Issues and Best Practices for the BPM and SOA Journey

Introduction

Over the last few years, both Business Process Management (BPM) and Service Oriented Architectures (SOA) have been touted as the core approaches that will help organizations improve business performance. Both approaches provide an evolutionary methodology for transforming the architecture of the enterprise by aggregating business change and IT development under a holistic umbrella. While they have subtly different objectives, they share a lot of common ground and complement each other well.

- BPM is a continuous performance improvement methodology. It is primarily a management discipline based around the iterative improvement of business processes. Underpinning that objective is a technology infrastructure that drives work through the firm, enabling a regime of monitoring, optimization and traceability.
- SOA is an approach to systems development that delivers applications through the composition and orchestration of discrete, independent components, or "services." The design philosophy driving SOA is process-oriented, business-centered, architecture-driven and set-up to support change from the outset.

Some vendors are trying to position BPM as a subset of SOA, regarding it as the way in which systems are orchestrated together (ignoring the human element of business processes). This is the wrong way to position a business driven approach. BPM is about driving improvements in business performance, yet benefits from SOA thinking. On the other hand, SOA aspires to underpin business agility, yet it is fundamentally an approach to IT integration.

The aim of this paper is to outline a holistic blueprint that will help firms succeed on the BPM-SOA journey. It includes a number of distinct techniques and approaches that can help overcome some of the challenges associated with rolling out BPM-SOA across the enterprise. Generally, the paper explores SOA concepts from a BPM point of view.

Why the Interest in BPM-SOA?

In a rapidly evolving business climate, the pressures on firms to adapt are immense. Driving the need for change are new business arrangements between firms and their clients, partners and suppliers, as well as new compliance regulations and new technology. The competitive dynamics are such that firms must be able to tease apart their business operations and outsource (or off-shore) aspects of the value chain that were once managed internally. Moreover, new innovations are appearing almost daily, driven in part by ubiquitous access to the Internet, forcing the pace of change in both the physical supply chain and in services of every kind.

Customers have now become active participants in the business processes of the firm. They expect a multi-channel experience, coordinated across all touch points. But processes are usually fractured across functions and channels (sales, support, accounts, call center, mail room, partners, Internet, bricks and mortar, etc), creating all sorts of opportunities for things to get lost or mis-communicated when resolving the customer's case. More specifically, different processes for each channel inevitably lead to uncoordinated and inconsistent behavior that will, in the end, lose customers.

Yet the systems of today are often highly structured, with fixed integration points that make them brittle and difficult to change. The technology infrastructure in major organizations is more like a poorly constructed quilt, full of patches from past silo-based development efforts; shot through with holes and delicate interconnections. At the same time, the business processes that these systems support are internally focused, control-oriented and embedded into rigid legacy systems and enterprise applications. BPM and SOA applied together provide the ideal infrastructural approach to resolve these thorny issues.

According to a recent McKinsey study, successful SOA initiatives within large companies can provide significant benefits, including cutting the cost of integration by 60-70% in the medium term and dramatically increasing the firm's adaptability. For example, a retail bank cut its time to market by 30% and its ongoing development and maintenance costs by 10-20%.[1] SOA alone though is not enough.

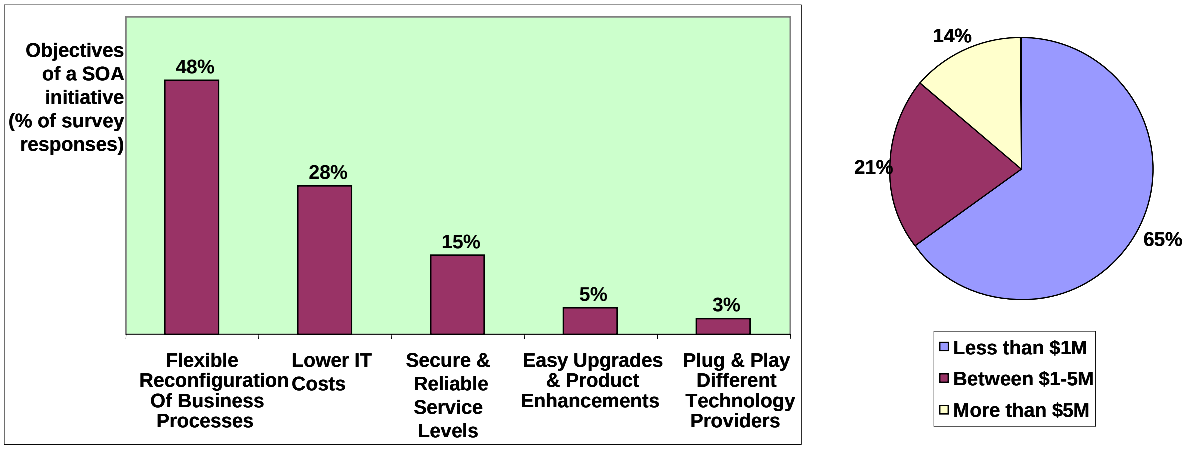

Figure 1 - Summary results from a recent AMR Survey on the benefits of SOA (average spending on SOA was $667,000)[2]

Combining the two approaches is essential to ensuring the effectiveness of this approach. For example, despite heavy emphasis on business processes in the AMR survey (Figure 1), about a third of those already undertaking SOA initiatives found that they could not reconfigure business processes as required. This was largely because they had not deployed a BPM environment to provide the orchestration layer.

