SYS-CON

THE WORLD'S LEADING MAGAZINE DEDICATED TO **WEB SERVICES TECHNOLOGIES**

FOR ALL THE LATEST NEWS AND INFORMATION VISIT:

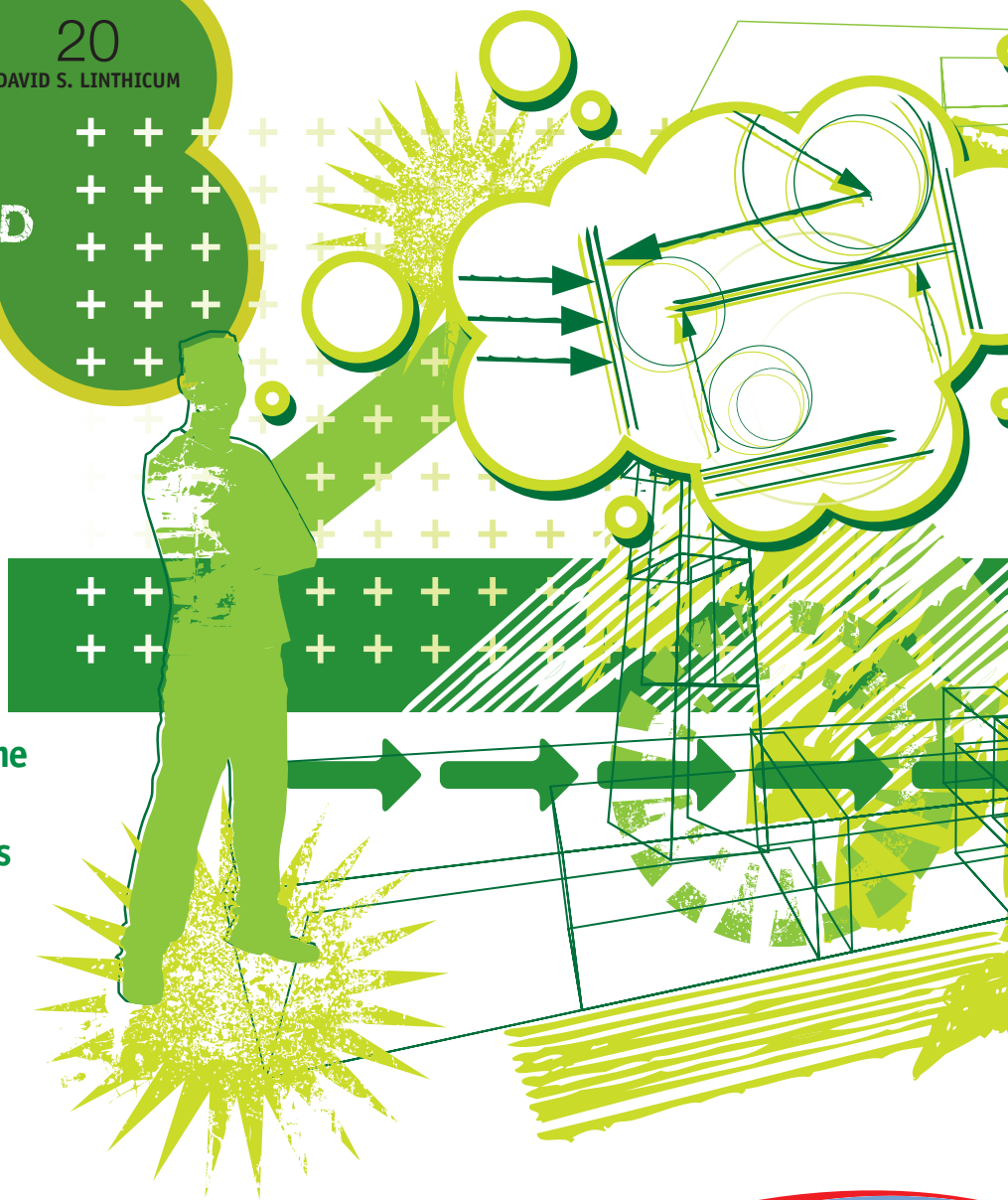
www.SOA.SYS-CON.com

SOA WORLD™ MAGAZINE

MAY 2007 / VOLUME: 7 ISSUE 5

BUILDING A SOA: BEST APPROACH AND BEST PRACTICES

20 DAVID S. LINTHICUM



3 **Ready, Fire, Aim**
SEAN RHODY

4 **SOA Architectures and Models**
DAVID S. LINTHICUM

6 **Alleviate SOA Headaches**
KEN HAMILTON AND MARK LAJEUNESSE

10 **My Design Time Is Your Runtime**
JIM MURPHY

14 **Service Management Paradigms**
NAVEEN KULKARNI AND KRISHNENDU KUNTI

26 **The Blueprint for SOA Success**
HON WONG

30 **The Flesh and Bone of SOA**
MICHAEL HAVEY

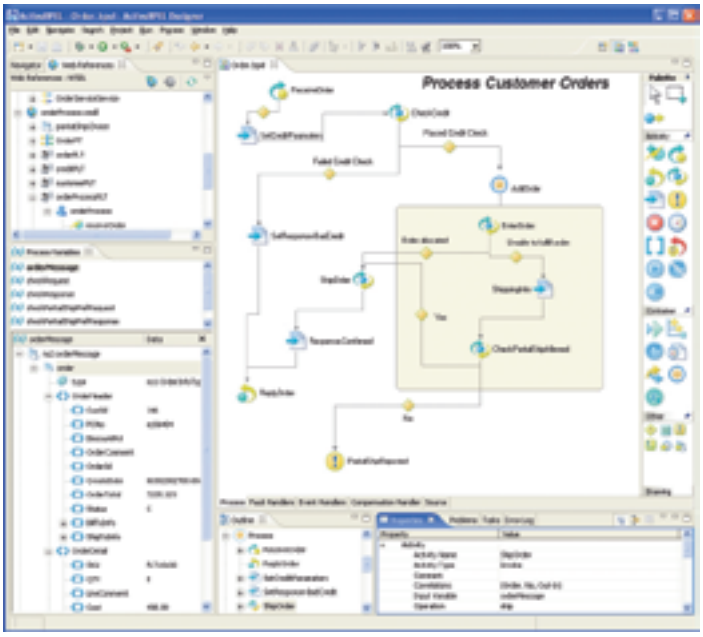
PLEASE DISPLAY UNTIL JUNE 30, 2007

$6.99US $7.99CAN

05>

0 71486 03420 9

Get Started with BPEL 2.0



Build next-generation SOA applications with the leader in BPEL technologies

Download BPEL tooling & server software today

active-endpoints.com/soa

BPEL consulting and training.
