Cutter
IT Journal

The Journal of
Information Technology Management

Vol. 20, No. 6
June 2007

"If we accept that SOA is the royal road to better alignment between business and IT, we must face the question: exactly how will this alignment come about?"

— Tom Welsh,
Guest Editor

Service-Oriented Architecture and Its Implications for Governance

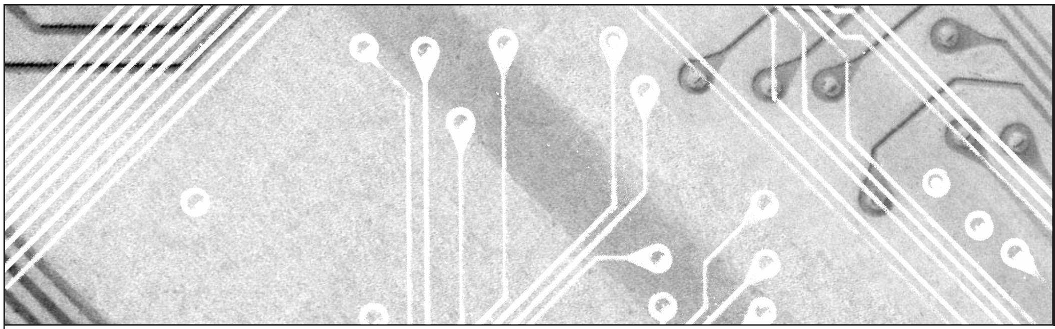
Automated Management

A successful SOA initiative soon brings forth dozens, or even hundreds, of services. No human being can possibly keep track of how all these are working. SOA governance requires powerful suites of tools to monitor, record, and analyze service traffic in order to identify bottlenecks, clashes, and dependencies before they cause serious problems.

A Meeting of Minds

SOA involves an apparent contradiction. Services can only be implemented bottom-up by developers, but they must be specified and fine-tuned top-down by business professionals. The only solution is to place an intelligent layer in the middle to reconcile technical means with business ends. That intelligent layer is SOA governance.

CUTTER
CONSORTIUM



Cutter
IT Journal

About Cutter IT Journal

Part of Cutter Consortium's mission is to foster the debate of, and dialogue on, the business technology issues challenging enterprises today, to help organizations leverage IT for competitive advantage and business success. Cutter's philosophy is that most of the issues that managers face are complex enough to merit examination that goes beyond simple pronouncements. Founded in 1987 as *American Programmer* by Cutter Fellow Ed Yourdon, *Cutter IT Journal* is one of Cutter's key venues for debate.

The monthly *Cutter IT Journal* and its weekly companion *Cutter IT E-Mail Advisor* offer a variety of perspectives on the issues you're dealing with today. Armed with opinion, data, and advice, you'll be able to make the best decisions, employ the best practices, and choose the right strategies for your organization.

Unlike academic journals, *Cutter IT Journal* doesn't water down or delay its coverage of timely issues with lengthy peer reviews. Each month, our expert Guest Editor delivers articles by internationally known IT practitioners that include case studies, research findings, and experience-based opinion on the IT topics enterprises face today — not issues you were dealing with six months ago, or those that are so esoteric you might not ever need to learn from others' experiences. No other journal brings together so many cutting-edge thinkers or lets them speak so bluntly on issues such as:

- Proving the value of IT ROI
- The give and take of offshore outsourcing
- The value of high-quality data
- The evolution of agile project management
- When and how to kill a dying project

Cutter IT Journal subscribers consider the Journal a "consultancy in print" and liken each month's five or six articles exploring a single topic to the debates they participate in at the end of a day at a conference — with every participant in the conversation forcefully expressing his or her viewpoint, backing it up with data and real-life experience.

Every facet of IT — application integration, security, portfolio management, and testing, to name a few — plays a role in the success or failure of your organization's IT efforts. Only *Cutter IT Journal* and the *Cutter IT E-Mail Advisor* deliver a comprehensive treatment of these critical issues and help you make informed decisions about the strategies that can improve IT's performance.

Cutter IT Journal is unique in that it is written by IT professionals — people like you who face the same challenges and are under the same pressures to get the job done. The *Journal* brings you frank, honest accounts of what works, what doesn't, and why.

Competition remains intense in the high-tech world. Budget cutbacks and short deadlines have become the norm. You need advice and experience you can rely on. Put your IT concerns in a business context. Discover the best ways to pitch new ideas to executive management. Ensure the success of your IT organization in an economy that encourages outsourcing and intense international competition. Avoid the common pitfalls and work smarter while under tighter constraints. You'll learn how to do all this and more when you subscribe to *Cutter IT Journal*.

Cutter IT Journal®

*Cutter Business Technology Council:
Rob Austin, Ron Blitstein, Christine Davis, Tom DeMarco, Lynne Ellyn, Jim Highsmith, Tim Lister, Lou Mazzucchelli, Ken Orr, Ed Yourdon*

*Editorial Board:
Larry L. Constantine, Bill Curtis, Tom DeMarco, Peter Hruschka, Tomoo Matsubara, Navyug Mohnot, Roger Pressman, Howard Rubin, Rob Thomsett, George Westerman*

*Editor Emeritus: Ed Yourdon
Publisher: Karen Fine Coburn
Group Publisher: Chris Generali
Managing Editor: Karen Pasley
Production Editor: Linda M. Dias
Client Services: service@cutter.com*

Cutter IT Journal® (ISSN 1522-7383) is published 12 times a year by Cutter Information LLC, 37 Broadway, Suite 1, Arlington, MA 02474-5552, USA (Tel: +1 781 648 8700 or, within North America, +1 800 964 5118; Fax: +1 781 648 1950 or, within North America, +1 800 888 1816; E-mail: citjeditorial@cutter.com; Web site: www.cutter.com).

Cutter IT Journal® covers the software scene, with particular emphasis on those events that will impact the careers of IT professionals around the world.

Subscription rates are US $485 a year in North America, US $585 elsewhere, payable to Cutter Information LLC. Reprints, bulk purchases, past issues, and multiple subscription and site license rates are available on request.

Opening Statement

by Tom Welsh, Guest Editor

Service-oriented architecture (SOA) has been the hottest topic in software for the past two or three years, and it looks as though it will continue to enjoy that status for some time to come. Like Java, XML, and Web services, it has attained buzzword superstardom — membership in that select clique of terms that seem fated to be continuously analyzed and debated by analysts, bloggers, and the media. Under the circumstances, software vendors have naturally hastened to talk up their products' potential contribution to SOA success.

Unlike most innovative ideas, however, SOA is deemed to be of immediate and pressing interest to business decision makers, from CxOs on down. This is because (in spite of its name) it is not just another software architecture, but a new and powerful way of reorganizing business enterprises. The theory is that, with SOA, all corporate software applications are decomposed into logically related groups of services. Each department or division — HR, finance, marketing, purchasing, and so on — takes responsibility for creating and maintaining those services that implement its business activities.

With the advent of SOA, it is harder to consign all technical questions to the IT department and leave it to work away in isolation. Indeed, this should theoretically be impossible, as services cannot be properly designed or administered without substantial input from the relevant business groups.

SOA is intrinsically distributed, whether network-wide within an enterprise or reaching out beyond the firewall to embrace external partners, suppliers, and even customers. Moreover, SOA experts stress the importance of breaking up the monolithic control that central IT has usually exercised over corporate computing in order to put separate divisions, lines of business, and departments in charge of their own services. For these reasons, together with its strategic nature and the size of the investments likely to be required, SOA is believed to encourage convergence between business and IT planning.

Thus, SOA is deeply linked with enterprise architecture (EA) and governance. Indeed, it could be seen as the answer to an enterprise architect's dream, as it comes complete with answers to many knotty questions that would otherwise have to be worked out ad hoc. If you have a robust, viable SOA that is capable of growing in response to corporate business requirements and opportunities, you automatically have most of your enterprise architecture, too.

All this is well and good, but if we accept that SOA is the royal road to better alignment between business and IT, we must face the question: exactly how will this alignment come about? How can we establish SOA governance so that business and IT professionals cooperate smoothly to deliver maximum business value while minimizing risk?

SOA governance is an extremely important, and highly sensitive, topic. After all, it sits at the intersection of two zones of controversy, neither of which has yet been resolved. On the one hand, we have SOA — a new technology revolution that poses difficult IT problems and even more exacting organizational ones. On the other, we have IT governance, a discipline that, to be polite, we may characterize as still more of an art than a science. Neither can get by without the support of the other. IT governance is being transformed by the radically new demands of service-based IT, and SOA cannot hope to succeed without firm and consistent governance.

Given that SOA compels business and IT to cooperate organically, it follows that SOA governance must also blend requirements, processes, and techniques from both these domains. Questions inevitably arise as to the distribution of responsibility, the degree and manner of cooperation, and the ways in which decisions are made and — just as important — enforced. At one extreme, a committee of business managers might interview IT specialists before deciding what services are needed, how they are to be funded and built, and what feedback metrics are to be obtained. At the other — a far more frequent scenario — business groups inform the IT department of their requirements and then leave the "techies" to do the rest. Between them, the authors of the six articles that make up this issue of *Cutter IT Journal* thoroughly address all these questions.

Michael Kunz, Dirk Krafzig, and Dirk Slama proceed from the logical view that SOA governance cannot be dealt with as a static issue. Instead, they describe the ongoing challenges posed by a legacy-to-SOA transformation, noting that "effective governance raises the probability of SOA success because it enables big organizations to conduct change programs over many years." Not only does enterprise SOA force a reevaluation of how IT interacts with business, it also imposes organizational changes.

According to the authors, the transformation to SOA takes place on six levels: business, organization, people, IT processes, functional architecture, and technical architecture. They provide a sustainable roadmap for the transition to SOA at each of these levels and claim that this framework helps enterprises take a holistic approach to SOA adoption. This contention is supported by a brief case study of an international services company whose reuse rate in IT projects rose by 120% in a single year after it decided to take a global approach to SOA.

In our second article, Jorge Ronchese takes a step back to get a broader perspective on the origin and desirability of SOA governance initiatives. As every organization does not necessarily need SOA, he proposes eight criteria for assessing the likely benefits. There follows an analysis of inadequate reasons for embracing SOA governance, such as a desire to impose stronger controls or to compensate for weak management without ruffling too many feathers. Resistance to change is inevitable, but, Ronchese argues, it is not necessarily a bad thing. He explains how listening carefully to people's objections can be the first step toward defining problems and finding solutions to them, and he shows how the Theory of Constraints can help with this analysis. Given that "SOA governance is important if it solves some of the problems or constraints that are preventing your company from reaching its goals," Ronchese concludes by asking bluntly, "Do you have them clearly identified?"

Keith Swenson, VP of R&D at Fujitsu's Enterprise Software and Solutions Group, states a vital prerequisite for effective SOA creation: understanding the essence of the business. After all, without a clear insight into the enterprise's purpose and core activities, how can an SOA even be designed? Gaining that insight in turn demands a shift in the way business and IT work together. To this end, Swenson tells us, "Companies need to build an architectural roadmap with SOA governance woven into the fabric of daily operations." In short, someone must understand what the business is doing well enough to see changes coming, and then understand IT well enough to adapt the SOA accordingly. The trick is to make top-down business goals mesh with bottom-up IT development, which Swenson contends is the role of SOA governance. He also urges the instrumentation of SOAs with appropriate tools in order to know which services are available, who is using them, which services depend on others, and what quality of service is being provided.

So far, we have seen a variety of different approaches. Cutter Senior Consultant Tushar Hazra seeks to knit these together. He begins by comparing SOA governance with corporate governance, EA governance, and IT governance, seeing to what extent these existing disciplines can be co-opted to support the new dispensation. One clearcut distinction is that SOA governance has to cope with the impact of distributed services across one or more business organization(s), based on service-level agreements. Hazra agrees with Swenson

IN NEXT MONTH'S ISSUE

How Should IT Enable Business Strategy?

Guest Editor: Victor Rosenberg

Back in the 1990s, IT was considered the genie in the lamp. Want to revise a business process or achieve some other information goal? Just turn the problem over to IT, which will somehow magically make it happen.

Over a decade later, the bloom is definitely off the rose. Today there are many views as to how involved IT should be in business strategy. Nicholas Carr and his acolytes say IT's role is merely to do what the business says should be done as cheaply and efficiently as possible. Others believe that IT can enable competitive advantage by converting new capabilities into better products faster than competitors and by quickly translating know-how into business value.

Join us next month as we debate the ways IT can — or can't — enable business strategy. Learn how to determine whether your IT organization is a strategic enabler or a tactical one — and how those roles can change over time. Discover how one IT group went from providing basic services to participating in a true business partnership thanks to its "well-formed strategy statements" and savvy industry scanning. If you'd rather be contributing innovative ideas instead of simply executing what you've been handed, don't miss next month's issue!

that SOA governance should form a connecting layer between the complementary and equally important bottom-up and top-down approaches. After surveying some useful policies, principles, and best practices based on his own practical experience, Hazra offers some advice about the tools and techniques available for implementing governance processes.

At first glance, Hao He and Brett McDowall seem not to be addressing SOA governance itself, as their chosen topic — the Universal Business Identifier (UBI) — has more to do with technical architecture. On closer inspection, it turns out that the UBI has a significant contribution to make. He and McDowall start from the thesis that the goal of SOA governance is to align the business with IT by providing a suitable decision framework. The sharing of models between business and IT, especially a common taxonomy, materially contributes to that alignment. What He and McDowall propose is that the Universal Resource Identifier (URI), which we routinely use to specify Web addresses, should be co-opted in the guise of the UBI. They argue that "absolutely anything important in an enterprise should be identified by a URI." At a stroke, this would fulfill a number of SOA requirements, such as identification, addressing, ownership, resource lookup, and transparency. Above all, the UBI would help to keep things as simple as possible.

In our last article, Piotr Szabelak, Jan Topinski, and Cutter Senior Consultant Borys Stokalski identify a fundamental polarity between the formal, orderly processes of EA management and the innovative, potentially chaotic forces of SOA, which they dub the "yin" and "yang," respectively, of corporate computing. Indeed, the mere introduction of SOA can lead to growing demand for changes in governance, as established processes no longer cut the mustard. For example, many organizations currently seeking to roll out SOA have found themselves forced to rely on traditional techniques such as large-scale release management. These, however, militate directly against the fine granularity, ad hoc nature, and responsiveness that are supposed to be among SOA's greatest strengths.

This can be avoided, say the authors, by restructuring IT governance and EA processes along SOA lines. Everything — from business planning to software release processes — must fit in with the services metaphor. Trickiest of all, practices should be devised that allow project teams to coordinate their work on services without introducing any cumbersome and restrictive corporate chokepoints, and the services portfolio must become a truly shareable IT asset.

Some key ideas recur in article after article. Almost everyone emphasizes the fact that SOA's reason for existence is to further corporate business goals — so SOA governance must be based upon a thorough understanding of those goals. There is also a strong consensus that building an effective enterprise SOA cannot be done piecemeal; instead, it should flow from a coherent central plan. On the other hand, the whole thrust of SOA is to accommodate different platforms, languages, and protocols. So we arrive at the classic dichotomy between top-down design and bottom-up implementation, with SOA governance as a "glue" layer in the middle holding it all together.

While tools are necessary for effective SOA governance, they are definitely not sufficient. As governance relies heavily on the behavior and decisions of human beings, it cannot be wholly automated. Moreover, the part of it that can be automated is, as we might expect, the lower-level, relatively mechanical aspects.

Whatever your interests and concerns or the special requirements of your particular organization, there is sure to be something in this issue that will interest and inform you. You may also discern some overlap and, perhaps, a measure of constructive disagreement. But that is inevitable, as the art of SOA governance is still in its infancy. One thing is for sure: if you are responsible for setting up or administering a corporate SOA, you will find stimulating and useful ideas aplenty in what follows.

Tom Welsh is a Senior Consultant with Cutter Consortium's Enterprise Architecture advisory service and former Editor of Cutter Consortium's monthly Web Services Strategies*. He is an independent consultant and analyst specializing in middleware, object technology, and software engineering. At Digital Equipment Corporation, Mr. Welsh was a hardware technician, software support specialist, corporate software developer, and senior technology consultant before taking on the task of marketing Digital's OO software products in the UK. In his role as a principal analyst at ComputerWire, Mr. Welsh has been a leading contributor to* Client/Server Tools Bulletin*,* Internet Tools Bulletin*, and* Object-Oriented Tools Bulletin*. He has written many reports and papers as well as chaired and spoken at conferences and seminars. Since 1992, Mr. Welsh has closely followed the work of the OMG and its specifications, including CORBA, UML, XMI, and CWM. He can be reached at twelsh@cutter.com*

Governing the Legacy-to-SOA Transformation

by Michael Kunz, Dirk Krafzig, and Dirk Slama

ENTERPRISE SOA AND WHY IT MATTERS

For decades now, IT experts have been devising new concepts and methods for improving the structure of application systems in order to get a better grip on the ever-increasing complexity found within them. We invented hierarchical, relational, and object databases to manage persistent data more efficiently, and we could observe how the evolution of programming paradigms from structured over functional to object-oriented programming introduced new ways of improving the management of complex application systems.

However, a new programming paradigm is unlikely to help us in managing the complexity that is found in today's historically grown, highly heterogeneous application landscapes, simply because the cost and risk of replacing all the existing application systems is too high. Even if these systems are badly structured and costly to maintain, they *are* working — and that is what matters.

Enterprise service-oriented architecture (SOA) offers an architecture blueprint that is specifically designed to work on the enterprise level — not on the level of individual applications — to help improve the overall structure of heterogeneous application landscapes. In an enterprise SOA, *basic services* provide a core set of relatively stable application functionality, which is then aggregated to provide individual *application services* (see Figure 1). Enterprise SOA is based on the assumption that business processes drive this aggregation. Business process management (BPM) is a key concept of SOA. It helps make business processes explicit, which is the basis for managing, monitoring, and continuously optimizing them. Business rules management (BRM) allows for centralized control over the way business decisions are made within these processes. Because stable code in basic services is separated from more frequently changing business process definitions and business rules, changes in these critical areas can be made much faster, and business stakeholders have more control over these areas.

As we can see, the benefits of an enterprise SOA are manifold. However, transforming an unstructured legacy application landscape into an enterprise SOA is not an easy task, and perhaps more importantly, it is not a technical task alone.

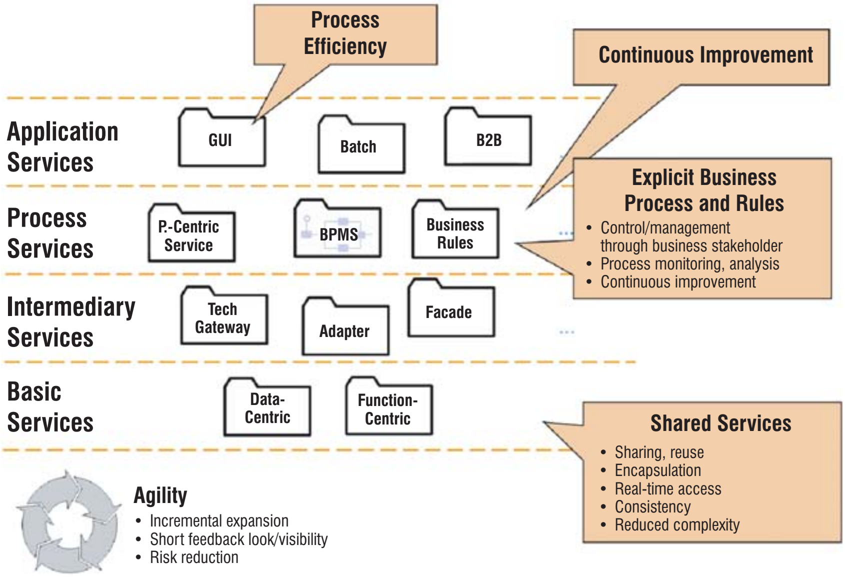


Figure 1 — The conceptual view of enterprise SOA facilitates communication between various stakeholders in the enterprise and fosters business/IT alignment.