The specific benefits of BPM initiatives are well documented. In a recent review of BPM case studies,[3] enhancing business performance was the most common result highlighted (lowering costs and faster cycle time). Enhancing customer service by freeing up staff time from administrative duties was also a common theme. In parallel, some firms had integrated their disparate customer channels to provide a more consistent customer experience. Others had distilled the processes of the firm into an independent layer, above the level of legacy system integration to provide a platform for greater agility. Others had used the approach to outsource parts of their business, or redistribute the work to the customer (self-service portal).

The numbers that go with these case studies are impressive. For example, a division of Allianz, the global insurance group, achieved an 80 percent increase in efficiency, with the average time to process a claim reduced from weeks to days. The Driver and Vehicle Licensing Authority in the UK increased case management productivity by more than 50 percent, allowing the DVLA to redeploy staff to other areas. The Dutch telecoms operator KPN reduced the average time for order processing by 90 percent and cost savings of between €150,000 and €200,000 per month. Alongside that they estimate that the solution enabled a 95 percent reduction in errors, 80 percent straight through order automation, and a 50 percent productivity improvement.

From a business perspective BPM-SOA technologies provide a number of related benefits:

- They reduce the time to market, and enhance the ability of the firm to respond quickly and efficiently to business change. If approached properly, they provide the mechanism to both exploit

[1] Extracted from "Designing IT for Business" (McKinsey Quarterly #3, 2003)

[2] Reported in "The SOA Future" http://www.line56.com/articles/default.asp?articleID=6881&TopicID=4

[3] Survey of over 30 BPM case studies carried out by the author.

new market opportunities, and then to leverage this speed of response for competitive advantage – much like a top sports star makes it look easy as they beat the competition. Indeed, they can enable a major enterprise to copy a market innovation and then quickly leverage its resources and reach to counter the competitive threat from more nimble players.

- Over time, they offer the ability to reduce cost massively. Through automation of the process itself, employees are freed up from the drudgery of non-value adding tasks, allowing them to concentrate on higher value, customer facing activity. Moreover, the cost of IT development is significantly reduced as applications move from the design-compile-run paradigm of traditional software development, to one based around assemble-configure-monitor (common in BPM–SOA deployments). Model Driven Architectures play a role here also.
- They encourage predictable and consistent performance through the re-use of standardized, process-based business capabilities (business services). They also deliver better visibility and robust traceability on the status of work within the firm.
- The need to coordinate business processes and improve productivity does not end at the firewall. BPM (and SOA) provides the opportunity to choreograph the participants of the processes across the value chain, whether they are customers, partners or suppliers.
- Insulation – changes made to processes need not affect the underlying services and the line-of-business applications with which they interact (and vice versa). This aspect significantly increases process agility by isolating the impact of change.

But these benefits do not come cheaply. According to McKinsey, a SOA initiative within a large company can involve an initial investment of $10m and take over two years.[4]

Issues on the BPM-SOA Journey

Converting to a BPM-SOA infrastructure is not a silver bullet. Before diving headlong into a multi-million dollar BPM-SOA program, the organization needs to recognize and deal with a range of issues related to the interplay between the two approaches. To derive the maximum benefits, and leverage the investment involved, one must understand that both BPM and SOA involve a journey, not a destination. This paper highlights a number of areas that are potential roadblocks along this journey and provide suggestions for best practices to avoid or get around these roadblocks.

At their core, both approaches are iterative, requiring a new state of mind in both the business and the IT organization. Moreover, rather than having a comfortable start and end-point), the BPM-SOA journey is never complete. To succeed in BPM (and to leverage the benefits of SOA more widely), requires an effective partnership between the business and IT as they together explore how the business is organized, right up to, and including, questioning the established political power bases of the firm. This involves developing an effective governance model with both the business and IT collaborating to ensure a proper foundation. Moreover, achieving the maximum benefit requires careful planning (developing effective process architectures) and a recognition of the fundamentally different methods employed. This paper then goes on to explore the issues associated with modeling processes as services, suggesting a couple of alternatives for dealing with this problem. It concludes with a short discussion on infrastructure and a summary of best practices.

A New State Of Mind

Both BPM and SOA can be thought of as a state of mind – a way of thinking about how the business and IT assets work together; how the business and governance model should be designed and a way of delivering the technology and applications to support that design. However, they are two *different* states of mind. BPM is fundamentally a top-down approach whereas SOA is usually bottom-up. The challenge is in marrying these two perspectives, identifying the commonalities, and resolving the differences.

Common to both approaches is the concept of loose coupling. With BPM tools, this is reflected through loose coupling between processes. In SOA, loose coupling enables both internal and external

[4] Extracted from “Designing IT for Business” (McKinsey Quarterly #3, 2003)

applications to be spread across a distributed technology infrastructure. They both encourage reuse as a mechanism to increase business agility and flexibility. If designed appropriately, business systems can adapt in line with changing needs of the prevailing competitive environment, rather than being constrained by brittle, tightly integrated applications that restrict agility. The response to a given business need is composed from a set of loosely coupled components, rather than tightly integrated at design time. Moreover, if approached correctly the cost of development and maintenance is considerably reduced.

While both approaches promise an alluring future and can be pursued independently, deriving the maximum value of either technique involves leveraging methods of the other. Services often require orchestration in the wider context of a business process, and business processes can incorporate a number of cooperating business services.

To undertake a BPM initiative without employing SOA based principles is questionable. It means either that the team ignored best practice, or there was no significant integration, (which, in turn, probably highlights a flawed approach to the management and control of business processes). Of course, it is possible to employ SOA principles without using BPM, since there are plenty of software projects that require integration yet have no need of process management.