BPEL design tools, servers and source code for Eclipse, Apache Tomcat, JBoss, WebSphere, WebLogic, BizTalk and Microsoft .NET.

activeBPEL

active endpoints

SOA WORLD MAGAZINE
www.SOA.SYS-CON.com

EDITORIAL OFFICES
SYS-CON MEDIA
577 CHESTNUT RIDGE ROAD, WOODCLIFF LAKE, NJ 07677
TELEPHONE: 201 802-3000 FAX: 201 782-9637
SOA World Magazine (ISSN# 1535-6906)
Is published monthly (12 times a year)
By SYS-CON Publications, Inc.
Periodicals postage pending
Woodcliff Lake, NJ 07677 and additional mailing offices
POSTMASTER: Send address changes to:
SOA World Magazine, SYS-CON Publications, Inc.
577 Chestnut Ridge Road, Woodcliff Lake, NJ 07677

Ready, Fire, Aim

WRITTEN BY **SEAN RHODY**

We all do it from time to time – forget something, get it out of sequence – and experience that annoying feeling that we've just done something incredibly stupid. I usually arrive at the dry cleaners to pick up my clothes, only to realize I'd left the next batch behind. Fortunately, it's not a long drive. But it's usually because I'm in a rush, trying to accomplish multiple things. If I slowed down just a second, I might accomplish more.