Enterprise SOA has a fundamental impact on the organizational structure of an enterprise. Take, for example, the transformation from functional to object-oriented design and programming. This was a methodology shift that happened predominantly within the development organization. Enterprise SOA, on the other hand, requires fundamental changes on the organizational level in many areas. Consider the introduction of business rules management. BRM not only requires us to carefully separate business rules from application code and manage it under the control of a business rules management system (BRMS), it also requires us to redefine the way IT interacts with business. Business stakeholders are suddenly much more directly involved in the management of application systems, because they take on direct ownership of the business rules. Furthermore, changes to business rules are typically executed within a matter of days. While this new level of agility and flexibility is great from a business point of view, it will have a dramatic impact on the IT operations side, which is used to release cycles that are counted in months, not days.

Let's take another example — the reuse of application logic, which is one of the holy grails of SOA advocates. Many articles on SOA describe how reuse in an SOA can be achieved by managing SOA interfaces and other artifacts in shared registries or repositories. However, history has proved that reuse is typically not achieved through goodwill or information sharing alone. In 1968, Melvin Conway made an observation that we believe is still true today. According to Conway's Law, organizations are constrained to produce designs that are copies of the communication structures of these organizations. For instance, if you have four groups working on a compiler, you'll get a four-pass compiler. Conway's Law argues that enterprise architecture cannot fundamentally be changed unless the IT organization is changed in parallel. The IT organization, therefore, not only conducts the transformation, but is also subject to it.

Taking into account that many application landscapes comprise several hundreds or even thousands of different applications and an almost unmanageable number of logical and technical dependencies between these applications, it becomes obvious that modernizing these IT landscapes is an enormous challenge that could take many years and will require the full dedication of the IT organization to accomplish. Effective governance raises the probability of SOA success because it enables big organizations to conduct change programs over many years. Governance is therefore the umbrella that makes enterprise SOA feasible.

THE LEGACY-TO-SOA ROADMAP

In our experience, the transformation of an application landscape into an enterprise SOA requires a holistic view of various aspects of an enterprise's business, organization, people, IT processes, functional architecture, and technology (see Figure 2). Also, it is important to understand the dependencies between the business architecture, the SOA architecture, the software architecture, and the technical system architecture. These dependencies have to be identified for the "as-is" architecture, and potential changes on all levels have to be reflected in the "to-be" architecture. The transition from as-is to to-be then has to happen on the business, organizational, people, IT process, functional architecture, and system levels. In this section, we will provide a brief overview of some key aspects of each of these levels.

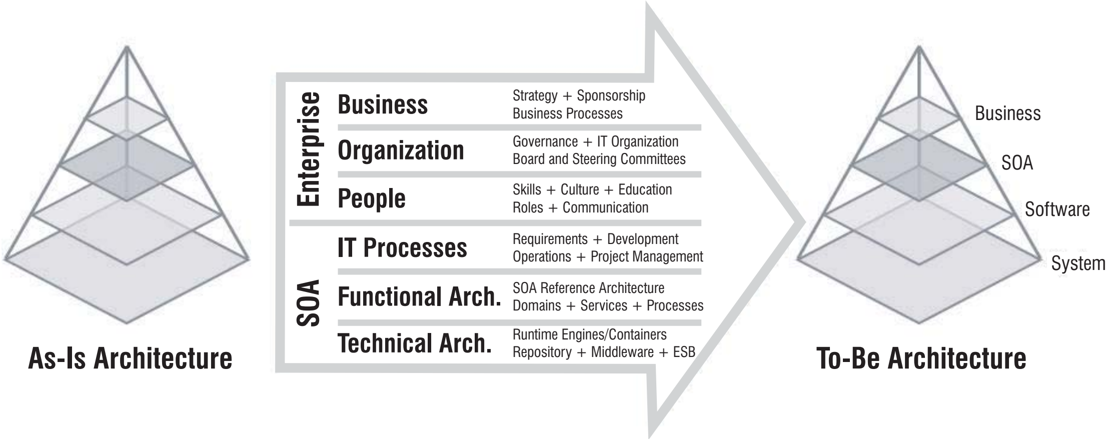

Enterprise
Business
Strategy + Sponsorship
Business Processes
Organization
Governance + IT Organization
Board and Steering Committees
People
Skills + Culture + Education
Roles + Communication
SOA
IT Processes
Requirements + Development
Operations + Project Management
Functional Arch.
SOA Reference Architecture
Domains + Services + Processes
Technical Arch.
Runtime Engines/Containers
Repository + Middleware + ESB
As-Is Architecture
To-Be Architecture
Business
SOA
Software
System

Figure 2 — A sustainable roadmap for the renovation of existing enterprise application landscapes must cope with challenges on several levels.

Business

To see how SOA impacts the business level, we consider two examples:

1. **IT funding.** Traditional funding models for enterprise IT typically distinguish costs for maintenance and projects on the one hand and costs for infrastructure and business functionality on the other. The budgets for infrastructure (such as mainframes, networks, or operating systems) are typically located in IT, while the budgets for the development and maintenance of business functionality are owned by business lines. SOA destabilizes the equilibrium of these cost categories because reusable SOA assets do not fit into this schema. SOA services have characteristics of business functionality and infrastructure at the same time. Traditional funding models are therefore not prepared to allocate the costs of SOA properly.

2. **Process orientation.** Process orientation would be extremely useful for the efficient implementation of SOA. Explicit process models enable service designers to develop service functionality that can be reused across several processes. For obvious reasons, process orientation hardly makes sense as a pure IT initiative. On the contrary, it is the business side that must drive any process initiative.

Organization

As we mentioned above, the IT department is not only the driver of the legacy-to-SOA transformation but also the subject of this change process. We usually see changes on three levels:

1. **SOA board.** A successful SOA organization is typically governed by an SOA board. The SOA board is an executive team, composed of business and IT management. It acts as the steering committee for the SOA program. The goals of the SOA board include aligning business and IT strategy, assessing project proposals, ensuring that SOA relevance and impact are considered appropriately in IT projects, and prioritizing business initiatives and opportunities for SOA.

2. **SOA competence centers.** The day-to-day work is done by SOA competence centers. By definition, a competence center is a team that represents leading-edge knowledge and differentiated capabilities. An SOA competence center is responsible for providing strategic SOA skills. It functions as a resource pool that can be tapped as new SOA projects arise.

3. **Maintenance teams.** The most far-reaching decision would be the setup of permanent maintenance teams for shared SOA services. While SOA boards and SOA competence centers are complementary to existing organizational units, setting up maintenance teams for SOA services would impact the structure of established departments.

People

Obviously, an SOA initiative requires support of the organization's employees. It is not surprising that this support does not come for free. SOA can have a dramatic impact on many individuals, and thus SOA initiatives must be carried out with both empathy and systematic change management.

Let us again have a look at an example. A typical COBOL developer in the maintenance organization of a big enterprise is usually responsible for a piece of business functionality and a specific end customer. He takes responsibility for the complete IT process chain, including requirements gathering, design, development, testing, and deployment. Furthermore, he is responsible for all layers of the software stack, from 3270 presentation down to VSAM files. In a service-oriented world, the job profile of this person changes dramatically. This must not be underestimated, because many people may react fearfully to change or be paralyzed by it. It is therefore vital that those conducting an SOA initiative take explicit measures to gain the support of all stakeholders.

An SOA initiative must be accompanied by an information campaign. This campaign can involve regular newsletters, road shows, intranets, and so on. The main objective is to keep the staff informed and promote the ideas of SOA and its benefits for the organization. Education programs are also necessary. These programs should involve both technical and communication skills. For example, your average COBOL developer will have to acquire skills in new technologies such as WSDL or middleware, and she also has to learn how to collaborate with new team members, such as repository administrators or developers of service consumers. From the perspective of the HR department, these changes would also involve the definition of new role profiles, updated incentive schemes, and so forth.

IT Processes

With the increasing progress of the legacy-to-SOA transformation, the IT processes have to be adapted to the needs of SOA — particularly due to reuse, the smaller size of projects, and the role of the service contract.

While reuse is more or less an incidental event in traditional IT environments, it is a key objective of SOA.

Writing software with reuse in mind is different from writing software for one specific use case. Writing reusable software requires, for example, broader requirements gathering and/or designs that take different concurrent usage patterns into account.

The average size of projects also plays a major role. In an SOA world, the average size of projects is shrinking. This has two major consequences. First, the overhead costs for project initialization have to be reduced. Second, many design activities have to be moved from the projects to program management.

The third driver for change is the service contract. The service contract is a specification document that describes the interface and purpose of an SOA service. In traditional projects, data models and GUI specifications are often the major design artifacts that are used to manage the implementation process. In an SOA environment, the service contract can provide a worthwhile additional contribution, if managed properly.

So let's consider, for example, the relation of an enterprise customer and a provider of implementation services. Traditional work orders were often related to particular applications that should be developed or changed in self-contained projects. A great deal of the responsibility for the architecture was delegated to the implementation partner. In a service-oriented world, the relation between these partners and the processes in question will change significantly. This is particularly the case because in a service-oriented world, the responsibility for the high-level architecture remains with the end customer. As a consequence, the high-level architecture becomes a major milestone in service-oriented projects.

Functional Architecture

Well-designed domain architectures for a large enterprise typically comprise about 20 functional domains. Each functional domain represents a major concept of the business and encapsulates both functionality and data that are related to this business concept. A domain also provides SOA services that allow service consumers to access the functionality and the data of the domain. Any executable programs, such as user interfaces or batch programs, are separated from the SOA domains.

Nowadays applications are fundamentally different from SOA domains. A typical application is a mixture of several domains, various user interfaces, and batch programs. Different applications often implement functionality and data redundantly.

The major goals of SOA are to:

- Work toward the alignment of applications and SOA domains
- Extract service functionality out of applications and move it to proper SOA services
- Reduce redundancies and inconsistencies

In an SOA world, the average size of projects is shrinking.

Technical Architecture

Last but not least, the legacy-to-SOA roadmap has to be applied on the level of the technical architecture. An Enterprise Service Bus (ESB), for example, is a new type of middleware that enables efficient communication between the various communication end points within an SOA (e.g., SOA services or business processes).

ESB product vendors typically argue that the first step toward an SOA is the implementation of an ESB infrastructure. Once this infrastructure is in place, they say, enterprises can start implementing services on top of it. Unfortunately, the situation is much more complex. Typically, enterprises already have several middleware products in their infrastructure portfolio. Each middleware product is deeply integrated into the existing application landscape and cannot be replaced overnight. At the same time, many of these products provide their own migration path toward ESB technology. This is one of the reasons why enterprises will have to manage ESB heterogeneity in the future as they have managed traditional middleware heterogeneity in the past. The modernization and consolidation of the middleware portfolio toward a standardized ESB infrastructure is consequently an incremental, long-lasting undertaking that must be governed properly.

All these goals cannot be achieved in an instant. Addressing the most pressing issues on each of the six levels of the legacy-to-SOA transformation takes considerable time. SOA governance is needed to guide this undertaking.

CASE STUDY: SOA IN A GLOBAL COMPANY

In 2002, a major international company in the service industry launched regional SOA initiatives for several countries. Aiming to address the problems caused by its heterogeneous, nonstandardized IT landscape, the company tried to increase standardization substantially and to improve efficiency within IT.

Building on the success of the local SOA initiatives, management decided later to take the SOA approach to the global level, aiming to address global harmonization and integration issues. As the regional SOA initiatives had done, the global SOA project started with the development of improvements on the technological and architectural levels. These included the development of a service repository and the fully automatic configuration of the global runtime infrastructure by means of model-driven software development.

While this new approach had many technical benefits, adoption in the different country organizations was slow. In order to improve the adoption rate, management decided to launch a global IT architecture board, which had the task of coaching the different country organizations on the use of the new SOA infrastructure as well as reviewing all projects with respect to architectural compliance. In order to enforce the SOA concepts more rigorously, architectural compliance was also included as a key performance indicator (KPI) in the balanced scorecard the company used to measure the success of individual IT projects.

In parallel, management launched an enterprise architecture management (EAM) initiative, which aimed to help increase transparency and align business and IT more closely by defining mappings between business and software architectures. The concept of SOA was included as a key instrument for structuring the business and software domain architectures.

Finally, management reviewed the project portfolio management and demand management processes. Previously, projects were seen as largely independent, which prevented the company from identifying areas of possible reuse between different projects. The company leveraged the information created in the EAM initiative to identify dependencies between different projects on the functional and technical levels. This was done on the level of granularity of SOA domains.

Today the company is able to identify potential synergies between projects in the planning phase and to react to this information much more efficiently. For example, it might split two planned projects into three projects, where the first project delivers reusable SOA functionality that is then used by the other two projects. Over the last year, the company has improved its reuse rate in IT projects by 120%.

CONCLUSION

SOA provides a set of very powerful tools, but it can only be implemented efficiently if it is addressed on the technical level *and* the business and organizational levels. The six levels of the legacy-to-SOA transformation provide a proven framework that guides enterprises on their way toward SOA adoption and helps them to take a holistic approach.

Probably the most critical element of the legacy-to-SOA transformation is the organizational level. As Conway's Law dictates, organizations are constrained to produce designs that are copies of their communication structures. Consequently, the successful adoption of SOA requires would-be adopters to very carefully review their own organizations.

Since finishing his studies of business management in Cologne, Germany, Michael Kunz has worked for eight years as an architect, consultant, project manager, and general manager in the IT industry, working for both IT service providers as well as independent software vendors (ISVs). His main vertical focus is in the insurance industry, where he worked for several years in the life and property and casualty (P&C) areas, managing complex projects and developing modern architectures. Mr. Kunz has been focusing on EAM and SOAs since 1997. Mr. Kunz can be reached at Michael.Kunz@businessglue.de.

Dirk Krafzig has worked for more than 15 years as an architect, consultant, and manager in the IT industry, both for IT service providers and ISVs. Dr. Krafzig focuses primarily on the insurance industry, particularly the life, health, and P&C areas, where he has managed complex projects and developed modern architectures. Since 1996, he has focused on EAM and SOAs. Dr. Krafzig holds an MSc and PhD in computer science. Dr. Krafzig can be reached at dirk.krafzig@businessglue.de.

Dirk Slama holds an MSc in computer science from TU Berlin, as well as an MBA from IMD International in Lausanne, Switzerland. Mr. Slama has more than 12 years of international IT experience, having worked in the US, the Asia/Pacific region, and Europe. Mr. Slama has helped customers such as Boeing, AT&T, NTT DoCoMo, and Halifax Bank of Scotland to implement large IT projects. Mr. Slama can be reached at dirk.slama@businessglue.de.

Who Cares About Governance and SOA?

by Jorge V.A. Ronchese

The term "SOA governance" is appealing. It seems like something good to have. Are you attracted by these two words, "SOA" and "governance"? Do you think that they naturally fit each other nicely? Do you think that your company needs SOA or governance to improve?

If you feel the temptation to push forward and start an SOA governance project, I recommend you think twice. There are two main sources of any SOA governance initiative. Either IT pushes it because adding governance to an SOA initiative makes it sound less risky and should assure the company's involvement, or the company pushes it because the SOA initiative is not well suited to the situation, it's risky, or it's not totally understood, and the company wants to better control it. These are just two sides of the same coin. Both are clear symptoms that something is not going as expected. And I would bet the initiative has not started properly.

Let me share with you some thoughts and experiences about how, if needed, you should handle your SOA governance initiative.

FACE REALITY FIRST

I have found that SOA produces mixed feelings in IT departments. Can you divide your personnel into SOA fanatics and SOA skeptics? Have you done it already? Have you heard the skeptical arguments properly?

Your homework starts with discussing those issues with your team or teammates. Face them as your first exercise. They are a source of invaluable information that neither vendors nor consultants can provide.

To the rest of the company, SOA still sounds like technical stuff (i.e., not related to their jobs or concerns). Our gut feeling tells us that *nobody in the company really cares how we deal with technical complexity*, or which three letters we use to name our new silver bullet. It is part of our job description. As IT professionals, we might also be tired of the "Three Letters Syndrome" (TLS) in the IT market, but SOA still seduces us in one way or another.

Sooner or later, we will start to spread the word of the newest panacea to our internal clients (name it the "business side" if you feel that your IT department is outside the business). Have you done it already? What reactions did you get?

THE "REAL" PROBLEM

The SOA concepts and paradigm assume some conditions, which I was not able to find in every organization undertaking SOA. Before pursuing an SOA initiative, you should first check to see whether these assumptions are valid for your company:

- There are processes defined. Someone understands them with a global view and, ideally, owns them.
- The internal client sees his processes change faster than IT might be able respond to them.
- Internal clients dislike the current information systems.
- Internal clients understand that there is a problem to solve.
- It is a well-known fact that external customers experience some unpleasantness dealing with the company due to "disconnected" processes or systems.
- Internal users have to access several applications to do their jobs.
- Internal users have to access external applications (from customers, suppliers, or services companies) to do their jobs.
- Future needs are not easy to predict.

I could continue listing more assumptions, but I think we can agree that these describe the situation you may be facing today. Some of the problems listed might be solved by a service-oriented architecture or other alternatives. However, is it critical to solve them as soon as possible? Are these problems real problems or merely symptoms?

Face it: *first, you have to understand clearly which problems you are facing in your daily and future business environment*. In terms of priorities, SOA could come in second place, or maybe third. It will naturally fit if it is really needed. Instead of trying to solve every problem as separate entities, my colleagues and I recommend using a systemic approach or method for discovering them. The Current Reality Tree (CRT) diagram from the Theory of Constraints' Thinking Processes (TOC TP)[1] can be used to help you discover the root causes of your symptoms or problems by means of cause-effect-cause relationships (see Figure 1). The process of building the CRT is best done in a working meeting with the people who are experiencing the problems. You have to listen to them carefully when they express their claims.

Avoid the mistake of selling a solution to no one's problem. Remember the old adage: "People who buy shovels are not interested in shovels; they want to make holes or to fill in holes as quickly and easily as possible." What would occur if you went ahead with the SOA message without considering the real problems that need to be solved? What will occur? Will your internal clients understand why you are talking about it?

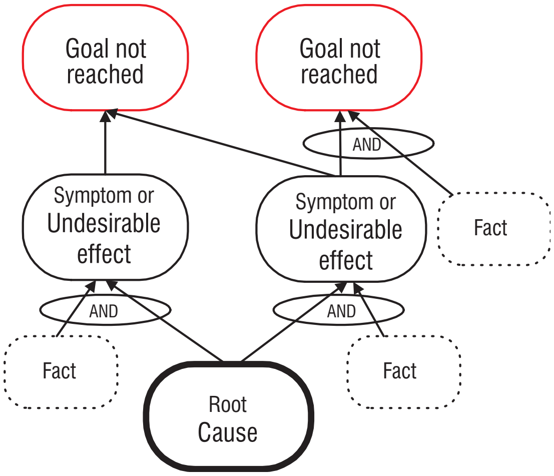


Figure 1 — A CRT diagram can help you find root causes from symptoms or problems.

