The Split Personality of BPM

Derek Miers
Enix Consulting

This paper explores the link between Business Strategy, BPM Technology, and Process Architectures—specifically, highlighting the need for, and difficulty in achieving both procedural 'control' and business 'agility'.

Part I outlines the need for 'Procedures' and evolving business 'Practices.' Part II explores the need to understand and model 'what' you are doing before detailing 'how' it is done. Part III discusses some of the challenges for Business Process Management Systems (BPMS)—how they can support revolutionary objectives, yet provide an evolutionary journey. It highlights some of the issues in BPMS selection and suggests an approach for designing flexible and adaptable process architectures—providing a balance between procedural control and business agility.

A key objective of this paper is to outline effective approaches and techniques that support the design of process frameworks—especially where a BPMS implementation is the intended target.

PART 1: 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) 'decided' that the standard for business processes will be Business Process Definition Language For Web Services (BPEL4WS: shortened to BPEL).[1] This followed the Business Process Management Initiative's (www.bpmi.org) release of the Business Process Management Language (BPML).[2] Further, under the auspices of the BPMI, we have seen the development of 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.[3] 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 important to pause and consider 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.[4]

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.[5] We also have to bear in mind the subtly differing agendas of those involved. Some see business processes as the way to impose 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 the 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—stable internal processes 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.

Employees 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 of resources needed (the denominator side of the productivity equation).[6] 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 being 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 toward 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 of the spectrum, the term 'Practices' is perhaps a more accurate than the rather vague word 'Process.'

Another way of thinking about it is that Practices often reflect some of the 'vision' of management, whereas all we usually end up with are Procedures (the interpretation by technologists). If you unpick the vision end of process, it becomes possible to describe the capabilities and behaviors required (i.e., the Role behaviors along with their competencies and capabilities), from which, one could 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.

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 and that there is a continuum between the two poles.[7] As outlined in

Figure **1**, 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.[8] 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.

Figure 1—The dialect of business processes is often confusing to those involved. Practices and Procedures offer more accurate descriptions.
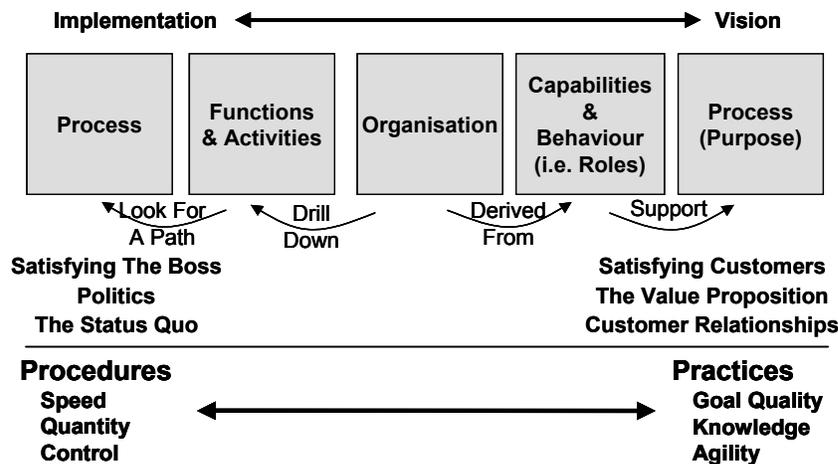
As an example, consider an international bank—everyone would agree that the teller should not get creative with bank drafts. Tellers have very strict procedures (and systems) for handling money directly. On the other hand, try applying a rigorous procedural approach to those developing new trading instruments or those managing the bank's 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 that allowed the dynamic recombination of procedural fragments that, in turn, 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. Job experience is turning procedural exceptions into routine—allowing you to do more and more in a routine way.[9] At the same time, in the same department even, there are rules—policies, goals, and principles that continually evolve—Practices. These Practices might

be thought of as *sets* of procedures that are under the control and development of individual business users. They evolve and usually have a strong collaborative component.