Service-oriented architecture faces similar challenges. In particular, the concept of architecture seems to be the one that gets left behind, or thrown out with the bathwater. Yet it's really the architecture part of SOA that enables it to provide the highest return on investment from an implementation.

I've gone on and on about architecture – what it is, why it's important, how to do it – in numerous issues of *SOAWorld Magazine*. Yet in many cases I still see organizations with the same old problem: "Ready, Fire, Aim!"

This is partially due to the challenges of providing an ROI on technology that is in many cases considered pure infrastructure. It's an old problem. Remember when you didn't have Internet access on the job, or it was a slow dial-up? Making the case that Internet access was important to the employee base was difficult – especially if you had to justify a move from limited dial-up to universal broadband. Sure, we all thought it was a good idea, but what was the monetary value to the organization? It was much easier to quantize the costs than to enumerate the benefits and place a value on them. And yet, eventually, the overwhelming majority of organizations have moved to broadband.

SOA is very similar. It's nearly impossible to quantify the business benefit of an SOA migration. The benefits are clear in an environment with multiple applications and strong integration needs, but at the same time, it's still a challenge to pin a number to them. Will an SOA increase productivity? Will it reduce maintenance costs? How much? How long will it take to reduce those costs? Yes, the questions sound like you're being pecked to death by the accounting duck, but they are also real issues. Bottom line in today's economy is that an SOA migration needs clear ROI justification.

Which is a challenge. Instead, organizations are using the grassroots approach to SOA. New projects have to include SOA. Old applications that are being renovated are encouraged to adopt SOA aspects. And slowly, through the process of accretion, an SOA arises from the muck.

The challenge is that by building SOA a system at a time, we're concentrating on the wrong end of the process and ignoring the role of architecture. We're building the same old silo – perhaps a little more open, but not a whole lot better than the system as it existed before the introduction of an SOA element.

This is why architecture is important. Without it, services arise at all levels of granularity. Processes have to be cobbled together haphazardly. Service rationalization and consolidation does not take place, so duplicate or nearly duplicate services exist long after they should have been rationalized to a single service. The retirement of redundant systems is delayed long after it could be accomplished.

Although it's rare to be able to fund a complete SOA overhaul, a best practice that should be followed is to set up an SOA Governance and Architecture group. The group doesn't need to boil the ocean in order to solve every problem, but as systems are reworked or introduced they can be the coordination point for rationalizing the enterprise to a set of identified services. They can also serve as the mediator in various negotiations around design and guide the IT organization as to how best to achieve an SOA given multiple options.

In the same way that I need someone to remind me to take my dry cleaning with me, we all need to have Architecture as part of SOA. Now where did I put that cleaning ticket? ■

About the Author

Sean Rhody is the editor-in-chief of **SOA World Magazine**. *He is a respected industry expert and a consultant with a leading consulting services company.* *sean@sys-con.com*

Understanding SOA Architectures and Models

Part III: Enterprise Architecture and SOA

WRITTEN BY **DAVID S. LINTHICUM**

This is the larger issue, as I see it, and is very visible to me working both in the world of SOA and the world of enterprise architecture. So, why are they different worlds? Moreover, what is enterprise architecture, and how does it fit with reference models and reference architectures, as discussed here before?

From Wikipedia: "Enterprise architecture is the practice of applying a comprehensive and rigorous method for describing a current and/or future structure and behavior for an organization's processes, information systems, personnel and organizational sub-units, so that they align with the organization's core goals and strategic direction. Although often associated strictly with information technology, it relates more broadly to the practice of business optimization in that it addresses business architecture, performance management, organizational structure, and process architecture as well."

Thus, the notion has morphed from something that's been more of a technology concept to a management concept, if you're using this definition. I've blogged about that issue several times. Indeed I think the government is driving some of this, thus expanding the definition and reach of the notion of enterprise architecture.

