

The SOA Magazine

Feature Article



Dynamic SOA and BPM

by Marc Fiammante

Published: September 25, 2009 (SOA Magazine Issue XXXII: September 2009)

[Download PDF](#)

[Digg This](#) • [De.licio.us](#) • [Slashdot](#) • [Technorati](#) • [StumbleUpon](#) • [Google Bookmark](#)

Abstract: After several years of companies industry-wide combining Services Oriented Architecture (SOA) and Business Process Management (BPM), the results are mixed. Some companies have had substantial benefits moving to SOA, while others have had average results. All these companies used the appropriate technologies, such as Web Services and Business Process Execution Language (BPEL) for processes, so the outcome should, in theory, be more predictable. It is now a good time for companies to extract the best practices and learn from others' experiences. This article is an introduction to "Dynamic SOA and BPM", a book which provides an exhaustive exploration of the best practices for delivering dynamic business processes and business services in order to quickly absorb market condition changes.

Introduction

When SOA was initiated a few years ago, the simplified integration capabilities brought hopes of a simplified Business and IT landscape with reusable business components enabled by open technologies. There were however several reasons for not receiving the full business value of this services approach, including these essential ones:

- Business semantic tight coupling on a technical loose coupling: Many projects just used Web Services to implement a Client/Server approach. Any change in the server interfaces leads to client changes and high change costs.
- Business Processes reintroducing the tight coupling of flexible business services. Enterprises have implemented end-to-end processes such as order management in very large business processes, but then faced the lack of reusability of some sequences that could have been modularized and exposed as services.
- Rigid information models used to expose business services. In many cases services are only viewed as operations, forgetting that the business information structure that they carry will as well vary with the evolution of the business and lead to costly changes of services and processes.

Each of these experiences induced a deeper thinking process around what should be the essence of a business variable approach and how we could reach a true business/IT alignment with the expected shortened IT cycle enabling faster and cheaper business reactions.

How Other Industries Approach Varying Conditions

The IT industry is not the first industry to face varying conditions.

Whether it is in the automotive industry with shock absorbers, or the building industry with expansion joints, there always is a mediation layer that absorbs the variations or allows one part to move or to have varying characteristics without affecting the other part.

The integrated elements may not need to vary or move internally, but at the end, a dynamic assembly is realized with tightly-coupled parts flexibly linked with other parts. Similarly with an IT approach, there must be a notion of levels of assembly and of granularity in components.

But the final assembly is always designed based on a global architecture that looks at requirements, operating conditions and desired capabilities. Dynamicity is not a goal by itself; rather, it has to respond to a specific context and desired states. Similarly the enterprise and business architects must look at how the enterprise business model needs to evolve and what market conditions it will face to understand what the variability needs to be.

Streamlining the Enterprise Architecture

Every industry creates plans to address new markets, customers or products and it is essential to define which of the components will be tightly-coupled and where the flexibility needs to occur, and with what limits.

However, it is a common pitfall to merge the business strategy, the implementation, the information and the infrastructure aspects into a single Enterprise Architecture Map. In a dynamic enterprise approach, using BPM and SOA, the Business, Applications, Information and Technology domains, will each require a specific mapping and domain decomposition leading to a four-layer view of the enterprise.