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 Fall 2003 issue of the MIT Sloan Management Review. "The Performance Variability Dilemma" by Eric Matson and Laurence Prusak discusses the need for evolving Business Practices from the perspective of Business Performance Management (another interpretation of the BPM acronym). 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 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 situations.

When constructing process models of work occurring 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 involved. We are, in effect, making assumptions about and 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. 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 from 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 sufficient, 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.

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 was another step toward simplicity. 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.

Problems Modeling Processes

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 better ways of capturing the interactions between the business entities involved—the internal actors, customers, partners, suppliers, and systems. Technology applications 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.

And this is synonymous with the primary underlying aim of the BPMI—empowering business people to model their business processes, the way they want them, and then supporting their execution, management, and evolution over time. Rather than always having to revert to an IT function to write code, **the process models are the system**. To achieve this aim, rich models are needed to allow the BPMS to automatically handle the required interactions with third party applications and generate any portal(s) required for human interaction.[10]

The aim of BPMN is to provide that modeling notation. BPMN does provide a rigorous and effective modeling notation for describing detailed implementation procedures. Crucially, it separates the control flow of the procedure from the data flow, showing how distinct procedures interact through messages (the data flow).[11]

But when we think of process models, we must remember that we are building 'models'—mere representations of the real world. Further, we need to remember that, in the course of modeling, we make some assumptions about how the real world is structured. With graphical models, 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. These semantics and the icons used to represent them 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. And given that the term 'process' is poorly defined at best, everyone has subtly different perceptions of what constitutes business processes and, more importantly, everyone has many different reasons for wanting to construct process models.

Despite the 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. They are often used to support discussions between potential business partners on how the two organizations could work together.[12]

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, the measures usually require a process model as a framework against which analysis is undertaken.

Either way, the primary need is to support communication among people. There are two aspects here: a) supporting the management of processes, and b) helping people to understand the context of a process.

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 that all appropriate business rules are observed. We also need models that define how a series of individual processes are orchestrated 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.

Over the years, 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 grew to include as many as 24 different modeling notations and over 80 different types of icon available on just one of those diagram types.[13]

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 to suit short-term requirements. Even with a rigorous observation of the modeling approach, diagrams are often only understood fully by another expert of that modeling technique.

Some of the modeling tool vendors felt that the lack of a common notation was inhibiting market growth. They formed a distinct workgroup within the auspices of the BPMI to develop such a notation—BPMN.[14]

Given this background, the approaches that were considered when defining BPMN all have a strong, procedural/control flavor.[15] As a result, the proposed standard concerns itself mainly with the definition of procedures. On the other hand, because BPMN separates control flow from the data flow of a procedure, it becomes possible to model a process as a set of inter-communicating procedures (see
Figure **2**). This is a very powerful way of understanding a business process (as against an internal, computerized procedure).[16] [17]