Further, the business itself usually thinks in terms of top down projects that effect organizational change. Yet a services-oriented paradigm is at odds with traditional functional decomposition approaches used in many IT development environments. Instead, services are “composed” to deliver the desired functionality – an approach that is entirely consistent with the best practice of process architecture. Many SOA-oriented initiatives begin bottom-up, encapsulating existing application functionality in wrappers, using Web Services and middleware technology to link these components together. For many in the IT function, the ultimate goal of SOA is to turn that plate of brittle spaghetti into a series of modular building blocks, from which it is then possible to compose new applications and services.

On the other hand, the business needs an effective top-down design methodology that aligns business level capabilities and yet, at the same time, leverages discrete and independent service components as mechanisms for implementing those capabilities. Further, it is worth noting that the benefits of BPM-SOA extend well beyond the IT organization, which typically equates to between 2% and 10% of the cost base of a business. Designing “business-oriented services” allows optimization of business operations across the firm (the other 90% to 98% of the cost base).[5] At the enterprise level, BPM-SOA then becomes an aggregated architecture that encompasses all IT systems, crossed with the business capabilities that the organization wants to deliver. In turn, this emphasis on business oriented services may push the boundaries into the extended enterprise. And when one considers the opportunities for value innovation, the biggest impact is when it is applied across the entire value chain.

For example, Optus, one of Australia's largest telecommunications services providers, used a BPM solution to support its regulatory obligations around local number portability. The system supports the various business units inside Optus, along with a number of independent external contractors, and ties to other carriers (i.e. competitors). As a result, Optus has already reduced time to “port” local numbers by 30%. But once fully optimized, the firm anticipates a 60% improvement in process performance. But this is not just a benefit to customers; an increase in process efficiency of just 10% can translate into millions of dollars on the bottom line. In another example, Dell is able to squeeze cost out of its delivery logistics by actively integrating the services of its third party suppliers into its own processes. If one supplier is not performing against agreed service level agreements, Dell is able to switch to another immediately. As a result, it now has tremendous bargaining power, further reducing cost for the consumer.

Ensuring a Proper Foundation

The fastest way to derail a well-intentioned BPM-SOA initiative is to forget about the business. Before leaping into the depths of implementation, it is essential to ensure that the initiative is going to have the backing and support of the business itself. Given the potential impact on the organization itself, executive sponsorship is absolutely critical. The business needs to own both the long-term change program and the solutions that come out the other end. Otherwise, through a lack of buy in, people will not fully engage in

[5] Forrester Research Best Practices Series– “A Taxonomy Of Service Types For SOA” by Randy Heffner with Jost Hoppermann, Alex Cullen, and Carl Zetie, October, 2005

delivering success, and they will not give their full support to the project. However, there is no one size fits all solution to establishing appropriate governance policies. Firms have to figure it out for themselves.

Depending on the history and organizational maturity, the best start point is usually to create a high-level, cross-functional “Steering Group”. This group will oversee the creation of a BPM-SOA framework and the individual projects undertaken within it. The idea is that the Steering Group provides a respected, business-centric body that affords an objective view, setting priorities appropriately. It also ensures business ownership of the overall program, creating a clear organizational context for change. The Steering Group is primarily concerned with ensuring that the change program underpins and aligns with the Key Business Objectives (KBOs) of the firm.

Other organizations may opt for a central “Process Management Office” or “BPM Center of Excellence” (BPM CoE), reporting up through some sort of Process Czar either to the CIO, or more appropriately, to the Executive Board or CEO of the firm. While the Steering Group option is a good start point for those just getting into BPM-SOA (focusing on ensuring success of the first project), the BPM CoE route is a more appropriate organizational form when rolling out BPM-SOA across the enterprise (having been successful in the first project or two).

In this enterprise-wide context, the BPM CoE is normally responsible for developing common principles, language, frameworks and methodologies for process development. It is also responsible for developing the process architecture itself, along with supporting individual change projects and ensuring that the architecture reflects the needs and operations of the various business units (which may imply organizational change).

Establishing an Appropriate Process Architecture

With the organizational context for the BPM-SOA initiative established, it is necessary to take a step back and assess the overall architecture. When it comes to explaining the role of process architecture an analogy helps. Let’s say you want to build a hotel. Naturally you would employ an architect. The city imposes a set of building regulations on the development to which all hotels must conform. These are the “rules” that are developed taking into account the prevailing conditions of the area and reflect a collective wisdom on good building practice. For example, in Finland buildings must have triple insulation on the windows and doors (it gets cold). In Wellington, New Zealand or Osaka Japan, the foundations must be particularly strong and robust (there are lots of earthquakes there).

When building your hotel you employ an architect to interpret your own needs and ensure that the proposed design is compliant with the local building regulations. The architect is responsible for overseeing the construction and ensuring the building is fit for purpose. In the same way, a business manager describes his or her needs for process and this is interpreted by the process architect in the context of the prevailing architectural rules of the firm (established at a global level). The interfaces of the process might equate to the interfaces of your building when it hooks up to the services of the city.

Taking this analogy a little further, what about a pre-fabricated house? There is a lot of architecture that goes into ensuring all the component parts fit together, such that its cost of manufacture is low and, at the same time, allowing rapid construction on site. But what if you are building a shed in the back yard? You probably don’t want an architect to help you. Instead you will no doubt purchase flat-pack from the DIY store. But again, a certain degree of design has gone into the flat-pack.

The point is that different types of processes require different types of architectural support. As long as the business need fits certain usage patterns, then it is possible to design and construct bespoke applications, or even buy them off the shelf, and put it together in-house. On the other hand, as process outsourcing grows, firms are realizing the benefits of not needing to own either the process or the infrastructure to support it. In a sense, these are very much “business services”. We are used to thinking of them externally, but there is also a great deal of benefit from designing the internal organization in a similar fashion.