"Enterprise architecture is becoming a common practice within the U.S. federal government to inform the Capital Planning and Investment Control (CPIC) process. Federal Enterprise Architecture (FEA) reference models serve as a framework to guide federal agencies in the development of their architectures. The primary purpose of creating an enterprise architecture is to ensure that business strategy and IT investments are aligned. As such, enterprise architecture allows traceability from the business strategy down to the underlying technology."

So, if we go with this definition we are moving from a high-level management notion, including strategy, business optimization, budgets, etc. to information technology, including SOA. EA is more holistic, where SOA is a more specific approach for building an IT infrastructure.

My Take

I'm not sure that I can do anything to change the definitions here, or eliminate the confusion when considering the number of ways everyone is slicing and dicing SOA and EA. At the end of the day, I'm not sure it matters as long as enterprises find the right approach that works for them. Typically that takes some effort and planning. Truth be told, there are no canned SOA solutions out there, at least, not ones that will have significant strategic value.

Besides ending the debate about unified semantics and vision, it's also helpful to consider the "loop" as well or mechanisms we put in place to understand and create best practices based on experiences with building and deploying SOAs. The key issue here is

—continued on page 8

About the Author

David S. Linthicum is an internationally known application integration and Service Oriented Architecture expert. In his career Dave has assisted in the formation of many of the ideas behind modern distributed computing including Enterprise Application Integration, B2B Application Integration, and Service Oriented Architecture, approaches and technologies in wide use today. Currently, he is CEO of the Linthicum Group, LLC, (www.linthicumgroup.com) a consulting organization dedicated to excellence in Service Oriented Architecture planning, implementation, and strategy.

david@linthicumgroup.com

SOAWORLD MAGAZINE
www.SOA.SYS-CON.com

CORPORATE

President and CEO
Fuat Kircaali fuat@sys-con.com

Group Publisher
Roger Strukhoff roger@sys-con.com

ADVERTISING

Senior VP, Sales & Marketing
Carmen Gonzalez carmen@sys-con.com

Advertising Sales Director
Megan Mussa megan@sys-con.com

Associate Sales Manager
Corinna Melcon corinna@sys-con.com

SYS-CON EVENTS

Event Manager
Lauren Orsi lauren@sys-con.com

Event Associate
Sharmonique Shade sharmonique@sys-con.com

CUSTOMER RELATIONS

Circulation Service Coordinators
Edna Earle Russell edna@sys-con.com
Alicia Nolan alicia@sys-con.com

SYS-CON.COM

VP information systems
Robert Diamond robert@sys-con.com

Web Designers
Stephen Kilmurray stephen@sys-con.com
Richard Walter richard@sys-con.com

ACCOUNTING

Financial Analyst
Joan LaRose joan@sys-con.com

Accounts Payable
Betty White betty@sys-con.com

SUBSCRIPTIONS

SUBSCRIBE@SYS-CON.COM
1-201-802-3012 or 1-888-303-5282
For subscriptions and requests for bulk orders, please send your letters to Subscription Department
Cover Price: $6.99/issue
Domestic: $69.99/yr (12 issues)
Canada/Mexico: $89.99/yr
All other countries: $99.99/yr
(U.S. Banks or Money Orders)

Worldwide Newsstand Distribution:
Curtis Circulation Company, New Milford, NJ

For list rental information:
Kevin Collopy: 845 731-2684, kevin.collopy@edithroman.com;
Frank Cipolla: 845 731-3832, frank.cipolla@epostdirect.com

At Last, a Mission-critical Ajax Framework

FREE 30 DAY EVALUATION visit dev.nexaweb.com

Ease of Development

- Declarative UI Design, Event Handling, and Data Binding for Ajax & Java
- Eclipse-based Visual Ajax & Java Editor
- User Interface & Business Logic separation
- Extensible Widget Library (Struts, JSF, SWING, and Flash compatibility)
- Drag & Drop data source integration (databases, Web services, XML, JSON)