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

It will be interesting to see how many modeling tool vendors eventually support the notation. Privately, some of these vendors have pointed out that there is little in terms of competitive advantage for them in supporting this standard.[18] The future for these vendors is to concentrate on model and repository management. It will take pressure from user organizations to ensure that vendors quickly incorporate BPMN support into their products.
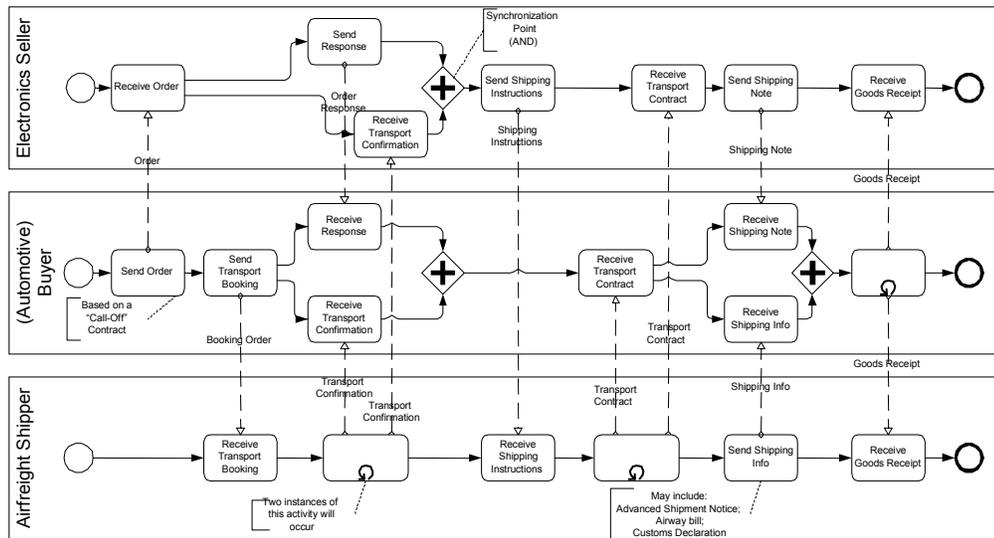

Figure 2—Here the modeler uses a Collaboration Process model to show how the entities interact in an e-commerce scenario (source BPMN Working Group)

One of the key problems with this sort of procedural approach is that it does not help to identify bad processes. Sure, a few activities can be combined and responsibilities moved around, but with a flow diagram it is possible to prove almost 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 five-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.

When thinking about organizational design, we need the capability to model practices and describe the sorts of competencies and capabilities required (even the behaviors of the roles involved). We also need techniques that allow a high level vision to be modeled and explored—examining the 'what,' without resorting to defining the 'how'.

During the 80s and 90s, several alternative approaches to process representation were developed that sought to provide an effective solution in these areas. The best of these was Role Activity Diagrams (RADs)[19] [20] [21] 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

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). At the highest level, most commercial organizations would recognize customers as a strategic influence, along with other external entities such as suppliers, technology, regulation, etc (as in *Figure 3*).
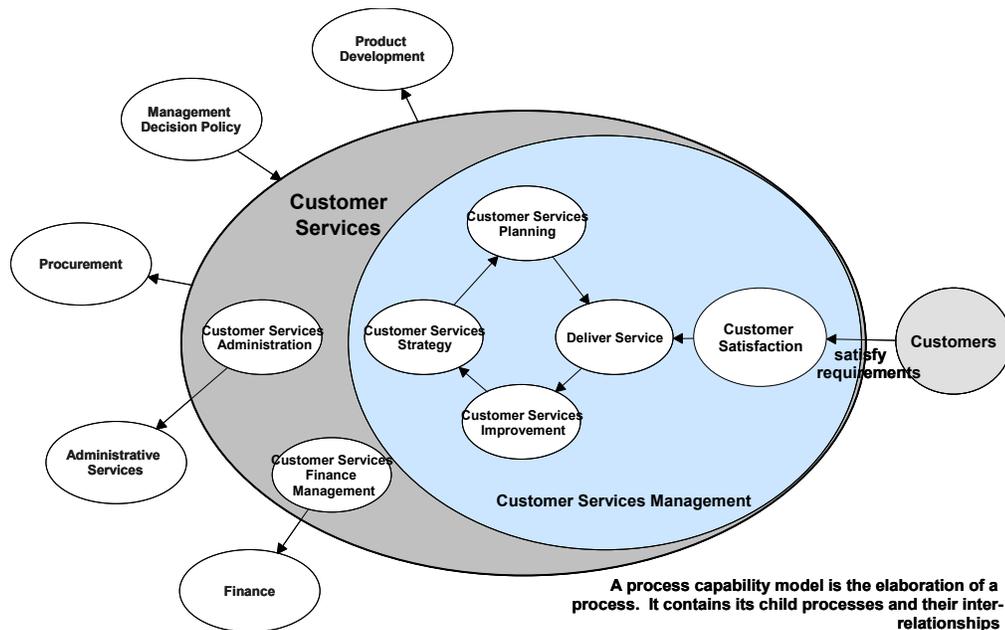