However, significant care and expertise is still required to design appropriate process-service architectures. However this is achieved, effective modeling is needed to capture and communicate the nature of the process architecture framework. But here is where the problems often begin. Too often, the IT folks tasked with this work revert to form, looking at traditional modeling approaches to represent the service components of a process. They struggle to apply familiar functional decomposition approaches to the new paradigm. The point is that processes are not *decomposed*; they are *composed* from constituent parts as needed, rather than being contained, one within another.

Fundamentally Different From Traditional IT Approaches

Combining BPM and SOA forces two worlds to collide, each of with has a very different perspective. In the BPM world, the key role is that of the business analyst. To a Business Analyst, everything is a process. The goal is usually driving efficiency and performance improvement. But in IT driven SOA initiatives, it is the Software Developer who has the driving role. Once they get the idea of SOA, everything appears as a potential service. A successful marriage follows not from an "either-or" decision. Instead, it revolves around developing a new approach that leverages the strengths of each technique.

As was noted above, successful BPM programs are iterative in nature (as are SOA programs). They differ from SOA programs in that they don't assume that it is possible to tie down all necessary functionality and deliver it in a given system; rather, the best practice is to focus on the 20-40% of functionality that delivers the majority of the value. Thereafter, the process models that drive the application are refined and embellished over time to deliver the desired functionality. This approach is fundamentally different from the traditional waterfall-style systems development methodology that is still common in many IT functions. Indeed, business users have become so accustomed to waterfall-style approaches that they need re-educating as to what is possible (i.e. to help them understand it is important not to force all desired functionality into the first release).

Another aspect of successful BPM projects is that they initially focus very tightly to ensure project success. This allows delivery in a short time frame, and thereby encouraging the business to build confidence in the approach (and the team to build experience). More complex and contentious projects are deferred until later on the roadmap. Having experienced the success of the first project, the business is more willing to get behind the change program, taking control of the project and driving it forward.

Usually, the emphasis in a BPM project is on automating the process wherever possible, facilitating the interaction of the users, guaranteeing traceability of cases and enabling management visibility into metrics. And here there is a fundamental difference from SOA. With SOA oriented initiatives, the unwritten assumption (in the IT department) is based on orchestrating system to system processes, where as BPM projects focus on the needs of humans involved – their interactions with each other and the systems that they use.

However, when one explores SOA case studies, one finds that they often start with a very different proposition. While they might set out to support the needs of the high-level business process, they focus on wrapping data access and transaction mechanisms at a low level. As one moves up the services stack, collections of services create re-usable application-centric components that can be orchestrated into more recognizable end-user oriented systems. Some have gone so far as to leverage the capabilities delivered by third party organizations (e.g. the Dell example above), orchestrating not only internal services, but those of external service providers.

The library of services that is created can be invaluable in a BPM-SOA project. However, getting the level of granularity right for services is critical. IT developers will usually begin with fine grained services. On the other hand, the BPM project team will identify the need for business level components (coarse-grained services). Services need to be coarse enough to be managed appropriately (i.e. have meta-data associated with them that allows them to be located and re-used), yet value creating. In the end, the degrees of flexibility of the resulting applications are dependent on having the right level of granularity. Too small, and it becomes hard to build anything meaningful; too large, and it becomes unwieldy (i.e. it is already working as an application).

The meta-data must hide the technical details and yet expose a description of the functionality allowing it to be recognized by non-technical users. This meta-data will include its inputs and outputs, as well as the quality of service it will deliver (SLAs). It may also need to include information on security and infrastructure requirements.

BPM-SOA Modeling - A Problem of Dialect

Part of the problem one runs into in the BPM-SOA discussion is one of terminology, dialect and applying the traditional mindset that created the mess in the first place. “Process”, “Component”, “Service” are widely used terms that are usually interpreted subtly differently by different groups (i.e. meaning subtly different things). Indeed, we need to think about how processes might be composed of a set of discrete services, and vice versa – how a business service might be implemented as a set of processes.

Process ≈ Service ≈ Capability

But the term process means different things to different people. The interpretation by a senior manager is usually quite at odds from that used in the trenches of software development. When thinking about “process” at a high level, one is normally thinking of “process as purpose”. For example, the Finance process is there to ensure that the company does not go bust. Similarly the Customer Relationship Management process is about ensuring the firm has happy customers. Unpicking that high-level notion of process involves identifying the “capabilities” that are necessary to satisfy the purpose. From those high level capabilities, one can then design an appropriate organizational structure to deliver that purpose. But the problem comes when one says the word “process” to an existing functional organization. Rather than thinking in terms of the capabilities necessary to deliver on the purpose, people tend to look at the existing functional hierarchy, drilling down through the various departments that make up that fiefdom, until they identify lists of activities. The various paths through the activities (Standard Operating Procedures) are then referred to as a “processes”. But this is not the same notion of process as we started out with (process as purpose).

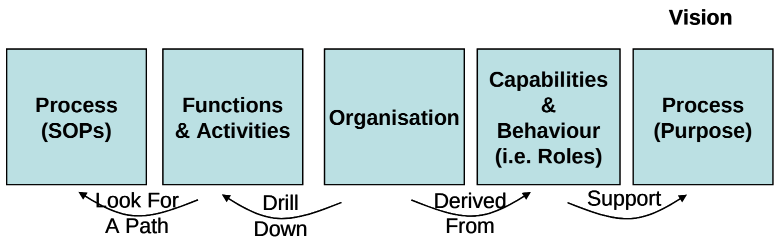

Figure 2 – “Process as Purpose” is not the same as “Standard Operating Procedures”

Indeed, the term process is better interpreted as a spectrum – “procedures” at one end, and “practices” at the other. Procedures are usually imposed and are about standardization and control, whereas practices tend to evolve and are used as a guide. When one looks closely at a process (as purpose), one finds a mixture of both procedure and practice. Over time, as practices are better “understood”, they are often “proceduralized”.

