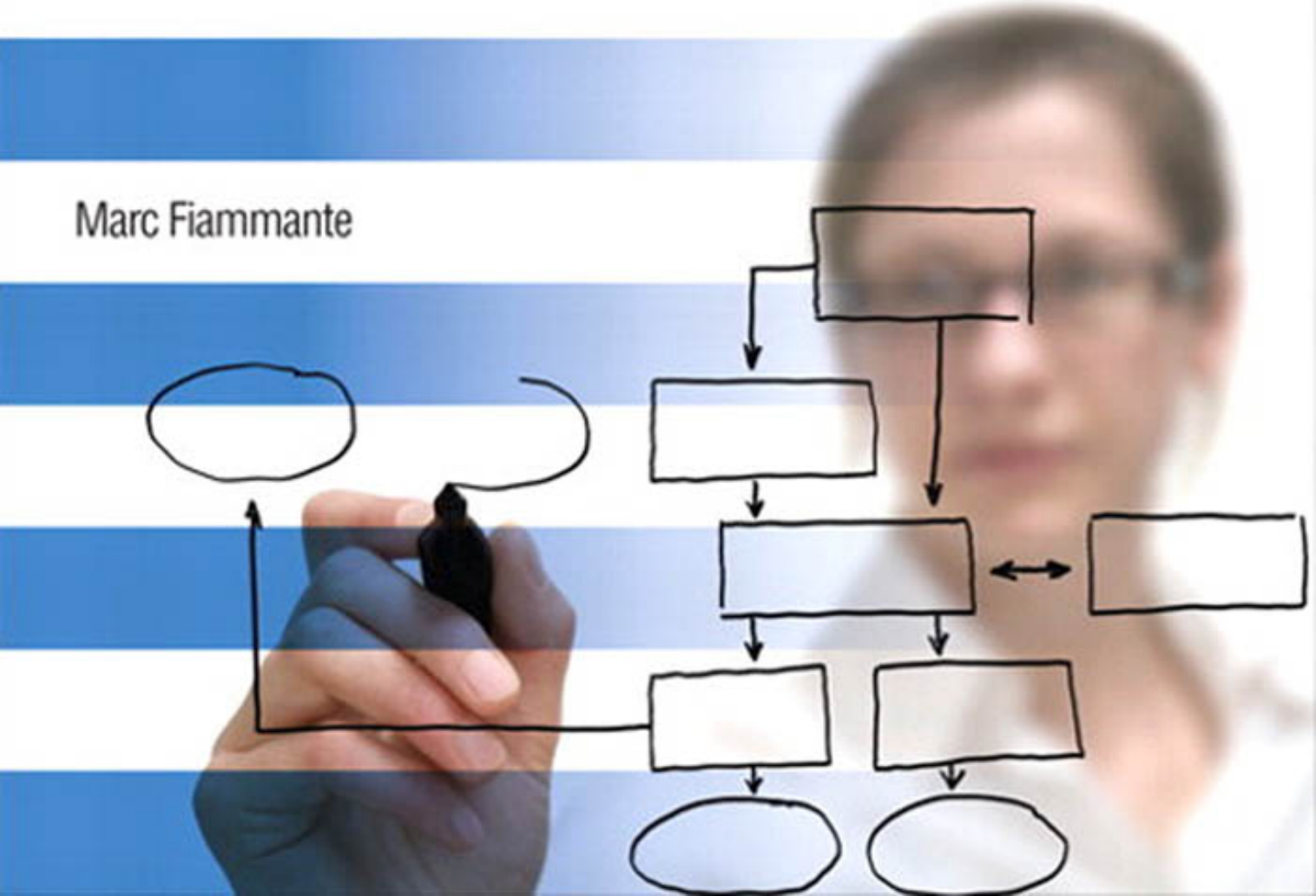
IBM PRESS

Dynamic SOA and BPM

Best Practices for Business Process Management and SOA Agility

Marc Fiammante



The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

© Copyright 2010 by International Business Machines Corporation. All rights reserved.

Note to U.S. Government Users: Documentation related to restricted right. Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

IBM Press Program Managers: Steven Stansel, Ellice Uffer
Cover design: IBM Corporation

Associate Publisher: Greg Wiegand
Marketing Manager: Kourtnaye Sturgeon
Acquisitions Editor: Katherine Bull
Publicist: Heather Fox
Development Editor: Kendell Lumsden
Managing Editor: Kristy Hart
Designer: Alan Clements
Project Editor: Anne Goebel
Copy Editor: Geneil Breeze
Indexer: Lisa Stumpf
Compositor: Jake McFarland
Proofreader: Sheri Cain
Manufacturing Buyer: Dan Uhrig

Published by Pearson plc
Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U. S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside the U. S., please contact:

International Sales
international@pearsoned.com

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, the IBM logo, IBM Press, CICS, DataPower, developerWorks, IMS, Rational, Redbooks, RequisitePro, Tivoli, WebSphere, z/OS. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.

Library of Congress Cataloging-in-Publication Data

Fiammante, Marc.

Dynamic SOA and BPM : best practices for business process management and SOA agility/Marc Fiammante.

p. cm.

ISBN-13: 978-0-13-701891-8 (hardback : alk. paper)

ISBN-10: 0-13-713084-8 (hardback : alk. paper) 1. Service oriented architecture (Computer science) 2. Business--Data processing. 3. Industrial management--Data processing. 4. Business logistics--Data processing. I. Title.

TK5105.5828.F53 2009

658.500285--dc22

2009020223

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671-3447

ISBN-13: 978-0-13-701891-8

ISBN-10: 0-13-701891-6

Text printed in the United States on recycled paper at R.R. Donnelley in Crawfordsville, Indiana.

First printing July 2009

Foreword

Confronted by competitive pressures and the worst economic climate in decades, today's enterprises need to be increasingly focused on leveraging SOA and BPM to meet their business objectives and capitalize on market opportunities. With mashups and Web 2.0 capabilities becoming ever more prevalent, SOA has reached a level of pervasiveness that requires businesses to quickly adapt. Nowhere is the need more evident than with the millions of mobile workers who toil each day outside the bounds of the traditional enterprise. These workers depend on their devices to consume services and enable business processes cleanly and efficiently while being flexible enough to integrate behavior and business logic changes without requiring new deployments; they expect changes in backend enterprise systems to be totally opaque.

Approaches that increase flexibility help limit the impact on the consuming side when changes are made on the provider side, which leads to greater business value, better development practices, and faster time to market. Dexterra is leading enterprise mobility by example. At Dexterra, we embrace service-orientation in building composite mobile solutions that deliver SOA and BPM best practices to help demanding companies automate previously manual processes, empower employees with the information they need to improve job performance, and transform the enterprise with real-time visibility and control. Our Dexterra Concert mobile platform embodies this SOA approach to enable enterprises to drive dramatic improvements in productivity, performance, flexibility, and responsiveness.

Marc Fiammante is a proven authority on the topic of SOA, having spent years leading field projects and delivering SOA and BPM adaptability. The insights he has accumulated in this book make it an extremely useful asset and reference point for enterprises looking to implement or gain greater value from their SOA and BPM strategy and to extend the capability to workers of all types, mobile or not.

—**Michael S. Liebow**
CEO, Dexterra

CHAPTER 1

From Simplified Integration to Dynamic Processes

> He that will not apply new remedies must expect new evils; for time is the greatest innovator.
>
> *Francis Bacon*

Common Pitfalls Limiting the Value of SOA and BPM

When Services Oriented Architecture (SOA) was initiated, the simplified integration capabilities brought hopes of a simplified business and IT landscape with reusable business components enabled by open technologies. A few years later, the results are uneven; some have experienced substantial benefits from a move to SOA, while others experienced more average value, even though they've used the appropriate technologies.

There are multiple reasons for not realizing the full business value of this services approach. Let's look at three essential reasons:

- **Many projects simply use advanced technologies to implement a client/server approach on Web Services**—Even though the protocol and technical adaptation is no longer a problem, they faced the same issues as a decade ago: changes to the server interfaces leading to client changes. My team faced such a scenario when a large bank decided to expose all its mainframe transactions as Web Services, but the operation semantic remained the same as for mainframe transactions. This pure bottom-up approach did not bring much business value, and in some cases, it led to worse performance.
- **Business processes reintroduce the tight coupling of flexible business services**—In many cases, Business Process Management (BPM) has been established as a disconnect approach from the SOA approach in which business services were limited to the exposure

of existing application functions. This model fails to address the modularization and variability of the business processes, and the processes act like glue rigidifying the services. For example, a large telecommunications operator implemented an end-to-end order management in a very large, single Business Process Execution Language (BPEL), including more than 300 activities, and ultimately faced a lack of reusability when some sequences could have been modularized and exposed as services.

- **Rigid information models are used to expose business services**—In many cases, services are only viewed as operations, forgetting that the business information structure they carry will vary as well with the evolution of the business. For example, another telecommunication operator was updating its product catalog weekly and generating a new schema at each change. This led to instability of the exposed services and integration processes, and delayed IT projects rather than enabling faster cycles.

Each of these experiences induced a deeper thinking process. What should be the essence of a business variable approach? How can we 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. Even though computing technology itself has had some approaches to variability in the past, such as overloading or rules engines, here we expose some solutions for which we can find an implementation analogy in the IT industry.