Enterprise-Class

- Lightweight Universal Client Framework (Ajax, Java, and Desktop client < 200 kb)
- Real-time, Secure Messaging
- Server-side Push, Publish/Subscribe, and Broadcasting capabilities
- Clustered & Persistent Session Management
- Section 508 Accessibility compliance

Get it all, Get it today, Get it only with Nexaweb!

nexaweb

ITSM and ITIL
Alleviate SOA Headaches

WRITTEN BY **KEN HAMILTON AND MARK LAJEUNESSE**

As IT departments continue to prioritize their investment initiatives, many will find Service Oriented Architecture (SOA) is a critical item to tackle. Although key business projects must always take precedence, McKinsey & Company reports that more than 60% of surveyed CIOs plan to use SOA projects as a way to achieve associated business objectives.

For good reason SOA is being heralded as a major architectural shift that enables IT teams to reduce costs and improve responsiveness and flexibility – two key business objectives for CIOs today. With SOA, IT ultimately becomes a service provider to the business itself — delivering infrastructure, applications, and processes as services throughout a company.

SOA certainly offers promise; but at the same time, successfully navigating the people, process, and technology changes associated with it can be risky. Of course, companies want to reduce this risk. According to Gartner a successful SOA program "necessitates new processes, ranging from governance, through development, to operations." SOA deployments require a commitment to planning and strategy, but the entire adoption process can be streamlined if the Information Technology Infrastructure Library (ITIL) framework is already in place. In fact, companies that adhere to ITIL best practices prior to implementation will minimize many common issues, such as the complexity of change management, in an SOA deployment because of the rigor and discipline of their ITIL experience. The SOA adoption curve becomes much flatter with specific IT service support processes and tools locked and loaded.

At the heart of a solid ITIL implementation is IT Service Management, or ITSM. ITSM is the integration of people, processes, and technology into a well-designed system based on best industry practices. With companies' IT processes already in place, ITSM provides the roadmap to help maximize business value with controlled costs, while mitigating business risks. The goal of ITSM is to implement and manage IT services in an organization that meet the needs of the business. It is a long-term approach to the way an organization, whether it is technology-focused or not, operates. For example, an overnight delivery package being sent to a business partner may seem non-technical in nature. However, to deliver the product, the "back office" is dependent on a successful ITSM implementation that ensures there is a process in place to deliver the product on time and with maximum efficiency.

The power and benefits of SOA increase as its adoption across the enterprise increases. To obtain the maximum value of SOA adoption, it's important to drive SOA into all aspects of IT operations. Some people call this "operationalizing SOA." Companies that have a solid track record of implementing ITSM processes will likely find it easier and quicker to adopt the new processes required to successfully adopt SOA.

The Power of ITIL Frameworks

Developed by the British government in the 1980s, ITIL is a framework of best practices intended to facilitate the delivery of high-quality IT services. ITIL outlines an extensive set of management procedures intended to support businesses in achieving both quality and value in IT operations. ITIL is supported by a comprehensive qualifications scheme, accredited training organizations, consultancies, user groups, and implementation and assessment tools.

The "library" is currently in its second iteration but is being enhanced with a newer version slated for release on May 30, 2007. While ITIL v2 offers a consolidated view of IT processes, ITIL v3 will provide a more comprehensive approach that takes business lifecycle management into account and includes real-world examples, best practices models, and metrics.

The adoption of ITIL is gaining speed in the United States as companies find that ITIL can help cut costs, improve IT services through proven best practices, and increase productivity by providing a common IT language. These benefits are similar to those of SOA, and companies that adopt both approaches may see a multiplier effect as the benefits are self-reinforcing. Executives from the IT Service Management Forum (itSMF) project that the number of companies adopting ITIL in the U.S. could grow to about 75% by the end of 2007. Additionally, Forrester Research projects that 80% of the Global 2000 companies will adopt the ITIL framework by 2010. Similarly, Gartner predicts that SOA will be used in more than 80% of new mission-critical applications and business processes by 2010.