To understand the rich reality of a firm’s processes we need better ways of thinking about both procedures and practices. The problem is that when a flow diagram is the only representation of processes, they always seem to look correct. It is very difficult to identify errors or parts of a process that could benefit from redevelopment. Flow diagrams also take a very procedural view of process, focusing on the order of activities while ignoring, or at least struggling to represent, flexible business practices. Moreover, with flow diagrams as the only modeling concept, the temptation is to model everything down to the minutest level of detail (precisely the point where firms become stuck in analysis paralysis).

If chunks of processes can be represented as discrete business capabilities, then services can be used to compose high level business processes (process as purpose). One could think of business capabilities as the Lego blocks that can be used to create a high level business process.

Modeling Processes as Services

The challenge is in agreeing the granularity of the different levels and naming conventions adopted for each level. There are many potential solutions to this problem and here we present a couple of examples. The simplest approach may be to have a clear separation between an end-to-end process layer and a services layer. But this design tactic can quickly become difficult to manage and does not lend itself to large scale deployment.

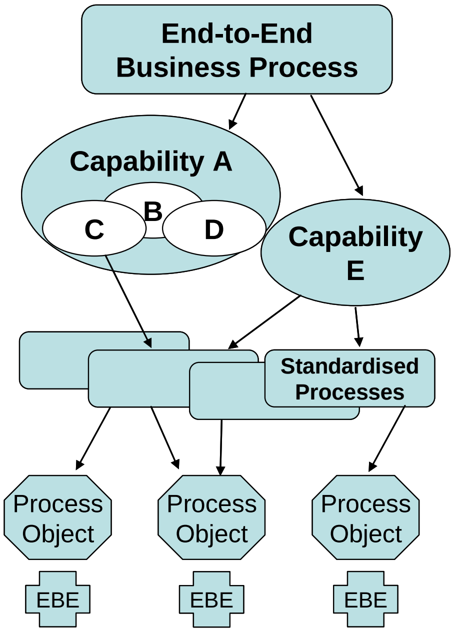

Figure 3 – Capabilities re-use standard Process Objects (services) through Standardized Processes

A large consumer electronics firm approached it in a more sophisticated fashion (see Figure 3). An "End-to-End Business Process" was implemented as collection of "Business Capabilities". A Business Capability could contain other Business Capabilities and could also be implemented by a number of Normalized Processes. A Normalized Process re-uses component Process Objects that change the state of an Essential Business Entity. EBEs are the essential elements of the domain.

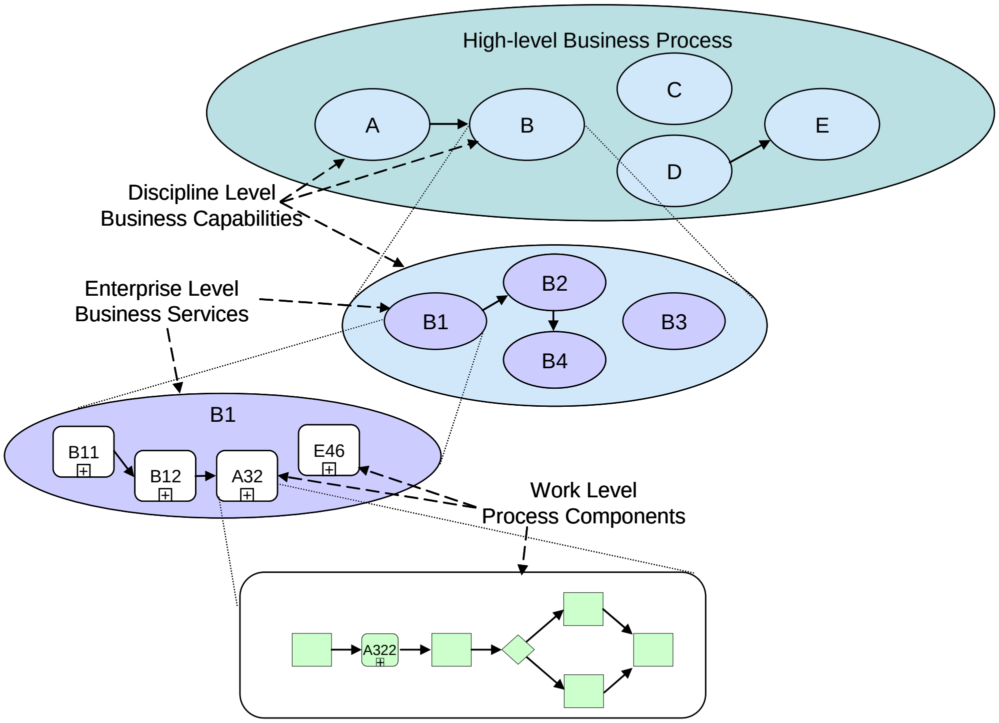
High-level Business Process
A
B
C
D
E
Discipline Level Business Capabilities
Enterprise Level Business Services
B1
B2
B3
B4
B1
B11
B12
A32
E46
Work Level Process Components
A322

Figure 4 – High-level business processes are composed of various levels of business capabilities and services

The Financial Management Enterprise Architecture (FMEA) project of the US General Services Administration (GSA) used a different approach (see Figure 4). What was particularly interesting was the way in which they extended the EDOC standard to provide multiple levels of business components. A number of "Discipline Level" business capabilities, which are really business roles, fulfill the needs of the highest-level business process (Process as Purpose). In turn, some number of "Enterprise Level" business services supports each Discipline Level capability. Each Enterprise Level business service is composed of a set of "Work Level" process components.[6]

For example, the Finance process is composed of a set of disciplines (Invoicing, Paying, Auditing, Cash Management, Budgeting, etc). These Discipline Level business capabilities were re-used in other processes, with each providing a set of "Protocols" that describe its external interfaces. The Acquisition Process was made up of Contracting, Transferring, Delivering, Ordering, Invoicing, Acquiring and Acquisition Accounting capabilities (amongst others). The way in which the Acquisition Process fulfils Acquisition Accounting for the Financial Management Process would include all the sub-protocols offered.