Whether it is in the automotive industry with shock absorbers and suspension, or the building industry with expansion joints, or the mechanical industry with washers, a mediation layer always absorbs the variations of one part without affecting the other part. In these industries, 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. Components can be subassemblies with their own internal variability using similar technologies, but the higher level assembler only cares about the external characteristics of the components. Similarly, in the IT approach, there must be a notion of the levels of assembly and the granularity of components.

In all industries, the final assembly is designed based on a global architecture that looks at requirements, operating conditions, and desired capabilities. Dynamicity is not a goal by itself; it has to respond to a specific context and desired states. Take for instance, buildings in Tokyo, which require more flexibility to withstand earthquakes than buildings in Paris. A 4x4 car has to absorb more landscape variations than a limo. 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 the variability necessary.

A bank we worked with had to implement higher controls on its loan origination processes because of the Sarbanes-Oxley requirements. However, this bank operates in four different countries with four different legislations and four different IT systems. In addition, this bank plans to expand to Eastern Europe in the future. This specific case led to a common process trunk with

variable subprocesses handling specific legislative constraints and a common top-down but variable services definition for account, products, customer information, and so on.

A Streamlined Enterprise Architecture for BPM and SOA

Every industry creates plans to address new markets, customers, or products. Similarly, such a business plan is required for an IT-enabled industry to identify potential evolutions and targeted domain boundaries. Such planning is essential to define which of the components will be tightly coupled as well as where and to what extent flexibility is necessary.

Merging the business strategy, implementation, information, and infrastructure aspects in a single enterprise architecture map is a common pitfall. Good architecture involves separation of concerns. Wikipedia provides a good definition of enterprise architecture that clearly separates the concerns:

> The architecture process addresses documenting and understanding the discrete enterprise structural components, typically within the following four categories: Business, Applications, Information, and Technology.[1]

In a dynamic enterprise approach using BPM and SOA, each domain requires a specific mapping and domain decomposition leading to a four-layer view of the enterprise with a simple mean to remember them.

There are recognized enterprise architecture frameworks or metamodels in the industry, such as The Open Group Architecture Framework (TOGAF)[2] and the Zachman Framework.[3] TOGAF is a method that addresses reification of the enterprise architecture (A to H) and the "why" (requirements), but does not separate as clearly as the Zachman Framework the "what," "how," "where," "who," and "when." The business information domain is covered by the "what," the business logic and functions performed are described by the "how," the various locations and topologies where the enterprise performs business are analyzed in the "where," the enterprise organization and the various actors are formalized in the "who," the events and dynamics of the business flows are modeled in the "when," and finally, the enterprise goals and requirement are captured in the "why."

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

The four domains—Business, Applications, Information, and Technology—are particularly important to an SOA and BPM approach as business processes consume business services exposed from applications or other business process modules. These business services apply functions on core business entities and information that allow their flexible integration at the enterprise level. The applications also consume technical services from the infrastructure layer. *Technical services* refer to services that do not pertain to a particular business domain but are common technical functions made available by the IT to the application layer. Examples of these

technical services are "send eMail," "audit interaction," and so on. Each of these four domains has policies that define the rules and regulations that apply to a particular domain.

Figure 1-1 shows these four domain layers creating a mnemonic structure (a capital "E"). The top layer addresses business, the middle layer the applications (including the service components that they package), the bottom layer the infrastructure, and all are connected vertically by information, or in other words, the business logic.

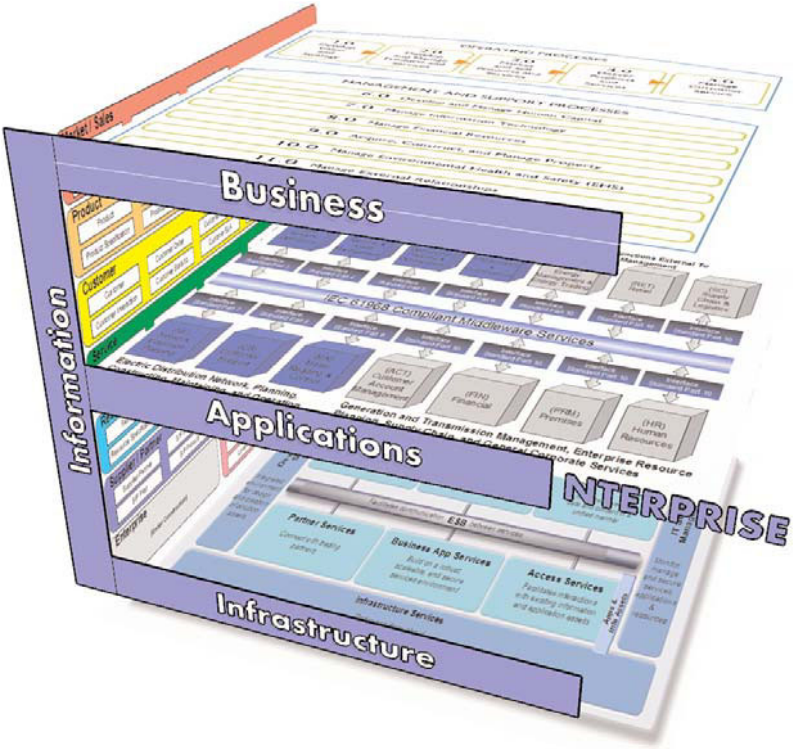


Figure 1-1 Mnemonic for the "E"nterprise architecture layers

In Chapter 2, "Streamlining the Enterprise Architecture for Dynamic BPM and SOA," we address techniques to achieve an efficient approach including horizon based enterprise architecture, enterprise architecture staged zooming, and accelerated implementations using template business and technical architectures.

But first, let's take a high-level view of the four enterprise architecture layers by looking at the business aspects.

Mapping the Enterprise Business

The use of an enterprise map in a dynamic BPM approach allows you to identify the boundaries of process modules that will have value for the enterprise. The detailed practice is discussed in

subsequent chapters of this book. In this chapter, we look at bridging the enterprise architecture with this business modularization.

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

The need to operate independently necessitates that a business component be at the intersection of several axes, which could potentially include

- A business area or a decomposition of a business 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 for defining a business map, the business components, and the business model. However, they all provide a decomposition of the business of an enterprise. A business process map is the static landscape on which business processes and business events provide dynamic behavior. A few standards are available, and in the next sections, we walk through some public examples of such enterprise maps and decompositions.

NGOSS Business Process Framework (eTOM)[4]

The enhanced Telecom Operations Map (eTOM) from the New Generation Operations Systems and Software (NGOSS) 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 of a functional decomposition tree. In Figure 1-2, you have a first level of decomposition with the verticals such as Fulfillment and an orthogonal first level of decomposition, Customer Relationship Management. At their intersection, there will be further decomposition in level 2 process catalogs such as Order Handling, which itself contains a further decomposition into a level 3 modularization.

This top-level map is complemented with a level 3 functional decomposition in the 300+ paged GB921D document, "The Business Process Framework (eTOM)."

Even though the standard says in business process decomposition no workflows are standardized, only a static functional decomposition of the business operations is provided. This is a good starting point, but the telecommunications business is highly dynamic, and static workflows are necessary for the modularization but insufficient for real business operations.

APQC Process Classification Framework[5]

The APQC (American Productivity and Quality Center, although now worldwide) is a standards body developed a classification of enterprise most common processes and captured that classification as a common language and open standard.

Particularly the classification looks at decomposition of the enterprise into finer grained elements and creating a tree of four levels, where each level addresses a particular semantic of the enterprise business (see Figure 1-3).

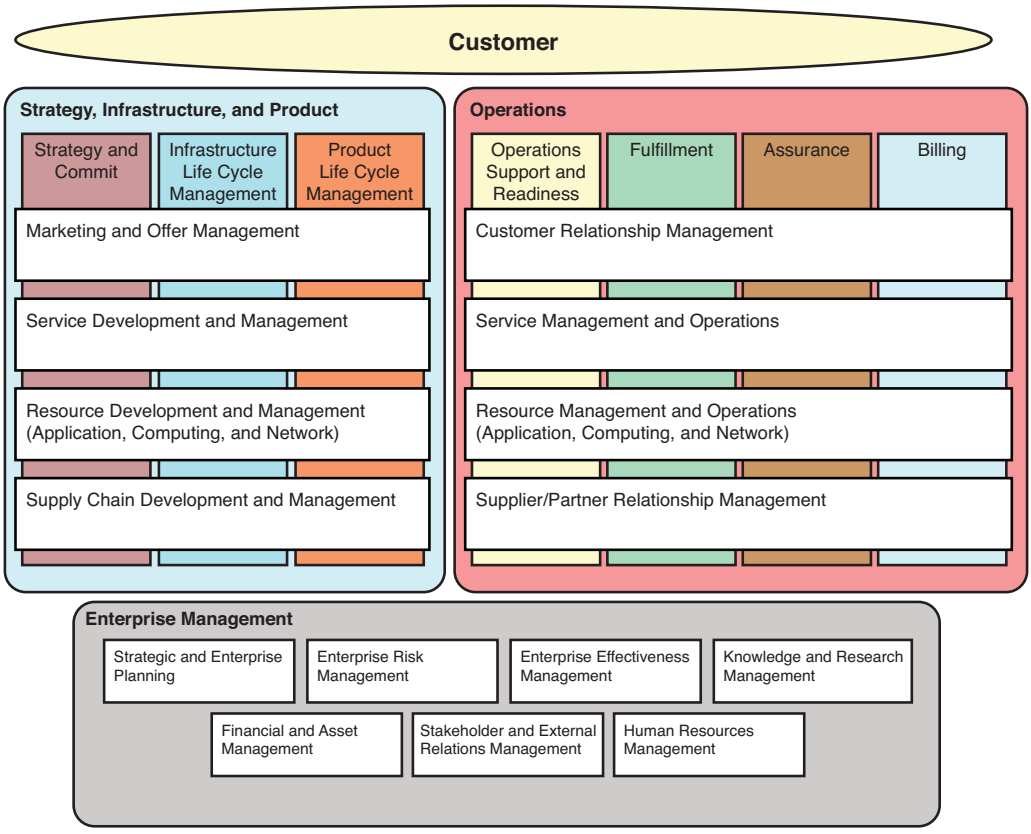


Figure 1-2 The Enhanced Telecom Operations Map (eTOM)

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

An essential aspect of the PCF is that it provides a standard for naming the different levels, which interestingly enough can be matched with the three eTOM levels. Similarly, even though 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.