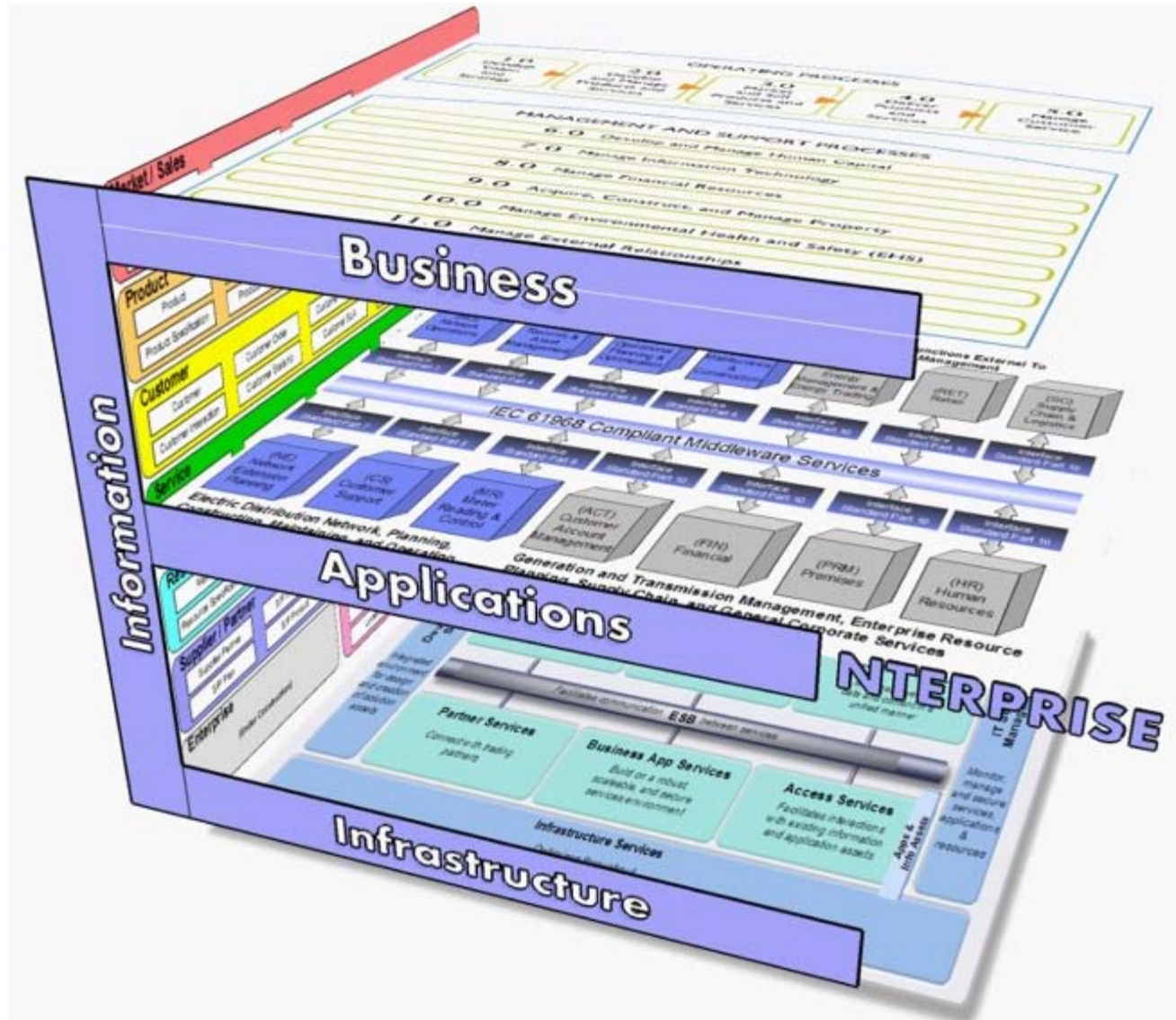


Figure 1: 'E'nterprise architecture layers Mnemonic

However, a full-fledged enterprise architecture with an exhaustive approach for each of the above layers is perceived as a costly and lengthy process that does not always provide quick wins and pragmatic results. Applying these methods and this framework to their full extent for an enterprise can be an important effort, and in a dynamic BPM and SOA approach we need to find a streamlined approach that still has the necessary architectural models and work products while being affordable.

In the book, I expose techniques to deliver a streamlined enterprise architecture including "Horizon-Based Enterprise Architecture", "Enterprise Architecture Staged Zooming" and accelerated implementations using template business and technical architectures.

Enterprise Business Layer

The purpose of an enterprise map in a dynamic BPM approach is to identify the boundaries of process modules that will have value for the enterprise, and explore the bridge of enterprise architecture with the business modularization.

It is essential to identify the business components that deliver business capabilities and act as service centers in the enterprise with potential to operate independently through their ability to deliver a unique set of business capabilities.

Because of the need to operate independently, a business component will be at the intersection of several axes which can potentially be:

- A business area or a decomposition of this area
- An organizational structure operating within a business area
- An accountability level
- Business life cycle phases

Based on these different views, there are several approaches to define a business map and the business components and to model the business. A common aspect is that they all provide a decomposition of the business of an enterprise.

There are a few standards available for each of the layers; here are some public examples of such enterprise maps and decompositions.

NGOSS Business Process Framework (eTOM)

This standard provides a business process functional decomposition framework for Telecommunication Service Providers, and serves as a neutral reference point and blueprint for decomposing processes to a level 3.



Figure 2: The Enhanced Telecom Operations Map© (eTOM)

This "level 2" map is complemented by a "level 3" functional decomposition in a 300+-page document named GB921D.

Even though the standard states it is a business process decomposition, it provides no standardization of workflows but only a static functional decomposition of the business operations is provided.

APQC Process Classification Framework

The APQC standards body developed a classification of an enterprise's most usual processes as a common language and open standard.



Figure 3: The PCF enterprise-level categories

The Process Classification Framework (PCF) includes process groups and more than 1,500 processes and associated activities down to a level 4 of decomposition.

