions for the Customer and Advising Roles were left blank following the rejection by the lending authority (because they were not of interest to the modeler or the audience).

Figure 5 - A RAD showing the quality process for an online repository of
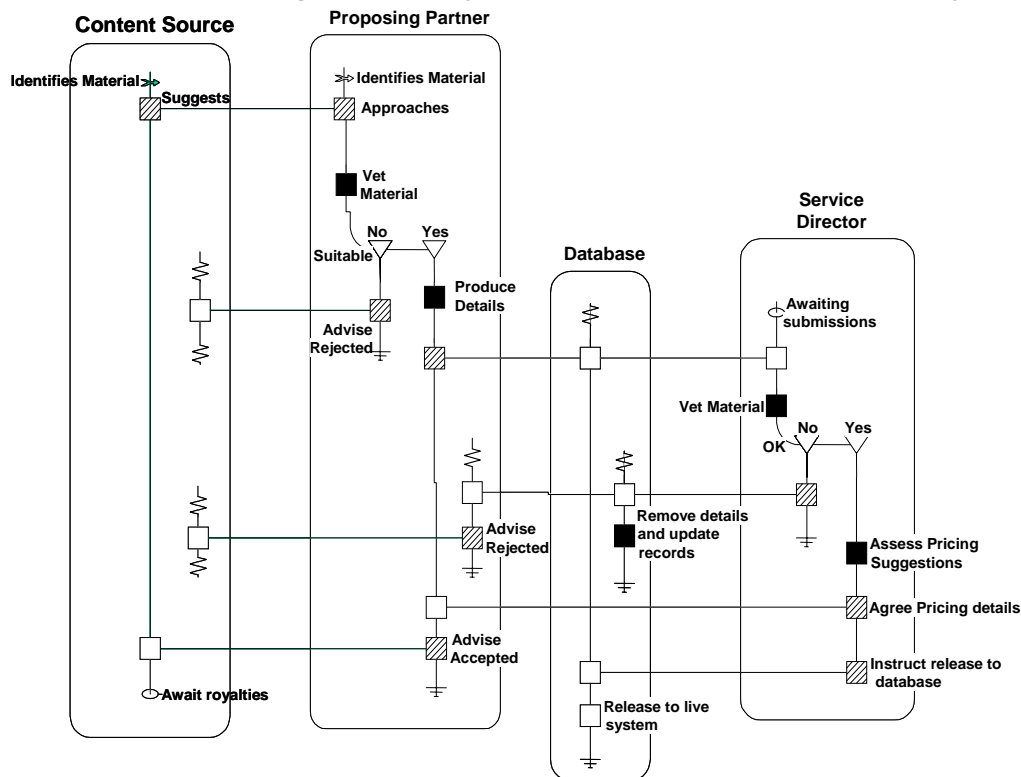
benchmarking content

Superficially, RADs appear similar to the communication diagram type of BPMN models (and LOVeM, Rummler Brache, etc). The similarity however is entirely superficial – i.e. with BPMN, the swimlane is used to represent the organizational entity (function) that is responsible for carrying out an activity.[19] On the other hand, RADs focus on the **Role as a set of behaviors**. Interaction, concurrency and activities are the first class modeling constructs that change the Role's state (rather than merely showing how a process moves on through a series of activities).

Figure 4 shows a simple RAD representing the customer perspective of the mortgage application process. A different model might be used to understand how the internal roles of the lending institution might interact with each other and the rules that govern the decision. On the other hand, if the organization pursued a one-stop shop approach, with empowered customer service representatives, this model could easily provide sufficient detail to help the individual understand how the process really works.

In Figure 5, an online benchmarking service attempted to model their approach to the ensuring quality of content that would appear. The process is deliberately loose – it applies equally well to book reviews or case studies, technology product reviews or conference listings. Notice that either the Content Source or a Partner in the Network might trigger the process. It makes no difference which. Either way, the proposal is validated (against a set of principles that are appropriate for that service) before assessing access costs and agreeing pricing details before release onto the live system.

Of course, because RADs do not share the same underlying paradigm as that used by traditional flow diagrams (or even those using swimlanes for role attribution), the models should not be read in the same way. Another key point is that RADs are capable of expressing all of the synchronization and concurrency required between roles (which flow diagrams struggle to achieve).[20]

RADs also suggest where the interfaces between processes are required and where process fragments might be invoked for specific purposes. The 'black box' construct offers a way of structuring a process – strictly speaking, a black box cannot 'contain' an interaction with other roles on the diagram. But when 'opened up', one might discover an entirely separate process with different roles (than those shown on the model). Moreover, the black box can also show where the actor fulfilling the role may use his or her own judgment, on how to complete the activity. Process measures and guidelines can also be attached to the framework outlined in the diagram.