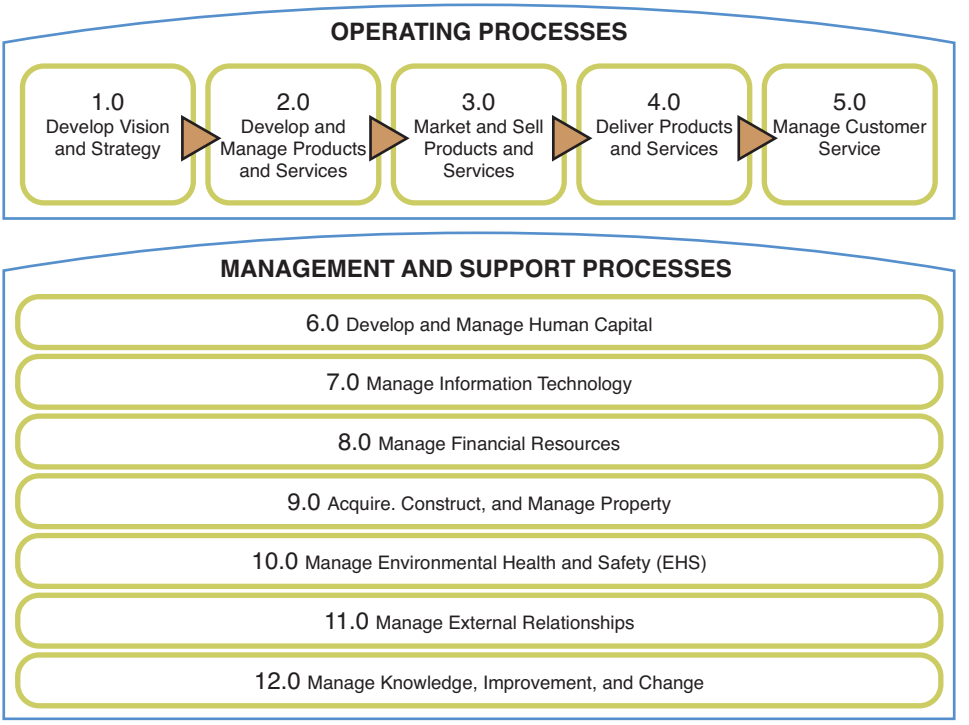


Figure 1-3 The PCF organizes processes into 12 enterprise-level categories.

This classification serves as the basis for further discussions in this book, particularly in the business service granularity discussion.

IBM®'s Component Business Modeling[6]

This approach from IBM is a new way of looking at an enterprise business. The Component Business Modeling (CBM) methodology identifies the basic building blocks of an enterprise business, as shown in Figure 1-4, to help address the future challenges of the enterprise.

This map is a mean to identify business components and create a business strategy for the enterprise. These business components have functionality that occurs once within the enterprise, and they act as business service centers that have the potential to operate independently. This enables the replacement of one business component by another implementation. For example, the product fulfillment can be delivered by many factories around the world, but the enterprise expects the same services from all product fulfillment instantiations around the globe. In one country or for one given product, the enterprise can own the factory while in another country, it can be a contractor. However, each business component delivers value to the enterprise through its ability to deliver a unique set of common services per business component function.

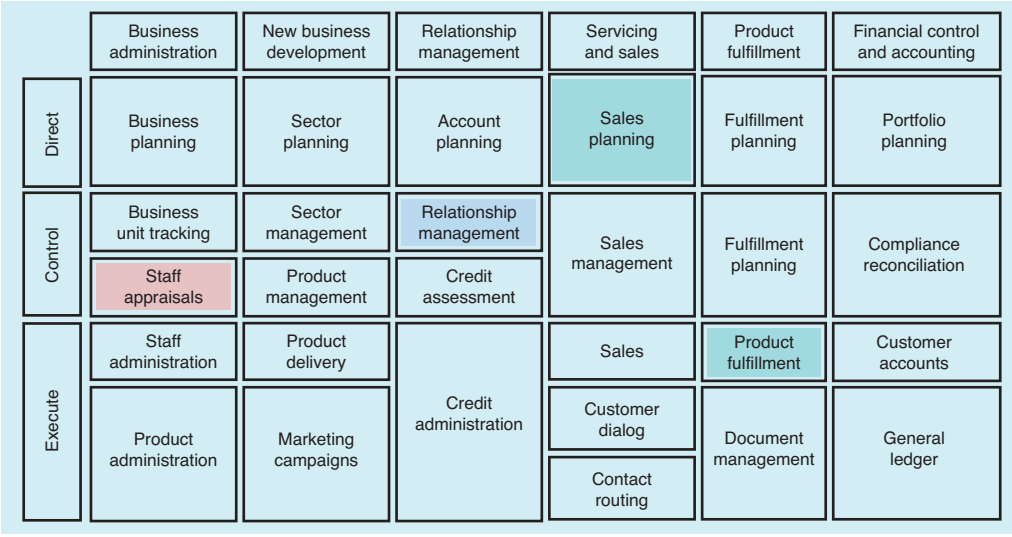


Figure 1-4 CBM map showing basic business building blocks

The implication is that explicit processes operate within the boundaries of a business component and use other business components' services to chain into the realization of processes that span the enterprise.

The granularity of business components is, however, often too high. In a dynamic process approach, these maps serve as the basis for further functional decomposition, and they can be considered as level 1 and level 2 of the APQC Process Classification Framework.

SCOR—The Supply-Chain Council

The Supply-Chain Council is a nonprofit global independent organization, composed of many companies around the world, which has produced the Supply-Chain Operations Reference (SCOR) model. This model is a process reference 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.

In addition, the Council defines four levels of decomposition but only addresses the first three levels in the standard. The fourth level is the business differentiator of each company.

These four levels are

1. **Process Type**—This level 1 or top level provides a balanced cross business domains and cross process categorization. It defines the scope and content of the previously mentioned management processes.
2. **Process Category**—This level 2 or configuration level provides a reconfigurable level that companies will adjust to their own business plans.

3. **Process Element**—This 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.
4. **Activities**—This 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 use 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.

As a confirmation that tree decompositions can be applied to process decomposition let me quote "A Coordination-Theoretic Approach to Understanding Process Differences" from MIT Sloan School of Management to show that derivation trees are common ways of designing processes for modularity and reusability:

> By applying top-down coordination-theoretic modeling, supported by a handbook of generic coordination mechanisms and exceptions handlers, we were able to create derivation trees for all three processes that made the source of their similarities and differences much easier to identify.[7]

Usually significant variations appear below the third level of decomposition in being consistent with APQC at the task level defined by its categorization.

Defining Service Granularity from Business Maps

My customers frequently ask, "What is a good service granularity?" Business maps and decompositions are a particularly powerful tool to identify a service granularity and, thus, answer this question. I regularly use them in customer meetings and receive excellent feedback.

However, to answer this question, first we need to precisely define a service. In the following discussion and throughout this book, we use these definitions:

- **Business service**—The grouping of repeatable business tasks that address the same functional domain, for example, Process Customer Order. Business policies are defined at this business service level, such as policies to handle various business services for corporate customers or individuals. The term we use is specifically looking at software components that deliver business logic and not business activities at the level of "sell cars" or "provide mobile telephone services."
- **Web Service**—A technical representation of the elements of a business service, grouping discrete business tasks together for technical management such as versioning in a technical descriptor using either Web Services Description Language (WSDL) or in a more abstract fashion Service Component Architecture (SCA) as technical standards. Looking at the Process Customer Order business service, the included Web Services might be Customer Order Life Cycle Management and Customer Order Information Management. There can be one or more Web Services per business service, but in many cases there will be only one.

- **Service operation**—A repeatable business task. In the previous case, a service operation might be Perform Order Feasibility Check, Submit Customer Order, 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. Given these definitions, let's 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. For example, 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 have around five to seven times 400, or 2,500 tasks.

  We see that this is already a very large number of tasks, and a large portion of these tasks can potentially 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, 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 operation has reusability value at the enterprise line of business levels.

The conclusions on granularity are

1. A good business service granularity, to maintain value and manageability, is between level 3 and level 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 as detailed in the section, "Mapping the Enterprise Applications," later in the chapter.
4. Any too fine-grained Application Programming Interface (API) at level 5 or below should be aggregated into a coarser grained service by applying variability on the interfaces and resolving the variability between the façade and the API.

Mapping the Enterprise Applications

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

At level 3 eTOM defines a Track and Manage Order Handling process, part of the Order Handling group. However this process requires accessing services from a Customer Information Management, a Product Catalog Management, a Billing Management, and a Customer Order Life Cycle Management application, which is implementing the services.