Another approach is outlined in a McKinsey article of 2003 (See Figure 5). The authors discuss the notion of business-driven "Domains", which are roughly synonymous with the capability oriented approaches discussed above. Domains have a common business context. They are used to control how development and infrastructure work is governed and funded. Services are then used as the interface to each Domain.

One could think of these Domains as portfolios of services aligned with the needs of a given community of usage. Having the business itself own these Domains ensures that the boundaries are not technology dependent.

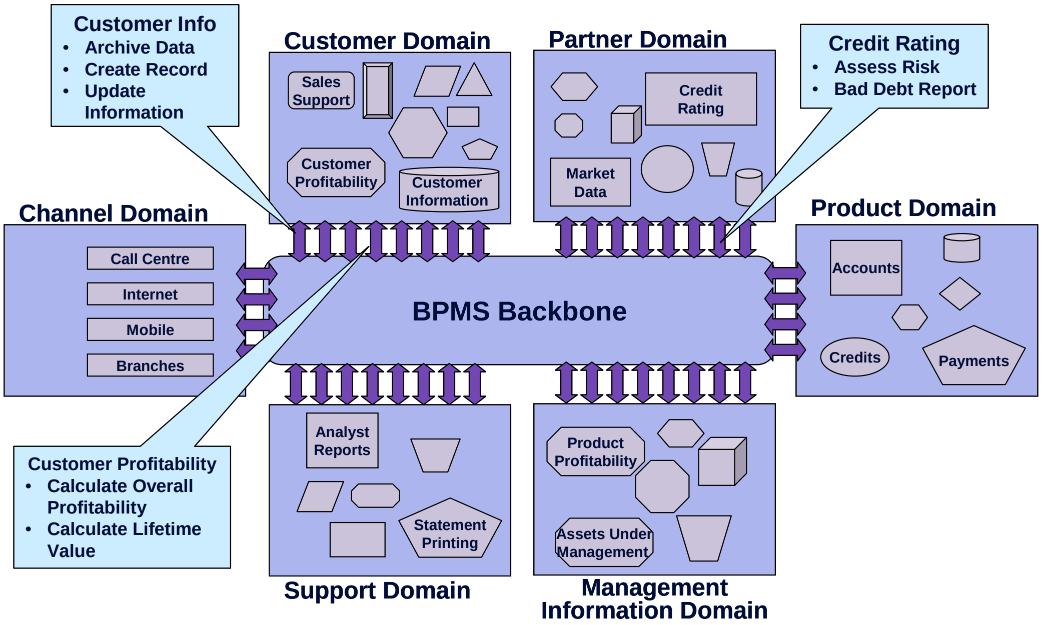

Figure 5 - The business is dissected into a number of cooperating Domains **(source adapted from "Designing IT For Business" McKinsey Quarterly #3, 2003 (p85))**

The point is that it is not so much *which* methodology is used; the important issue is to ensure that one approach is adopted within the firm (to aid communication). That also implies defining exactly what the terms actually mean. The ultimate aim is to enable business analysts to focus on the selection of the correct service and the mapping of data models between process and service. They should not have to concern themselves with the technical underpinnings of how a service is implemented. Moreover, this allows the business analyst to focus on improving the performance of one service at a time, rather than worrying about how each different business unit utilizes the process.

[6] This segmentation was inspired by a case study presented on the OMG Tampa meeting in February 2006. The abstract shown in Figure 4 is an adaptation (and simplification) of the approach that was presented, combining their language with our own views on Business Capability Modeling.

Effective BPM-SOA Platforms

Selecting the appropriate technology for *your* BPM-SOA initiative is also very important. There are a myriad of products and approaches, some of which will fit your plan, and some that will not. Some organizations will opt for a mix and match approach, choosing components from a range of vendors. Some will look for a BPM Suite that incorporates enough functionality for them to start managing sub-processes as services. Others will prefer to implement their infrastructure on top of a fully integrated suite, where the BPM Engine and services layer are purchased from a single vendor committed to their ongoing integration. For very large businesses this latter option is probably going to take preference.

But for most organizations, there are still some challenges in unlocking the functionality buried in existing systems. Most legacy systems were never developed with SOA or BPM in mind. When these legacy systems were developed they mixed the manual and automated process together, with people undertaking parts of the work and the systems doing other parts, i.e. the people are part of the system.

Indeed, in some firms, business processes are often seen in terms of the systems that were originally designed to support them. They are no longer part of the "how"; they have become the "what", with the system being more important than the process. So it is not a simple matter of just turning off parts of that legacy application and replacing it with new functionality. The legacy application ripples its way throughout the entire process, making it difficult to change one function at a time or even one process at a time, as many systems incorporate support for multiple processes.

This is a huge problem for large businesses that are looking to leverage their legacy applications in modern applications. As a start point, it is good to know what you have and what problems you are up against. For example, at General Motors they began a program of legacy replacement and realized it required effort across a number of fronts. As they assessed each system, they categorized them against a spectrum of need. In some cases they needed to "relearn" what the application actually did. Others needed "re-facing" (to allow their functionality to be re-used). Many of the related business processes needed "re-engineering", while some applications needed "replacing" and others needed "retiring".