In our experience, RADs are extremely intuitive and require little more than a few minutes training to read and understand. A RAD of a 'procedure' would probably have one start point, flow through a detailed network to one or two outcomes, and would probably involve concrete roles. On the other hand, a RAD of a 'practice' could have many triggered threads for different situations. It would be characterized in a general way and would probably involve 'abstract' role titles.

Further, modeling using RADs can be learnt in a day or so. They compress a lot of information into a simple diagram.[21] They are extremely good at supporting discussion or looking at a part of a process.[22] In a workshop scenario, end users often become extremely vocal when discussing a process modeled with a RAD, which is in stark contrast to the situation where flow diagrams are used.[23]

In their simplest form, RADs are mere pencil and paper diagrams but it is also worth noting that RADs have an extremely formal mathematical underpinning.[24] It is possible to 'prove' a process using the technique (something that is virtually impossible with flow diagrams). Certainly, they are better when it comes to identifying flaws and logical inconsistencies.[25] For instance, in Figure 5, it just would not make sense to make a connection from somewhere on the bottom right to the state of the customer at the top left (although it would make sense for the customer and advising role to find an alternative lender should they receive a rejection).

Using RADs it becomes relatively easy to understand the perspective of your customer (or any other role for that matter). One can easily model the interactions that a customer has with your business (say via a customized corporate portal), then extend in either direction to think about the customers own processes or how the systems and applications of the firm support these interactions.

In the UK, a wide variety of organizations use RADs to help them better understand and communicate their processes.[26] Some businesses use them to help new employees understand the overall process and where they fit, allowing them to suggest changes since, in the end, employees know their jobs best).[27] [28]

Conclusion to Part III

A key value of process models (to some extent overlooked by the BPM industry) lies in their ability to support education and communication (of ideas). This in turn, depends on the ease and speed with which non-specialists can pick them up and use them.[29] Technology (and the procedures that it can handle) should not be the starting point – it is getting human beings

aligned around common objective, which implies developing a deep appreciation of the process.

The truth is that there is no single process-modeling notation that is applicable to all needs. It is always a good start point to understand what you are modeling, why you are doing it and who the audience is for the model. This will help dictate the notation.

Certainly, there is a need to better understand processes (especially their interactions) before getting to implementation - flow diagrams, no matter how rigorous the notation, simply do not deliver. We believe that Capability models and Role Activity Diagrams offer an effective contrast to the procedural approaches that dwell on the sequence of Activities and assignment of their responsibility.

Part IV – Revolutionary Objectives, Evolutionary Journey

Enterprises have long sought a technological silver bullet – an approach to supporting work (using IT) that delivers rapid time-to-market, consistent customer service and control, yet without sacrificing flexibility, agility and corporate adaptability. The ultimate technological nirvana is an approach that can deliver working systems based on the ideas of business people – without the endless involvement of an army of programmers. We look for the capability to re-use concepts and ideas – our corporate intellectual property that exists in many forms, at low cost and with minimum risk.

It is an appealing idea and, for a time during the 1990's, some people thought Enterprise Resource Planning (ERP) systems offered just such a solution. But the reality did not live up to the expectation. An assumption was made – that the generic processes embedded within the system would suit all businesses. Quite apart from the initial expense of implementation and the straightjacket imposed by the cost of modifying a system afterwards, some firms are now starting to balk at the prospect of costly upgrades to the core software.[30]

Business Process Management (BPM) technologies promise a similarly tantalizing future. Initially billed as the future of workflow management, now every vendor has a BPM flavor to its marketing. They all promise business agility combined with fast implementation and efficient processes. To some extent, they all provide the capability of driving down the cost of execution against a given business vision.

But, as our processes and value chains continue to evolve (ever-more rapidly), firms need more than empty marketing hype. As firms change and expand, they want absolute stability and efficiency in their day to day operations. Yet at the same time, the organization also needs its people to explore and re-design their processes, instantly deploying new approaches to reflect alterations in strategy or exploit new opportunities. Organizations want standardization and compliance, but alongside that, the ability to unlock change - in particular areas, for specific roles or even just to replace a discrete application (without affecting the rest of the process).

Outsourcing non-core activities, moving back office functions off-shore[31], implementing Sarbanes Oxley (SOX), working with new customers and partnership arrangements, radical changes in the supply chain – all of these business level changes require radically redesigned processes.

But from an IT perspective, radical change usually implies increased risk – we saw what happened to the out-of-control ERP implementation where the core business processes was petrified inside of the core application. The IT function needs the capability to roll out revolutionary change – but to do that incrementally, in evolutionary steps.

The key to achieving this sort of evolutionary change is BPM. BPM technologies enable the firm to build an independent layer of business processes (the procedures), maintaining them separately from the functional applications and the organizational structure. As new procedures and practices are developed, the firm can deploy them safely – without upsetting other infrastructure components.