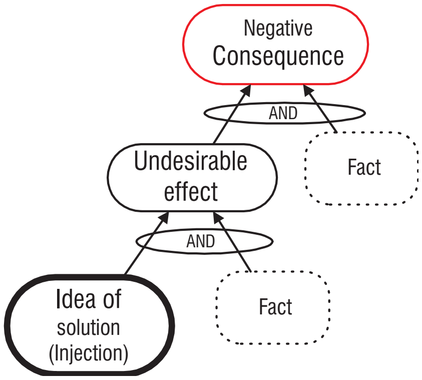


Figure 2 — An NBR diagram can be used to clearly understand negative consequences of an idea.

Eventually, the inevitable happens: someone begins to see black holes, black boxes, security threats, and maybe a new workload for her department. She faces you, or even quarrels with you, because you are trying to change her way of doing things without any good reason.

You are facing resistance; people are raising obstacles on your way. They can appear in the form of any company process to review (and thus delay acting on) your suggestions. Can the company's procedures be a real obstacle to brilliant ideas?

You can find resistance at different levels, which implies different levels of understanding about your initiative. Maybe people don't see that there is a problem to solve, or they understand the problem but feel that this initiative doesn't solve it, and so on. The Negative Branch Reservation (NBR) diagram from the TOC TP can be used to help you understand how others see your solution driving negative results or undesired consequences (see Figure 2). Once this relationship is clear, you can work out ideas to solve or mitigate these consequences.

Resistance these days is hidden behind the newest holy grail, something that everybody seems to look for but nobody is able to find: governance. When someone mentions it, suddenly any initiative is diverted onto a corporate, bureaucratic path. You may feel this as a break in communication, a misunderstanding, a conflict that puts IT and the so-called business areas on opposite sides.

The Cloud diagram from the TOC TP can be used to help you understand the components of the conflict (see Figure 3). Questioning the assumptions hidden behind the arrows will help lead you to solutions for this conflict.

TO GOVERN OR NOT TO GOVERN?

The inherent complexity and risks of service-oriented architectures automatically recall to everybody the control aspect of governance. This response is disappointing to SOA proponents, who want their business clients to appreciate the potential benefits of the initiative. Is this reflexive focus on control a sign that they haven't realized SOA is beneficial for the company?

Fortunately, good governance mechanisms and structures are beneficial for IT and non-IT areas (and, of course, for the company). Unfortunately, few companies have embraced governance objectives as a whole. Most

[1] The Theory of Constraints and the thinking processes were originated by Dr. Eliyahu Goldratt.

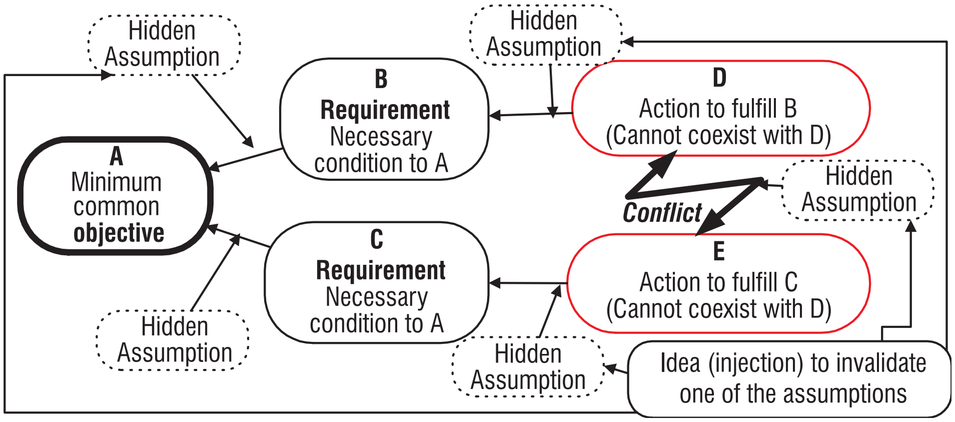

Hidden Assumption
Hidden Assumption
B Requirement Necessary condition to A
D Action to fulfill B (Cannot coexist with D)
A Minimum common objective
Conflict
Hidden Assumption
C Requirement Necessary condition to A
E Action to fulfill C (Cannot coexist with D)
Hidden Assumption
Hidden Assumption
Idea (injection) to invalidate one of the assumptions

Figure 3 — A Cloud diagram can be used to better spot wrong assumptions and find alternatives for solving conflicts.

of them fund governance for compliance reasons; others confuse it with security or even auditing; and my colleagues and I have also found some enthusiastic companies that identify governance as the source of quality requirements. Sometimes talking about governance is a diplomatic way to analyze and uncover weak management practices within a department.

Every company redefines the meaning of governance. Initially, governance addressed the steering and control mechanisms within the company, but Sarbanes-Oxley efforts reinforced the control view. Then, good governance meant having an adequate balance of risk and value in any company initiative. Lately, governance includes decision-making and conflict-resolution processes and structures.

But why do we need governance in the first place? We need it because it helps direct the company's efforts to achieve its goals while taking risks into account. (Remember, goals are important!) Having good governance means that we put in place processes and structures to establish the rules, directions, and controls needed to execute what has been agreed upon and to judge and resolve the conflicts that will arise. To be of any good, governance should help the company to achieve the goal of value while also caring about the associated risks.

SOA environments introduce a whole new set of governance questions that need to be answered. Who owns the services? How do we select services? Which services do we publish to the external community? How should the services be orchestrated? How do we control the service operation? How do we control the service lifecycle? And so on.

Why can we not conclude that SOA governance is a very strong value proposition? Have we clearly defined the problem(s) that SOA governance processes will solve?

SOA needs to be identified as necessary before we open the governance door. We want to introduce SOA governance as something that is good to have, not as a means of delaying the SOA initiative or avoiding having to deal with IT's suggestions. We also need to separate companies' IT governance processes and structures from our internal IT management practices.

THE EVIL INSIDE

How can we return our focus to the business benefits of our SOA initiatives so we can work out the objectives with our business peers (and then all be part of the business)? Maybe we should look back and remember that neither SOA nor governance is a good starting point for business improvement initiatives or for bettering the business-IT relationship. While SOA can have a positive business impact, it remains a technical concept that our business partners may have trouble grasping.

This means we need to get back to basics and figure out how to talk in business value terms about SOA. We also need to show that we welcome risk concerns. Then we can move forward together in a joint effort to establish better governance of the solution. From this point on, we can be walking on the strategic alignment side of the governance concept and delving into the ways we can deliver real value for the company. While these are not easy tasks for most IT departments, the Future Reality Tree (FRT) from the TOC TP can be used to help you show how your solution drives positives or desired effects in the company (see Figure 4).

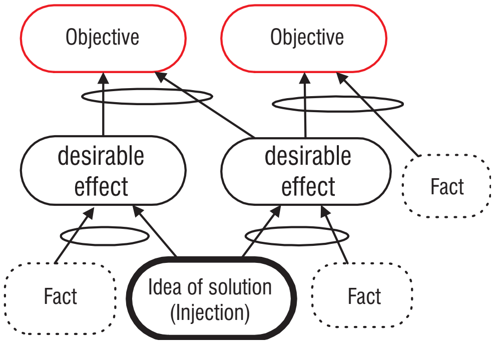


Figure 4 — An FRT diagram can be used to better show the desirable effects that will result if the solution is implemented.

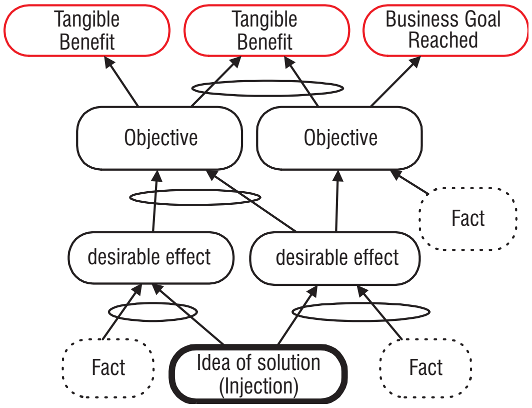


Figure 5 — The FVT shows the expected business outcome if the solution is implemented.

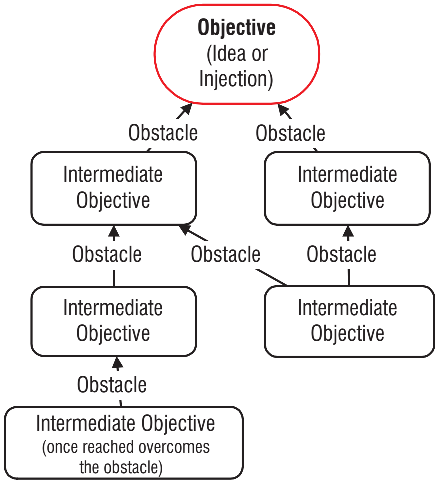


Figure 6 — A PrT diagram can help you develop an agreed-upon project strategy for realizing the solution.

One step further is what I have named the Future Value Tree (FVT), which pushes the cause-effect relationship until we understand which key process indicator (KPI) we are affecting with the solution or which tangible measurement(s) will be improved (see Figure 5). You can later use this diagram to feed your business case.

If SOA is good, we should be able to realize it. For that purpose, we need the non-IT areas to assume their roles and responsibilities. How do we do this?

The Prerequisite Tree (PrT) from the TOC TP can be used to help you involve everyone who should do something to implement the solution. The process of building the PrT should be done in a working meeting with all of needed participants (see Figure 6).

All this being said, we should realize that we are our own major obstacle to doing the right things. Our way of thinking and communicating our ideas or solutions is the biggest impediment to implementing them while involving others. As I have pointed out, we in IT often underestimate the real critical problems of our company by not facing them properly — that is, together with the business areas. Consequently, we are not able to positively overcome the resistance we encounter.

As shown in Figures 1-6, the use of TOC tools and processes can help business and IT tackle business problems in a systematic way, better define the solution so as to avoid any negative effects, and find an effective path to make it happen. The detailed explanation of each TOC thinking process is outside the scope of this article, but you can find a rich explanation in [1].

RESISTANCE IS GOOD

Resistance to change has a meaning, and you have to discover it in order to produce effective and lasting results. Facing resistance is good, because it gives us the opportunity to better define our problems and find more effective solutions.

The Theory of Constraints identifies six layers of resistance to change. Each one of them reflects the stage of understanding and involvement your peer or client has. Each layer comes with a thinking process to overcome the resistance in a proper manner, one that moves you closer to a change with a positive impact on the goals of your company. You will have to learn to walk through each one of the layers, as any shortcuts will put your future project at risk.

Each of these layers is related to the following three major questions:

1. WHAT to change?

2. TO WHAT to change?

3. HOW to cause the change?

The easiest way to determine what layer you are stuck in is to identify the concerns and questions of each one.

Layer 1: WHAT to Change?

Facing Layer 1 resistance, you will find disagreement about the problem. Conversations will include expressions like "What is this for?"; "Why should we bother about this?"; and "We cannot do anything." At this layer, forget about business benefits or features and functions. You need to reach an agreement on what the actual problem is.

Layer 2: From "WHAT to Change?" to "TO WHAT to Change?"

Facing Layer 2 resistance, you will find disagreement about the initial conceptual idea of the solution. You will hear expressions like "Why do you say this will help?"; "How this can help us?"; or even "Is this the newest IT trend we must follow?" (At least you have agreement on the problem, don't you?)

Layer 3: TO WHAT to Change?

Facing Layer 3 resistance, you will find reservations about the results to be obtained because of the solution found. You will know you are at this layer when you hear things like "Will it really impact our numbers (or goal)?" or "Maybe it will never work, just like the last technology migration." You will have to find a way to relate the solution to the future reality you predicted ("desirable effects" in terms of TOC).

Layer 4: TO WHAT to Change? (Part II)

Facing Layer 4 resistance, you will find constructive criticism. Don't give up! It should be constructive, after all. People at this layer are worried about negative side effects. This is their way of expressing risks ("undesirable effects" in terms of TOC). When evaluating the chances of the solution, they might say, "Yes, but …" or "Company X has thrown a humongous amount of money at the solution to make it work." You will need to work out the "but …" part in order to overcome this layer of resistance and complete your solution.

Layer 5: From "TO WHAT to Change?" to "HOW to Cause the Change?"

Facing Layer 5 resistance, you will hear people tell you why the solution will never be achieved in your environment. Don't panic. This information is very useful. You need to involve everybody to find the obstacles (which is the proper TOC term). Two heads think better than one. Welcome obstacles and involve people to help you overcome them.

Layer 6: HOW to Cause the Change?

When exhibiting Layer 6 resistance, people will find difficulties in collaborating with others to get the job done. This means that it is not clear yet how everybody should act and what tasks should be done by whom and why.

Each of the TOC TP tools mentioned above is meant to be used at a particular layer of resistance (see Table 1).

Of course, there are some assumptions used by the method, too:

- Your company or business area has a clear goal.
- You will involve other people in the improvement process, working together as a team through each layer of resistance to change.
- You are an open-minded person, able to find root cause problems and work out solutions that may be different from your initial ideas.

RESPONSIBILITY MATTERS

Chances are the problems your company is facing can be solved via a service-oriented architecture, and that to make it a good solution, some minimal governance requirements must be fulfilled. However, following the method outlined above is not a way to "overcome resistance to a SOA governance project." Rather, it is a

Table 1 — Relationship Between TOC TP Tools and Layers of Resistance

TOC TP Tool	Layer of Resistance
Current Reality Tree (CRT)	Layer 1: You work to understand the root problem.
(Evaporating) Cloud	Layer 2: You work to find a real solution.
Future Reality Tree (FRT)	Layer 3: The solution produces the desired effects.
Negative Branch Reservation (NBR)	Layer 4: The refined solution blocks or mitigates undesirable effects.
Prerequisite Tree (PrT)	Layer 5: The strategic plan helps overcome the obstacles.
Transition Tree	Layer 6: Procedures are devised to clearly involve others.

method for understanding real obstacles to achieving your company goals, finding good enough solutions (value), developing a plan everybody is committed to executing, and arriving at a clear understanding of how everybody fits together and what has to be done in order to make the solution happen.

Although risk and value are two sides of the same coin, the value side gives us the chance to exploit business-oriented SOA alternatives, innovate, and really nail problems with solutions that everybody cares about. When business cares about it, we can then fill the governance holes with responsibility. As Cutter Senior Consultant Christopher Avery says, "Taking responsibility is the first step in leading others and solving problems." When most of us act responsibly, then our processes will be effective, and good governance can truly help us to achieve the goals of the company.

CLOSING REMARKS

If nobody cares about SOA governance within your company, it could be because SOA is still perceived as an IT issue. In this context, SOA governance means "managing the development and infrastructure processes related to SOA" — which *does* make it sound solely like an IT issue. Show them why it is a company issue using business terms, not technical jargon.

If you are trying to involve people to better achieve an SOA initiative and they claim they need governance processes (the ones that may not be in place), then you are facing resistance. A good understanding of the resistance you face is a means of improving solutions and processes, but it may require you to develop some new skills and be open-minded. Use the TOC TP tools to face each resistance level, improve the processes and solutions, and induce responsible behavior. Use them within your governance process to establish a better decision-making process.

SOA governance is important if it solves some of the problems or constraints that are preventing your company from reaching its goals. Do you have them clearly identified?

REFERENCE

1. Scheinkopf, Lisa J. *Thinking for a Change: Putting the TOC Thinking Processes to Use*. The St. Lucie Press/APICS Series on Constraints Management. CRC, 1999.

ADDITIONAL SOURCES

Avery, Christopher. "Responsible Change." Cutter Consortium Agile Project Management *Executive Report*, Vol. 6, No. 10, October 2005.

Avery, Christopher. "Responsibility Redefined: IT Leadership." Cutter Consortium Agile Project Management Webinar, 18 August 2005.

Debernardo, Héctor, and Margarita Hurtado Hernandez. *El Puente*. Ediciones Granica, 2006.

Goldratt, Elihayu M. *It's Not Luck*. North River Press, 1994.

Goldratt, Elihayu M. *Theory of Constraints*. North River Press, 1990.

Hazra, Tushar K. "Employing Sourcing Through an SOA Model: Utilizing Pragmatic IT Governance for Improvements." Cutter Consortium Sourcing & Vendor Relationships *Executive Update*, Vol. 7, No. 9, September 2006.

IT Governance Institute (ITGI). COBIT 4.1 (www.isaca.org/cobit).

Jensen, Bill. *Simplicity: The New Competitive Advantage in a World of More, Better, Faster*. Perseus Books Group, 2001.

Rosen, Mike. "Theory and Practice of SOA." Cutter Consortium Enterprise Architecture *E-Mail Advisor*, 31 May 2006.

Weill, Peter, and Jeanne W. Ross. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business School Press, 2004.

Jorge V.A. Ronchese is a Senior Consultant with Cutter Consortium's Business-IT Strategies and Enterprise Architecture practices. He is also CEO of Suivant, an Argentina-based business-IT consulting firm, where he focuses on innovative ways to help IT departments in their improvement efforts and to solve complex business issues with adequate technology. He is an international speaker and well-known consultant throughout South America in such subjects as continuous improvement processes, IT governance, enterprise architecture, ITIL, and the Theory of Constraints as applied to the IT area. Mr. Ronchese's wide view of IT management has inspired the Suivant Unified Model (SUM), which focuses on the evolution of IT management and related best practices. He has developed the Future Value Tree (FVT) thinking tool, which is a simple graphical method that enables organizations to make tangible the value of projects and sustain successful business cases. He has served as VP of HSO Business Systems for Latin America; Pre-Sales Manager for South America of Ariba, Inc.; and Regional Manager of MRO Software, Inc.

Mr. Ronchese has a degree in systems information engineering from the Universidad Tecnológica Nacional of Rosario, Argentina, where he was also a professor. He has undertaken postgraduate studies in EU laws, politics, and economy (University of Padova, Italy); IT management (University of Belgrano, Argentina); and business management (University of Bologna, Italy). Mr. Ronchese can be reached at jronchese@cutter.com.

The Key to SOA Governance: Understanding the Essence of Business

by Keith Swenson

Kaname is a Japanese term meaning "essence." In a Japanese fan, the bottom piece that keeps the fan together is the kaname. The kaname of a business is what keeps it all together, what defines it, its essence. The kaname of a business must be identified so that all activities influencing the kaname can be identified and improved. This decomposition into individual activities, into "business services," is the first step in realizing the benefits of a service-oriented architecture (SOA).

The promise of an SOA is a modularized system that can flexibly address changes, enhance business performance, and enable business agility in an increasingly competitive corporate climate. But determining which business services need to be created, how they should function, and how the hundreds or thousands of individual services should be managed can be a daunting task. Do all the services actually work together to support the essence of the business?

And once the business services are created, further complications arise. Which businesses are using what services, and which services are available to them? Are the right people accessing the right services? Are there rules for changing, validating, and approving services? If a service is changed, who and which other services will be affected? How can things be fixed if something goes wrong? Are the service levels meeting the quality of service (QoS) requirements?

Answering these questions requires adequate governance. Governance is about visibility and accountability and involves the processes and policies of both business and IT. In an SOA, governance becomes even more important, as reusable, autonomous components are created to consume or be consumed by other reusable, autonomous components, potentially from other vendors. Issues such as security, reliability, and availability are also important as mission-critical applications and business processes are being developed using SOA.

Determining the kaname of a business and putting in place adequate SOA governance requires not only technology, but also a shift in the way business and IT work together. Companies need to adopt clearly defined roles within their organizations, allowing the stakeholders to understand each other's goals and tasks. Only then can they understand the essence of the business and put the proper governance mechanisms in place for optimal SOA performance. Without the support and participation of IT architects, managers, and development teams, an SOA initiative is likely to fail. By working together, however, this cross-functional group can develop a sound strategy, best practices, and a methodology that can help design a flexible SOA to address change and optimize reuse.