If the intention is to look at a future state architecture, the **ideal application map** must be defined. This is the map that an enterprise would create if there were no financial, time, or technology restrictions. These maps are often delivered by IT consulting companies and often referred to as "city planning." This application map is the superstructure 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 consume services from the real implementations in the existing applications. This top-down approach needs 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. The best practices for delivering such adaptations are discussed in later chapters of this book.

One standards body standardizes this application map, and it's no surprise the TeleManagement Forum (TmForum) is the most advanced in that space, as telecommunication operators are faced with exponential growth and dynamic market evolutions, requiring higher levels of standardization for lowering costs.

TmForum Telecom Applications Map[8]

The Telecom Applications Map provides the bridge between the NGOSS standardized business process and information framework building blocks (eTOM and the SID, Shared Information and Data model) 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, the map provided in Figure 1-5 presents a functional and information guideline for implementing a layer of reusable business services exposed by application.

Mapping the Enterprise IT Infrastructure

The business processes and applications require an IT and middleware infrastructure to operate. It is important that this infrastructure also provides support for variability and flexibility at a core technical level. Even though some standards like the Open Grid Services Infrastructure (OGSI)[9] are looking at standardizing services exposed from infrastructure components, there are no commonly accepted standards for middleware and operating systems services.

One approach to a product-agnostic map is the SOA Reference Model that IBM provides. This model, shown in Figure 1-6, serves as a reference point to position both middleware and functionality within a service-oriented architecture. For further details, see "IBM SOA Foundation: An Architectural Introduction and Overview."[10]

Mapping the Enterprise Information

Information is the essential "logical" asset that links together all layers of the enterprise. As processes do not run in vacuum but carry, transform, and refer to information, the structure of that

information reflects the way the enterprise wants to operate. Figure 1-7 shows how telecommunication operators relate business domains to core entity domains.

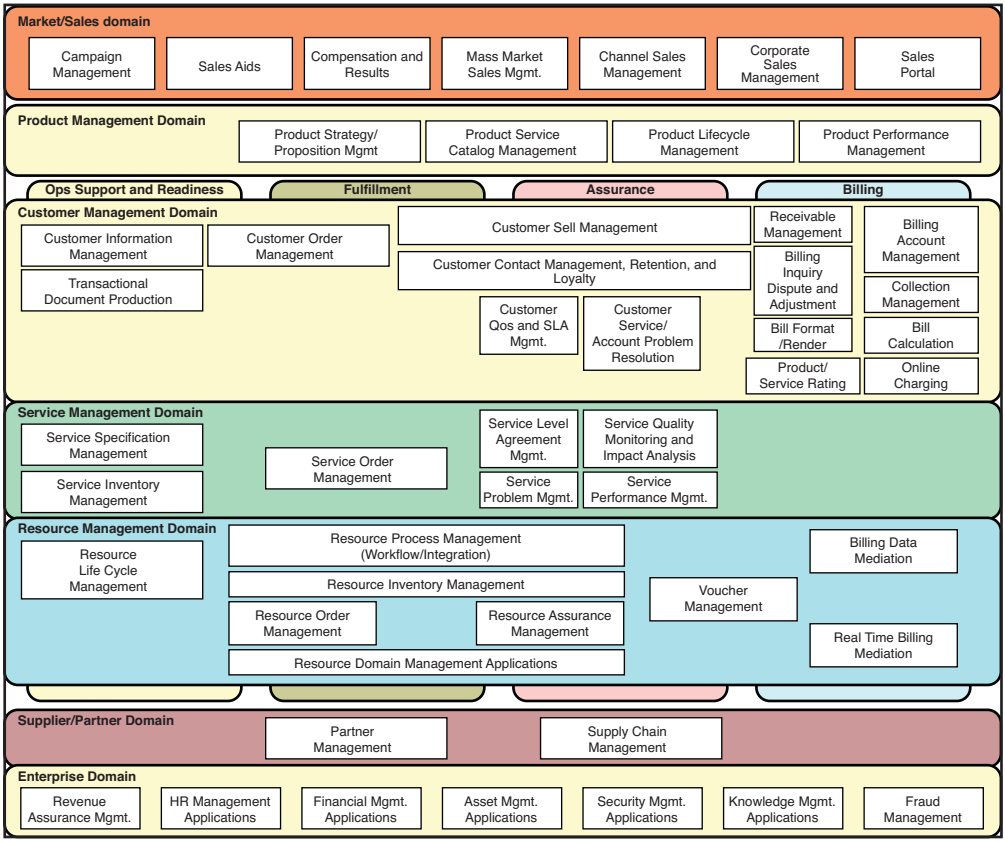


Figure 1-5 The TmForum Telecommunication Application Map GB929 R2.1

The business information model, domain maps, and core entities represent the business view of the information as opposed to data models, which address the technical implementation of managed objects data (see IETF RFC 3444[11]).

As an example of the influence of business on the information model, let's look at a retailer that used to sell single products and now wants to sell bundles. Should the bundle business object be considered a product with its own pricing superseding included products, or should it be another type of business entity with a global pricing deduced by applying discounts to the products included in the bundle? This is typically a business question that relies on the business model and needs to be reflected in the information model.

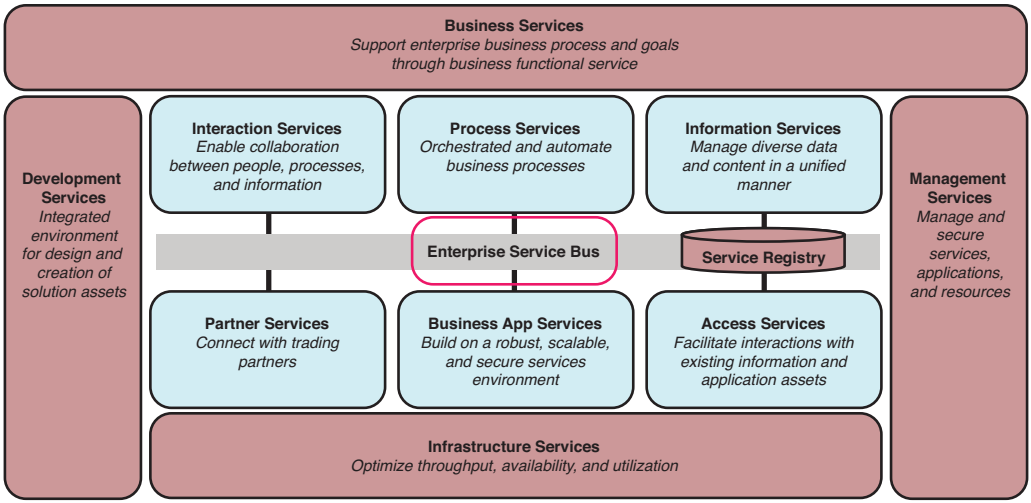


Figure 1-6 The IBM SOA Foundation Reference Model

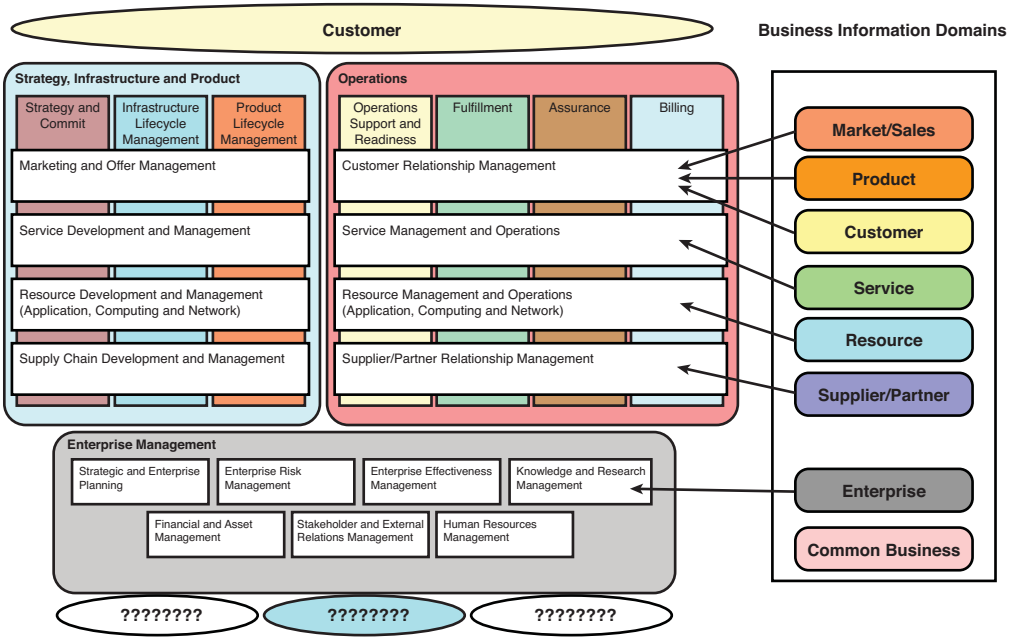


Figure 1-7 TeleManagement Forum eTOM and SID relationships

The Unified Modeling Language (UML) diagram in Figure 1-8 shows inheritance on the left or none on the right, a consequence of the way the enterprise intends to address markets.

A question arises: Can we make the model flexible enough to address both cases and allow variations of the business model in the future? We address approaches to a realization of this variability in Chapter 3, "Implementing Dynamic Enterprise Information."