Taking this a little further, BPM technologies allow the firm to package its process capabilities into discrete services – services that are re-used to create ever more sophisticated facilities and offerings.

Most enterprise technology vendors see the opportunity and are busy attempting to manipulate the categories and definitions – arguing about what is, and what is not a BPM technology.[32] Indeed, those making the most noise about the need for standards are often those that are least open to the wider interpretation of Business Process Management.[33]

Humans in the Equation

Following the interactions between the protagonists inside the standards bodies, one might think that this was only about allowing several sets of computer systems to interact with little or no human involvement. But when one looks at the broader picture of business

processes, BPM implementations clearly need to support a much wider set of usage scenarios.

When you include humans in the equation, problems start to arise. Customers seldom behave in a neatly ordered fashion and many application areas just do not lend themselves easily to procedural control – the procedures become just too complex to maintain.

Many who sought to standardize processes and downsize their operations have now started to realize that business processes show a greater similarity to the complex and chaotic behavior of fluids, than the orderly progression of Henry Fords' production lines.

This divergence implies a number of strategic questions for process architects and technologists. Is there a single 'best way' of doing something? In a world where the only sustainable competitive advantage is the ability to innovate and adapt, what role has a control-centric view of work? How should the organization be structured to enable maximum responsiveness in an unpredictable market place? How can the firm ensure compliance (a la SOX) in an empowerment oriented culture? Culture is like concrete – it sets and is difficult to shift. How can the framework of processes support an evolving culture (as surely it will when the company is "un-bundled, out-sourced and off-shored")?

A key point, made earlier in this paper, is that every organization and every department within it, needs a mixture of Practices and Procedures – a unique mixture of control and efficiency at one end contrasted with agility and knowledge oriented work at the other. The question for management is where to draw the lines and how will this evolve over time. The interpretation of where to draw the lines is different for every firm, depending on their culture, management predisposition toward risk, the degree of control required, etc.[34] Answers to some of these questions and where the boundary lies on the continuum process adaptability are not immediately obvious.

BPMS Capabilities

For all of this discussion on business practices and evolution, we are still stuck with what computers can do today – i.e. manage procedures, something that is firmly oriented toward the efficiency end of things. To a varying extent, products also enable the firm to evolve their processes. And it is how these products enable such evolution that is interesting – what ever you put in place will require modification at many different levels - probably very quickly. So you need to be able to unlock change in certain areas and for a given set of roles.

BPMS products share one common feature – the ability to route work throughout the business based on activity sequences (procedural rules), leveraging some sort of organizational structuring mechanism – connecting people, applications and legacy systems.[35] They all keep track of what happened (allowing detailed audit) and are now starting to incorporate business activity monitoring suites that allow process owners to assess the performance of their operation against key performance indicators (KPIs). Further, some of these environments integrate with process simulation toolsets, supporting business managers in assessing the potential implications of a proposed change (leveraging existing performance data).

However, even with the most flexible of underlying architectures, a lot of people initially design highly mechanistic process architectures. They seem to have an irresistible urge to connect everything together into some vast network. Control and perceived efficiency are foremost in their mind, with agility and evolution an afterthought.

With this sort of approach, past experience shows us that complexity (and with it the Total Cost of Ownership), grow exponentially, as new functions are added and as processes are re-designed or re-implemented. While a modern BPMS can enable the firm to change the underlying framework of processes, this can still be an expensive exercise.

Historically, workflow products required that all potential paths and exceptions were predicted and built into the process definitions beforehand. But, it is virtually impossible to take into account every potential eventuality and circumstance. Over time, as more

'exceptions' are built in, process maps become horrendously complex (and with that, a higher TCO). These complex multi-step processes are often simplified (graphically) through the use of sub maps (often called sub-procedures), which are called from the main process map.

With most traditional workflow engines, these sub-procedures are usually embedded in the parent process (rather than maintained as separate components called at runtime). Parallel paths of activity are often involved in both the parent and the sub-procedure, which may need to synchronize at certain points.[36] Most products also tend to embed any code needed to integrate with third party applications in the tasks and activities themselves. Change the third party application and each piece of code must be revisited and re-developed. The potential for spaghetti code is enormous.

Moreover, such systems usually struggled to support the flexibility needed by front office and knowledge workers. The challenge, for vendors and organizations alike, was how to handle exceptions and who should make the appropriate changes (the Supervisor quickly became the bottleneck).

State Engines represent a subtly different approach, one that seems difficult for many to comprehend. Engines designed around this type of architecture are usually based around the notion of processes as 'objects' or 'agents'. They manage the persistent state of the process (and its instances) over time, adjusting to each new event and ensuring that all participants in the process (humans and software systems) are guided towards a shared goal.[37] For the purposes of this paper we will not discuss this type of architecture in any depth as products are few and far between (although they offer greater promise and flexibility – see Note 37).