As a result of increased awareness of ITIL, there's been a corresponding increase in the implementation of ITSM projects. As companies define and manage IT services that are critical to enterprise business process, they need to be able to improve service quality with real-time information about IT impact on service level agreements, and then understand how to optimize service support. This mission-critical management is delivered through ITSM solutions that enable CIOs to run IT as a service delivery business. The ability to deliver a high quality of service is especially important when SOA is embraced as your architectural approach, as a single service may be used in, and therefore impacts a multitude of business-critical applications.

ITIL Paves the ITSM Road for SOA Implementations

SOA programs can deliver tangible cost and response benefits but the challenges for an SOA implementation shouldn't be underestimated. For starters, the task of identifying processes that are used by multiple groups in the organization can be complex. Getting those groups to agree on a common process definition can be even more complex. When you add ensuring consistency of understanding the underlying SOA technology stack along with the significant number of architectural choices that also have to be made, the complexity grows geometrically.

However, there are a series of other challenges that have to be addressed if SOA adoption is going to be successful. These challenges can be grouped into three buckets: governance, quality, and management.

SOA governance is commonly meant to be the process of increasing the collaboration between the major stakeholders: service providers, service consumers, and the SOA Center of Excellence. The SOA Center of Excellence, led by the enterprise architect or the CTO, is interested in balancing flexibility for the providers and consumers versus control so that standards are followed and SOA adoption is successful. Questions that have to be answered in the governance area include: "How do I find and trust services?" "Is the enterprise working on the *right* services?" "Can we establish a consistent and repeatable implementation process?" and "Can we effectively control the services that are in production?" In essence, the chief SOA architect is responsible for making sure that SOA doesn't come to mean "State of Anarchy."

SOA quality is the process of ensuring that services meet the functional and performance requirement of the organization, with a minimal amount of effort in the testing process itself. Since services used by multiple business processes can be single points of failure, it's critical to extend enterprise testing processes to address the specific SOA needs . Without extended quality processes, adopting a service that's created by someone else is just too risky for developers to do. Further, since the use cases for a generalized service are numerous, any change to a service must be tested against a seemingly unlimited number of test cases. This can strain already thin testing resources to the breaking point, both in test management and test execution.

Finally, *SOA management* is the process of managing services that are in production. There are multiple aspects to SOA management:

- The process of monitoring the availability and performance of the services and applications that use them.
- Rapidly restoring availability when issues are detected – or even finding problems before they impact an end user.
- Setting service level agreements between the consumer and provider, and reporting on the achievement of the service levels.
- Managing the policies that control the interaction of the services at runtime.

ITSM has been adopted by companies to improve the management processes listed above. When ITSM is implemented using the ITIL framework, the SOA implementation process is both accelerated and simplified. This is critical because *SOA only has benefits when it is successfully deployed, and ITSM helps make SOA deployment successful.*

Managing traditional applications is difficult enough – how long has it been since you've been affected by a performance or availability issue in a computer system? SOA management multiplies the complexity. As anyone who's deployed SOA in their IT environment knows, one hiccup in a system can cause serious degradation across several previously unrelated applications and business processes. Tracking down system issues can be tedious and consume a considerable amount of budget. Because ITIL focuses on monitoring and reporting IT resources, mature incident and problem management processes and tools can reduce the time needed to find a "needle in the haystack" when faced with problems in the system.

Better Business & IT Alignment

A major catalyst to SOA deployment is the promise of more efficient and cost-effective systems and services. Since IT operations are optimized for greater responsiveness, this comes in the form of agility. However, successful adoption of SOA requires updated processes. These processes, across governance, quality, and management, should be deployed following best practices.