DETERMINING KANAME

The first step to creating an SOA is to understand the essence of the business so that a clear vision of what the SOA will be and what value it will provide can be established. Too often, companies rush to implement an SOA without clearly identifying the business value or the ideal end state. Once expectations are misaligned, the success of the overall SOA implementation is jeopardized.

To reach this clear vision and build the many services that support this vision, companies need to understand both the human aspects of an SOA transformation and the lifecycle management of the services. This can be done by establishing a roadmap for processes and policies and by clearly defining the stakeholders and required architectural standards. To do this, many organizations create a center of excellence or similar cross-functional group to provide resources and guidance, to serve as a repository for best-practice information, and to operate tools that support the SOA governance process. The main function of such a group is to share thoughts, experiences, and knowledge. By improving the way they relate to each other and communicate, group members are more likely to succeed in understanding the kaname, attaining the right vision for the SOA, and establishing adequate governance to ensure success.

A recent article in *Bank Systems & Technology* describes Bank of America's approach to governance and quotes Bill Conroy, the bank's enterprise architecture senior business executive:

> Bank of America delegated governance for SOA to the four CIOs within the organization, the bank's Conroy says. An architecture council, which Conroy chairs, made up of the CIOs controls new technology and new products. "We're very controlled on the products we let in — we let in on a very specific business case," he relates, adding that new technology has to pass a litmus test: It has to drive a significant amount of revenue, reduce a significant amount of risk or clean up the environment. No new technology can be purchased throughout the bank if the council has not approved it, Conroy stresses. [1]

Companies need to build an architecture roadmap with SOA governance woven into the fabric of daily operations, using both a top-down business approach for understanding business goals and a bottom-up IT approach for building individual services (see Figure 1). This requires new organizational models for bridging the gap between business and IT; it also means being able to visualize the fundamental structure and behavior of the business to understand what changes occur, quickly identify the changes, and just as quickly adapt the system to the changes.

At this stage, management support is most crucial, since management drives IT strategy. Therefore, a clear business and financial case that accurately maps to the reality of the business is essential. To achieve this, the stakeholders must:

- Reach a consensus on the essence of the business
- Visualize and model the business using process simulation to estimate efficiencies
- Construct the system so it closely reflects the business model in an architecture that is responsive to changes

VISUALIZING THE ESSENCE

The essence of a business can be identified only through a common understanding of the business objectives (increasing revenue, complying with regulations, contending with globalization, promoting corporate and social responsibility, etc.) and the related IT systems (see Figure 2).

A primary goal at this stage is to understand how people work, who owns what responsibilities, and which interdependencies link business processes and IT resources. The goal is for business people to discuss and agree on the business elements of an application and for the IT people to discuss and agree on how to manage the technological underpinnings. The outcome of such discussions should form the roadmap for the SOA solution. The value of this approach is that it doesn't start out focused on IT and the integration of systems, but instead focuses on understanding what the business is or wants to be — in other words, the kaname.

BUILDING UPON THE VISION

Once an architecture roadmap of the solution is agreed upon, processes and services that align to the business can be built out. This includes not only processes that govern how systems interact, but also processes that

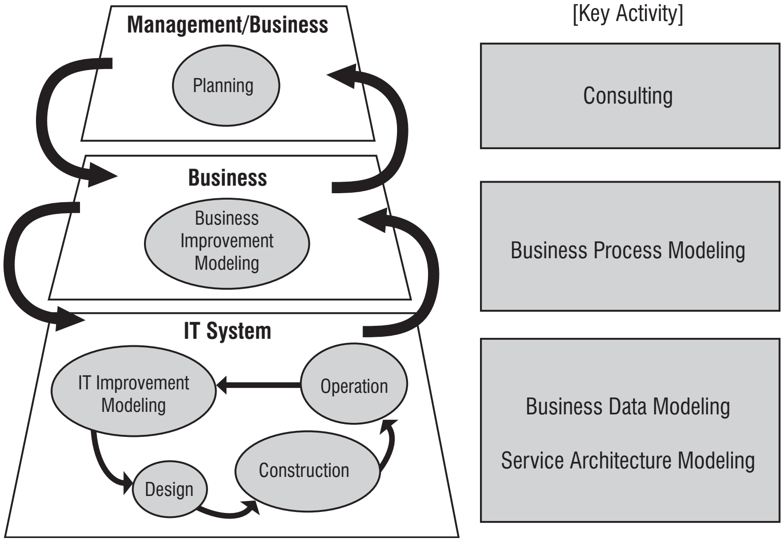


Figure 1 — Business goals drive IT models.

govern and reflect how people work — the human-centric business processes. A process-driven approach uses modeling tools targeted at the business analyst, tools such as Visio and IDS Scheer's Business Architect, or business process modeling (BPM) vendor-specific design environments that not only include a BPM notation–based modeling tool, but also:

- Execution of the model
- Process performance monitoring and analysis of the actual running business process or a simulated business process
- Optimization for interactive process improvements

These tools enable the business to define the human factors involved in a business process, and these activities can then be decomposed into reusable, autonomous services that can be orchestrated and delivered back to the modeling tool.

The actual integration hooks can be created using the Business Process Execution Language (BPEL) for service orchestration or an Enterprise Service Bus (ESB) for mediation. For example, consider an online store that uses a credit card company, a shipping company, and various suppliers of the goods sold. The basic business process of selling goods, getting paid for them, and shipping them may be fairly straightforward. But the details — which companies to deal with, what data formats they use, and the precise Web address at which they can be reached — are very likely to change on short notice. An IT professional can augment the business process with execution details using an ESB so that these differences can be resolved as they become apparent, without any Java programming or even recompilation and library deployment.

The challenge is for business people to transform their vision into processes that can be easily deployed by IT while leveraging their existing systems and infrastructure without significant rework. A combination of BPM and an ESB does just that. BPM promotes a model that ensures business analysts can define processes in business terms, resulting in processes that nontechnical users can own and manage throughout the process lifecycle. This saves time and minimizes rework.

SOA: READY FOR PRIMETIME?

An SOA supports reusable services that perform business functions and provide an excellent foundation for implementing process-driven integration scenarios that solve complex business process management and orchestration problems. Just as business processes can leverage the services within the enterprise, these same business processes can also be exposed as services to be consumed within applications. The end result is that BPM becomes part of the SOA fabric, in which the business processes are viewed as nothing more than a new kind of service. These reusable services may reflect business tasks, such as opening a checking account, verifying a credit card transaction, or processing a purchase order.

Let's take a simple example. Say you are a provider of a leading business rules management system (BRMS) in which complex rules make risk assessment

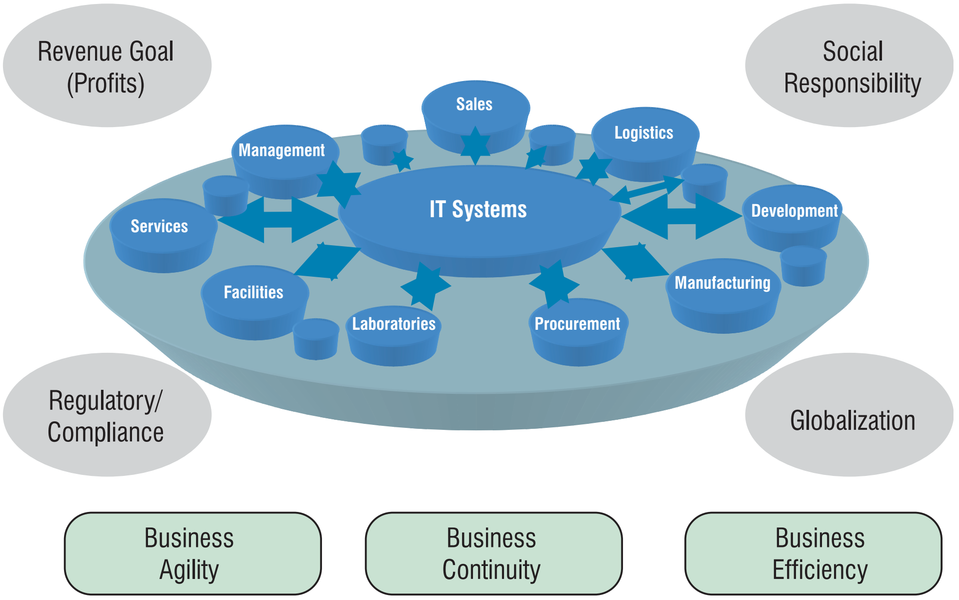


Figure 2 — IT resources provide value by linking the various departments in an organization.

decisions. In order for an application or BPM product to use an existing complex rule, it needs to be published or cataloged so it can be discovered by rule name, description, availability, cost, or other attributes. In the old days, this kind of functionality was buried in legacy code. Any change to the rules was delayed by lengthy IT development cycles. The SOA development approach focuses on leveraging the investments that organizations have already made in technology by providing the tools required to expose both business logic and data in existing or new systems in a standardized way. A service, such as "risk rating," takes into consideration many complex business rules to generate an individual's risk rating. (For example, if someone has poor credit, has a payment history [days outstanding] of greater than three months, and has defaulted on a number of loans, then his or her risk rating will be high.) When someone applies for a loan, the loan approval business process could make a risk rating service request on demand.

Good governance reduces the risk of mismatched services and redundant development efforts.

Most important, if these complex rules need to change, the BRMS can manage the business rules independent of the business process or application, allowing policy managers to make changes to the rules without heavy IT involvement. As for the business process, it doesn't need to know about the change if it doesn't affect the agreed-upon service.

Leading BPM vendors have already recognized the need to expose their functionality as services for reuse. By structuring applications in this manner, IT assets become more agile, and organizations are better able to align their investments in dynamic business environments. A business analyst, integration expert, or developer can then use the BPM tool to snap together business processes exposed as services to create new business processes, thus reusing existing investments to create new value across the enterprise.

THE NEED FOR GOVERNANCE

As BPM and SOA provide a way to respond quickly to changing business requirements, businesses need to quickly discover, manage, monitor, and analyze the use of SOA artifacts through a centralized SOA registry/repository. Such a repository provides a control point for governing service availability, versioning, and compliance with internal and external systems.

Take, for example, the business process just described. An existing asset is consumed, but that existing asset happens to be a business rule from a different system that is exposed as a service. What if this service utilized another business process and so on and so on? You would find the relationships shown in Figure 3.

Now imagine you have hundreds of services calling other services that might be legacy code, business rules, ESB sequences, or others. How do you know what is actually happening at runtime; who is actually using the service; and what security enforcement, performance enforcement, and availability enforcement need to be in place? You could easily end up with the situation shown in Figure 4.

An SOA repository provides a mechanism for keeping track of an organization's SOA assets, including all the dependencies and relationships. Services can be published into the repository, listing them according to categories that make sense to the organization, and they can then be discovered via a search mechanism. Services are associated with other artifacts that are described through dependency relationships and grouped through an extensible taxonomy. Good governance reduces the risk of mismatched services and redundant development efforts (see Figure 5).

As more and more demands begin to build, so will the importance of SOA management and governance, especially when multiple providers of services make changes. SOA repositories should allow users to subscribe to any SOA artifact and be notified via callouts or e-mail of any changes to that artifact.

In addition to cataloging services to enable reuse, an SOA repository also, through corporate design standards, encourages the use of common guidelines so service development remains consistent among different architecture groups within an organization. For

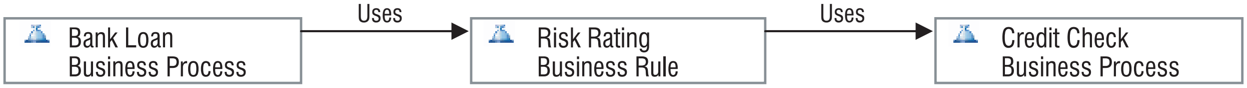
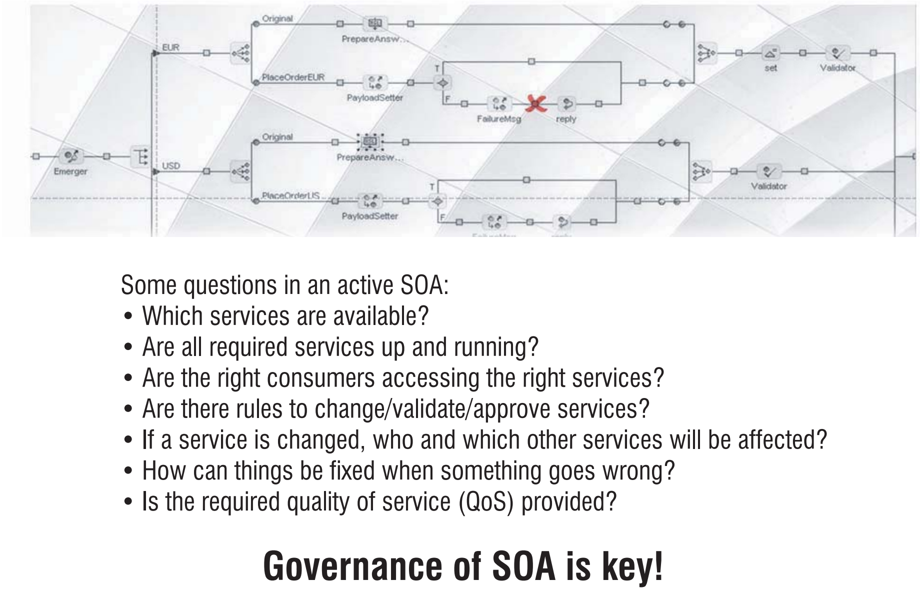

Bank Loan Business Process
Uses
Risk Rating Business Rule
Uses
Credit Check Business Process

Figure 3 — A service usage graph.

example, consistency can be maintained throughout a service lifecycle, from conception through production, by implementing appropriate approvals and attaching appropriate documentation, specs, and test plans at each phase. Businesses should be able to configure the lifecycle of not only services, but also business rules, business processes, and basically any SOA asset through a set of lifecycle stages and states. The authorization and approvals required as a service moves through the different stages and states also need to include roles and privileges.

The SOA repository provides storage for metadata for services and any other artifacts related to an SOA asset, including WSDLs, XML schemas, XSLT stylesheets, policies, and so on.

Once the services are invoked, organizations need to keep track of how they are being used. Real-time tracking of services includes monitoring of performance, availability, usage, and more, so that alerts and adjustments can be made. Governance becomes important to ensure that a service is functioning according to defined



Some questions in an active SOA:

- Which services are available?
- Are all required services up and running?
- Are the right consumers accessing the right services?
- Are there rules to change/validate/approve services?
- If a service is changed, who and which other services will be affected?
- How can things be fixed when something goes wrong?
- Is the required quality of service (QoS) provided?

Governance of SOA is key!

Figure 4 — A service orchestration diagram.

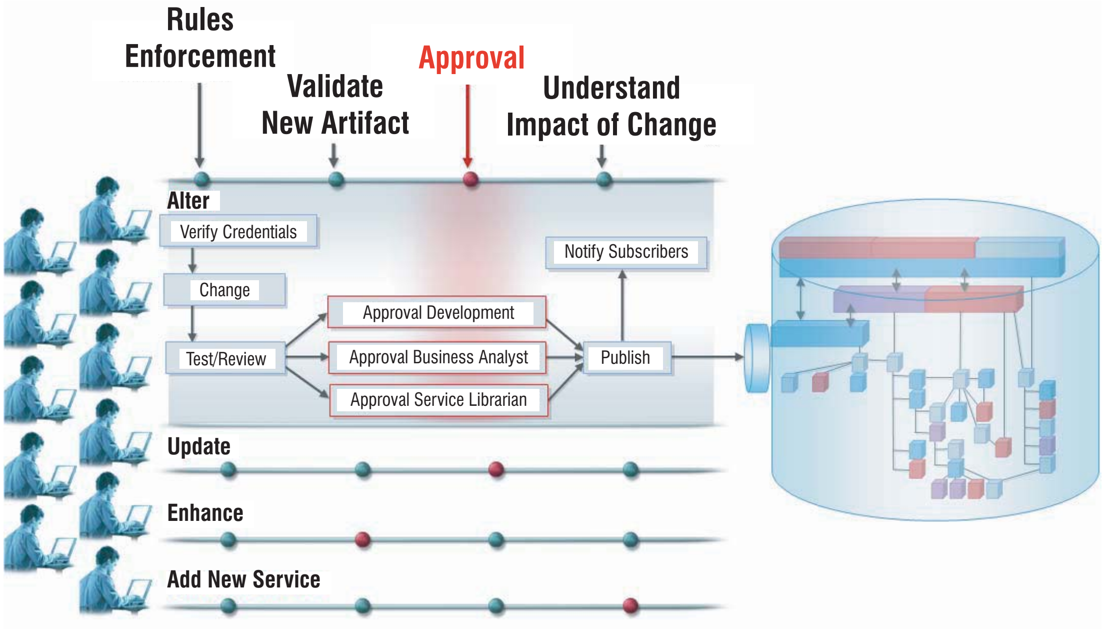


Figure 5 — The governance process is key to ensuring that only high-quality services are listed in the repository.

service levels. If not, a notification occurs to change the way the service runs. Services should be associated with service-level agreements that define the required performance characteristics of the services and spell out penalties for failures.

> **The true benefits of an SOA are achieved when top-down business goals meet bottom-up IT development, with an SOA governance and management solution that seamlessly joins the two.**

Services also require security levels to control who can use which service for what purpose. This involves defining and verifying users who are authorized to use certain services. For example, a company might define a class of users who can transfer more than $1 million across various accounts. Also, the privacy of sensitive data must be protected to meet internal and external compliance regulations and standards that require a complete audit trail. For example, the US Health Insurance Portability and Accountability Act (HIPAA) requires all transactions to be SSL-encrypted.

While establishing adequate governance is not trivial, the tools do exist to provide the operational-level visibility needed to make solid governance possible. The SOA development team must ensure these tools are in place.

Operational-level visibility is also important to understanding how to manage the services at the business level. For example, should a gold-level customer have priority over other customers for certain transactions? Should there be two levels of service offered at basic and premium prices?

Because an SOA repository maintains the dependency relationships of services and associates all services with their artifacts, such as policies, it provides a level of visibility that enables organizations to adapt quickly to change while minimizing risks. This results in greater business agility and cost effectiveness.

IT ALL BEGINS WITH KANAME

While SOA governance requires an investment in people and technology to establish the appropriate context for an SOA, the benefits of this investment are tremendous. The true benefits of an SOA are achieved when top-down business goals meet bottom-up IT development, with an SOA governance and management solution, such as an SOA repository, that seamlessly joins the two. The goal of any SOA initiative is to improve communication, processes, and efficiency within an enterprise to deliver superior products and services at lower costs. But the start of any SOA initiative should be better collaboration and communication between business and IT on objectives, processes, implementation plans, and optimization. Only then can the kaname of a business be understood. And only when the kaname is understood can an organization develop a consistent vision and methodology throughout the process and ensure that the services composing the SOA will optimally support the business.

REFERENCE

1. Feig, Nancy A. “New Foundation: SOA Implementation and Bank Transformation.” *Bank Systems & Technology*, 30 March 2007 (www.banktech.com/news/showArticle.jhtml?articleID=198700451&pgno=4).

Keith Swenson is VP of Research and Development in Fujitsu Computer Systems Corporation’s Enterprise Software and Solutions Group. He is known for having been a pioneer in Web services, and he has helped the development of standards such as WfMC Interface 2, OMG Workflow Interface, SWAP, Wf-XML, AWSP, and WSCI. He is currently working on standards such as XPDL and ASAP and is the chairman of the Technical Committee of the Workflow Management Coalition. Mr. Swenson can be reached at Fujitsu Computer Systems Corporation, 1250 E. Arques Ave., Sunnyvale, CA 94085, USA; Tel: +1 800 831 3183; E-mail: KSwenson@us.fujitsu.com.