The latest BPMS environments are, for the most part, still based on the core premise of standardized procedures. But they are much more flexible while also providing the capability to extend and reuse existing technology assets. According to IDC, BPMS environments can reduce TCO by up to 50% over the lifecycle of a particular process.[38]

BPMS environments normally have better ad hoc functionality to get around those show stopping scenarios (where an unpredicted exception occurs), allowing qualified users to move the work item on to another point in the process definition (or jumping back to a previous activity). By freeing up the ability to step around the process (outside of a pre-defined path), it means that the process description itself becomes an order of magnitude easier to develop. You no longer have to describe every possible path or navigation route. Instead, you concentrate on the core – leaving the exceptions to be worked out after deployment. Gradually, these exceptions are then incorporated into the process definition. This 'skinny' approach to process development has a dramatic impact on TCO and time-to-market.

Some products provide mechanisms that allow new procedural fragments to be added to the process instance in real time, binding the new fragment to the parent definition for this specific case. Products also now tend to allow the integration of business rules engines, further simplifying the potential paths through a process.[39]

Deciding on a BPMS environment is not a trivial exercise.[40] Each different engine has unique facilities that the process design can leverage. But regardless of the chosen engine, firms should be careful to think about the ways in which their processes will work within it.

A Process Framework for Implementation

Most importantly, firms need to think about how the entire process framework will evolve over time and how it will support the more difficult knowledge worker based business practices. Developing an appropriate process design requires a deep understanding of the issues – a set of perspectives that are normally only gained through experience.

However, if the process framework is thought through carefully, TCO is reduced significantly and, perhaps more importantly, overall flexibility and adaptability enhanced. By breaking down processes into simpler reusable components (process fragments), they become easier

to understand and evolve. Their interfaces are tightly defined allowing individual process sub-services to take on a life of their own. Change that enterprise application at the back end for another compound service – no problem for the user interface; it need not know anything about it.

This of course assumes that the BPMS environment supports the notion of procedural fragments. Many of them do not. When some vendors talk about sub-procedures, they really mean sub-maps of a parent procedure (which is not quite the same thing).

To leverage the capabilities of the modern BPMS – process designers should forget about predicting everything in advance. Instead they should look for systems that can support the dynamic binding of process fragments at runtime. There are a few flavors here – Business Process Orchestration, others talk about incremental development of Service Oriented Architectures[41], or use the language of Case Handling.

- Business Process Orchestration implies identifying and cataloguing relevant business services (processes), then combining them as needed based on a defined and flexible process model. It makes little difference to the engine whether these services are internal – they could just as easily combine several external Web Services with a set of processes that integrate the employees of the firm and leverage existing IT investments in such areas as ERP and CRM systems.

- The language of Service Oriented Architectures revolves event-driven IT services that are constructed from smaller, yet entirely discrete, services. Such architectures are generally more adaptable to change over time. Whilst this is not especially new (think OSI, CORBA, COM, etc.), Web Services provide a standards based solution.

- Case Handling concepts are based around the familiar metaphor of a desk file, with all its related documents and customer interactions. Most professions such as architecture, legal and accounting operate on cases. Similarly, one could regard virtually all government and utility processes as case handling. Insurance claims processing is a clear example of Case Handling. In the words of the Head of Claims for Commercial Union - "every claim is an exception".[42] Whilst most claims of a given type can share a common approach to their resolution, there are often specific circumstances that were not catered for in the standard operating procedures. Indeed, virtually every CRM situation could be thought of as Case Handling. While cases are handled manually (without computer support), humans interpret the rules and do what is best. It becomes much harder to do the right thing when we have to rely on codified procedures inside a BPMS (even if they are linked to a business rules engine)

Think about how these sorts of capabilities can affect the design of processes for use in the front office. Customer service representatives can now choose those elements that are appropriate (to a customer's problem). Flexibility is increased yet work is still carried out in a compliant and consistent manner.[43]

The business can now consider supporting that letter from hell scenario – where a bank customer writes in to notify the bank of a recent divorce. Freeze the joint current account, there's a change of address to handle. Oh and there's the mortgage, two trust funds to be set up for the children, pension implications and … instead of thinking that you must have an end-to-end process for all of that, you should have a case management capability that deals with multiply instantiated process fragments.

What we need is a network of multiply-instantiated processes within which there are networks of multiply-instantiated threads (process fragments). If designed correctly, expert end-users can recombine these fragments in novel ways, allowing them to produce new products and services much more quickly than would otherwise have been possible with traditional approaches. They can even try out alternative process designs side by side if need be (assuming a high degree of deployment control).