One of its essential aspects is that it provides a standard for naming the different levels. Similarly even though some of the level 4 activities are usually sequential there is no workflow standardization in the PCF.

These four levels are:

1. Categories: classifying business competency domains of the enterprise
2. Process Group: Grouping processes in business components that apply to similar business entities
3. Process: Formalized list of tasks and activities required to achieve a specific aspect of a business component
4. Activity: The granular business definition of an element that is chained by a higher level process.

This classification will serve as the base for the further discussions in this book particularly in the business service granularity discussion.

SCOR Supply Chain Council

The Supply-Chain Council is a non-profit global independent organization, composed of many companies around the world, which has produced the Supply-Chain Operations Reference: a model that integrates the best practices for supply-chain operations business process reengineering.

SCOR addresses five distinct business domains called "Management Processes" which are plan, source, make, deliver and return and defines the following four levels of decomposition:

- Process Type: "level 1" or "Top Level" provides a balance across business domains and cross-process categorization. It defines the scope and content of the above-mentioned management processes.
- Process Category: "level 2" or "Configuration Level" provides a reconfigurable level which companies will adjust to their own business plans.
- Process Element: "level 3" or "Process Element Level" contains the process modules that operate within a business component and can be reconfigured to achieve the higher-level processes.
- Activities: Finally, "level 4" or "Implementation Level" contains the necessary tasks and granular business services. SCOR standard considers this level to be specific to each and every enterprise and does not standardize this level.

Process model decomposition appears as a common practice among the publicly available models and standards. They do not all take the exact same terminology but they all tend to modularize the representation of the business as a hierarchy and sequences of lower sets of business tasks.

Defining Service Granularity from Business Maps

A frequent question is, what is a good service granularity? The business maps and decomposition are particularly powerful tools to identify a service granularity and answer this question, but for the purpose of a common understanding, let's provide some definitions:

- Business Service: A business service is the grouping of repeatable business tasks that address the same functional domain. (e.g. Process Customer Order). Business policies will be defined at this business service level, such as policies to handle variations of business services for corporate customers or individuals.
- Web Service: A technical representation of elements of a business service, grouping discrete business tasks together for technical management such as versioning in a technical descriptor; using as technical standards, either Web Services Description Language (WSDL) or in a more abstract fashion, Service Component Architecture (SCA).
- Service operation: Is a repeatable business task, and in the above case it can be "Perform Order Feasibility Check" or "Receive Order Status Update". These are usually qualified as operations in the WSDLs or SCA descriptors. The number of operations range from one to ten or more, and a good

average is seven.

Given these definitions, let us do a simple computation. At a level 2, an Enterprise usually has between 50 to 100 Business Components or Process Groups. Down one level there are between 300 and 1,000 processes in the enterprise. The eTOM telecommunication business process decomposition gives around 400 processes at level 3.

At level 4 the number of tasks will be between 1,000 and 10,000. APQC has 1,500 activities defined at level 4 and if we pushed eTOM to a further decomposition at a level 4, we would get around five to seven times 400, or around 2,500.

We see that this already is a very large number of tasks, of which a large portion has the potential to become business services. If we compute the number of operations, this gives us a potential of 5,000 to 20,000 service operations for the enterprise, which is a huge effort to manage. However, only a fraction of the enterprise has value to be exposed as business services and processes, and from our experience, between 1% and 5% of the potential services operations have reusability value at the enterprise line of business levels.

The conclusions on granularity are:

1. A Business Service granularity that maintains value and manageability is between level 3 and 4 of the enterprise functional decomposition. This usually corresponds to the APQC level 4 activities.
2. Without a functional decomposition there is no way to know the granularity level of a given service.
3. Processes should be modularized using a functional decomposition to make the modules reusable as business services at level 3. At level 4 or under, the Business Services are exposed from applications.
4. There must be a variability approach to aggregate any existing API at level 5 or below, to the large enough granularity to be exposed at the correct level.

Enterprise Applications Layer

There is a difference between the business components and the implementation of the services that they require. Let us look at a concrete example.

At the level of an "Order Handling" process, part of the "Order Handling" group will require accessing services from a "customer information management", a "product catalog management" and a "Customer Order Lifecycle management" application, which is implementing the services.

Looking at the future state, there is a need to define the "ideal application map"; the one that an enterprise would create if there were no financial, time and technology restrictions. These maps are often delivered by IT consulting companies and often referred to as "city planning" and are the super structure of interfaces that will be exposed as the reusable business services layer.

Defining these maps in a top-down approach is essential to isolate the business processes that will consume services from the real implementation in the existing applications.

This top-down approach will need to be complemented by a bottom-up approach that defines how existing or new applications and packages will be adapted to expose these interfaces in a flexible and consistent manner.