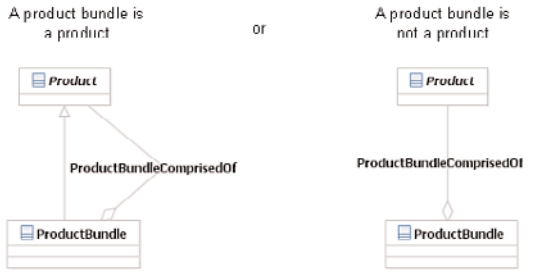


Figure 1-8 Business model affecting the information model

Basic Principles for Enterprise Dynamicity

Now that we have set the static scene for the enterprise architecture, how do we use all these elements in a way that allows a dynamic business and IT approach? Dynamicity implies flows and movements, with events and messages propagating through the enterprise and variations between and within the business components. Flows, such as the 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 event's 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 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, we use the categorization of basic types of processes derived from the Business Process Modeling Notation (BPMN) standard.

Categorizing the Processes

The BPMN standard introduces three basic types of submodels within an end-to-end BPMN model:[12]

1. **Collaboration (global) processes**—Business to business between enterprises or organizations
2. **Abstract (public) processes**—Inside an enterprise across business components or internal organizations
3. **Private (internal) business processes**—Within business components

In addition, BPMN introduces the notions of **pools** and **lanes**, which are also helpful in decomposing processes in modules and grains, allowing them to 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. Cross-industry standards, such as ebXML BPSS, RosettaNet, or to some extent, industry-specific standards, such as ACORD, describe such interactions sequences. The

example collaboration process in Figure 1-9 shows a sequence of purchase order processing between two companies.

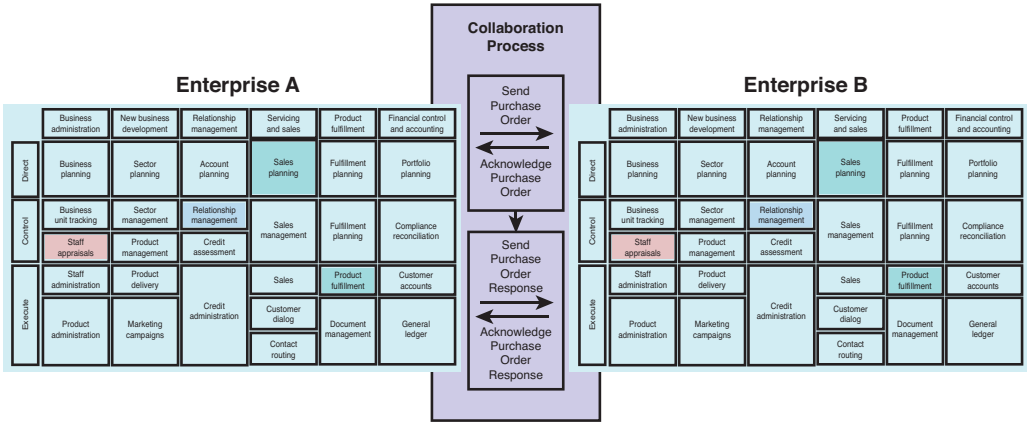


Figure 1-9 Example collaboration process

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. EDI VANs are an early form of these collaboration processes, which are progressively disappearing because they inhibited flexibility.

Abstract (Public) Processes

These processes represent the overall interactions between different internal organizations in the enterprise owning their internal business rules or sequences. Consequently, 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 as defined previously, but looking at the APQC decomposition, these abstract processes also occur between the processes at level 3 of the Process Classification Framework. Just as for the collaboration processes, there cannot be an 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. The overall sequence is the result of services calls and events occurring between each of the level 3 processes or business components. 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 Figure 1-10, the "sales to bill" abstract process is represented by the interactions between the business components.

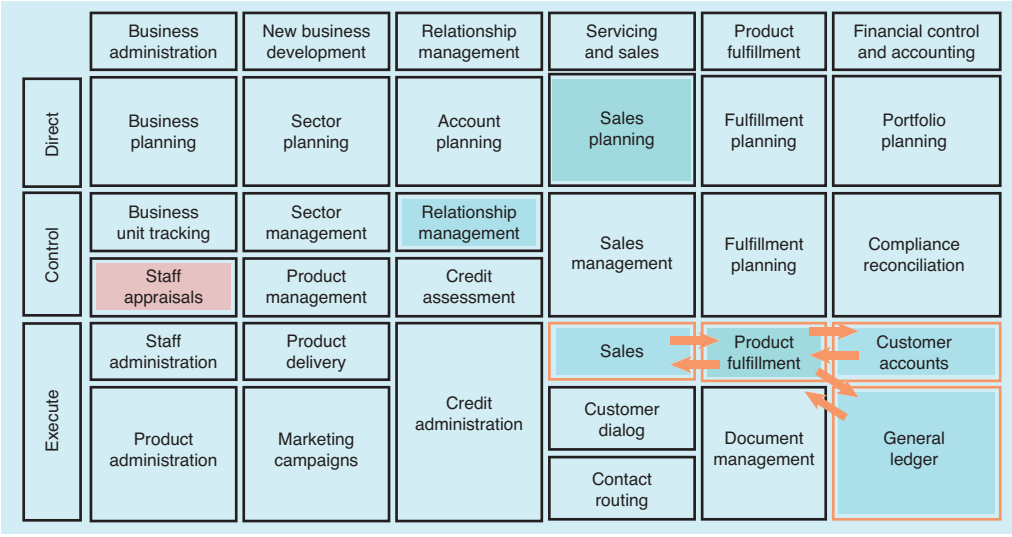


Figure 1-10 Abstract sales to bill process

Private (Internal) Business Processes

Within a single organization with a single identified business owner, the owner can manage the rules and sequences of tasks, which are then private to that "owner" or his delegates.

As a result, private processes are preferable for explicit flow control and automation because they are manageable assets of the enterprise, for which stake holders and life cycle can define.

They can be either explicit, if expressed in a workflow language or service choreography/orchestration language such as XML Process Definition Language (XPDL) or Web Services Business Process Execution Language (BPEL), or implicit, if resulting from the code running inside a particular application or package such as the ones provided by companies like SAP, Oracle, Amdocs, and so on. These processes chain to other private processes either by the means, events, or services calls. If they are exposed as services and consume services, they then can become components in an assembly model, such as standardized by the Service Component Architecture standard from the Open SOA organization.[13] These private processes match the APQC processes that are at decomposition level 3. 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. Figure 1-11 shows an SCA assembly diagram where the customer order process (ProcessCustomerOrder) is exposed as a service and consumes other services, one of which is another process validating the customer order.

This private process is exposed from within the eTOM level 3 as shown in Figure 1-12.

As you can see the service component in Figure 1-11 has more interfaces than the level 3 of the process decomposition in Figure 1-12, highlighting the difference between a component

providing an implementation in the application map and a private process identified from the process decomposition.

Figure 1-11 Private process exposed as a component and consuming other processes

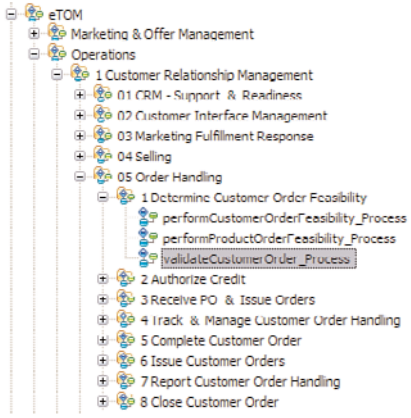


Figure 1-12 Private process in eTOM level 3

Pools/Lanes

Within a private process, a **lane** represents a sequence of tasks or activities driven by a specific participant. A participant can be any specific party role such as a buyer, seller, or call center operator, but also it can be a technical participation such a bus mediation or a program routine. The identification of a lane's nature has an impact on the selection of the appropriate technologies for implementing particular portions, services, or a private process. **Pools** are group of lanes and seem to imply multiple actors. However, this is not explicitly stated in the BPMN standard.

Applying Decomposition to End-to-End Process

As a consequence of decomposing processes in categories, an end-to-end process is the abstract or collaboration process resulting from the assembly of private processes that group the lanes that interact within their boundaries.

Here is an example or "order to cash" end-to-end abstract process for telecommunication operators, composed of a chain of private processes contained within the boundaries of a business component.

In Figure 1-13, we identify variations of private processes, such as the various service configurations and resource provisioning. This example also looks at private processes that are common successors in an abstract flow, such as the Set Top Box configuration required by the three services of a triple play—that is, Internet, television, and voice-over-IP calling services.