Figure 3—A Business Capability Model (source MooD sample repository) [22]

Within this strategic context, it is then possible to represent a set of constituent capabilities that are required to make up the organization. In turn, each of these capabilities can then be opened up and modeled. Capabilities describe organizational proficiencies or competencies—things the firm needs to be able to do on an ongoing basis. They might be defined in terms of the relationships they have with each other and the outside world (behaviors), their objectives, KPIs, and issues or required improvements. From a modeling point of view, a defined business capability is then reusable 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 having to initially specify sequences or functions. Such an assessment of a firm's capability facilitates a richer discussion on business needs, around which stakeholders can more easily reach consensus without getting bogged down in the details of activity sequences or functional decomposition.

Conclusion to Part II

Before getting involved in implementation, it is essential to understand the organization as it fits into the value chains within which it operates. An appropriate structure can be derived by modeling the organization as a set of collaborating capabilities. Looking inside a capability, one can explore its constituent parts and their interactions.

Technology (and the procedures that it can handle) should not be the starting point. The primary requirement is to get humans aligned around a common objective. This implies developing a deep appreciation of the process—understanding it both in terms of the necessary Procedures and the flexibility required (its Practices). Building such insights is best achieved by contrasting alternative perspectives rather than insisting on a single modeling style or notation.

The capacity 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. But 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 needs.

The truth is that there is no single process-modeling notation that is applicable to all needs. When looking for appropriate techniques, a good start point is to define 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—simple flow diagrams plainly do not deliver. Paraphrasing Aaron Levenstein (talking about statistics), "Process Models (Statistics) are like a bikini. What they reveal is suggestive, but what they conceal is vital." And flow diagrams can conceal much of the richness required to understand the fundamentals of "how things get done around here".

PART III: REVOLUTIONARY OBJECTIVES, EVOLUTIONARY JOURNEY

Introduction

Enterprises have long sought a technological silver bullet—an approach to supporting work (using IT) that delivers rapid time-to-market, and 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 1990s, 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.[23]

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,[24] 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 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 the emerging category of modern BPMS engines. These 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, these products 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.[25] 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.[26]

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 process, 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 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 Ford's 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?
- 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')? Culture is like concrete—it sets and is difficult to shift.

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 this will evolve over time. Placing the proper emphasis on these issues is likely to be different for every firm, depending on their culture, management predisposition toward risk, the degree of control required, etc.[27] 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 largely 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—whatever you put in place will require modification at many different levels—probably very quickly. So you need to be able to manage how you unlock change in certain areas of the business and for a given set of roles.

BPMS products share one common feature—the ability to route work throughout the firm based on activity sequences (procedural rules), leveraging some sort of organizational structuring mechanism—connecting people, applications, and legacy systems.[28] 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), grows 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, experience has shown us that it is virtually impossible to take into account every potential eventuality and circumstance. Over time, as more 'exceptions' were built in, process maps became horrendously complex. And as a result, the TCO grew ever higher. These complex multi-step processes were often simplified graphically through the use of sub maps, often called sub-procedures, which were called from the main process map.

With most traditional workflow engines, these sub-procedures were usually embedded in the parent process (rather than maintained as separate components called at runtime). Parallel paths of activity were often involved in both the parent and the sub-procedure, which may have needed to synchronize at certain points.[29] Most products also tended to embed any code needed to integrate with third party applications in the tasks and activities themselves. If you changed the third party application, each piece of code had to be revisited and re-developed. The potential for spaghetti code was 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. 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). [30]

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.[31]

BPMS environments have better ad hoc functionality to get around those show 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. Indeed, some of these modern BPMS engines allow the business users to modify their process models and instantly see the changes in the operating environment—i.e., the models that end-users produce drive the system in real time.