SOA Governance: Building on the Old, Embracing the New

by Tushar K. Hazra

A plethora of publications today emphasize the significance of business drivers in adopting service-oriented architecture (SOA). It is also evident that embracing SOA requires significant (if not equal) investments of time and energy on the part of decision-making business leaders and IT practitioners. Hence, practitioners need a mechanism (aka governance) to consistently orchestrate and execute the activities that can help them manage, monitor, and maintain SOA initiatives, while aligning their business goals with IT services and solutions. This mechanism consists of two primary elements: a methodology defined by the business and IT leaders and a process that can be automated to employ the mechanism in practice.

The first part of this mechanism is usually substantiated by a set of procedures, policies, industry standards, and best practices. For the second part of this mechanism, many leading software vendors today either directly or indirectly support SOA governance with their tools. I strongly believe, however, that SOA governance cannot be achieved by simply using commercial tools. I think it is important to revisit the basics of IT governance already in place for many midsized and large companies. In my opinion, IT governance provides the foundation for enterprise architecture (EA) and SOA governance. I submit that it is essential for us to recognize: (1) the uniqueness of the mechanism, (2) major complexities and challenges involved in using the mechanism, and (3) the processes practitioners plan to use in order to leverage the mechanism, before getting started with SOA governance. Most available tools can then enable practitioners to expedite the performance of their governance activities.

ORGANIZATION

Over the past five years, I have assisted many practitioners from a number of midsized and large companies and US government agencies in establishing their SOA governance. During this period, I have witnessed few distinct situations in which visionary leaders opted for a “strictly” consensus-driven approach to defining or formalizing their SOA governance. In most other cases, it was not so much about formalizing the SOA governance as recognizing the practical challenges associated with it — and then making decisions to transform the organization’s existing governance principles to accommodate SOA. In this article, I will draw upon my own experiences to address five questions related to SOA governance:

1. What is unique about SOA governance?
2. What must be done to transform corporate IT governance in order to deal with SOA governance?
3. How does SOA governance handle most changes involved with business (in practice)?
4. What are some of the policies, principles, and best practices known today for SOA governance?
5. What do most available tools offer that can expedite this governance?

WHAT MAKES SOA GOVERNANCE UNIQUE?

In order for us to obtain a clear, consistent, and yet pragmatic perspective on SOA governance, I feel compelled to review some terms (see Table 1). While it may be difficult to acquire a consensus amongst practitioners about these definitions, at least they provide a baseline for discussion.

I have observed from experience that SOA governance can coexist with corporate, IT, and EA forms of governance. In some cases, it has grown organically from IT and EA governance. In many cases, companies have their SOA governance interact with existing EA governance and support overall IT governance.

Figure 1 presents a simplified, high-level view of the connection SOA governance has with EA, IT, and corporate governance. As SOA is more of a business-driven approach, it has direct responsibilities to recognize the business strategies, values, and goals of an

Table 1 — Governance-Related Terms

Governance	The word "governance" is often used synonymously with "government." Therefore, fundamentally it is more of a political concept than a technical or business-oriented one. Many dictionaries define governance as an exercise of authority or control — a method or system of government. Some definitions emphasize the people and process aspects, too.
Corporate Governance	Wikipedia defines corporate governance as a set of processes, customs, policies, laws, and institutions affecting the way a corporation is directed, administered, or controlled. Corporate governance typically includes the relationships between and among various stakeholders and the goals for which the corporation is governed. The principal stakeholders are the shareholders, senior management, and the board of directors. Other stakeholders may include employees, suppliers, customers, banks and other lenders, regulators, the environment, and the community at large.
IT Governance	Wikipedia defines IT governance as a subset discipline of corporate governance. IT governance focuses on information technology assets and systems and deals with their performance and risk management issues, as well as compliance with various government regulations and industry mandates. In their famous book *IT Governance*, Peter Weill and Jeanne W. Ross define IT governance as "specifying the decision rights and accountability framework to encourage desirable behavior in using IT" [2].
EA Governance	For most midsized and large companies, EA governance builds upon the corporate and organizational culture in delivering corporate business value. In general, EA governance primarily presents new or future technology blueprints and enterprise-wide reusable standards and develops EA compliance policies and procedures. EA governance helps in the review of service-level and contractual agreements with vendors and IT partners and recommends capital investments for IT to the company's senior leadership.
SOA Governance	According to Wikipedia, SOA governance is an emerging discipline that enables organizations to guide and control their SOA initiatives and programs. I suggest that SOA governance provides a key element for the success of any SOA initiative. It governs the interaction between business and IT teams (as well as between two or more IT teams) as they incorporate distinct policies, procedures, and best practices that are applicable to services throughout the entire SOA initiative lifecycle.

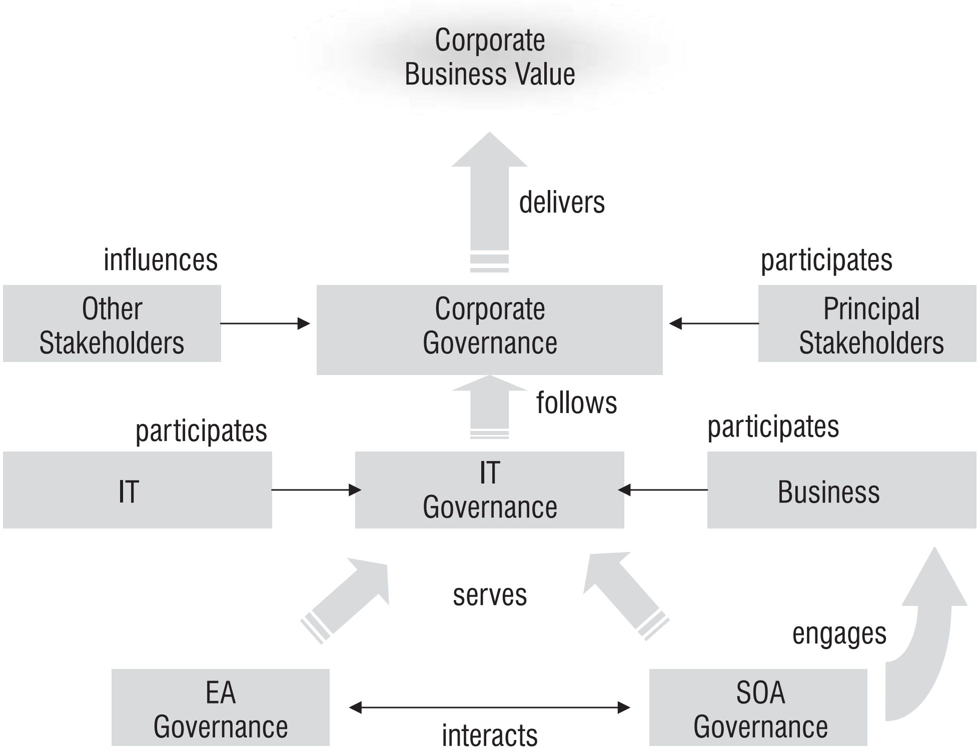


Figure 1 — A simplified view of various governance structures.

enterprise. So what really makes SOA governance unique? From a business perspective, SOA governance emphasizes guidance and control of the pay-per-use concept of invoking a service, while formalizing the policies and procedures that enable service consumers and providers to connect with each other and conduct business using a set of standardized principles. Another aspect of this pay-per-use concept relates to discovering, identifying, and procuring a suitable service, and then subsequently determining if the service provider is reliable enough to be trusted for use.

Once a service is procured, other issues such as versioning, upgrading, or retiring a service must be addressed as well. When dealing with internal service providers, this part of formalizing the contract or service-level agreement (SLA) could be relatively simple. With external service providers, the contract may not be so easy to formulate. From a technical perspective, SOA governance focuses on the service design, development, and deployment aspects of SOA. It also relates to formalizing the coupling and interfaces between services. Therefore, SOA governance is unique in the way it concentrates on the impacts of distributed services across multiple (or a single) business organization(s) based on specified SLAs.

First off, SOA governance provides transparency in the usage of a service. This means a service consumer can search for, discover, and locate a desired service and obtain consistent service (response) from any specific service provider. Second, SOA governance offers the business flexibility in choosing a service from a multitude of services catered to meet the same business requirements with the desired quality of service (QoS). And finally, SOA governance offers a mechanism to help practitioners carry out three major activities throughout an SOA initiative lifecycle ("lifecycle" here includes design as well as runtime use of SOA):

- Providing oversight of associates or employees involved in business and IT alignment efforts
- Maintaining continuity of business operations and/or supporting disaster recovery projects for the entire organization
- Reducing the associated cost of operations while existing infrastructure is transformed to support SOA

After looking at both business and technical perspectives, I submit that SOA governance is really unique because of its theme — service, and more specifically business service. For SOA governance, a business service must be the "unit of work." And SOA governance will be unique only if the practitioners can harness the power of existing kinds of governance while reusing business services across the business organizations, and, subsequently, across the enterprise.

CONNECTIONS WITH EXISTING FORMS OF GOVERNANCE

In their famous book *IT Governance*, Peter Weill and Jeanne W. Ross define IT governance as "specifying the decision rights and accountability framework to encourage desirable behavior in using IT" [2]. According to the authors, use of IT encompasses achieving corporate performance goals. I interpret the concept of desirable behavior in terms of people and how the delivery of business value is promised, proposed, and then delivered by IT. I visualize the decision rights and accountability framework as a part of the process and methodology instilled in the enterprise by the governance body. Finally, I find that available tools and techniques can facilitate delivering this desirable behavior.

In a recent Cutter Consortium *Executive Report*, I discussed EA governance in terms of its significance for people, process, and technology [1]. Practitioners use EA governance to go beyond their strategic principles (i.e., "doing the right things") to actually delivering effective solutions (i.e.,"doing things right") for their business operations. The EA governance board (an organizational structure with members from various business and IT organizations, and the senior leadership team) usually formalizes a charter that includes a set of EA principles, frameworks, reference models, and standards. The EA governance mechanism can offer the appropriate practitioners a means of monitoring the compliance of a specific organization with these principles, frameworks, reference models, and standards. Furthermore, the metrics associated with EA governance can be directed toward influencing enterprise application projects to follow the established principles while delivering efficient business solutions.

In working with IT and EA governance teams, I observe that SOA governance has multiple links to the existing forms of governance:

- **IT governance focuses on utilizing IT assets to deliver business value.** Subsequently, EA governance has gotten more specific about utilizing standards, best practices, and IT policies to deliver enterprise-wide business value. SOA governance is a natural extension of and progression from both IT and EA governance. It delivers business value by focusing on business services.

- **For most companies, IT governance is heavily influenced by the business organizations.** To be successful, EA governance also engages business organizations from the early stages of its lifecycle. Of course, SOA is a business-driven initiative to begin with, so it is not necessary for practitioners to make SOA governance a brand-new affair. Nevertheless, SOA governance must have more business influence and ownership than IT governance.
- **IT and EA governance mechanisms have evolved over the past few years.** In comparison, SOA governance is a relatively new concept. Therefore, practitioners can leverage their experience with existing governance concepts to build effective SOA governance.
- **Without a doubt, SOA harnesses a new "service-centric" mindset.** However, the need for IT and EA governance structures is not going away anytime soon. Unless and until all the business functions in an enterprise can be delivered as services (which is very unlikely), there will be other IT assets that must be governed using IT and EA governance policies.

In the final analysis, SOA governance will always be linked with IT and EA governance principles, policies, and best practices — if not with their organizational structures. And in most companies where stable IT and EA governance practices already exist, practitioners will benefit from their prior experience with these existing types of governance. Over a period of time, practitioners can transition their current governance principles to create an SOA governance mechanism.

FROM THE TRENCHES

As I've mentioned, SOA is a business notion to begin with. Over the past few years, most corporate environments have been undergoing progressive change. Greater competition, faster and more effective interactions with customers, and stricter government regulations and industry mandates are among the factors that are driving the change. Most companies try to cope with these challenges by becoming more responsive, innovative, and agile, and service orientation is the latest means for doing so. The SOA approach offers seamless integration (and interaction) of business services in order to deliver the desired flexibility to the enterprise. Therefore, it is imperative that the inception of SOA governance be business-driven, its vision be business-focused, and its mission be business-oriented.

For most companies, service orientation starts with identifying candidate business functions that can be transformed into business services. Once transformed, these business services can deliver more efficient and effective results. The key impetus for creating these services is that they can be reused and shared across multiple organizations.

As a number of services are created, the need arises to organize and manage these services using a registry or repository. Most companies establish SOA governance to define a mechanism that helps practitioners govern these services. However, it is worth mentioning that the reusability and sharing of services require a change in organizational mindset. This mindset is business service–centric and not technology-driven. As a result, one of the primary responsibilities of SOA governance in practice is directed toward formalizing policies that deal with the existing or new organizational culture to embrace the concept of business services. The next set of responsibilities for SOA governance involves creating policies and principles for service design. Once the services are designed and developed, they can be deployed in any SOA environment — whether it be heterogeneous, distributed, or federated. The SOA governance policies must account for the deployment. Finally, SOA governance must address the challenges of employing services in business operations and necessary business-IT alignments.

During the past few years, while working with other practitioners in the field, I have cultivated a very simple model for SOA governance (see Figure 2). As shown, once the SOA governance policies, principles, and guidelines are created, they are handed over to SOA management. SOA management is responsible for enforcing any necessary changes to existing organizational policies for individual SOA initiatives. Based on experience with individual projects, SOA management collects best practices and instills them in the governance mechanism. SOA management activities are synchronized and supported by the portfolio management office. SOA management and SOA governance work together concurrently and synchronously with individual SOA initiatives. SOA management enforces new governance policies or changes to existing ones on individual SOA initiatives. Meanwhile, SOA initiatives influence changes to SOA governance policies, principles, or practices as appropriate by connecting directly with SOA governance.

In addition to enforcing compliance, this model promotes communication and collaboration between the individual SOA initiatives and the SOA governance.

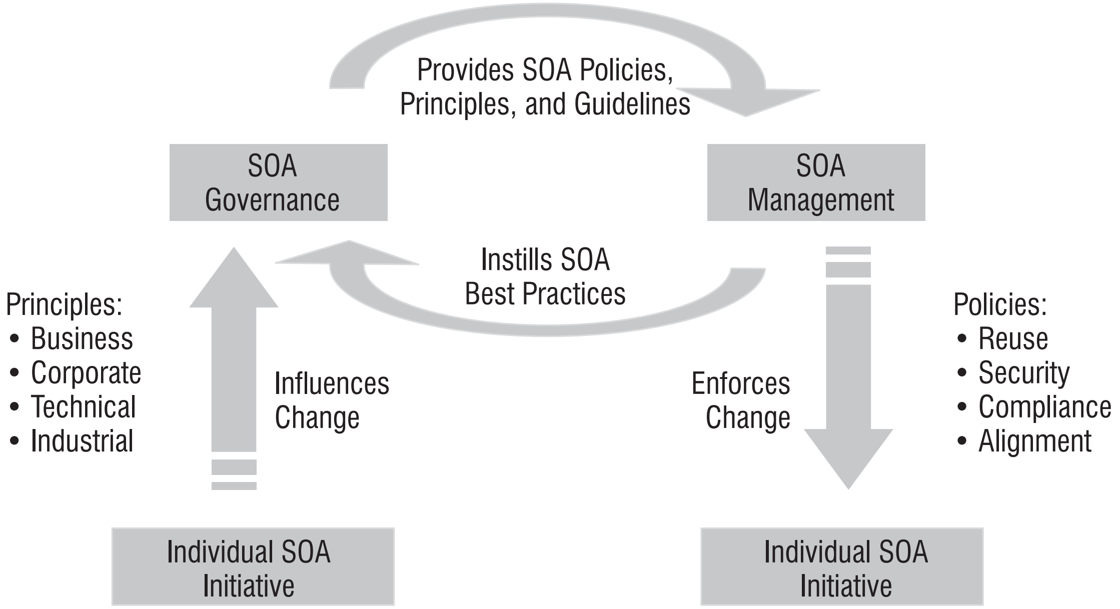


Figure 2 — A simplified view of SOA governance in practice.

This in turn makes businesses more flexible and helps them cope with change together with IT.

POLICIES, PRINCIPLES, AND BEST PRACTICES

From my experience, I have found that companies that are successfully establishing SOA initiatives usually involve both top-down and bottom-up approaches concurrently. The top-down approach allows the company to incorporate strategic business objectives, goals, and visions, as well as to plan for specific business needs. The bottom-up approach involves service development at an organization level. These approaches are then connected or intertwined via SOA governance and management with a set of policies, principles, and best practices to deliver the goals of SOA (see Figure 2). While I am focusing on the significance of SOA governance in this article, I would like to point out that SOA management also plays an important part in SOA initiatives. It is accountable for successfully employing the policies, principles, and best practices across the organization(s).

Policies

I have noticed that most companies working on SOA governance establish policies in four major areas:

1. **Reuse.** In my opinion, this is one of the most important policies. It encompasses not just how the services must be designed to meet certain specific business needs, but also how services may interoperate with each other. As a result, organizations can avoid creating multiple versions of the same service. Such a policy can also make sure that existing services are used before new services are built to meet the same or similar business needs.
2. **Security.** This is another important policy that defines how a service can be accessed, who can access the service, and when one can access the service. This policy also prescribes the security standards that must be incorporated while designing and developing services. In some cases, the security policy also identifies how various security-related risk factors can be recognized and the measures that can be taken to mitigate such issues.
3. **Compliance.** There are at least two major compliance aspects addressed by this policy. One aspect is related to various government regulations, industry mandates, and business constraints that the organization must follow (business point of view). The other aspect is related to various industry standards other practitioners follow in designing and developing services (technical point of view).
4. **Alignment.** This policy is related to building the collaboration and understanding between business and IT. While most companies have a strategic plan in place for establishing business-IT alignment, it is important to lay out a set of rules or contracts between the business and IT while building business services to be utilized in an SOA environment.

Most practitioners prefer to refine these policies iteratively and incrementally over multiple SOA initiatives.

Principles

SOA governance principles are mostly guidelines that are established to support various SOA-related activities across the business organizations. Obviously, these principles are directed toward achieving certain specific objectives. I categorize them under four major focus areas:

1. **Business goal–specific.** This set of principles is essential to ascertaining the capabilities that IT must possess in order to address strategic business goals. These principles are primarily focused on supporting the business management teams in making their decisions. They consist of processes to utilize resources effectively to continue business operations, to support disaster recovery, and to meet government and industry regulations.
2. **Corporate process–specific.** These principles are related to corporate policies, procedures, and standards relevant to formalizing contractual agreements with service providers (internal or external partners). Many companies start with a small set of contracts (i.e., SLAs) that ultimately encompasses the evolving needs of services enterprise-wide.
3. **Technology-specific.** These principles concern the architectural and other IT asset–related requirements for an SOA initiative and how organizations must embrace them. I have seen companies create technical principles formalizing the level of reuse, granularity, and coupling of services. I have also noticed companies creating metrics to measure the way services are used or shared across the organization(s). Some companies also consider a set of technical principles to integrate existing technology investments with SOA.
4. **Industry-specific.** These principles are important for establishing mechanisms to share services across multiple business entities and also to create composite services from multiple basic services. In some cases, practitioners utilized these principles to manage the entire lifecycle of a service and measure the service performance as appropriate. In my opinion, industry-specific principles are most useful in building a set of frameworks that can be reused by multiple organizations across an industry for deploying repeatable business services with minimal or no significant changes.