An example of a standardized "application map" is the one provided by the TeleManagement forum that is the most advanced in that space, as telecommunication operators are faced with exponential growth and very dynamic market evolutions, requiring higher levels of standardization for lowering costs.

TmForum Telecom Applications Map

The Telecom Applications Map provides the bridge between the standardized business process and information framework building blocks and real, deployable, potentially procurable applications by grouping together process functions and information data into recognized operation support and business support applications or services.

Even though no standard can ever represent a perfect systems infrastructure for an operator, it provides a

functional and information guideline to implement a layer of reusable business services exposed by an application.

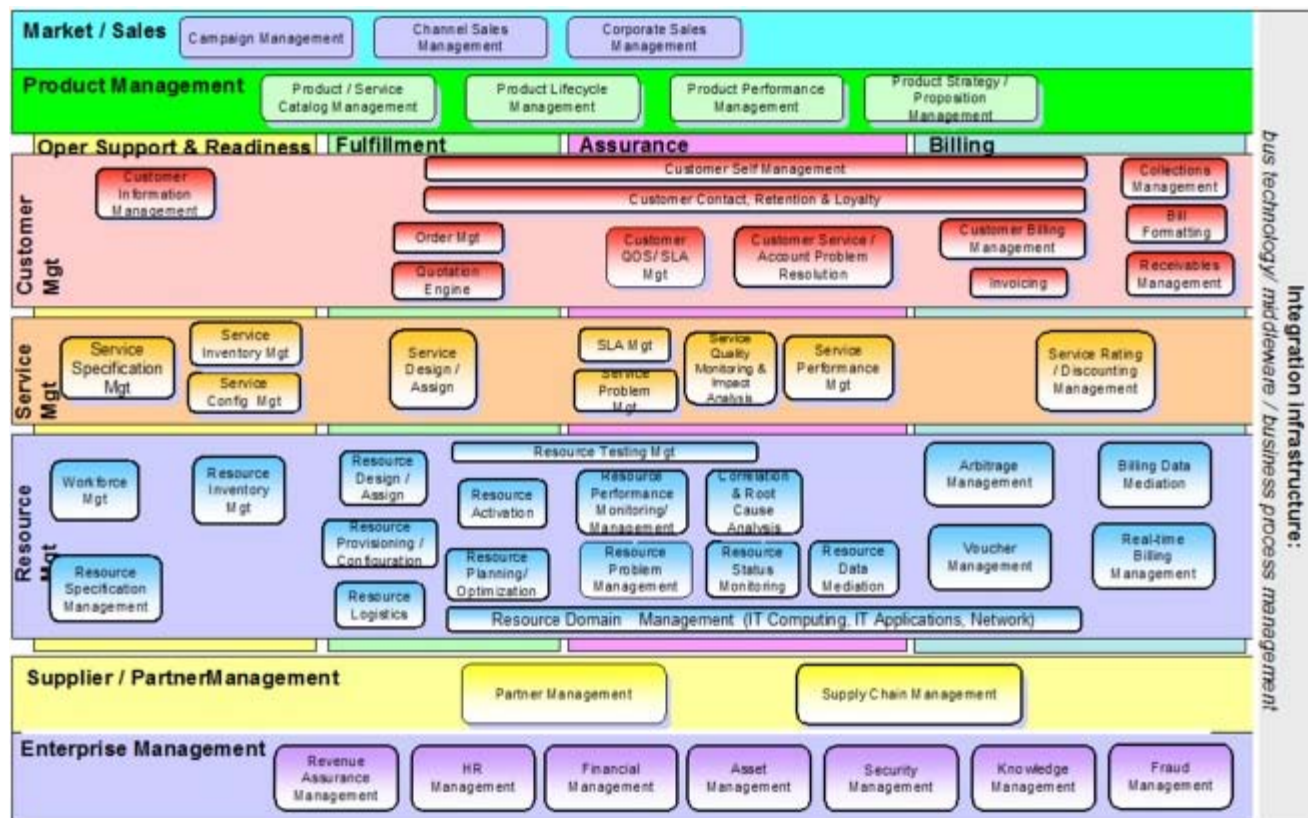


Figure 4: The TmForum Telecommunication Application Map GB929 R2.1

IT Infrastructure Layer

The business processes and applications will require an IT and middleware infrastructure to operate, with support for variability and flexibility at a core technical level. There is no commonly accepted standard for middleware and operating systems services.

An approach to an agnostic map is the following vendor-independent SOA Reference Model that serves as a reference point to position both middleware and functionality within a service-oriented architecture.