Products also now tend to allow the integration of Business Rules Engines, further simplifying the potential paths through a process. Integration of a third party rules engine allows the relationships between one task and the next to be described in terms of Boolean 'if then else' rules, rather than graphically captured as ever more complex linear connections 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 Engine should result in greater flexibility, more easily understood processes, and a dramatically lower TCO.

Deciding on a BPMS environment is not a trivial exercise.[32] [33] 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. If broken down into simpler reusable components (process fragments), processes become easier to understand and evolve. If their interfaces are tightly defined, individual process sub-services to take on a life of their own. Change that enterprise application at the back end for another 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: Some talk about Business Process Orchestration; others talk about incremental development of Service Oriented Architectures,[34] 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. While this is not especially new (think OSI, CORBA, COM, etc.), Web Services provide a standards based solution around which such architectures are constructed.

- Case Handling concepts are based around the familiar metaphor of a desk file, with all its related documents and customer interactions. Professions such as architecture, legal services, and accounting operate on cases. Similarly, one could regard virtually all government and utility processes as Case Handling. Insurance claims processing is also a clear example of Case Handling. In the words of the Head of Claims for Commercial Union— "every claim is an exception".[35] While 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.[36]

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 is that 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, providing compensating activities when a failure has occurred.

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. During development, these 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 best of these products 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 available to messaging oriented architectures and mainframe legacy applications.

Once these Components have been catalogued, they are then available to end-users to call from within their process designs. Change or upgrade the third party application, and all you need do is rebuild the Component and redeploy. There is no other impact on the processes.

More traditional integration associated with many workflow engines requires that each activity or task that calls a third party application must be revisited and rebuilt (leading to higher TCO). A slightly more modern approach is to employ static 'adapters' that act as the bridge to the third party application. The only problem with this approach is that as back end transactional systems are changed, a new adapter must be purchased and implemented.

Conclusion

This paper has sought to provide more than a call to arms (around BPM). Instead, it pointed to the conflicting language used within the BPM community that has only fuelled the confusion within the marketplace.

We began by highlighting the difficulty in balancing efficiency and agility. Achieving a successful BPMS implementation implies making hard decisions around culture and structure—before leaping into the morass of activity sequences. Issues will inevitably arise around the tension between **procedural control** and **knowledge-based practices**. Where the boundary lies is different for every industry, company, and organizational grouping.

In Part II we sought to outline a practical methodology for approaching BPMS implementation. This involves understanding **what** you are doing before getting into the **how**. Data and documents are merely the implementation details. In the online version of this paper we explore graphical approaches that provide a fundamentally different set of perspectives from that offered by flow diagrams.[37] True insights only come from contrasted perspectives rather than re-applying the same mind-set.

In Part III, we attempted to cure the reader of a tendency to rush in and 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 not involved.[38] 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.[39] 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, support the business unit, and ensure that the overall process framework works appropriately. 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 center of excellence that provides additional process modeling and rules management resources to the operating units.

End-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] The BPML specification predated BPEL by about 18 months. It is accepted that BPEL is largely based on BPML but did not include much of its sophistication (e.g. around longer running processes and transactions). Some would argue that this reflected the fact that the key players could not implement this sort of sophistication—limiting the scope to internally focused processes and the integration of Web Services. In a sense, BPEL slowed innovation in the market while, at the same time, the benefits were in allowing the technologies of the biggest vendors to work together. In the interests of standardization, BPMI has since adopted BPEL as the execution language (a

retrograde step). It has been suggested that if the Workflow Management Coalition (WfMC) would do the same (adopt BPEL), then the path would be open toward a normalization of the various standards initiatives in the domain. Both the WfMC and BPMI could then work together on influencing the future direction of BPEL.