I find these SOA governance principles extremely useful in spreading the use of SOA across multiple lines of business (where the same business function can be replicated as a service) or across the same business domain (where one service can be used multiple times) while complying with a set of consistent business rules.

Practices

Earlier, I mentioned that it is imperative that the inception of SOA governance be business-driven, its vision be business-focused, and its mission be business-oriented. I would like to substantiate my comments with a set of best practices that I have collected from the trenches over the past few years:

1. **Get business engaged from the beginning.** Since business owns the processes and the functions that are potential candidates for services, it is essential that the business drive the SOA governance. Most companies start their SOA governance small and then let it evolve as the business requirements grow.
2. **Establish regular business and IT collaboration.** Since business formalizes the corporate vision and IT enables the business, frequent collaboration between the two is imperative. The vision of SOA governance must be aligned with the business needs and must support the business vision.
3. **Formulate policies, procedures, and frameworks to support business goals.** Reuse of services is extremely important in achieving such business goals as competitive advantage, cost reduction, and customer satisfaction. SOA governance must promote service reuse to fulfill its mission to deliver business flexibility.
4. **Capitalize on emerging standards incrementally and periodically.** SOA is evolving, and so are the industry standards associated with it. It is essential to take the time needed to absorb the lessons learned in order to better understand the SOA and its governance.
5. **Prepare a set of SOA governance processes and procedures to expedite the change in organizational culture.** While SOA is evolving, existing IT assets and infrastructure can be utilized and leveraged for initial SOA initiatives. Yet the change in organizational culture is inevitable and must not be ignored.

I strongly believe that SOA governance is about (to paraphrase Weill and Ross) "encouraging desirable behavior in using business services." For this to happen, companies must make consistent use of a set of policies, principles, and best practices.

TOOLS AND TECHNIQUES

SOA governance involves a combination of two concurrent functions. The first one allows practitioners to establish the processes that can help them make the right decisions. The other function involves the use of tools to implement the processes defined by the practitioners, so that their decision making can be expedited throughout the service lifecycle (i.e., during design, deployment, and runtime use of a service). Many commercially available SOA governance tools can help practitioners create visual models; monitor service performance; or control, maintain, or modify a service. Another set of tools is available to support using service registries for service registration, discovery, or binding. These registries may also connect to Enterprise Service Bus (ESB) products to support SOA integration with existing technologies. Another emerging area of focus for SOA governance tools is service-level management. These tools enable various activities such as monitoring, optimization, and control of SLA-based use of services that are registered with a registry across the organization(s).

Although most leading SOA vendors claim to offer tools that either directly or indirectly support SOA governance, most of these tools only support a small part of the overall governance mechanism. While assisting various companies with their SOA governance efforts, I have had experience with Systinet's SOA governance tools, webMethods' X-Registry and X-Broker, and LogicLibrary's Logidex. Practitioners using the abovementioned products have achieved satisfactory results in delivering necessary governance support, visibility, and oversight to their customers. Many other vendors are pursuing the development of similar registries and repositories to allow practitioners to enforce and automate policies during the service lifecycle. Some of these products are also worth considering for SOA governance support.

CONCLUSION

Here are some of my observations on SOA governance, which may resonate with your own:

- **It is unique in the sense that it primarily focuses on business services.** However, the concept of governance is nothing new to the business and IT communities. In fact, most of the tenets of corporate, IT, and EA governance are extremely useful for building SOA governance.
- **SOA governance is a practice that is evolving.** While it involves a number of business-driven policies, principles, and established best practices, it is not fully mature. In crafting an SOA governance mechanism, practitioners will want to take advantage of their prior knowledge of and experience with existing governance mechanisms.
- **Most available tools support only a small part of the solution.** While most of the techniques derived or simply employed from these tools offer satisfactory results, they must be integrated with a multitude of practices (i.e., best practices that work in a specific environment) to make them highly effective and scalable in delivering true SOA governance.

It is clear that most of the policies, principles, and best practices pertaining to SOA governance are built upon the premise of leveraging existing business investments and IT assets. However, as the business environment continues to evolve and new technologies emerge, SOA governance will mature, and it will strengthen its already prevalent position in pursuing business-driven SOA initiatives.

REFERENCES

1. Hazra, Tushar. "Getting Your Enterprise Architecture Metrics Right." Cutter Consortium Enterprise Architecture *Executive Report*, Vol. 9, No. 9, September 2006.

2. Weill, Peter, and Jeanne W. Ross. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results.* Harvard Business School Press, 2004.

Tushar K. Hazra, PhD, is a Senior Consultant with Cutter Consortium's Enterprise Architecture and Sourcing & Vendor Relationships practices. In December 2000, Dr. Hazra founded EpitomiOne, a strategic consulting company specializing in EA, SOA deployment, portals, Web services development, model-driven solution delivery, IT governance and service delivery management, and global sourcing strategy facilitation. Dr. Hazra has 16 years of practical experience in developing and integrating service-oriented enterprise (SOE) and mission-critical component-based systems in collaborative environments for the insurance, healthcare, retail, banking, telecom, and aerospace industries.

During the past 12 years, Dr. Hazra has served as CTO and VP of several Fortune *100 companies. Currently, as President and CTO of EpitomiOne, he helps business and IT leaders incorporate their strategic sourcing visions for enterprise-level initiatives while complying with commercial and government standards, best practices, and mandates. As a hands-on practitioner, Dr. Hazra gets involved in everything from building strategies to developing EA and SOA to guiding and deploying MDA-based application integration initiatives. He can be reached at thazra@cutter.com.*

SOA Governance Using the Universal Business Identifier

by Hao He and Brett McDowall

In recent years, service-oriented architecture (SOA) has become the hottest topic in the industry because of its promise of reuse, cost savings, and faster time to market, all of which are very appealing to a highly competitive marketplace. While the promises touted by SOA are not new — as older generations of technologies such as CORBA remind us — the means of achieving those goals through SOA must be fundamentally different. If they are not, SOA will be limited to providing some better plumbing for enterprise application integration.

The recent focus of SOA has been on providing a standard way to integrate existing systems and applications with Web services. Although this is a necessary step toward an SOA, many issues have been encountered as well. For example, it is quite common that many Web services have been created by individual projects but reuse has been very low. Frequently, the major issue is that those new Web services applications lack business context and are system-centric. Even if someone tries to reuse these Web services, he would find it very difficult to understand the business context and system-specific semantics — not to mention the poor interoperability. It appears that the effort needed to reuse these services is still greater than the effort required to reimplement them. Thus, the whole purpose of SOA is defeated.

The answer to the problem described above, according to many, is SOA governance. As one industry researcher observes, "Doing lots of little Web services projects all over the place with no governance isn't SOA, it's just playing" [5].

THE ROLE OF GOVERNANCE AND ENTERPRISE ARCHITECTURE

The goal of SOA governance is to align the business with IT. By providing a decision framework rather than making actual decisions, good SOA governance helps individuals make sound decisions more easily. This leads to consistent results that meet the expectations of the business while at the same time complying with policies, regulations, and other constraints set by authorities. To pull this off, SOA governance must:

- Be able to manage the full lifecycle of services, from their inception, design, development, deployment, and operation to their termination
- Ensure accountability, which means that roles and responsibilities, ownership, and processes must be well defined

However, these are hard things to do. SOA is new, but governance is not, and people try to use the traditional governance approach for SOA. This leads to the following issues:

- **Lack of trust between the business and IT.** The business always feels that IT is lagging behind and is unsure of what IT can and cannot do.
- **Silo development is rampant.** Each business unit does its own development, and there is little or no coordination at the enterprise level.
- **Lack of accountability.** If something goes wrong, it is either everyone's fault or nobody's fault.

Some have argued that enterprise architecture plays an important role in SOA governance because it encompasses both business architecture and technical architecture. It defines the roles and responsibilities for people, who will then make decisions and implement IT systems according to the architecture roadmap. In order to achieve this goal, the following factors are important:

- Providing a common model and shared understanding between the business and IT
- Clearly identifying ownership for each business asset (i.e., anything valuable in a business, including business data, processes, capabilities, and people)
- Promoting transparency across all processes and artifacts

BUILDING A COMMON MODEL BETWEEN THE BUSINESS AND IT

When building a common enterprise model, it is intuitive to think about creating an enterprise taxonomy so business entities can be modeled with a common understanding. However, this is very difficult to do in practice because of the need to agree on a single taxonomy while business continues as normal. The fact is that almost every business domain has its own taxonomy, and mapping between a domain taxonomy and an enterprise taxonomy is a nontrivial task. So instead of working with a common taxonomy, what else can we start with?

If someone has worked in the content management space, she will know that the single most critical element is having a unique identifier for any content to be managed. The importance of this identifier far outweighs that of metadata, because once a piece of content is identified, it can be located and retrieved. Similarly, before describing business entities with a taxonomy, it is far more important to identify each of them with a unique identifier.

This may sound trivial, since surely there is an ID for anything in just about any system. Of course there is, but it is native to each of the hosting systems and cannot be shared outside of the system boundary. In addition, since SOA governance is not just about systems and applications but also needs to work for people, those IDs must be meaningful to humans.

In other words, we want a universally distributed identifier. Coming up with a scheme to create those IDs is not so easy, but luckily this problem has already been solved. The Universal Resource Identifier (URI) was created for exactly this purpose and has proven to be highly successful on the Web [1]. Imagine trying to define a taxonomy or model for information on the Web! Instead, as Tim Berners-Lee has pointed out:

> The most fundamental specification of Web architecture, while one of the simpler, is that of the Universal Resource Identifier, or URI. The principle that anything, absolutely anything, "on the Web" should [be] identified distinctly by an otherwise opaque string of characters (a URI and possibly a fragment identifier) is core to the universality. [3]

Following the same principle, anything, absolutely anything important in an enterprise should be identified by a URI. The immediate benefits are obvious:

- Once a business asset in an enterprise is identified with a unique URI, it becomes an enterprise resource that can be managed.
- As Berners-Lee has written, "Great multiplicative power of reuse derives from the facts that all languages use URIs as identifiers: This allows things written in one language to refer to things defined in another language. The use of URIs allows a language [to] leverage the many forms of persistence, identity and various forms of equivalence" [3]. We can rephrase this as "Great multiplicative power of reuse derives from the fact that all systems, applications, and people use URIs as identifiers: This allows things defined in one business domain to refer to things defined in another business domain. The use of URIs allows systems, applications, and people to leverage the many forms of persistence, identity, and various forms of equivalence."

Before describing business entities with a taxonomy, it is far more important to identify each of them with a unique identifier.

- Once an enterprise resource is given a URI, its ownership is also identified, because the structure of a URI explicitly supports the notation of authority. Clear ownership improves accountability and can be readily reassigned.
- Just as with the Web, an enterprise resource with a URI should be provided with a representation that others can GET (the very simple action with which one opens a URI with a browser). This reduces the enterprise taxonomy problem because multiple representations can be provided at the same URI. Additional semantics can be derived from links to other enterprise resources. This also prompts transparency because enterprise resources are no longer hidden deeply behind a system or systems.
- An enterprise can leverage many of its existing tools and infrastructure elements to manage enterprise resources. This eliminates the need for a centralized service registry and catalogs.
- The common model can be built by resource owners providing links to other resources incrementally. This reduces the burden on a centralized enterprise integration center. More importantly, this allows resource owners to negotiate relationships among each other for maximum profits, which is also an important aspect of SOA governance [4].

To make the URI friendlier in a business environment, let's create a new convention and call it the Universal Business Identifier (UBI). Figure 1 highlights how the UBI provides a standard addressable container for all aspects of the business asset, as against a Web service, which is much more complicated. The UBI approach provides the basis for standardizing how all of the

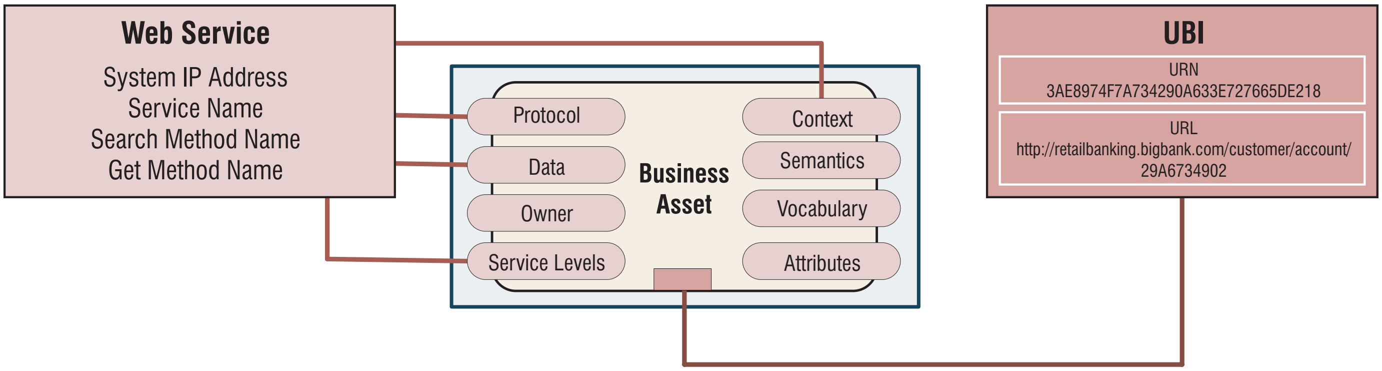


Figure 1 — A UBI provides a standard container for all aspects of a business asset.

different aspects of an asset are handled and, in the same way as the Web, allows convergence on standard protocols, naming schemes, and so on.

By contrast, think about what is needed to access a business asset (e.g., a customer account transaction resource) using a Web service. While the underlying communications and data-handling protocols are well supported by TCP/IP and protocols such as SOAP, there is a still a lot to do to negotiate between the supplier or owner of the business asset and the consumer. The Web service must know a range of system-specific addresses, how to access information on that specific system, and which set of steps to follow to locate the desired resource and then access it.

Of course, there is no magic involved here, and the UBI approach doesn't make these things disappear. It does, however, allow us to provide useful defaults so complexity can be hidden and dealt with where it needs to be. A UBI can have useful and extensible defaults for all the aspects that define a business asset, and both people and systems can go straight to an asset if they have the full address (as in a Uniform Resource Name [URN]), or they can use default search and get properties if they don't.

WORKING WITH THE UBI

Given that a UBI is just like a URI, it would consist of the following parts:

> http://{business sector domain name}.{enterprise domain name}/{business domain name}/{sub business domain name}…/{business resource id}

The first part is a top-level business domain, and the second part is the usual enterprise domain name. The URI path contains a number of sub-business domain names, forming a hierarchy. The last part of the UBI is an ID referring uniquely to a business resource. For example, the following UBI:

> http://retailbanking.bigbank.com/customer/1234

uniquely identifies a customer of bigbank. Depending on who is opening the UBI, the most appropriate representation is returned. For example, if the URI is opened by the customer it refers to, then an HTML page showing the customer's profile is returned. If the URI is opened by a service, an XML document about the customer details is returned. By default, if any part of the UBI is missing, a list or a search view is returned. For example, the following UBI:

> http://retailbanking.bigbank.com/customer/

would return a search form for a bigbank staff member so he can search for a customer. If the same UBI were accessed by a system, it would return an XML document with a list of UBIs to individual customers. A common practice could be that the document only contains a preset number of customers and the list can be paged through by supplying a start parameter, which means that default behavior is both sensible and useful.

Among the good things about the UBI is that it enables partial understanding and free extension — the very elements that enabled the rapid evolution of the Web [2]. People can create their own UBIs. If they don't understand an existing UBI, they can either study it by opening the UBI or just ignore it. Over time, the most-used UBI will become the standard, while the least-used UBIs will simply die out.

Another interesting attribute of UBIs is that they can link to each other just as Web pages do. Let's consider a case in which a customer deposits some amount of money to her account. This simply means that a link is created between two UBIs — a UBI pointing to the customer and a UBI pointing to her account. In a more traditional approach, such links are actually created implicitly in systems but buried deeply in system logs that no one can easily inspect.

With UBIs, the immediate concerns for IT and the business are the same: business resources and their relationships. In other words, business resources and relationships become the first-class citizens of enterprise architecture instead of systems and applications. This decouples the lifecycle management of business resources from that of systems and applications. From the business's point of view, the essence of governance is only to do with the lifecycle management of business resources.

So what does this mean for governance? Standardization is one of the great friends of compliance, the cornerstone of governance. With more visibility, better default behavior, less complexity, and more business-oriented computing solutions, both SOA and enterprise governance become much easier. More interestingly, this results in an enterprise that is like a healthy, self-governing ecosystem — new development and initiatives will concentrate around business assets that deliver real value to the business. Those that don't will be selected out, thus improving the agility of the enterprise. The task of enterprise governance also becomes simpler, as its focus is now on channeling funding to accelerate innovations while doing less policing. After all, the best governance is the one of which the people are least aware, as Lao-Tse said a couple thousand years ago.

CONCLUSION

The need to define every aspect of interfaces, protocols, and models is a facet of SOA that is often ignored. There is a common feeling that interfaces can hide everything, but the reality is that the interfaces must pass data, agree on attribute names, agree on the context of data, and agree on the protocols for using specific services. All of this is a severe impediment to the adoption of SOA and, in a typical enterprise, the major barrier to adoption.

While the ways to transition most effectively to this model are a subject for another day, there is little doubt that the current possibilities offered by SOA and modern Web technologies need to be used in a way that fits in with the business and IT realities of the modern enterprise. Adopting the UBI approach will allow the sort of evolutionary explosion we have already seen with the Web.

REFERENCES

1. "Architecture of the World Wide Web, Volume One." W3C, 15 December 2004 (www.w3.org/TR/webarch).

2. Berners-Lee, Tim. "Evolvability." W3C, March 1998 (www.w3.org/DesignIssues/Evolution.html).

3. Berners-Lee, Tim. "Web Architecture from 50,000 Feet." W3C, September 1998 (www.w3.org/DesignIssues/Architecture.html).

4. Wall, Quinton. "Rethinking SOA Governance." Arch2Arch, 14 March 2007 (http://dev2dev.bea.com/pub/a/2007/03/soa-governance.html).

5. Windley, Philip J. "Governing SOA." *InfoWorld*, 19 January 2006 (www.infoworld.com/article/06/01/19/73698_04FEsoagov_1.html).

Hao He is a Senior Architect and an expert in SOA and Web architecture with Object Consulting, Australia's leader in delivering enterprise business solutions using component-based technologies such as Enterprise Java, .NET, and open source. Dr. He is also experienced in Java/J2EE, XML, and all major XML-related technologies. He has a solid application modeling background using UML and has extensive knowledge of access control, cryptography, and system security. Dr. He is experienced in software design and verification and development process optimization. He is currently a member of JSR 311, which produces a new API for creating RESTful Web services; he was also a contributing member and editor of W3C's Web Service Architecture Group during its existence. Dr. He can be reached at Hao.He@objectconsulting.com.au.

Brett McDowall is an expert in the fields of enterprise architecture, distributed Internet-based systems, and object-oriented systems architecture. He is currently Chief Architect at Object Consulting. Mr. McDowall has been with Object Consulting for 15 years and has over 20 years' experience in the visioning, architecture, analysis, design, management, and implementation of major projects. He is a specialist in telecoms and distributed systems, with experience spanning the insurance, banking, finance, and government sectors. He has also played a major role in the creation and ongoing development of Process MeNtOR, Object's world-class software engineering methodology.