One rule of thumb in designing your process framework, the greater the level of empowerment, the lower the cost of ownership. Here the assumption is that the 'expert' end-users themselves (those who really understand the way work gets done around here) can model and deploy processes that meet their needs.

Managing Data & Integration

Even if products support dynamic binding of procedural fragments, there are still issues with regard to the way in which data is handled. When dealing with sub-procedures and process fragments, there are problems around how variables and information are mapped from the calling process. This information provides the 'context' to users and third party applications.

To ensure consistency, some vendors provide a template mechanism to guarantee that all related information in a sub-procedure is mapped correctly to the parent procedure. In this way, a consistent, 'shared data space' is created across all the potential steps in the process.

Others provide visual mapping tools to enable the mapping of data associated with the case onto the functionality of an activity or sub-process. While this can be extremely flexible, care is still needed to ensure that all potential sub-procedures (in a dynamic binding situation) are able to make use of the case relevant data in the calling procedure.

A separate issue relates to transactional control. Typically, a business process involves a number of transactions – e.g. looking up information from a customer database, taking an order, adding a new order to a database, updating prices, debiting an account, etc. Virtually all of this data exists in third party applications. To ensure that all elements in these systems are updated appropriately, the BPMS needs to keep track of each system level transaction. If all steps are completed successfully, then the entire data set is committed. However, if one or more elements fail (say a database is offline), no data should be committed and the overall transaction rolled back. Some modern BPMS environments now provide the capability to ring fence a group of steps and treat them as one composite transaction.

When it comes to integrating with third party environments, best in class products provide the capability to dynamically 'introspect' an application and encapsulate discrete elements of functionality in a component.[44] During development, each application is introspected to identify discrete elements of functionality, which are then catalogued along with the stub program needed to call that functionality, its arguments and the expected results (collectively this information describes a 'Component'). Once these components have been catalogued, they are then available to end-users to call from within their process designs.

Conclusion

This paper has sought to provide more than a call to arms (around BPM). It has outlined a practical methodology for getting there. Getting a successful implementation means making hard decisions about culture and structure - before leaping into the morass of activity sequences. Issues will inevitably arise - about the tension between **procedural control** and **knowledge-based practices**. Where the boundary lies is different for every industry, company and organizational grouping.

The methodology covers:

- Understanding **what** you are doing before getting into the **how**.

- Some graphical approaches that provide a fundamentally different set of perspectives from that offered by flow diagrams. True insights only come from contrasted perspectives rather than re-applying the same mind-set.

- Some ways of thinking should be used when translating the 'what' and 'how' into modern BPMS environments.

In Part IV, we have also sought to cure the reader of a rushing in to automate the first thing in sight – with a tightly specified end-to-end procedure. These sorts of process constructs

only really work in straight through processing situations - where humans are usually not involved.[45] Normally, a Case Handling style approach is more appropriate - where a **parent process is dynamically calling sub-procedural fragments** that do specific jobs.

With careful design, integration with enterprise applications need only be done once. Each application can be introspected (explored) and adapters implemented.[46] Each element of required functionality (from the third party application) is wrapped in an executable program (to create an atomic service) that is callable from activities or tasks in procedural fragments - creating compound services that might involve humans. Process applications are built up by combining these procedural fragments – i.e. compound services may contain other compound services. Each component of functionality is maintained independently from any user interface requirements, ensuring portability and re-use.

Lowest TCO is achieved when you leave (expert) end-users in control of both modeling and, as much as possible, the implementation. Business rules engines (to which the business users have access) can also add value in simplifying process models. Most importantly, the business gets to control its own destiny rather than kowtowing to an IT agenda.

The IT department still has an important role - to set up the overall environment, supporting the business unit and ensuring that the overall process framework works appropriately. Secondly, they will still need to set up any integration with third party applications. Finally, in conjunction with the business, IT will need to develop a centre of excellence that provides additional process modeling and rules management resources to the operating units.

Notes

[1] While Web Services provide the capability to 'loosely couple' applications and functions together, they still have a long way to go before the much hyped, seamless dynamic trading vision is realized. For example, we don't yet know how to handle the semantic differences between messages that are coming from suppliers – i.e. for the foreseeable future, people will always be involved.

[2] Recent academic articles highlight some of the issues addressed here, including: C.K. Pralahad and M.S. Krishnan "The Dynamic Synchronization of Strategy and Information Technology" Sloan Management Review, Summer 2002, pp24-33; "Does IT Matter? An HBR Debate" (available free online) provides a number of responses to Nicholas Carr's controversial article in the May 2003 issue of Harvard Business Review. John Seely Brown and John Hagel III continue the discussion in "Flexible Information Technology, Better Strategy" The McKinsey Quarterly, 2003 Number 4.

[3] Real people working in business change teams provided these descriptions. A couple came from conversations with modeling tool vendors.