[3] 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; John Seely Brown and John Hagel III point out that gaining value from IT requires innovations in business practices in "Does IT Matter? An HBR Debate" (online - http://harvardbusinessonline.hbsp.harvard.edu/b01/en/files/misc/Web_Letters.pdf) which also provides other responses to Nicholas Carr's controversial article in the May 2003 issue of Harvard Business Review. Seely Brown and Hagel continue the discussion in "Flexible Information Technology, Better Strategy" The McKinsey Quarterly, 2003 Number 4.

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

[5] These ideas have been central to our analysis of BPM tools and technologies since the mid-90s. 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).

[6] 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 percent 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.

[7] In reality, Practices are often about the 'management' of multiple, evolving procedural interpretations of how work is done. A Business Process Management System should be able to support this sort of subtlety without reverting to the IT function.

[8] 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.

[9] 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 three 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.

[10] To a varying extent, this is possible today with some of the more advanced BPMS products.

[11] However, 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 inappropriate for exploring the ways in which a business relationship operates or the value chain functions. BPMN was never designed to support process discovery (the act of understanding the process).

[12] In a sense, the BPM industry is trying to move people beyond this. If one can make those models deployable then a great many advantages accrue.

[13] One should look on a new modeling approach and seek to understand what benefit it can bring, rather than trying to work out why an 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.

[14] Readers should be able to download the BPMN specification at http://www.bpmi.org.

[15] BPMN primarily drew upon the 'processes as activity sequence' perspective. It considered 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. It also appears to have strayed a little from its aim of providing a visual language for process execution. Since several of the vendors involved in its development have come from the software engineering end of things, it has become a little skewed towards the needs of software development.

[16] One of the diagram types supported by BPMN is a 'Collaboration Diagram' (see **Error! Reference source not found.**). It is very similar to Role Activity Diagrams (RADs) developed during the 90s. RADs modeled how a "Role changes state" rather than the sequence of activities. RADs provide a more compact notation than the BPMN Collaboration Diagram, allowing the compression of a lot more detail onto a single model. However, both techniques capture essentially the same content.

[17] This paper originally included a section that focused on the modeling of both Procedures and Practices—exploring both the differences, and similarities, between BPMN and RADs. It also described a technique for exploring a business vision through Business Capability models. However, due to limitations on space in this book it has been dropped. You can find an extended version of the paper on the Enix web site which includes this section—www.enix.co.uk

[18] 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, which in turn, could then 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 was to define a standard executable model for BPMS environments (a completely different objective 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).

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

[20] 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/).

[21] Work on Role Activity Diagrams was one of the conceptual inputs assessed during the development of the BPML execution language.

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

[23] 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 $5 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.

[24] 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 percent more transactions per hour with three percent better accuracy than similar processing in the UK. Source "Offshoring and Beyond", McKinsey Quarterly, 2003 Special Edition—Global Directions.

[25] 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).

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

[27] 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.

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

[29] 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.

[30] 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 than many workflow environments currently offer (as discussed by Howard Smith and Peter Fingar in their recent paper "Workflow is just a Pi process").

[31] Empirical evidence to back up this claim is hard to come by, although anecdotal user stories would tend to support it. Further, with 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 percent lower than those expected for established enterprise applications. During the coming year we intend undertaking a study on TCO, contrasting leading BPMS environments with each other and more traditional approaches such as workflow and major enterprise applications.

[32] 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

[33] Those considering product selection might also be interested in the BPM Technology Showcase events—24 vendors, six tracks of workshops. First event in London April 1st and 2nd, with follow on events planned for Washington and San Francisco in the summer. See http://www.enix.co.uk/Technology_Showcase.htm for an up to date list of participating vendors and event details.

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

[35] Relates to an advanced work handling environment implemented by CU during the mid-90s. 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.

[36] 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.

[37] Due to space limitations in this book we were unable to include the extended section on modeling. See the complete paper, which includes broader sections on process modeling with BPMN, RADs and Business Capability diagrams. www.enix.co.uk

[38] Studies have shown that only around four perecent 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.

[39] 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.