In most situations, it is a question of re-facing (service enabling) existing functionality and ensuring it can be leveraged in modern BPM-enabled applications. Each element of application functionality must be packaged as a discrete service that is then invoked as needed within the modeled business process definitions. In turn, these processes may be exposed as services that can then be re-used by other processes, allowing for recursive definition of processes. The end result is that the CFO can hold onto those old mainframes, while watching IT become gradually more responsive and quicker in the delivery of new business services. Given the potential complexity of the legacy infrastructure, synchronous Web Services are not the answer to all of these problems. Indeed, an asynchronous messaging layer may be far more appropriate for separating some elements of functionality.

In this way, the modern BPM Suite and its associated SOA middleware components, allow new and existing functionality to be encapsulated in re-usable components that are then sewn into business processes as needed. The critical difference from the old workflow-style approach to integration is that, instead of integrating with each application at every point it is needed, the BPM engine calls a "component" or Web Service that has "wrapped" the legacy application.

This is a key requirement of the SOA layer. It must be able to wrap the functionality of existing enterprise applications (such as ERP, CRM or SCM) and then deliver it to new applications without requiring custom extensions of the underlying application. Indeed, there is strong evidence that the major application vendors such as SAP and Oracle (now including PeopleSoft, Siebel, and JD Edwards), are busy redeveloping their monolithic application environments to make them more accessible to customization in the process layer. The functionality delivered by the application is being broken apart to make it more amenable to customization on an instance by instance basis.

Overall, this approach to service integration and business processes allows the firm to extend and re-use its existing IT investments, leveraging existing functionality while enabling the development of new process-oriented applications that better fit the evolving needs of users. The result is a more agile, more responsive technology environment that developers can use to build new business capabilities faster, cheaper as the systems and process are independent of each other (i.e. lower implementation risk). Perhaps more importantly, it is now in a vocabulary the business can understand

Best Practice Observations

- ✓ *It is only by seeing and experiencing the benefits of BPM-SOA that the organization will get behind the program and help drive the change. Moreover, as a result of seeing and experiencing iterative change, the business requirements themselves start changing more rapidly. While the long-term benefit will deliver massive returns, this is usually only achieved after several iterations as the processes and service frameworks evolve to more closely fit the needs of the business.*

- ✓ *Ensure a high-level Steering Group is formed with Executive level sponsorship. Otherwise, the initiative will probably be doomed to failure. It is important to make sure that the battles are fought at the right levels, taking into account the overall strategy and objectives of the firm. The key thing is to make sure that there is agreement around how the roles and responsibilities are divided. Decide who is accountable for what, and then make sure that all members of the management team are consistent.*

- ✓ *Establishing effective governance policies requires a significant investment in developing appropriate operational policies, which must be communicated and compliance measured.*

- ✓ *Working with the Steering Group, agree the scope of the initial project and develop a roadmap on the way ahead; highlighting how (and when) other parts of the business will be involved. Iterate on the roadmap after each project cycle. Target short term benefits (tactical projects to build success) while creating a consistent and flexible IT landscape that can support evolving business requirements.*

- ✓ *It is important to start small and demonstrate success, then to build on that success. As a general rule, the wider the scope, the more difficult it is to choreograph the initial project, but bigger the opportunity for value innovation. Choosing the right start point and forming an effective team are critical.*

- ✓ *Ensure that the business users expectations are set appropriately at the outset. Acknowledge the fact that BPM-SOA involves a fundamentally different methodology (than that pursued in traditional development project lifecycles) focused around iteration and adaptation. Ensure that the user population understands the implications of this change (there should be an intention of deploying the first iteration within weeks of the initial rollout). Otherwise, they will assume that this project is like all other IT projects that came before (with the next version always 2.5 years away). This is essential to stop scope creep and lower the risk of project failure.*

- ✓ *Do not overestimate the "maturity" of the organization. The transformation to a service-oriented, business process driven operating model requires change in the prevailing mindset of both the business and IT organization. For many individuals, this is a personal development journey.*

- ✓ *Undertake an exercise to characterize the high-level capabilities of the firm. These capabilities are the behaviors that the firm must exhibit in order to fulfill its "external relationship requirements" with customers, regulators, shareholders, partners, suppliers, etc. The objective is to identify all strategically significant influences on the firm such that suitable mechanisms can be installed and monitored. Capture the boundary conditions between these capabilities.*

- ✓ *When modeling business capabilities, think about the constituencies that are affected, and make sure understand how they fit onto the roadmap. In order to limit the dependencies between a capability and those processes that make use of it, express the interface as a service contract. Design the contract in such a way that it is not specific to any one type of request. This should be at the declarative level, rather programmatically defined. If it is hard-coded, then the benefits of loose coupling are lost.*

- ✓ *Instead of decomposing the desired functionality into chunks (which are then coded and tightly integrated), the best practice is to assemble a set of standalone services into processes, with each of these discrete services delivering some aspect of the desired functionality.*

- ✓ *When it comes to developing effective process architectures, focus on building families of process components (services) that can be instantiated and re-used as needed. Develop a series of coarse-grained integration services such as Retrieve Customer Profile, Update Address, Calculate Credit Score or Update Inventory.*

- ✓ *Start with a plan – not a complete plan for the entire organization, but one that allows the team to build experience and knowledge, pilot and then iterate. Tightly scope the initial project, define the intended timeline and then flesh out the details. Start with a few services – be prepared to add more services, but only within the context of the plan (the initial business process targeted). Keep an eye on the goal. If it is not core to the initial project, earmark it for a subsequent iteration or another project.*