[4] These ideas have been central to our analysis of BPM tools and technologies since the mid-90's. They took shape following work around differing perspectives on business processes and the language used to describe them. Customers involved in this research included Xerox Research Centre in Grenoble, France and TeamWare Group in the UK. Some of the original thoughts can be attributed to Lucy Suchman (then of Xerox Parc), Bob Snowden (then of TeamWare Group) and Peter Cropper (then MD/CIO of Nortel).

[5] Productivity = Value / Resources. The classic approach to pushing up productivity is to lower the numbers of resources required – i.e. downsizing while trying to maintain value. If you can get rid of 20% of the workforce because of the introduction of a technology solution then it is usually possible to get it funded. On the other hand, if your solution was likely to increase the value delivered by an equal amount, it tends to be a lot more difficult to gain funding. Potential increases in value are a lot more difficult to quantify than reducing the resources. One of the reasons that the BPR movement fell from grace was that it focused almost exclusively on reducing the denominator rather than paying enough attention to value innovation. For a recent article on the productivity equation, see "What high tech can learn from slow-growth industries" McKinsey Quarterly, Number 4, 2003.

[6] At a 1996 workshop at the Xerox research facility in Grenoble, Lucy Suchman (then of Xerox Parc) demonstrated that, at some level, virtually all work was knowledge work. Even in the most mundane of roles, individuals build up their knowledge of their working context. The research underpinning this perspective was based around observation of directory enquiry operators and ground staff at an airport.

[7] Think about starting a new job in a new company – you struggle to understand the rules and the culture. As you get used to things and develop your understanding of the job, you start to work out which rules are breakable and which ones are sacrosanct. By the time you have been in that job for 3 years, you know everything there is to know. If you are lucky, you haven't become bored and disillusioned by a management that has overdone command and control. Job experience is turning exceptions into routine – allowing you to do more and more in a routine way.

[8] Looking at the broadly stated aims of the BPM movement, we believe that the current standards initiatives are doomed to failure. They limit themselves to looking at the world in a very 1970's programmer-oriented way, seeing only procedures that are programmable into a linear sequence. While this approach is potentially appropriate for handling the interactions between computers in a physical supply chain (orders and the distribution of physical goods), it struggles to deliver an effective solution to managing the sophisticated problems of modern knowledge workers.

[9] BPMN is proposed as the modeling notation that will 'unite the BPM movement'. While these articles point to some of the shortcomings of BPMN, it is certainly worthwhile as a way of standardizing modeling of procedures. The core of our argument is based around the need to ensure that an implementation is structured appropriately.

[10] One should look on a new modeling approach and seek to understand what benefit it can bring rather than trying to work out why your existing favored approach is better. Different techniques bring different benefits, but perhaps most importantly, they provide a different perspective on the problem, allowing the organization to gain a deeper understanding.

[11] According to the Workflow Management Coalition, there are now 10 groups working on process management standards, 7 of them dealing with process modeling alone. The Business Process Management Initiative (www.bpmi.org) is proposing BPMN as a standard graphical notation for use in modeling procedures.

[12] Source: BPMN Draft 1.0 as at time of writing available on the BPMI web site.

[13] We hesitate to use the term 'Role' here since the graphical grouping merely indicates the responsibility for action. As we shall see, other notations allow for a greater behavioral emphasis around Roles.

[14] With generic modeling tools such as Microsoft Visio now able to produce XML output, we believe it is only a matter of time before a BPMN stencil for Visio is released that will allow the direct generation of BPEL. It is also true that the 'BPM Market' will continue to struggle while there is no effective (widely accepted) standard.

One could argue that the various standards of the Workflow Management Coalition provided a way of describing business processes in a neutral, non-proprietary fashion – allowing robust process definitions to be passed from one engine to another and import process models from other modeling environments. This objective contrasts with that of the BPMI where the aim is to define a standard executable model for BPM systems (subtly different from interoperability). The graphical notation is a separate issue – the WfMC never attempted to define a common graphical representation (as is the intention for BPMN).

In the end, relatively few vendors adopted the interoperability standards of the WfMC. Perhaps more importantly – very few users were that worried about a prospective vendor's adoption of these standards.

[15] The acronym RADs (for Role Activity Diagrams) should not be confused with so-called Rapid Application Development techniques (RAD).

[16] See A. Holt et al. "Coordination System Technology as the Basis for a Programming Environment". Electrical Communication, Vol. 57, No. 4 (1983), pp. 307-314. One of the researchers, Clive Roberts continued to pursue the approach and introduced it to Praxis Systems (subsequently bought by Deloitte & Touche) where it gained further support gaining major exposure under the IOPT program sponsored by the UK Department of Trade and Industry. Clive was involved in that research and continues to explore the uses of Role Activity Theory (see www.co-ordination.com). The same theoretical underpinning is behind the work of Terry Winograd and Fernando Flores of Action Technologies Inc when they proposed the ActionWorkflow Methodology (http://actiontech.com/).