In his current role, Mr. McDowall plays a major part in the architecture and development of enterprise systems. He liaises between business managers, software developers, and project managers at large organizations to help them understand the applicability of new technology to their business. He assists in making recommendations on if, and when, to adopt new technologies and creates appropriate new technology architectures where required. Mr. McDowall can be reached at Brett.McDowall@objectconsulting.com.au.

The Tao of SOA Governance

by Piotr Szabelak and Jan Topinski, with contribution from Borys Stokalski

Enterprise architecture management (EAM) and service-oriented architecture (SOA) mark two disciplines that belong to the IT *plat du jour*. Many organizations across the world are trying to implement SOA patterns and EA practices to reap the fruits promised by both disciplines. A typical implementation path for successful SOA adoption leads inevitably from the world of technology (using services as just another mechanism for implementing functional requirements) into the realm of architecture patterns that support strategic business-IT alignment (see [2, 3] and the "SOA Roadmap Template" sidebar). Yet when participating in large services portfolio implementations in complex, modern organizations, we have seen the gap that opens between EAM practices, such as enterprise release management, and the dynamic process of enterprise services portfolio development. One can experience the gap even in the case of a relatively agile approach to EA, based on iterative releases of enterprise architecture. Arguably, the disciplines in question have different origins that make them the "yin" and "yang" forces in the enterprise.

THE YIN AND THE YANG

EAM as a discipline was founded in 1987 by John Zachman. The client-server revolution and then the Internet resulted in a radical change in perception as to the impact IT has on business. Zachman argued that in order to regain control over the flood of IT investment, we have to understand the relationship between components that constitute the virtual architecture of an enterprise. This means that those who participate in IT investment decision making should be able to answer basic questions related to information assets and projects at various levels of detail. Thus, EA is seen (often rightly) as a "yin" force that subjects innovation to centralized decision making, or (often not so rightly) as a force that trades off creativity and tempo for stability and order.

In contrast, the biggest motivation behind the SOA movement is to provide IT users the means for more productive, faster, simpler business innovations. Services enable organizations to increase the granularity of IT investment, introducing standardization of software design and promoting reuse of functionality — including the functionality that has been long hidden inside intricate monolithic legacy systems. Standardization may sound like another "yin" initiative, but the symptoms of a successful SOA implementation are just the opposite. Once users understand that the services effectively change the complex IT landscape into a set of LEGO[1] blocks that can be quickly combined to deliver value for the organization and its customers, the demand for new services increases radically, often beyond IT's ability to manage it. Thus, SOA becomes the "yang" of corporate computing, promoting innovation, change, and — if not handled properly — chaos.

It is obvious that these disciplines have to be balanced to support a healthy process of enterprise innovation. It is less obvious how to do it. Let us shed some light by reviewing the challenges of SOA governance from the perspective of two key roles in the process: the solution architect and the enterprise architect.

THE SOLUTION ARCHITECT'S PERSPECTIVE

An organization emerges from the initial stages of SOA implementation (i.e., service delivery and service integration) having assembled an early services portfolio — the initial set of "software LEGO" pieces. The set usually consists of a number of business services plugged into a service bus built on top of robust application integration technologies. At this point in the SOA implementation, the organization may already have experienced benefits such as improved productivity in delivering new business functionality (see the "Managed Services Portfolio" sidebar on page 36).

[1] LEGO® is a registered trademark of the LEGO Group.

SOA ROADMAP TEMPLATE

Each milestone in the SOA roadmap template relates to different categories of architecture management issues and results in specific business benefits and opportunities.

SERVICE DELIVERY

Service delivery starts with applying service orientation on the application level. At this stage, services are just another way of delivering software functionality across the network, and the benefits of the investment in mastering the new paradigm are very much application-specific.

SERVICE INTEGRATION

To realize the broader benefits of SOA, it is important to change the focus of service implementation efforts from applications to the enterprise architecture. The first area in which the investment starts to pay off is the reuse of existing application functionality in new, composite applications. This step results in the assembly of service portfolios, defined and published using enterprise-level standards and mechanisms.

SERVICE INTEROPERABILITY

Once a productive SOA environment is in place, assembling new business functionality from existing services will often prove to be cheaper, faster, and more reliable than other approaches. This will lead to the creation of application ecosystems — sets of applications that share a significant portion of their functionality within the confines of the "value network," linking the enterprise with customers and business partners.

SERVICE MANAGEMENT

The effective and efficient collaboration of services across many, often unforeseen, contexts creates demand for more elaborate definition of service capabilities. Apart from the interface definition, which is the fundamental SOA metadata, applications have to be able to determine capabilities such as performance, reliability, and/or cost of service.

ADAPTIVE ENTERPRISE ARCHITECTURE

Adaptive enterprise architecture is a vision of the enterprise in which business processes, information, and knowledge assets can be quickly and effectively (re)organized and (re)deployed to support business decisions.

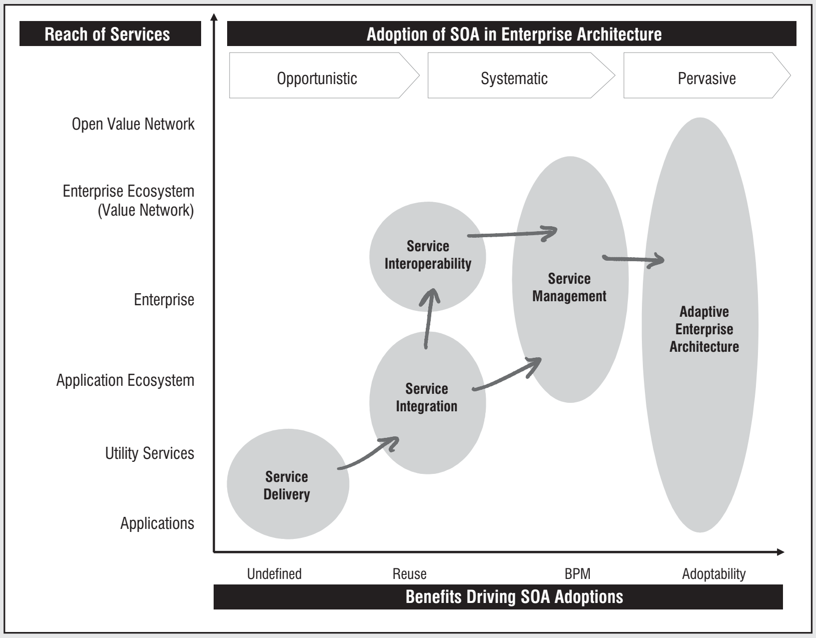


Figure A — SOA roadmap template.

MANAGED SERVICES PORTFOLIO

The services portfolio needs a solid architectural foundation to meet high and broad enterprise-level quality requirements related to availability, performance, and/or fault tolerance. Such a foundation (including the capabilities of the enterprise application integration platform) is often called an Enterprise Service Bus (ESB). The ESB and the portfolio of managed services are strategic IT assets shared between applications, connected through adapters and channels.

Services as seen in the "concentric model" of an ESB (see Figure B) are built in one of four tiers:

1. The **Managed Services** layer defines business services and implements message routing and transformation.
2. **Adapter Services** form the foundation for every service, connecting integrated systems to the ESB.
3. **Channel Services** translate native system invocations into enterprise service "language."
4. **Composite Applications**, an optional part of the architecture, are sometimes required to provide some additional functionality on top of the Managed Services and Adapter Services layers.

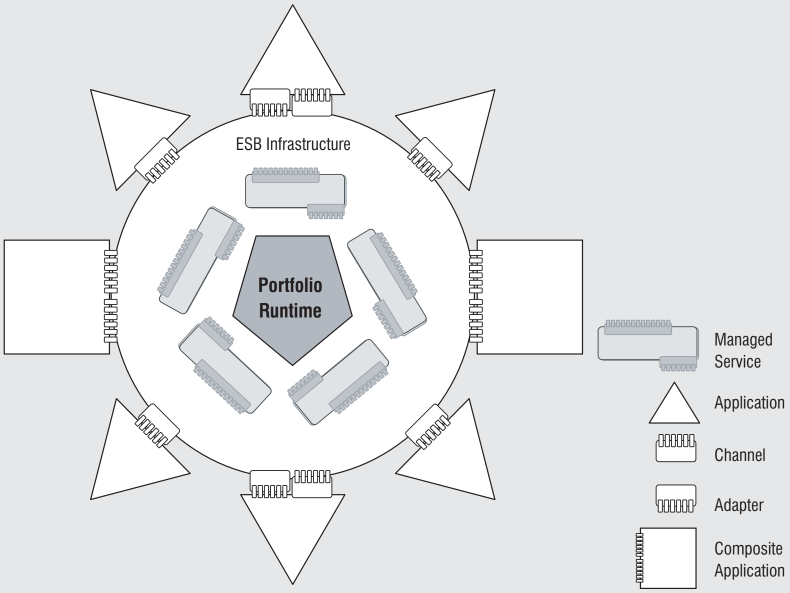


Figure B — The "concentric model" of Enterprise Service Bus.

Complementing ESB with additional components and tools that support service lifecycle management leads to the pattern defined as *Managed Services Portfolio*. The complete Managed Services Portfolio architecture, including the service lifecycle repository, is shown in Figure C.

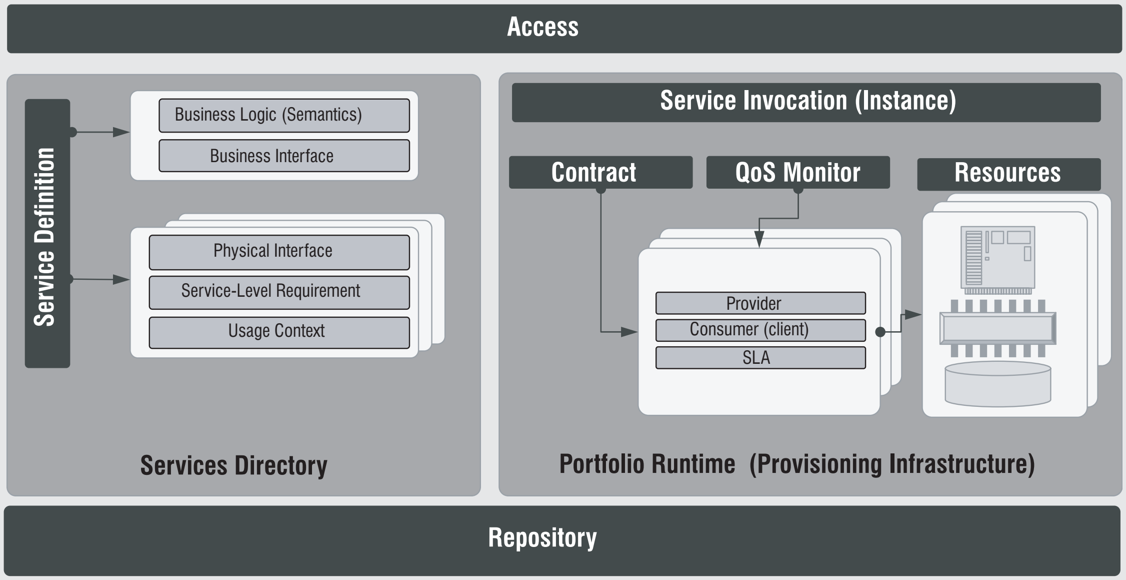


Figure C — The complete Managed Services Portfolio architecture.

As the pressure to leverage the asset increases, new releases of the enterprise architecture will include more and more requirements for new and improved services. The most immediate outcome is the increased complexity of the services portfolio. A less obvious result is the growing complexity of the process that is delivering services — more projects, more vendors, more subtle coordination requirements. SOA exhibits its transforming power by enabling very focused, agile process innovations, yet at the same time it is often constrained by the EA practices that promote monolithic releases of interdependent functionality. In a complex SOA environment, you eventually reach the limitations of management approaches that seemed to work before you achieved a certain complexity.

Consider the example of one major telecommunications company, which — two and a half years after starting its SOA program — now manages a portfolio of more than 200 services. The portfolio integrates functionality from 15 major systems (such as billing, CRM, provisioning platforms, and/or revenue assurance) and has been used as a foundation of a couple of nontrivial composite applications. Each release of the enterprise architecture (synchronized around CRM functionality) affects 30-50 services, including both new services development and changes in existing services. Apart from internal IT staff, there are some five major vendors responsible for various projects within the scope of each release.

If your organization is driving its SOA effort toward a point of similar complexity (which, you'll recall, is a byproduct of successful uptake of services by the business users), you'll need to understand some challenges in order to address them properly.

Understanding Service Usage Patterns

IT managers are often concerned about the quality of capacity planning. In the case of SOA, a reliable prediction of usage may not be possible by those who design and implement it. If the services really support business innovation, then they may often be used outside the initially envisioned usage context — after all, this is what innovation is about. That is why a well-designed portfolio allows for the collection of usage data in such a way that service architects may simulate potential effects of changes in the services portfolio and better forecast the required changes in services portfolio infrastructure. Apart from having access to usage data, a solution architect needs to understand the capabilities and architecture of the services portfolio.

Understanding the Services Portfolio Architecture

The production environment may be really well organized and services well described and structured, but in a complex SOA environment, you are likely to discover that to be productive, you need more information on the services portfolio than you actually get. Apart from knowing what services are available, you may need to know what kinds of commitments (scope of functionality, service levels, usage constraints) are associated with them. As a service developer, you might also be interested in tracking known issues in services that are outside your own scope of work. Equally important is visibility into the development and planning stages. If SOA is about reuse, then we must remember that a prerequisite for reuse is awareness of reuse opportunities.

> **SOA exhibits its transforming power by enabling very focused, agile process innovations, yet at the same time it is often constrained by the EA practices that promote monolithic releases of interdependent functionality.**

Understanding the Service Lifecycle

Visibility into the service development process leads us to the next issue. As SOA progresses within an enterprise, applications become decoupled into services, event-response patterns, and processes. If this is done right, these objects are to a large extent independent, so that one is able to manage them separately. The trouble is that few enterprises are capable of providing the fine-level project granularity enabled by SOA. Typical IT projects involve much coarser scope definitions. The same applies to releases of the enterprise architecture, which are usually managed as programs — sets of projects supporting high-level goals (see Figures 1 and 2). The style of central coordination applied in such situations can only be efficient if the requirements and the solution architecture are well aggregated. Such a program coordination style implicitly forces the projects to adopt a waterfall process even if an agile process would be more appropriate on the project level.

A Silent Game

Let us try to wrap up the issues. In order to take full advantage of the productivity and flexibility gains

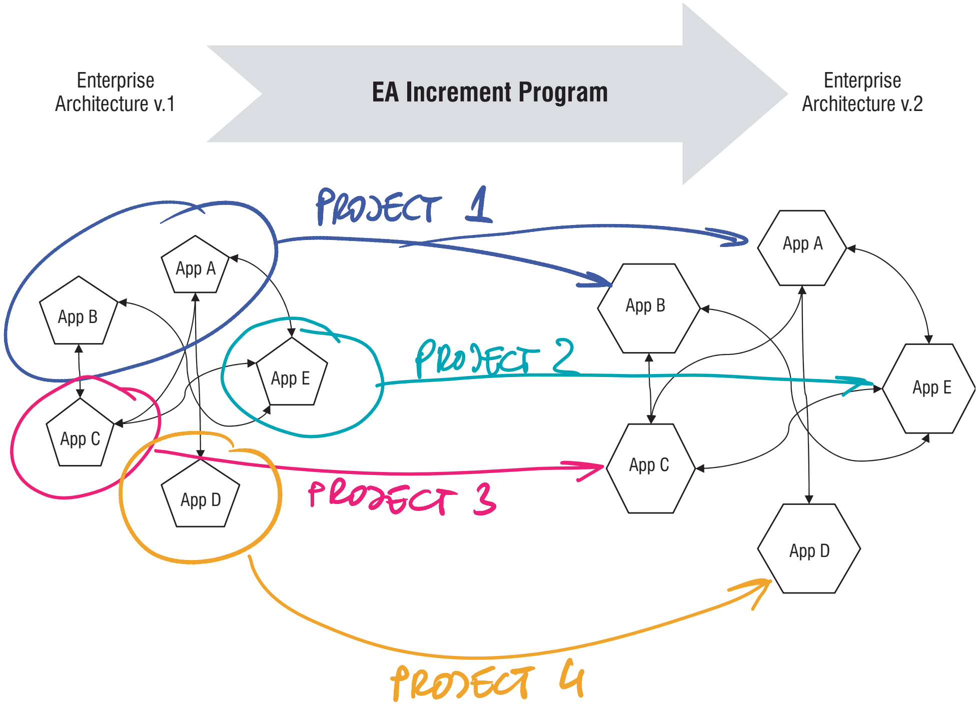


Figure 1 — Release management is simpler if the architecture consists of relatively coarsely defined, independent building blocks. Project-level interdependencies are simple and do not result in significant friction.

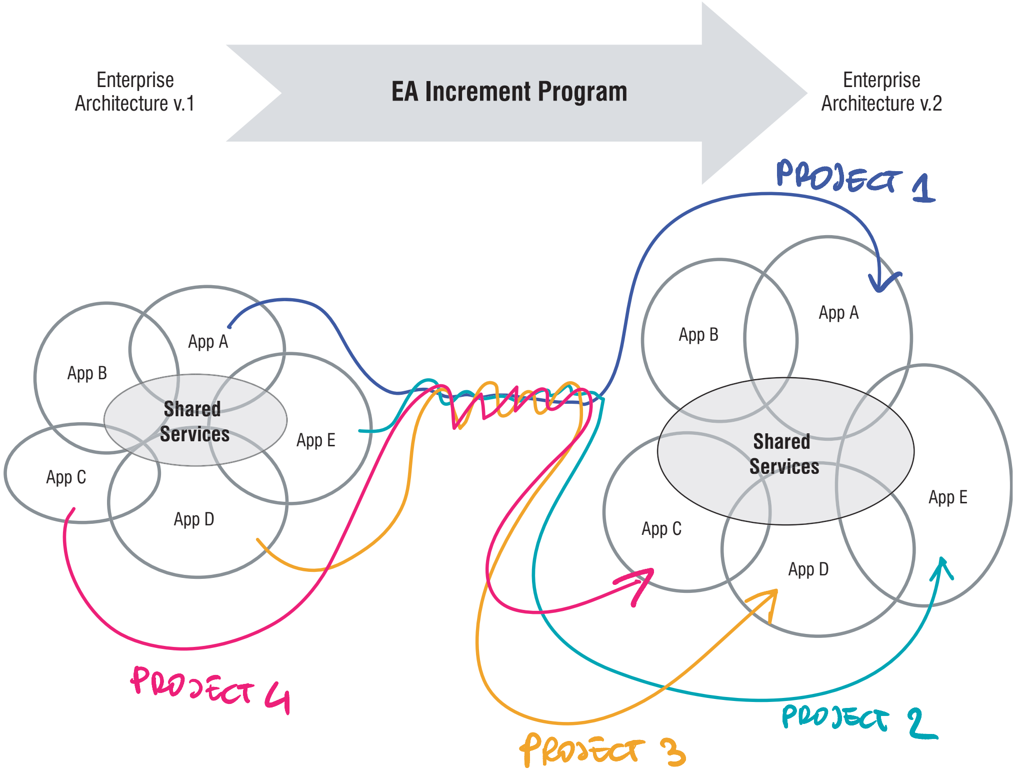


Figure 2 — If applications use and share a large number of interdependent services, then program-level coordination results in less-than-optimal efficiency of release management, due to deadlocks caused by complex project-level interdependencies.

offered by SOA, the delivery of interdependent services should be managed separately. Yet in large efforts, such as business-critical releases of the enterprise architecture, this is usually not happening. Such releases are managed from the top down. In an attempt to retain control, program management and enterprise architects divide the work into coarse chunks they can easily understand — that is, projects with a waterfall structure, synchronized by traditional program and project management rituals.