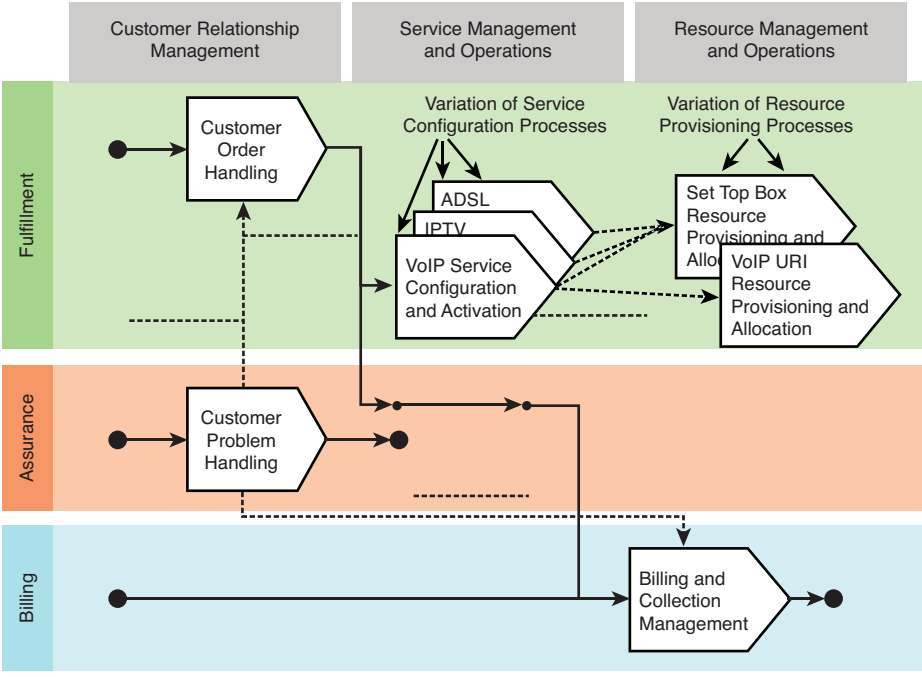


Figure 1-13 Abstract process of chaining private processes on a business map

In this specific case, the services can be submitServiceOrder with three implementations: submitServiceOrderForVoIP, submitServiceOrderForIPTV, and submitServiceOrderForADSL.

In a dynamic process approach, we must be able to add a quadruple play service such as gaming, with its specific service configuration and resources, without having to regenerate and test the "end to end" 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 variations of the 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 a processes that carries explicitly all the information to processes that refer to information preserved elsewhere. Then, we only pass the core of the decisional information through processes, 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.

This still does not solve the variation of the interface to that information if attributes are added. An enterprise information variability approach is the necessary complement to ensure that within acceptable limits, the information model characteristics can change without affecting the interfaces.

The Enterprise Expansion Joint

This process and information dynamicity requires flexibility, and a mediation layer between components is one of the elements that provides such flexibility. The initial solution to the mediation layer in IT and a services-oriented architecture has been the Enterprise Services Bus. However, these buses often address the technical variability, looking at protocols and messaging, leaving the content interpretation to the integrated consumers or providers.

There is a need to push the mediation layer capabilities to content and business semantic adaptation, thus removing the interface's tight coupling of consumers and providers. As a simple example, the IT solution should enable the evolution of the versions of provided business services without requiring an immediate regeneration of business service consumers.

All the additional capabilities lead 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 precisely identified limits that should be identified during the enterprise architecture phase. The enterprise architecture modeling method must then address the sizing of the business variability that this expansion layer can absorb, looking at foreseeable future business evolutions.

I had a concrete case where an enterprise wanted to deliver a new customer application and process while planning to change billing systems in a later phase. The target was to allow this change of billing systems for another packaged application without affecting any of the elements of the new customer care environment. So, the approach was to define what business services a billing system should expose to customer care in a top-down fashion and then use the mediation layer to perform the adaptation between exposed APIs and desired granularity and variability. In this specific case, the existing billing system was Tuxedo-based and required to use units of work in C for consistency, which led us to create a specific adapter, delivered in less than one month using the available adapter frameworks.

In addition to the granularity adaptation, the mediation layer may also have to address state as the reusable business services are designed to be independent of any implementation. Thus it may not expose specific data that is necessary for a specific application or package realizing the service.

In Chapter 6, "Implementing the Enterprise Expansion Joint," we discuss a set of techniques for realizing such a layer.

Summary

In this chapter, we saw multiple aspects for an enterprise to address to realize the value of a truly dynamic BPM and SOA approach. This realization 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. There is no "one size fits all" solution; however, a set of modeling steps and implementation techniques are part of an enterprise architecture approach and are discussed in Chapter 2.

Index

A

ABOs (Adaptive Business Objects), 62
abstract processes, BPMN, 14-15, 32
abstract processes dynamic binding approach, 159
ACORD (Association for Cooperative Operations Research and Development), 127
activities, WS-BPEL, 140
actors, defined, 29
ad-hoc polymorphism
operation patterns (WSDL), variability, 66-67
adaptation, dynamic adaptation (ESBs), 101
adapter pattern, 63
Adaptive Business Objects (ABOs), 62
adding business semantics in bus meditations, 105
AggregateControl, 111
AggregateReply, 111
AggregateRequest, 111
aggregation
WebSphere DataPower, 111-112
WebSphere ESB, 110-111
WebSphere Message Broker, 111
agility
enterprise business agility, 24-25
enterprise technical agility, 22-23
allocating use cases, into process tree decomposition, 81-86
American Productivity and Quality Center. *See* APQC (American Productivity and Quality Center), 5
annotating code for service exposure
SCA (Service Component Architecture), 73-74
"any," variability, 52
application components, as private processes, 86-89
application layers, modeling, 127-131
application management layer, operational management, 162
Applications layer (enterprise architecture), 10-11
TmForum Telecom Applications Map, 11

APQC (American Productivity and Quality Center), 5
Business layer (enterprise architecture), 5-7
architecture life cycles, 121
business context, 124-125
change cases, 123-124
tooling, 121-122
Arrangement, Data Concepts, 45
assertions, 104
Association for Cooperative Operations Research and Development (ACORD), 127
audit trails, managing, 171
authentication, 72
avoiding round-tripping problems, 145

B

bindings, intergrating with components (SCA), 71
blind replay navigation approach, 159
BPEL (Business Process Execution Language), 2
BPM (Business Process Management), 1
managing, 149-151
pitfalls limiting value of, 1-2
BPMN (Business Process Management Notation), 14, 126, 138
abstract (public) processes, 14-15
abstract processes, 32
collaboration (global) processes, 14-15
lanes, 17
pools, 17
private (internal) business processes, 14-17
to WS-BPEL implementation, 142-146
bridge pattern, 64
BTT (Branch Transformation Toolkit), 107
business analysis, variability, 44
business mapping, 44-49
core entity identification, 44-49
business competencies, CBM, 29
business components, CBM, 29
business context, 124
architecture life cycles, 124-125
business dashboards, dynamic business process management, 168
Business Direction Item, Data Concepts, 45
business horizon, 26
business information changes, impact on processes, 19
business information sharing, business information sharing, 50-52
Business layer (enterprise architecture), 4-5
APQC, 5-7
CBM, 7-8
eTOM, 5
SCOR, 8-9
service granularity, 9-10
business layers, modeling, 127-131
business mapping, variability, 44-49
business monitoring levels, dynamic business process management, 166-168
business owners, defined, 29
business ownership, 29
Business Process Execution Language (BPEL), 2
business process life cycle management, 158-160
Business Process Management (BPM), 1, 25
managing, 149-151
pitfalls limiting value of, 1-2
Business Process Management Notation (BPMN), 14, 126, 138
abstract (public) processes, 14-15

abstract processes, 32
collaboration (global) processes, 14-15
lanes, 17
pools, 17
private (internal) business processes, 14-17
to WS-BPEL implementation, 142-146
business process modeling, 134-138
business processes
event driven business processes, 95-96
extracting routing logic from, 92-93
limiting information model impact, 94-95
operational management, 164-165
spanning tree branches, 82-84
subprocesses, 84-86
business rule engines, variability, 89-92
business semantics, adding in bus meditation, 105
business service life cycle management, 156-158
business services, 9
business agility requirements, 24-25

C

capabilities
enterprise business agility, 24
Business Process Management, 25
business services, 24-25
enterprise technical agility, 22
integration, 22-23
software components, 23
capturing enterprise architecture models, tooling, 125-127
cascading model view controller, 118
CBM (Component Business Modeling), 7
business competencies, 29
business components, 29
Business layer (enterprise architecture), 7-8
change action selection, impact analysis (life cycle management), 155
change cases
architecture life cycles, 123-124
drivers for, 123
Characteristic Value, 55
CICS (Customer Information Control System), 23
CICS Service Flow Feature, 114
CICS service flow modeler
managing granularity from mainframe, 113-115
CIM (Common Information Model), 127
Classification, Data Concepts, 45
collaboration (global) processes, BPMN, 14-15
Common Business Entities Domain, SID, 46
Common Information Model (CIM), 127
Common Object Request Broker Architecture (CORBA), 63
communities of interest, 43
Component Business Modeling (CBM), 7
business competencies, 29
business components, 29
Business layer (enterprise architecture), 7-8
component concrete, SCA (Service Component Architecture), 71-72
component interfaces, SCA (Service Component Architecture), 70
components, 68
SCA (Service Component Architecture), 69-70
controlling behavior with policies, 72-73

composites, SCA (Service Component Architecture), 69-70
controlling behavior with policies, 72-73
Condition, Data Concepts, 45
confidentiality, 72
context area based integration
split model techniques, 117-118
context tree
split model techniques, 117-118
control policies, 150
controlling component and composite behavior with policies (SCA), 72-73
CORBA (Common Object Request Broker Architecture), 63
core entities, business information sharing (variability), 50-52
core entity identification, variability, 44-49
credentials, propagating, 170-171
cross-referencing, 58-59
CSR (Customer Sales Representative), 80
Customer Domain, SID, 46
Customer Information Control System (CICS), 23
Customer Sales Representative (CSR), 80