As with all IT projects, SOA implementation must meet all mandated compliance initiatives, such as Sarbanes-Oxley. These regulatory and compliance requirements demand solid reporting of processes, something that implementing these best practices helps facilitate. By providing consistency and predictability, the best practices of ITSM help businesses become stronger and organizations run more efficiently, enabling the company to apply additional effort on other initiatives. SOA, similarly, since it improves flexibility

The number of companies adopting ITIL in the U.S. could grow to about 75% by the end of 2007
— IT SERVICE MANAGEMENT FORUM (ITSMF)

80% of the Global 2000 companies will adopt the ITIL framework by 2010
— FORRESTER RESEARCH

SOA will be used in more than 80% of new mission-critical applications and business processes by 2010.
— GARTNER

and lowers cost, frees bandwidth for IT to focus on its core mission of helping the organization achieve its business objectives.

SOA implementations with proper governance result in customers that are most satisfied with their overall implementation strategy. Results from a recent Ovum Summit report show a high correlation between a customer's level of satisfaction with SOA and the depth and breadth of commitment to ITSM best practices and tools, which includes service desks, asset and configuration management tools, IT portfolio management tools and business service management (BSM) performance monitoring dashboards. SOA customers who emphasize traditional development and test issues without considering how SOA changes the way they need to manage IT on a day-in-day-out basis tend to be the least satisfied.

Furthermore, from our experience, customers who have made progress implementing end-to-end ITSM strategies tied to business-oriented SLAs and supported with appropriate enabling technologies are twice as likely to report that their SOA investments are meeting their goals.

Successfully implementing and operating SOA environments clearly requires sophisticated operational, runtime management tools and strategies. Enterprises' experiences show that implementing SOA in tandem with ITSM best practices and tools is one of the best ways to ensure the SOA solution meets their goals and is a value-add to the business.

The bottom line is that to meet expectations with an SOA investment, it's valuable and important for ITSM to be in place according to ITIL best practices. If you've begun adopting ITSM, you're going to be more successful adopting SOA. If you haven't adopted either, consider adopting ITSM in parallel with SOA to save yourself time, money, and headaches. ■

About the Authors

Ken Hamilton is founder and past chairman of itSMF USA, and currently a director of the ITSM practice at HP Education Services. He is a recipient of the itSMF USA Lifetime Achievement award. Ken has over 20 years of experience in IT, is a certified IT service manager, and has led the development of IT service management methods and ITIL/MOF best practices in the U.S. As a director of ITSM practice in HP Education Services, Ken drives business development and growth of ITSM training services. He also develops HP's ITSM strategy in the rapidly growing market in the United States.

Mark LaJeunesse is the Service Oriented Architecture (SOA) program manager for Hewlett-Packard Services' consulting and integration business. Mark is responsible for the development of SOA services, sales and delivery training, driving HP's EAS SOA visibility and differentiation. Mark has 20 years in the high-tech industry.

Understanding SOA Architectures and Models

—continued from page 4

sharing information, something we've not been good at in the past. I can understand in certain industry-specific competitive situations where it's not in the best interest of the company to share information.

There are a few issues as I see things: First, as already stated, we need to clearly define use cases for SOA-RA and SOA-RM, and learn to adapt these models to the very different problem domains I'm seeing out there. Like any new notion, this will come with use, and these models and architectures will only be as valuable as the data points that are fed back to the creators, and the changes made to accommodate the "real world." As a SOA practitioner I'm finding that a huge chasm still exists between those who define the concepts of SOA and those who actually do the work. That chasm needs to narrow significantly.

Second, what's the uptake thus far when considering the concepts of SOA Reference Architecture and SOA Reference Model? It appears to me that most of my clients are pressing forward with their SOAs, formal models and architectures be dammed. Unfortunately, defining something in a formal way doesn't mean the rank and file will use it. I think the recent debate is a clear indication of that. I can go back in history and point to many instances where good formal models existed, but nobody leveraged them, opting for models that were more ad hoc, understandable, and available. I do promote this effort, however. Please don't misunderstand me.