[17] Demo version available for download at www.morphix.com

[18] The best reference on Role Activity Diagrams is in the book "Business Processes: Modeling and Analysis for Re-Engineering and Improvement" by Martyn Ould (available on Amazon - expensive but worth it). His work on Role Activity Diagrams and related methodologies provide a refreshing alternative to procedural approaches to process modeling. We understand that an updated book is under development and should be available by mid-2004.

[19] Role attribution allows a wider audience to understand the model and, in the BPMS, provides routing information. BPMN uses "swimlanes" to help partition and/organize activities. "A Pool (also referred to as a "swimlane") is a graphical container for partitioning a set of activities from other Pools, when modeling business-to-business situations." This definition was drawn from the proposed BPMN specification and, like many definitions in standards documents, the term being defined is used itself in the definition.

[20] The BPMN Collaboration diagram

is outwardly similar to a RAD type structure. In a sense, they appear to show exactly the same sort of things – the only thing different is the modeling notation. However, the emphasis in the BPMN model is on the procedures in each swimlane and messages between them – the sequence of activities and the mechanisms of coordination. In contrast, RADs model how the roles change state and do not explicitly detail the mechanisms of coordination. Indeed, when modeling 'what' you are trying to achieve (rather than the 'how'), it is usually best to avoid specifying the mechanisms of coordination (the data and documents).

[21] For example, the model used to describe the core process of a mortgage lender was originally a 24-page flow diagram. Using a RAD modeling technique, the process model occupied just one single page of A4 (allowing all users to easily understand the entire scope of the process). Indeed, modeling the same process with a RAD quickly identified a flaw with the original that was just not visible with a flow diagramming technique.

[22] Many other approaches rely on populating an entire model repository. Simple RAD diagrams can quickly expose the issues under discussion.

[23] Indeed, in our experience (with a typical end-user workshop with 10 or 15 people in the room), using flow diagrams it is often difficult to get more than one person talking about a process as it is now. Everyone else concurs because the boxes and arrows seem to be about right. When RADs are used, arguments can quickly surface as different perspectives surface on how the process really operates.

[24] The mathematics of Pi Calculus can be applied to RAD diagrams. In the same way that a BPMN diagrams could be used to generate BPEL, so could RADs.

[25] During one risk assessment study on a BPM implementation for one of the world's largest banks, we reviewed the mortgage application process, which was represented by a flow diagram. Drawn as a RAD, which included all of

the internal roles, the model clearly demonstrated that proposed process had a major flaw. At one point, the process switched from dealing with single cases, to dealing with all cases in batch mode. While this looked perfectly natural as a flow diagram, it was not possible to represent this with a RAD. In fact, two related processes were at work.

[26] User organizations include: Glaxo Smith Kline (GSK), Shell, British Telecom, Lucent, Boots, Amerada Hess Gas, Barclays Bank, Deloitte & Touche, plus many, many more less well known companies.

[27] It is also worth remembering that the employees need to maintain their own processes. As Charles Handy put it "The assumption is that individuals will gain a greater degree of satisfaction from a job if they have a degree of freedom in determining how work should be done" - Understanding Organizations by Charles Handy, 1994. See also "In search of a balanced society" by Michael Skapinker, *Financial Times*, September 16, 2003

[28] Further reading on RADs is available on Martyn Ould's book "Business Processes: Modeling and Analysis for Re-Engineering and Improvement". We understand a more up to date version of the book is currently under development. Also, see his web site www.veniceconsulting.co.uk for a page of free resources on RADs, which includes a stencil for Visio and some guidelines on drawing RADs. See also Clive Roberts work in "Coordination System Technology as the basis for a Programming Environment" which can be accessed at: www.co-ordination.com.

[29] Experience shows that it takes no more than 5 minutes to explain to a business user how to read RADs.

[30] Around half of SAP's customer base bought its product between 1998 and 2000. It has been estimated that it will cost each of them around $15 million to upgrade to the latest version, re-applying their customizations – estimated ROI, approaching zero! Adapted from conversations with Ismael Ghalimi, Chief Strategy Officer, Intalio – November 18th 2003.

[31] According to McKinsey, "Business Process Off-Shoring" is estimated to be worth between $32-35 billion in 2002 and well over $100 billion by 2008. A worker who earns $2 per hour in India is likely to cost around $20 per hour in the US. One British bank gets 20% more transactions per hour with 3% better accuracy than similar processing in the UK. Source "Offshoring and Beyond", McKinsey Quarterly, 2003 Special Edition – Global Directions.