As a result, instead of asking, “Is the service xyz specification completed?” in most cases we can only ask, “Has project ABC completed its specification phase?” And if we need to call a service xyz from the ABC application delivered by another project, we cannot learn which version of service xyz we could use in our release, as the only information available for “other projects” is the version in the entire ABC application. It just seems so much simpler if, instead of analyzing complex relationships within the services portfolio, we could reduce the management effort to only a couple of applications and a dozen services included in our assignment. Unfortunately, this is not how it’s done, and thus we trade real agility — the ability to optimize the progress of work based on detailed design information shared between projects — for what pretends to be “more mature” management. This is just as if we asked a bunch of kids to jointly build a castle sharing the same set of LEGO blocks but ordered them not to talk to one another. Instead, they are just to talk to dad or mum, who is coordinating the construction. It might work, but it has little to do with either productivity or fun.

THE BUSINESS ARCHITECT’S PERSPECTIVE

Business architects use architecture concepts and patterns as organizing principles for key architecture building blocks, abstractions used primarily for decision making and — to some extent — high-level design. An important aspect of such tools is their ability to foster communication between the users and providers of technology. It is worth saying that concepts such as “services” and “services portfolios” seem to be equally appealing to business and technology communities, which is rather a rare case. Another appealing factor is the relative simplicity of the service metaphor, which makes it broadly applicable for various aspects of business-IT alignment.

But SOA is much more than just a metaphor. It is being directly implemented, and this implementation consumes resources — capital, money, effort. Thus, from the point of view of governance principles, one must start by asking some fundamental questions related to two basic “outcome areas” within a traditional IT governance model (as defined by the IT Governance Institute) [1]:

1. What value is being delivered by SOA?

2. What are the associated risks?

It is absolutely crucial to identify the business case for the SOA implementation, since (as with any other project) the identified requirements and SOA stakeholders are the starting and ending points for all SOA governance issues. An SOA roadmap template [2] is an example of an intellectual framework that can be used to address these fundamental questions.

An insightful reader may have already noticed how “frontline experience” shows that introducing SOA leads to a point when the evolution of architecture may demand a change in some established practices used for governance. We believe that this can only be achieved if the service metaphor is used consistently in all aspects of governance and EAM. Let us review some major acts performed by the enterprise architect in his or her work. These are: decomposition, abstraction, coordination, standardization, consolidation, and reuse.

This is just as if we asked a bunch of kids to jointly build a castle sharing the same set of LEGO blocks but ordered them not to talk to one another.

Decomposition

Services can be a very powerful tool for process decomposition and modeling. Processes can be decomposed at all levels of the enterprise architecture. This should be done mainly on the basis of the business operations model. If decomposition is done by business (which is seldom the case — formal modeling is usually part of the esoteric lore performed by business analysts within IT), problems may arise on different grounds:

- Lack of clear delegation of responsibility for the business activities comprising a service (business silos and uncoordinated redundancy of work) leads to problems in service decomposition.
- The same lack of clearly defined responsibility for business tasks also impedes services identification

and reconciliation in the "white spaces" between functional silos.

- A legacy of misalignment between IT and business may result in serious difficulties in matching business services with the applications or functions implemented in IT systems.
- A high level of "projectization" (work executed mostly as projects with few routine, process-oriented activities, objectives, and responsibilities) may undermine the identification and/or decomposition of services.

Implementing SOA is a transforming initiative. Services provide the foundation of the customer relationship, serve as the business process building blocks, and are often sourced from vendors and business partners. They can be measured, orchestrated, and benchmarked. They can be improved, recombined, and reconnected to support innovation and agility. If this is the vision, and it is generally accepted by the organization, then there is fertile ground for using IT architecture patterns such as *Managed Services Portfolio* to support such a transformation. (For a description of the Managed Services Portfolio pattern, see the sidebar on page 36.)

> **The services portfolio is a shared asset, and enterprises have little experience in solving the problem often described as the "tragedy of the commons."**

Abstraction (Decoupling)

If an organization recognizes the benefits of service-oriented architecture and accepts the consequences of the change it involves, abstraction becomes a critical act. It enables the organization to identify its core building blocks — its "functional DNA." This painstaking and crucial task may involve internal politics and consensus building. It is targeted at maximum decoupling of the defined services to allow them to be managed separately, to untie the existing organizational and functional knots, and to form a clear and uniform classification of services. This is the transition from a subjective view of services (typical for business stakeholders) to a logical view that constitutes a well-architected services portfolio.

As with decomposition, tensions may arise because of decisions made in decoupling services. Some examples are:

- Redundant functionalities in the current IT environment may require decisions not based on bilateral IT-business contacts, but rather on a business–IT development–IT operations triangle. Special governance solutions may need to be implemented to encourage IT development and IT operations to resolve their issues before beginning a discussion with business.
- Decoupling may reveal "orphan functions" in IT systems that create costs but have no business owners — or that have a business owner who does not accept the level of costs "discovered" at this stage.
- During initial portfolio creation, many existing areas of interest may intersect, leading to turf conflicts and difficulties in appointing and sharing responsibilities.

Coordination

Managed Services Portfolio, as a core SOA pattern, can be applied in many ways. Its most important role from a governance perspective is to support decision making related to changes in enterprise architecture. Decoupling leads us to a definition of a service backbone. This backbone — if well defined — serves as a managerial tool and is reflected directly in IT architecture. We believe that the service backbone could be the coordination axis for all levels of abstraction within the EA model, a sound basis on which to plan and/or implement business process changes, define service-level management parameters, and so on.

There are two reasons why EA initiatives — as seen from the frontline perspective of the solution architect — seem to ignore this role of the services portfolio. First, the services portfolio is a shared asset, and enterprises have little experience in solving the problem often described in management literature as the "tragedy of the commons." Second, the initial content of the services portfolio is usually relatively weak. Most of the support required by business is traditionally provided by other building blocks of the IT architecture — traditional, monolithic, isolated business applications. Thus, the goals of EA increments are not primarily associated with the evolution of the services portfolio. Portfolio infrastructure that could support the agile process of service delivery is not adequately supported by capital investment and management attention. As a result, the actual productivity trajectory related to EA transformation diverges from the optimal one.

Most of the issues the solution architect faces (as we discussed above) are found in the area of coordination.

Many governance issues can also be addressed here. Some examples are:

- Difficulties in providing an adequate service benchmarking framework and low awareness of reuse opportunities undermine the usefulness of the services portfolio as a decision-making tool.
- Organizations have difficulty assigning clear business responsibility for the quality, content, flexibility, and other important characteristics of the services portfolio (not just individual services).
- SOA must be integrated with the paradigms, methodologies, and approaches that have come before and remain in place (EA models, ITIL service-level agreements, etc.).

Standardization

In the process of decomposition and decoupling, a set of criteria for defining the services emerges. Supported with additional requirements (terms definitions, architecture rules, formats and notation definitions, etc.), it can be converted into a consistent and efficient set of standards and patterns.

The need to achieve both consistency and efficiency makes this task nontrivial. In the act of establishing standards, an enterprise architect forges tools that should help the organization bridge the gap between metaphor and the complexity of a real-world enterprise. Business users are confident in using the service metaphor, visualization (general business graphics), and numbers (budgets, business performance metrics), but they are not willing to handle formal architecture models such as UML constructs. On the other hand, any definition of services portfolio architecture must go beyond PowerPoint slides if it is to have any value for those doing the actual implementation work.

Thus, any set of standards developed for an enterprise-scale SOA implementation should support and promote basic motivations behind such initiatives (agility, reuse, efficiency, accountability, etc.). It should also promote — from the very beginning — the message that the services portfolio is a shared asset that can create value only if leveraged. Some relevant issues here are:

- Standards are required to share critical design and status information between projects at a level of granularity that supports grassroots coordination between projects. This mechanism will not work properly if project- and program-level governance promotes isolation of projects or, still worse, competition between projects and vendors.
- Introduction of optional-scope contracts for work on the services portfolio may optimize the business value that SOA projects deliver over time.
- Involving many vendors in an SOA program necessitates negotiations among many parties. Such negotiations are time-consuming and require diplomatic skills.

Any definition of services portfolio architecture must go beyond PowerPoint slides if it is to have any value for those doing the actual implementation work.

Consolidation

Coordination is the act of synchronizing disparate initiatives that transform the enterprise architecture. Standardization supports this act by creating an environment where design- and implementation-level decisions can be efficiently negotiated by the implementation teams. Consolidation is the act of implementing these mechanisms, patterns, and standards in order to deliver an integrated portfolio of manageable services through architecture, design, and implementation. This involves using release management for managing the subsequent increments of the envisioned enterprise architecture.

At this point, we believe that SOA governance should be supported on the project/release level by an information-sharing infrastructure, including:

- A UDDI-compliant service repository that is well documented, searchable, and user-friendly
- Automated design/deployment polices that support the chosen service lifecycle model, providing transparent access to project and service status information
- Runtime polices that enforce and measure service usage
- Other features such as automated services testing and versioning

Not all aspects of this act can be easily considered in advance. It is worth taking the time to recognize as many of them as possible and to build them into your SOA governance. Some examples are:

- In multibranch/multinational companies, it is not always clear how to implement standards.

Complex organizations will require a balanced mix of approaches — consensus building, negotiating, and sometimes even arbitrary decisions — in order to establish standards.

- If a company outsources selected operations or has close interactions with its clients or contractors, the system of communicating standards becomes quite a challenge (including indicating the responsible roles and their duties).
- Not all enterprise initiatives or existing solutions will be compliant with the centralized rules or standards. Despite the migration plan to the new standards, an architectural dispensations management approach should be implemented.
- Planning and implementing a system of restrictions for organizations, projects, and/or roles that do not comply with standards is a sensitive issue and should take into account a general future approach and current governance rules.

> **The dependencies between systems make typical releases what they are — complex and scary management and technical endeavors.**

Reuse

In the course of an SOA initiative, an organization will need to understand the many interdependencies related to its applications, systems, projects, and so forth. It may find that some solutions are redundant, some are orphans (and thus can be abandoned or radically simplified), and some will present opportunities for reuse. Apart from the suggestions we made above about treating the services portfolio as a shared IT asset, there are some other governance structures and standards that should be established to accommodate service reuse decisions at a higher level:

- Because it is not always clear when some service with a local or regional scope can/should be promoted, the organization should define a set of criteria for making this determination or, alternatively, appoint a person/role to do so.
- Some business services are so common and basic that they can be perceived as a part of the enterprise infrastructure rather than the business portfolio. Again, a set of criteria should be established for making this decision.

TOWARD SERVICE-ORIENTED GOVERNANCE

Employing large-scale release management to develop the enterprise architecture seems to be the state-of-the-art in EAM programs. As helpful as this technique is in synchronizing large development and deployment efforts, it forces an aggregation of the complex web of dependencies between services into much-simplified dependencies between components. As a result, otherwise independent building blocks often need to wait for one another, while some sophisticated interdependencies of requirements get lost in a crude "division of labor." The outcome is friction — in the form of quality problems, integration problems, and time wasted in delays caused by late delivery of components. And friction is the main enemy of agility.

One can view the problem as an unavoidable side effect of the complexity of modern IT architectures. Surely without the current discipline of release management, we could hardly hope for any systematic delivery of IT capability on the enterprise level. But is this really still true? What if what we have here is really a situation in which executive-focused management practices got out of sync with the new architectural paradigms brought about by the advent of enterprise SOA?

The dependencies between systems make typical releases what they are — complex and scary management and technical endeavors. As long as systems were mostly monolithic, we could reduce the management complexity by recognizing dependencies between application versions and related high-level business requirements. As we have said before, with SOA governance in place we might attempt to "zoom in" the scope to the level of dependencies of particular functionalities on particular components. Apart from the potential headache we can get trying to imagine the graph of interdependencies, such radical growth of scope complexity could actually blow up the release-based management system by making it too costly and inefficient. Furthermore, ignoring the opportunity of service-level dependency management creates inefficiencies in the development work.

We need to work this problem out in three steps:

1. We need to organize the IT governance and EA processes around the service metaphor. It is

comprehensible for non-IT people, it provides a good intellectual framework for all areas of governance, and it maps directly into software architectures and IT organizational principles and methodologies. One caveat is that such an approach will work only where this metaphor is adequate for the business style and architecture of a particular organization.

2. We have to introduce governance and management practices (business responsibility, funding mechanisms, operational responsibility, service benchmarking, etc.) that will make the services portfolio a truly shareable IT asset. The open source community may provide good examples of such collaborative management practices.

3. We need to establish operational practices that enable project teams to coordinate their work on the services portfolio without the involvement of program management bodies. This involves some technical solutions that enable collaboration on a shared IT asset (e.g., service repositories, shared regression testing environments, shared generators) as well as management solutions that will reduce the risk of inducing unethical or counterproductive behavior among the parties involved in service development (e.g., a federation of architects to establish and enforce standards on SOA projects).

This is the tao of SOA governance.

REFERENCES

1. IT Governance Institute (ITGI). *Board Briefing on IT Governance*, 2nd ed. ITGI, 2003 (www.itgi.org).

2. Kiepuszewski, Bartosz, Michal Paluskiewicz, Borys Stokalski, Sebastian Konkol, and Maciej Ruszynski. "Applying EA Roadmapping: An SOA Roadmap." Cutter Consortium Enterprise Architecture *Executive Report*, Vol. 7, No. 9, September 2004.

3. Topinski, Jan, Bartosz Kiepuszewski, and Borys Stokalski. "Service-Oriented Integration: A Report from the Trenches." Cutter Consortium Enterprise Architecture *Executive Report*, Vol. 9, No. 5, May 2006.

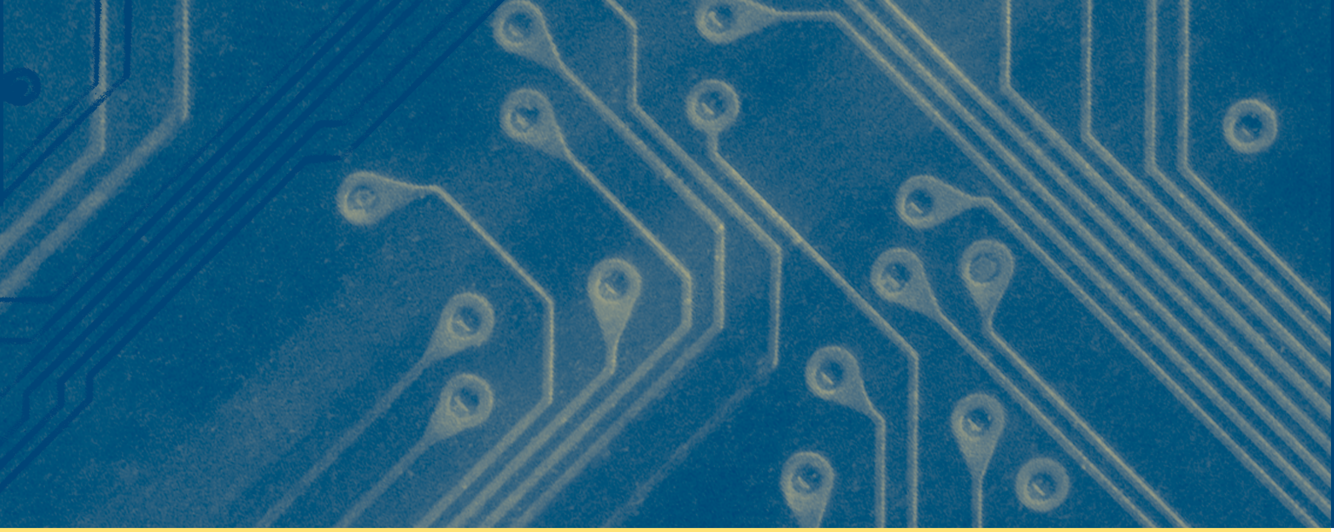
Piotr Szabelak is a Senior Enterprise Architect at Infovide-Matrix. He specializes in enterprise architecture and has worked on several successful EA implementation projects. He previously worked in IT in government, where he gained a background in EA, project management, project portfolio coordination, IT security, and data warehousing/BI. He can be reached at pszabelak@infovidematrix.pl.

Jan Topinski is a Senior Consultant at Infovide. Mr. Topinski advises on QA management and automation. He also works as an architect, specifically for integration platforms and SOAs. Mr. Topinski is an expert on development process automation environments and frameworks. He can be reached at jtopinski@infovidematrix.pl.

Borys Stokalski is a Senior Consultant with Cutter Consortium's Business-IT Strategies, Enterprise Architecture, and Innovation practices. He is VP of Infovide-Matrix S.A., a Polish company specializing in architecting and implementing IT-based business innovations. Infovide, which is the exclusive representative of Cutter Consortium in Poland, is ranked among the top CSI (Consulting & Solution Implementation) organizations in that marketplace.

Mr. Stokalski takes an active part in Infovide-Matrix initiatives targeted at new service areas and consulting products, such as adaptive business strategies and architectures, knowledge management, collaboration architecture, and training programs for IT managers. One of his top responsibilities is the stimulation of growth and efficiency in the Infovide-Matrix innovation network, an essential part of which is the knowledge collaboration with Cutter Consortium. Mr. Stokalski frequently shares his experiences and insights through articles and conference appearances on topics such as innovation management, the business impact of technology change, and EAM. He can be reached at bstokalski@cutter.com.

Cutter
IT Journal



About Cutter Consortium

Cutter Consortium is a truly unique IT advisory firm, comprising a group of more than 100 internationally recognized experts who have come together to offer content, consulting, and training to our clients. These experts are committed to delivering top-level, critical, and objective advice. They have done, and are doing, groundbreaking work in organizations worldwide, helping companies deal with issues in the core areas of software development and agile project management, enterprise architecture, business technology trends and strategies, enterprise risk management, metrics, and sourcing.

Cutter offers a different value proposition than other IT research firms: We give you Access to the Experts. You get practitioners' points of view, derived from hands-on experience with the same critical issues you are facing, not the perspective of a desk-bound analyst who can only make predictions and observations on what's happening in the marketplace. With Cutter Consortium, you get the best practices and lessons learned from the world's leading experts, experts who are implementing these techniques at companies like yours right now.

Cutter's clients are able to tap into its expertise in a variety of formats, including content via online advisory services and journals, mentoring, workshops, training, and consulting. And by customizing our information products and training/consulting services, you get the solutions you need, while staying within your budget.

Cutter Consortium's philosophy is that there is no single right solution for all enterprises, or all departments within one enterprise, or even all projects within a department. Cutter believes that the complexity of the business technology issues confronting corporations today demands multiple detailed perspectives from which a company can view its opportunities and risks in order to make the right strategic and tactical decisions. The simplistic pronouncements other analyst firms make do not take into account the unique situation of each organization. This is another reason to present the several sides to each issue: to enable clients to determine the course of action that best fits their unique situation.

For more information, contact Cutter Consortium at +1 781 648 8700 or sales@cutter.com.

The Cutter Business Technology Council

The Cutter Business Technology Council was established by Cutter Consortium to help spot emerging trends in IT, digital technology, and the marketplace. Its members are IT specialists whose ideas have become important building blocks of today's wide-band, digitally connected, global economy. This brain trust includes:

- Rob Austin
- Ron Blitstein
- Tom DeMarco
- Christine Davis
- Lynne Ellyn
- Jim Highsmith
- Tim Lister
- Lou Mazzucchelli
- Ken Orr
- Ed Yourdon