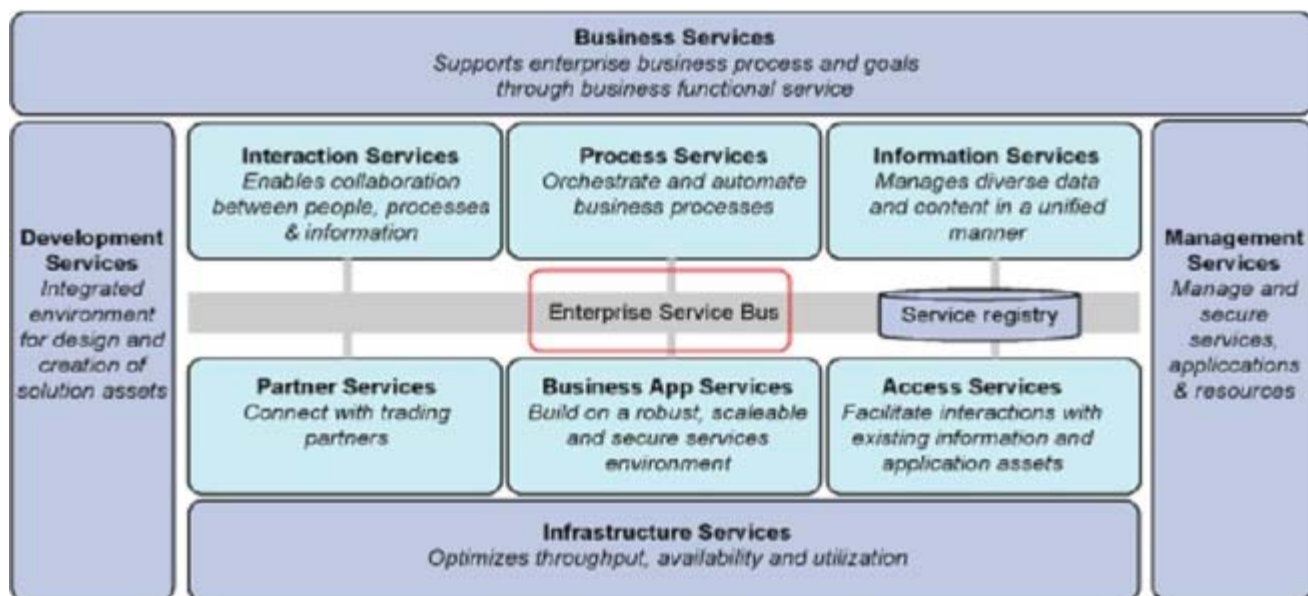


Figure 5: An agnostic SOA Foundation Reference Model

Enterprise Information layer

Information is the essential "logical" asset that links together all layers of the enterprise. As processes do not run in a vacuum, but carry, transform and refer to information, the structure of that information reflects the way the enterprise wants to operate. The following example shows how Telecommunication operators relate business domains to core entity domains:

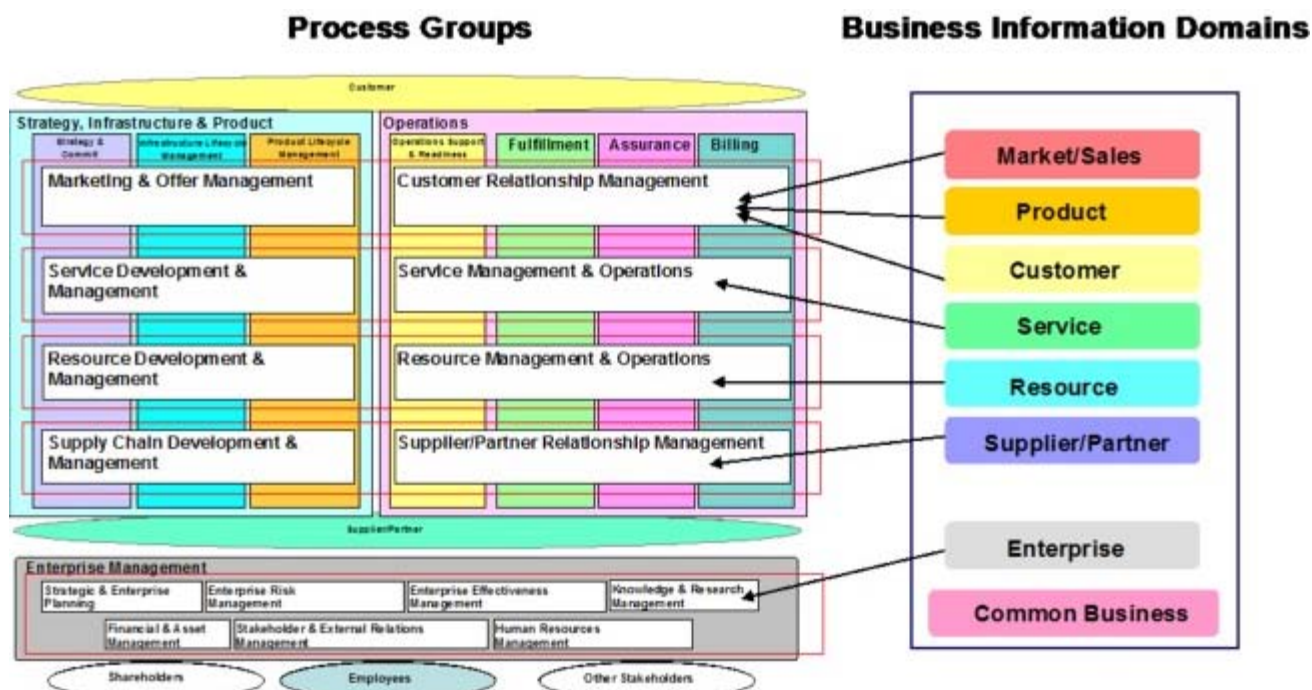


Figure 6: TeleManagement Forum eTOM & SID relationships

The business information model, domain maps and core entities represent the business view of the information as opposed to data models that address the technical implementation of managed objects data (see IETF RFC 3444).

This information model is decomposed to align ownership, but must include provisions for variability to

protect the business processes and services from structural changes of the information and its data representation.

Basic Principles for Enterprise Dynamicity

Now that we have set the static scene for the Enterprise Architecture, how do we use all of the above elements in a way that allows a dynamic business and IT approach? Dynamicity implies flows and movements, with events and message propagating through the enterprise and variations between and within the business components. Flows such as "Order to Bill" flow across enterprise components, and are often referred to as end-to-end processes. However, it is essential to differentiate the apparent effect of an events chain reaction from an explicit choreography of business services owned by a specific organization within the enterprise. Both are called business processes but are different in nature and implementation.

To differentiate these business processes we will use business decompositions and map to a common terminology with a precise definition of the different types of processes at each level. For this purpose I am going to use the categorization of basic types of processes derived from the BPMN standard.

Categorizing the Processes

The BPMN standard introduces three basic types of sub-models within an end-to-end BPMN model :

- Collaboration (global) Processes (business-to-business between enterprises or organizations)
- Abstract (public) processes (inside an enterprise across business components or internal organizations)
- Private (internal) business processes (within business components)

In addition, BPMN introduces the notions of "pools" and "lanes" that is also helpful in decomposing processes in modules and grains that can then be articulated in a flexible and dynamic fashion.

Collaboration (global) Processes

A collaboration process is the description of the interactions between two completely independent business entities. The following diagram shows a collaboration process described by a sequence of purchase order processing between 2 companies.