[32] A schism exists even in the use of the acronym – most seem to agree on BPM for Business Process Management. Others use the acronym to mean Business Performance Management - talking less about processes and more about activity monitoring and metrics. And the analyst community ensures that everyone continues to stay confused (Gartner have three different 'Magic Quadrants' that talk about BPM and processes).

[33] For the purposes of this paper, we do not intend to explore what is, or is not a BPM product.

[34] Another point to keep in mind – knowledge workers tend to react negatively to procedural controls that are imposed upon them. Of course, if they help develop these procedures and feel in control of how they were used, adapted and perhaps bypassed, the reaction is likely to be quite different.

[35] We have not sought to rule out workflow environments from this discussion of BPMS systems.

[36] It is on this sort of issue that most workflow oriented systems break down, resorting to cumbersome procedural approaches rather than more elegantly within the engine itself. Engines that manage the 'state' of the process are usually more adept at handling this sort of requirement.

[37] When defining the process in such systems, the desired behaviors of each class of participant are defined (this is where the analogy of 'Roles' as having behaviors becomes most powerful). These behaviors are designed to cater for expected (and unexpected) events that may occur during the life of the process. It is the interaction of these participants (Roles and their respective behaviors) that allow the engine to manage sophisticated processes with many disparate interactions and potential events. Supporting new functionality in the process description is merely a matter of adding new behaviors. Moreover, this type of architecture can more easily handle changes to the process description whilst running (for a class of processes or even just a given process instance) - offering the potential for true dynamic process enactment. Having said this, few vendors allow this sort of evolution.

In some products, the process is captured as a set of interacting procedures (one for each role) – both the customer and third party applications have procedures to model their own 'private' process. This allows a more accurate representation of the process and the interactions between these roles provide the context. This type of architecture is much closer to the conceptual underpinning of Pi Calculus proposed by Howard Smith, et al.

[38] Empirical evidence to back up this claim is hard to come by, although anecdotal user stories would tend to back up this figure. Indeed, we believe that, for some of the more advanced and powerful BPMS environments, the cost of ownership of an average process oriented application could be as much as 80% lower than those expected for established enterprise applications.

[39] Integration of a third party rules engine allows the relationships between one task and the next to be described in terms of 'if then else' rules rather than graphically captured as a linear connection on the map. For example, the next group of tasks in a process might be one of 10 different procedural fragments. Determining the precise fragment to invoke is decided via a single business rule (instead of 10 discrete routes in the parent process map). Deploying a combined BPMS and Business Rules environment should results in greater flexibility, more easily understood processes and a dramatically lower TCO.

[40] See Process Product Watch reviews available on the Enix web site at www.enix.co.uk. For advice on product selection contact the author Derek Miers - miers@enix.co.uk

[41] See "Making SOA A Reality" http://www.ebizq.net/hot_topics/soa/features/3142.html

[42] Relates to an advanced work handling environment implemented by CU during the mid-90's. Stephen Gould, the then head of claims, was trying to describe the problems associated with supporting claims handling within traditional procedural workflow environments. CU had adopted the ECHO Case Handling (originally developed by Philips in conjunction with several insurance businesses, sold to Digital Equipment and now defunct. Apart from the Case Handling capabilities of the product, the most interesting thing was the user interface. Based on a 'wave' concept, customer represents could view all work related to the case – depending on their role (as defined in the process) icons on the wave represented work to be done now, on the left of the wave were those tasks already completed, on the right were tasks and activities that could not yet be carried out.

[43] Some might argue that this sort of flexibility introduces security concerns and the opportunity for fraud. However, the firm has control over what those process fragments do and how they are deployed. From the fraud point of view, all systems track what actually happens, enabling detailed audit and review.

[44] During development, best in class BPMS environments allow each application to be introspected to identify discrete elements of functionality. Each required function is then catalogued along with the stub program needed to call that functionality, its arguments and the expected results (collectively this information describes a 'Component'). The BPMS can introspect any system that exposes an API through COM Automation, CORBA, Java, JavaBeans, Enterprise JavaBeans (EJBs), JNDI, XML, Web Services, SQL or XObject. Additional bridges are sometimes available to messaging oriented architectures and mainframe legacy applications. Once catalogued, the component is then available for integration into process definitions. Change or upgrade the third party application, all you need do is rebuild the component and redeploy.

[45] Studies have shown that only around 4% of processes are actually STP situations. As soon as humans are involved we start to run into exceptions – the more human participants the greater the potential for exceptions.

[46] Different vendors take different approaches. Some insist that this done via purchased extensions to their products. Others provide technology that allows developers to dynamically explore any third party application via its API set – negating the need for purchase of upgraded adapters from the vendor. The IT department only needs to ensure that this interface operates correctly when sewn into the system, rather than every individual linkage from every task or activity in the system.