D

Data Concepts, 44-45
Financial Service Data Model, 45
Telemanagement Forum Telecommunication Industry Shared Information, 46
Data Definition Language (DDL), 151
data persistence, micro-flows, 113
data sharing, 52
databases, state preservation, 106
DDL (Data Definition Language), 151
decomposition
applying to end-to-end processes, 17-18
end-to-end process, 166
levels of, 126
direct deployment, enterprise architecture tooling, 125-127
DPL (distributed program link), 114
drivers for change cases, 123
DRM (Data Reference Model), 42
dynamic adaptation, ESBs, 101
dynamic binding, business process modeling, 136
dynamic business process management, 166
business monitoring levels, 166-168
implementing business dashboards, 168
dynamic endpoint selection, 103
dynamic patterns, staged composition, 31-32
dynamic routing
creating routing context and adding business semantics in bus, 105
node-based routing, 102-103
policy driven dynamic assembly, 103-104
dynamicity, 14
applying decomposition to end-to-end processes, 17-18
BPMN, 14
abstract (public) processes, 14-15
collaboration (global) processes, 14-15
pools/lanes, 17
private (internal) business processes, 14-17

enterprise expansion joints, 19
impact of business information changes to processes, 19
modeling processes for, 80
allocating use cases into process tree decomposition, 81-86
private processes as application components, 86-89

E

EAI (Enterprise Application Integration), 115
EII (Enterprise Information Integration), 28
EJB (Enterprise Java Beans), 107
end-to-end processes, 14
applying decomposition to, 17-18
Endpoint Lookup meditation, 103
endpoints, dynamic endpoint selection, 103
enhanced Telecom Operations Map (eTOM), 5
Enterprise Application Integration (EAI), 115
enterprise applications, mapping, 10-11
TmForum Telecom Applications Map, 11
enterprise architecture, 3
Applications layer, 10-11
TmForum Telecom Applications Map, 11
Business layer, 4-5
APQC, 5-7
CBM, 7-8
eTOM, 5
SCOR, 8-9
service granularity, 9-10
defined, 3
focusing on variability, 21-22
horizon approach, 26
business horizon, 26
information horizon, 27-28
information, mapping, 11-13
infrastructure, mapping, 11
staged zooming, 29-31
dynamic patterns, 31-32
enterprise push patterns, 32-36
single actor pull patterns, 36-39
streamlined, 3-4
enterprise business agility, 24
Business Process Management, 25
business services, 24-25
enterprise expansion joints, 19
focusing on, 21
integration, 99-102
enterprise information, mapping, 11-13
Enterprise Information Integration (EII), 28
enterprise IT infrastructure, mapping, 11
Enterprise Java Beans (EJB), 107
enterprise push patterns, 31-33
multiple actor interactions with single owner business, 34-36
enterprise technical agility, 22
integration, 22-23
software components, 23
ESBs (Enterprise Services Bus), 99-100
aggregation, 110-111
dynamic adaptation, 101
eTOM (enhanced Telecom Operations Map), 127
Business layer (enterprise architecture), 5
Event, Data Concepts, 45

event driven business processes, 95-96
exchange payload, 52
eXtensible Business Reporting Language (XBRL), 57
extracting routing logic from business processes, 92-93

F

façade pattern, 64
facades, 83
Fan Out/Fan In mediation, 110
aggregation
WebSphere DataPower, 111-112
WebSphere ESB, 110-111
WebSphere Message Broker, 111
Financial Service Data Model, 44
fleet management, 47
flexibility, 56-57
cross-referencing, 58-59
keys, managing, 58-59
loose coupled keys, information models for modularity, 57-58
flexible integration, enterprise expansion joint, 99-102
flow nodes, 101
focusing, enterprise architecture on variability, 21-22

G

Gamma, Erich, 64
Gang of Four (GoF), 64
granularity, managing from mainframe with CICS Service Flow Modeler, 113-115
granularity adaptation
managing transactional units of works, 118-119
propagating faults, 119
single actor pull patterns, 38-39

H

Helm, Richard, 64
horizon approach, to enterprise architecture, 26
business horizon, 26
information horizon, 27-28
human screen navigation, 36

I

IaaS (Information as a Service), 37
IANA (Internet Assigned Numbers Authority), 75
IBM, CBM (Component Business Modeling), 7
IBM Business Services Dynamic Assembler, 104
IBM Master Data Management, 59
IBM WebSphere Multichannel Branch Transformation Toolkit (BTT), 107
ideal application map, 11
IDL (interface definition language), 70
impact analysis, life cycle management, 153-155
change action selection, 155
impact of business information changes to processes, 19
implementing business dashboards, 168
implicit process definition, 159
import models, service information model (life cycle management), 151-152
information architecture, variability, 43
Information as a Service (IaaS), 37
information changes, 41
information horizon, 27-28
Information layer (enterprise architecture), 11-13

information model flexibility, 56-57
loose coupled keys, 57-58
information model impact, limiting on business processes, 94-95
information modeling, 132-133
information models for modularity, loose coupled keys, 57-58
information variability. ***See also*** **variability**
metamodels, 59-60
infrastructure layers, modeling, 131
integrate models, service information model (life cycle management), 151-152
integrating components with bindings (SCA), 71
integration
enterprise expansion joint, 99-102
enterprise technical agility, 22-23
integrity, 73
interface definition language (IDL), 70
Internet Assigned Numbers Authority (IANA), 75
Involved Party, Data Concepts, 45
IT services, 22

J

J2EE, work area service (state preservation), 108-109
Johnson, Ralph, 64
JSON, REST payload information, 76-77

K

key-value-mode triplets, 108
keys, 56
loose coupled keys, information models for modularity, 57-58
managing, 58-59

L

labels, RDF, 49
lanes, BPMN, 17
Liberty Alliance Federation Framework, 170
life cycle management
business processes, 158-160
business service, 156-158
impact analysis, 153-155
change action selection, 155
model comparisons, 152-153
service information model, 151
life cycles, architecture life cycles, 121
business context, 124-125
change cases, 123-124
tooling, 121-122
limiting information model impact on business processes, 94-95
Location, Data Concepts, 45
loose coupled keys, information models for modularity, 57-58

M

mainframe with CICS Service Flow Modeler, managing granularity, 113-115
management
dynamic business process management, 166
business monitoring levels, 166-168
implementing business dashboards, 168
life cycle management
business processes, 158-160
business service, 156-158
impact analysis, 153-155
model comparisons, 152-153
service information model, 151
operational management, 160
application management layer, 162

business processes, 164-165
service management layer, 162-164
technology management layer, 161
management policies, 150
management processes, SCOR, 8
managing
audit trails, 171
BPM, 149-151
granularity, from mainframe with CICS Service Flow Modeler, 113-115
SOA, 149-151
transactional units of works in granularity adaptation, 118-119
managing state, 106
databases, 106
MQ message queuing, 106
structured hierarchical contexts, 106-107
work area service for J2EE, 108-109
mapping
enterprise applications, 10-11
TmForum Telecom Applications Map, 11
enterprise business, 5
APQC, 5-7
CBM, 7-8
eTOM, 5
SCOR, 8-9
service granularity, 9-10
enterprise information, 11-13
enterprise IT infrastructure, 11
objects in WebSphere Business Modeler, 143
Market domain, SID, 46
mediation, Fan Out/Fan In, 110
WebSphere DataPower for aggregation, 111-112
WebSphere ESB for aggregation, 110-111
WebSphere Message Broker for aggregation, 111
mediation flows, 101-102
dynamic routing, 102
creating routing context and adding business semantics, 105
node based routing, 103
policy driven dynamic assembly, 103-104
managing state, 106
databases, 106
message queuing (MQ), 106
structured hierarchical contexts, 106-107
work area service for J2EE, 108-109
mediations, 101
message flows, 102
message queuing (MQ), 106
messaging-oriented middleware (MOM), 111
metamodels, 125
information variability, 59-60
tooling, 126
micro-flows, WS-BPEL, 112-113
model comparision, life cycle management, 152-153
modeling
application layers, 127-131
business layers, 127-131
business process modeling, 134-138
information modeling, 132-133
infrastructure layers, 131
processes for dynamicity, 80
allocating use cases into process tree decomposition, 81-86
private processes as application components, 86-89
service modeling, 134
services layers, 127-131
models, 115
cascading model view controllers, 118

single pivot object oriented information model, 116-117
modularity, 68
module decomposition with a late binding approach, 159
MOM (messaging-oriented middleware), 111
MQ (message queuing), state preservation, 106
MTOSI (Multi-Technology Operations System Interface), 96
Multi-Technology Operations System Interface (MTOSI), 96
multiple ownership, 29

N

name/value pairs, variability, 54
structured approaches, 54-55
NGOSS (New Generation Operations Systems and Software), 5
node-based routing, 102-103
notification, WSDL operation patterns, 65

O

OAGIS (Open Applications Group Interface Specification), 52
name/value pairs, 54
objects
mapping in WebSphere Business Modeler, 143
variability, 52
OGSI (Open Grid Services Infrastructure), 11
one-way, WSDL operation patterns, 64
ontologies, 42
variability, 50
Open Applications Group Interface Specification (OAGIS), 52
Open Grid Services Infrastructure (OGSI), 11
operation patterns, WSDL, 64-65
operational management, 160
application management layer, 162
business processes, 164-165
service management layer, 162-164
technology management layer, 161
OSS/J (Operations Support System for Java), 152
OWL (Web Ontology Language), 34, 50