- ✓ *Pilot the SOA architecture and BPM approach, rather than piloting the services in the pilot. Keep in mind that the intention is to develop the overall architectural framework for the enterprise. Inevitably, organizational change is required. This must be carefully managed and synchronized with technology implementation.*

- ✓ *Rather than attempting to boil the ocean at the outset, ensure that a tactical project is undertaken so that the business can get a taste of success. Ensuring a tightly scoped initial project is one of the critical factors for success. For the initial project at least, avoid the temptation to spend time populating a repository oriented modeling tool. While this sort of effort may add value to a BPM CoE as it grapples with organization-wide issues, it is a distraction during that tactical experiment.*

- ✓ *Understand that the architecture of SOA is only partly responsible for the potential benefits, which more directly derive from service design. Granularity is extremely important as it is possible to kill any project through bad service design. A successful service is one that is re-used often and changes seldom. Start by designing from the customers' perspective, taking into account their needs for reliability, performance, on time delivery, service support, and transaction processing. Factor in multi-channel thinking early.*

Conclusion

While BPM and SOA based strategies can be pursued independently of each other, they make far more sense when integrated together. When BPM is deployed using SOA techniques, services are used as building blocks that can be orchestrated via the BPMS to support the needs of even the most complex business processes. In addition to creating new services, a key design principle of SOA is the ability to wrap components of application functionality, and then expose those components as services that can be called by different business processes. These reusable services can also be assembled to form new "composite" services and applications.

To regard BPM as a subset of SOA is a mistake. Such thinking will ensure that business agility continues to be dependent on IT. Rather than removing the barrier between IT and the business, it will help re-enforce it. BPM is about driving improvements in business performance, yet benefits from SOA thinking. While SOA aspires to underpin business agility, it is fundamentally an approach to IT integration. If anything, the BPM-SOA relationship is the other way around. SOA is a best practice design approach to developing and deploying a BPM initiative.

To be successful in the long term, BPM and SOA need to become a way of thinking at the business level. That will take time and cannot be rushed. The Steering Group must ensure that the initiative is resourced appropriately, with the right development infrastructure and delivery mechanism. Of course, an effective, architectural vision will help – reflected in both an evolving business roadmap and a set of supporting technology services. Alongside this vision, a management discipline is essential to ensure that the initiative continually drives toward the goal of enhanced business performance.

Appendix - SOA Core Concepts

Developing a Service Oriented Architecture is nothing particularly new. SOA is a software engineering best practice for any technology that requires integration. The core concept – loosely coupled layers (or components) that help insulate the overall system from change – has been around for a long time. Indeed, it has been central to most software development over the last two decades. At its heart, the approach relies on standardizing the communication interfaces between components (or layers).

Throughout the 1990's we saw the emergence of software standards such as CORBA and DCOM as mechanisms for distributing an application over a computer network. Every application had its own proprietary Application Programming Interfaces (APIs) that allowed others to access its functions. Yet the result still resembled a plate of brittle spaghetti. Everything was still connected to everything else.

What has changed over the last 10 years has been the widespread adoption of eXtensible Markup Language (XML) as the lingua franca for moving data between one system (application) and another. Web Services have also emerged as the latest method of remotely invoking application functionality. While Web Services are largely used inside the firewall, they also enhance the ability to collaborate with external partners. There has also been an explosion of middleware approaches that provide alternative (and sometimes superior) mechanisms for applications to interoperate (and support an effective SOA initiative).

Along the way, the best practice of software development has shifted. Rather than building monolithic applications that rely on a separation of data, application and user interface for scalability, we have a new concept underpinning the delivery of applications. Discrete elements of technology functionality are wrapped into services, with only their interface requirements exposed.

In a SOA based infrastructure, applications are delivered based on a set of re-usable service-oriented building blocks. Services that encapsulate and express business logic in an accurate and extensible manner, much like Lego blocks can be used to construct a myriad of different shapes. Instead of designing applications to last, they are now designed to change iteratively, with the intention that they will work in many different situations or contexts.

The functionality of a given service is often assembled from a number of other services. At runtime, after potentially several levels of service calls, some sort of procedure is executed. And even then, a step in the procedure will often call an infrastructure service (component) that wraps some existing (legacy) application to extract data, or update a database field. All that is needed at each level is the definition of the external interfaces and the mechanism that is used to call the service.

With careful design and development, a comprehensive portfolio of reusable business-oriented services can be delivered to business professionals who then re-use them in processes definitions, applying rules to how they are invoked and deployed. The business analyst, using a BPM Suite, can inspect the registry of available services, choose the correct one, drop it into the process flow and then map the relevant inputs and outputs. Change happens at the level of the business (rather than through another round of IT development).

But SOA itself does not necessarily deliver this chain of execution. SOA is a design approach. That is where a BPM Suite comes in, orchestrating the constituent services as described in some process model. Rather than driving the chain of services with inflexible code, an independent process layer governs the execution order, delivering a great deal of flexibility and the agility that business craves.

Of course, a BPM Suite (BPMS) can be deployed to drive continuous performance improvement, independently of SOA based concepts. But to enhance flexibility and re-use, processes often end up being broken up into a series of stand alone process components that are then called as needed by other processes. The BPMS coordinates the way in which composite processes re-use the functionality delivered by discrete process components, which in turn may wrap the functionality of third party applications. Effectively, this is SOA but from the other direction (top-down rather than bottom-up). But to make this work across the enterprise requires careful planning and an underlying process-service architecture. The development of that architecture is itself an iterative process.