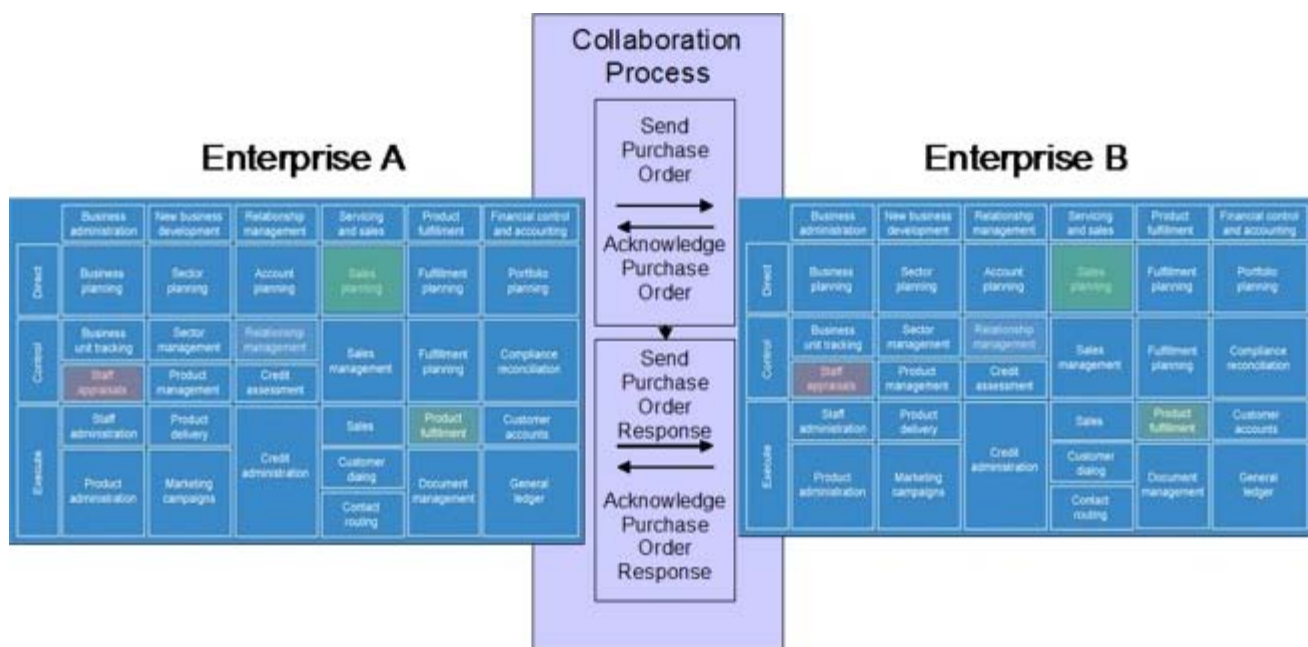


Figure 7: Example collaboration process

Even though there is a formal sequence, there is not an engine between the enterprises orchestrating that sequence. Each company/enterprise is responsible for sequencing its own part within its boundaries.

Abstract (public) Processes

These processes represent the overall interactions between different internal organizations in the enterprise owning their internal business rules or sequences. A consequence is that abstract processes do not have a single business owner, and would not be easily manageable if implemented as a single IT construct.

A first obvious level of abstract processes is between business components such as defined above, but they may also occur between the processes at level 3 of decomposition. Exactly as for the collaboration processes, there cannot be a single engine controlling the interactions between the business components or level 3 processes, because there would not be an owner of the logic in that engine.

Understanding this difference between an explicit control and an implicit realization of the business process is essential to understanding the approach for implementing dynamic business processes.

In the following picture the sales to bill abstract process is represented by the interactions between the business components.



Figure 8: Abstract Sales to Bill process

Private (internal) Business Processes

Within a single organization with a single identified business owner, that owner can manage the rules and sequences of tasks which then are private to that "owner" or his delegates. They operate within the boundaries of a business component and there can be several private processes in a given business component or within a finer grained decomposition.

As a consequence, "private processes" are the preferable subject to explicit flow control and automation because they are manageable assets of the enterprise, for which stakeholders and life cycles can define.

They can be either explicit- if expressed in a workflow language or service choreography/orchestration language such as XPD L or BPEL- or they can be implicit if resulting from the code running inside a particular application or package (e.g. SAP, Oracle, Amdocs provisioning etc.) These processes will chain to other private processes either by the means or events or services calls.

Within a private process, a "Pool" represents a sequence of tasks or activities driven by a specific participant and their identification has an impact on the selection of the appropriate technologies for implementing particular portions or services of a private process.

Applying Decomposition to End-To-End Process

As a consequence of the previous decomposition of processes in categories, an end-to-end process will then be the abstract or collaboration process resulting from the assembly of private processes that group the pools that interact within their boundaries.

Here is an example of "order to cash", an end-to-end abstract process composed of a chain of private processes each contained within the boundaries of a business component. It includes variations of private processes, such as the various service configurations, as well as private processes that are common successors in an abstract flow such as the Set Top Box configuration.