P

pairs
simple single pairs, variability, 54
structured approaches, variability, 54-55
participants, defined, 29
Partners, SID, 46
patterns
adapter pattern, 63
bridge pattern, 64
dynamic patterns, staged zooming, 31-32
enterprise push patterns, 31-33
multiple actor interactions with single owner business, 34-36
façade pattern, 64
operation patterns (WSDL), variability, 64-65
single actor pull patterns, 32, 36-38
granularity adaptation, 38-39
staged zooming, 36-39
payload information, REST (handling with JSON), 76-77
payload level services
security, 171-172
payloads, variability (any), 52

PCF (Process Classification Framework), 6, 28
policies, 104
controlling component and composite behavior, SCA, 72-73
Return All Matching Endpoints policy, 158
policy driven dynamic assembly, dynamic routing, 103-104
polymorphism, WSDL, 66-67
pools, BPMN, 17
private (internal) business processes, BPMN, 14-17
private processes, as application components, 86-89
process choreography integration, WS-BPEL, 138-140, 142
Process Classification Framework (PCF), 6, 28
process model decomposition, 9
process tree decomposition, allocating use cases, 81-86
processes
business processes, 92
modeling for dynamicity, 80
allocating use cases into process tree decomposition, 81-86
private processes as application components, 86-89
private processes, 86-89
security, 170-171
testing, 146-147
Product, Data Concepts, 45
Product Domain, SID, 46
propagating credentials, 170-171
propagating faults from granularity adaptation, 119
propagation blocking, 57

R

Ram Control, 26
Rational Data Architect, 127
Rational Software Architect (RSA), 152
Rational Software Architect for WebSphere, 122, 127
RDF (Resource Description Framework), 49
refactoring, 159
regulations, 123
remote procedure call (RPC), 96
REpresentational State Transfer (REST), 74
payload information, handling with JSON, 76-77
resource structures, 75-76
services, defining, 74-75
versus Web Services, 75
request-response, WSDL operation patterns, 64
Requirements Explorer window, 122
Resource Description Framework (RDF), 49
Resource Domain, SID, 46
Resource Item, Data Concepts, 45
resource structures, REST, 75-76
REST (REpresentational State Transfer), 74
Return All Matching Endpoints policy, 158
RIF (Rule Interchange Format), 90
round-tripping problems, avoiding, 145
routing, dynamic routing, 102
creating routing context and adding business semantics in bus, 105
node-based routing, 102-103
policy driven dynamic assembly, 103-104
routing context, creating, 105
routing logic, extracting from business processes, 92-93
RPC (remote procedure call), 96
RSA (Rational Software Architect), 152

Rule Interchange Format (RIF), 90
rules actions, 101
rules flows, 102

S

Sales domain, SID, 46
SAML (Security Assertion Markup Language), 170
SBVR (Semantics of Business Vocabulary and Business Rules), 90
SCA (Service Component Architecture), 9, 66-68
annotating code for service exposure, 73-74
component interfaces, 70
components and composites, 69-70
controlling behavior with policies, 72-73
integrating components with bindings, 71
making component concrete with, 71-72
security, 72
specifications, 68
tools, 73
SCDL (Service Component Description Language), 69-70
SCOR (Supply Chain Operations Reference), 8
Business layer (enterprise architecture), 8-9
management processes, 8
SCXML (State Chart XML), 62
SDO (Service Data Objects), 60
security
payload level services, 171-172
processes, 170-171
SCA (Service Component Architecture), 72
services, 169-170
Security Assertion Markup Language (SAML), 170
Semantics of Business Vocabulary and Business Rules (SBVR), 90
Service Component Architecture. *See* SCA (Service Component Architecture)
Service Component Description Language (SCDL), 69-70
Service Data Objects (SDO), 60
Service Domain, SID, 46
service exposure, annotating code for (SCA), 73-74
Service Flow Modeler, 113-115
service granularity, Business layer (enterprise architecture), 9-10
service information model, life cycle management, 151
service interaction activities, WS-BPEL, 140
service management layer, operational management, 162-164
service modeling, 134
service operations, 10
Service Oriented Architecture. *See* SOA (Service Oriented Architecture)
services
business services
business agility requirements, 24-25
defined, 9
defining using REST, 74-75
IT services, 22
payload level services, security, 171-172
security, 169-170
service operations, 10
software services (SOA), 22
testing, 146-147
Web Services, 9
services layers, modeling, 127-131
SID (Shared Information and Data), 46, 127
variability, structured approaches, 54-55
single actor pull patterns, 32, 36-38
granularity adaptation, 38-39
staged composition, 37

staged zooming, 36-38
granularity adaptation, 38-39
single pivot object oriented information model, 116-117
SOA (Service Oriented Architecture), 1
managing, 149-151
pitfalls limiting value of, 1-2
software services, 22
SOA Reference Model, 11
software components, enterprise technical agility, 23
software services (SOA), 22
solicit-response, WSDL operation patterns, 65
split model techniques, 115
cascading model view controllers, 118
context area based integration, 117-118
context tree, 117-118
single pivot object oriented information model, 116-117
staged zooming, 29-31
dynamic patterns, 31-32
enterprise push patterns, 32-33
multiple actor interactions with single owner, 34-36
single actor pull patterns, 36-38
granularity adaptation, 38-39
State Chart XML (SCXML), 62
state preservation, 106
databases, 106
message queuing (MQ), 106
structured hierarchical contexts, 106-107
work area service for J2EE, 108-109
streamlined enterprise architecture, 3-4
structured activities, WS-BPEL, 140
structured contexts, state preservation, 106-107
subprocesses, business processes, 84-86
Suppliers, SID, 46
Supply Chain Operations Reference (SCOR), 8-9
Supply-Chain Council, 8

T

taxonomies, 42
variability, 49-50
technical services, 3
technology management layer, operational management, 161
Telco TMF, standard SID model UML package structure, 133
Telecom Applications Map (TmForum), 11
TeleManagement Forum, 152
TeleManagement Forum Telecommunication Industry Shared Information and Data, 46
TeleManagement Forum's Multi-Technolgy Operations System Interface (TMF MTOSI), 143
TeleManagement Forum (TmForum), 11
testing, 146-147
The Open Group Architecture Framework (TOGAF), 3
TMF MTOSI (TeleManagement Forum's Multi-Technology Operations System Interface), 143
TmForum (TeleManagement Forum), 11
TOGAF (The Open Group Architecture Framework), 3
tooling
architecture life cycles, 121-122
capturing enterprise architecture models, 125-127
metamodel, 126
tools, SCA, 73
traceability, 122
trademarks, 41
transactional units of works, managing with granularity adaptation, 118-119

U

U.S. Federal Enterprise Architecture Data Reference Model (DRM), 42
UML (Unified Modeling Language), 12, 47, 50
import features, 152
information modeling, 133
service modeling, 134
UML packages, service modeling, 134
UML packaging, 124
URIs (Uniform Resource Identifiers), 74
URLs (Universal Resource Locators), 57
URN, 57
use cases, 122
allocating into process tree decomposition, 81-86
utility activities
WS-BPEL, 140

V

variability, 42-43, 64. *See also* information variability
"any," 52
business analysis, 44
business mapping, 44-49
core entity identification, 44-45, 47-49
business information sharing, 50-52
business rule engines, 89-92
extracting routing logic from business processes, 92-93
focusing on, 22
information architecture, 43
metamodel approaches, 59-60
object, 52
ontologies, 50
REST. *See* REST
simple single pairs, 54
structured approaches, SID, 54-55
taxonomies, 49-50
void*, 52
WSDL and, 64
ad-hoc polymorphism, 66-67
operation patterns, 64-65
variability, 60. *See also* information variability
varying conditions, as approached by other industries, 2-3
Vlissides, John, 64
"void*," variability, 52

W

Web Ontology Language (OWL), 34
Web Services, 9
versus REST, 75
Web Services Business Activity standard, 156
Web Services Description Language (WSDL), 9, 64
variability and, 64
ad-hoc polymorphism, 66-67
operation patterns, 64-65
Web Services-Interoperability (WS-I), 65
WebSphere, work area service (J2EE), 108
WebSphere Business Modeler, 127, 142
mapping objects, 143
WebSphere Business Process Execution Language. *See* WS-BPEL (WebSphere Business Process Execution Language)
WebSphere DataPower, aggregation, 111-112
WebSphere ESB, aggregation, 110-111
WebSphere Fabric Tool Pack, 152
WebSphere Integration Developer, 127, 144
WebSphere Message Broker, aggregation, 111
WebSphere Process Server, 144
WebSphere Process Server mode, 143
WebSphere Service Registry and Repository (WSRR), 103

work area service, J2EE (state preservation), 108-109
Workflow Management Coalition standards body, 160
WS-AT (WS-Atomic Transaction), 119
WS-BPEL (WebSphere Business Process Execution Language), 126
implementing from BPMN, 142-146
micro-flows, 112-113
process choreography, 138-142
WS-Federation, 170
WS-I (Web Services-Interoperability), 65
WSDL (Web Services Description Language), 9, 64
variability and, 64
ad-hoc polymorphism, 66-67
operation patterns, 64-65
WSRR (WebSphere Service Registry and Repository), 103

X-Y

XBRL (eXtensible Business Reporting Language), 57-58
XLink, 57-58
XML Canonicalization, 171
XML Pointer Language (XPointer), 58
XML Process Definition Language (XPDL), 126
XML schema, 132
XML Schema Declaration (XSD), 134
XPDL (XML Process Definition Language), 126
XPointer (XML Pointer Language), 58
XSD (XML Schema Declaration), 134

Z

Zachman Framework, 3