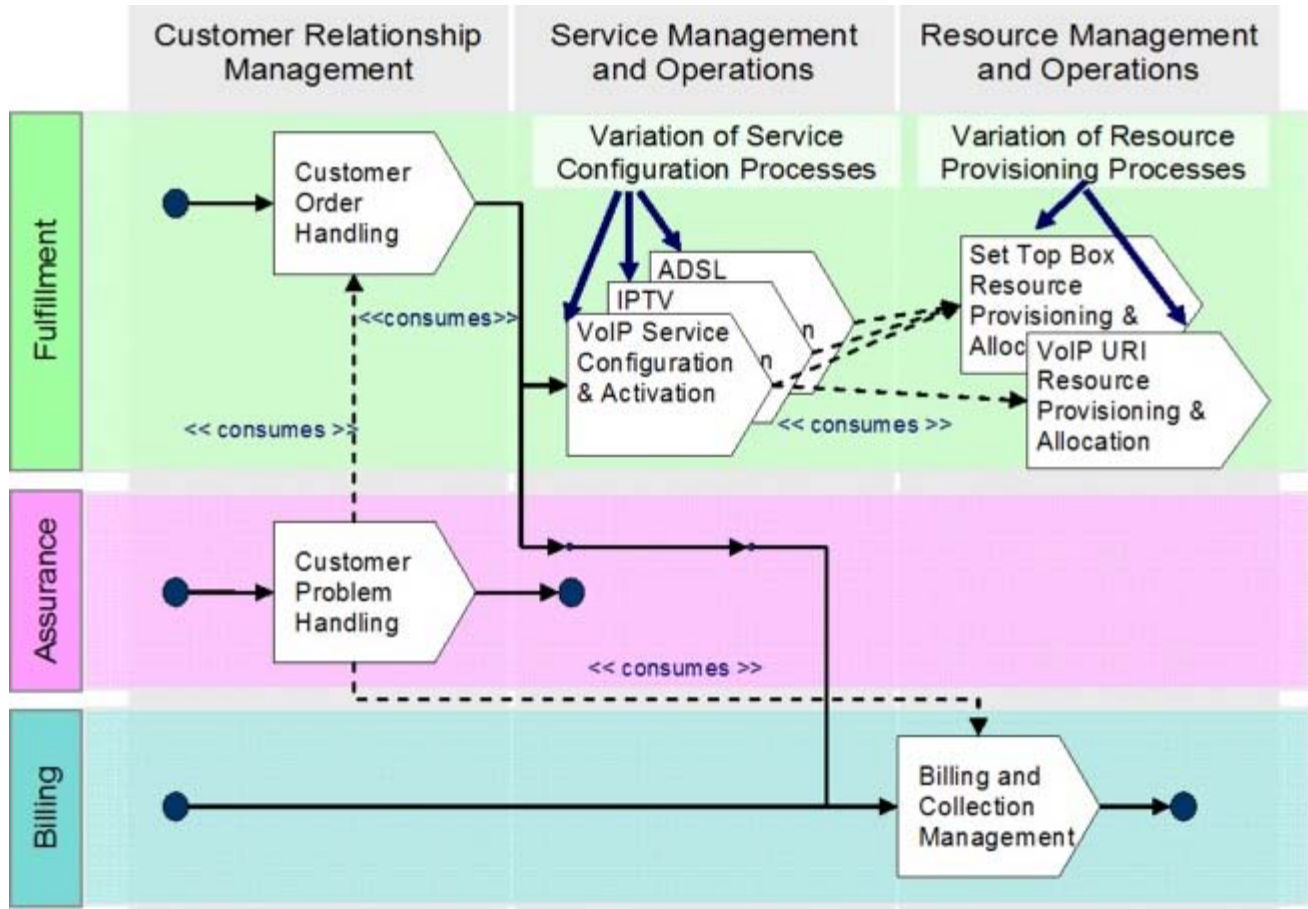


Figure 9: Abstract process chaining private processes on a business map

In a dynamic process approach we must be able to add a quadruple play service, such as "gaming", without having to regenerate and test the end-to-abstract process.

This can only be achieved by means of dynamic binding and coordination between the various private processes.

Impact of Business Information Changes to Processes

The less you carry the more agile you are. Similarly, the less information processes carry, the more they are able to handle variation of enterprise information. But that information still needs to be accessible from the various business organizations of the enterprise and the appropriate information must be available to the business components that require it for valid business needs.

To reach process agility we have to move from processes that carry explicitly all of the information to processes that refer to information preserved elsewhere. We then only pass through processes, the core of the decisional information and preserve the bulk of the information using a master data management approach and information services to access that data as required by the business components and processes.

To solve the interface variation for new attributes, an enterprise information variability approach is the necessary complement to ensure that within acceptable limits, information model characteristics can change without affecting the interfaces.

The Enterprise Expansion Joint

The mediation layer between components provides flexibility between services and processes and has initially been the Enterprise Services Buses.

There is a need to push the capabilities of this layer from technical variability, protocols, and messaging, to content and business semantic adaptation, thus removing the interface tight coupling between consumers and providers. As a simple example, IT should be able to evolve the versions of provided business services without requiring an immediate regeneration of business service consumers.

This leads to what I call the "Enterprise Expansion Joint": the capacity for the business to expand using existing IT capabilities, without having to enter a long delivery process. The business variations of this expansion joint have to be bound to limits identified during the Enterprise Architecture phase.

Conclusion

We have seen in this article that there are multiple aspects to address in order for an enterprise to derive the value of a truly dynamic BPM and SOA approach.

It requires structuring the business, defining the business goals to identify foreseeable business variation, mapping organizational ownership to process architecture, as well as structuring the business information and implementing a flexible business and technical joint. There is not a "one size fits all" solution but the need for a set of modeling steps and implementation techniques that are described in the book, resulting from our various experiences.

THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL



[Home](#) [Past Issues](#) [Contributors](#) [What is SOA?](#) [SOA Glossary](#) [SOA School](#) [SOA Books](#) [About/RSS](#)
[Legal](#)

Copyright © 2006-2009
SOA Systems Inc.