Finally, SOA, at least in my mind, is a bit more systemic to enterprise architecture than just an "architectural approach." Indeed, if SOA is to be successful, those responsible for enterprise architectures need to understand SOA. Many do, some do not. If SOA is going to provide the proper benefits, SOA needs to layer deeply into business processes, business strategy, information technology, and even capital planning. In essence, it becomes a new foundation in many respects, and as to the point I made many times before…a good SOA is a good enterprise architecture, and not a mere instance of technology or another silo. We need solutions, not another "bolt on" that just adds complexity, and hinders agility.

Hope this helps. On to more technical topics next. ■

Fiorano SOA™ 2006

The Quickest path to an SOA

- **FioranoMQ™ 2006 –** world's fastest, most scalable JMS
- **Fiorano ESB™ 2006 –** CAD/CAM for distributed applications
- **Fiorano BPEL Server –** simplifying business process orchestration
- **Fiorano Tools –** BPEL Studio, Mapper, FEPO, etc
- **Fiorano Components –** 60+ pre-built adapters

Benefits

- **Adherence to popular industry standards** - JMS, COM, .NET, JCA, JMX, BPEL, SOAP, etc.
- **Multi-language, Multi-platform, Multi-protocol**
- **Unmatched Scalability and High Performance**
- **Quick, Measurable ROI**

Download your copy of Fiorano today!

www.fiorano.com/downloadsoa

Fiorano®

Enabling Change at the speed of thought

My Design Time Is Your Runtime

The effects of decoupled lifecycles on SOA development & quality

WRITTEN BY **JIM MURPHY**

You've deployed a Web Service. It's been designed, built, tested, and published to your registry for use and is now a critical component in a strategic business application. Your SOA is beginning to show signs of return but what happens when other architects discover your service and begin to design new applications with it? On one hand, you're seeing the benefits of reuse, yet how can you manage a service that's undergoing continuous testing? What happens to your initial business application when one of its key services undergoes a load test for another composite application?

In a SOA, the characteristics of traditional design time and runtime are changed. The rules about ensuring quality are changing as well. No longer is it sufficient to follow design, build, test, deploy, and manage phases. Testing is continuous and can have a drastic impact on production applications. This article will examine the impact of SOA on traditional SDLC thinking and how testing can be done in production environments. We'll also explore the role collaborative SOA quality plays in lifecycle governance.

At the center of an SOA solution is the relationship between service consumers and service providers with distinct lifecycles. Separation of application functionality across lifecycles has profound effects on the development process and serious implications on managing system quality.

A Young Person's Guide to the SDLC

Wikipedia describes various process models that define a Software Development Lifecycle (SDLC) as a series of activities related to the development of software systems. The specific tasks vary between the process models that define specific software development processes but they are more the same than different. They boil down to the same basic steps:

* Analyze the Requirements
* Design the Architecture and Implementation
* Code and Build
* Test to Verify
* Deploy and Maintain

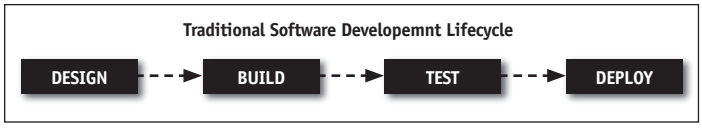


Figure 1: Typical activities of the SDLC

Make no mistake this *is* your father's software development process but it's the basis for many approaches to organized software development.

Agilists (AgileManifesto) have refined this model to make it fit the dynamics of software development and the needs of the business better by adding feedback loops, iteration, deferred requirements gathering, and continuous testing. Defined processes like the Agile Unified Process (AmblerAUP), XP Lifecycle (AmblerXPLifecycle) and the Agile SDLC (AmblerAgileSDLC) are examples of agile refinements to the basic software development process itself.

Traditional or agile the SDLC process has a start, a middle, and an end.

This is not the case with an SOA designed to meet business needs over time. An SOA has no practical notion of start and end much

like the Internet has no clearly delineated analysis, design, build, and test activities but is a composite of several interacting components that are constantly coming and going, changing and evolving, and to a greater or lesser extent interacting.

The Impact of SOA on the SDLC

It's important to realize that the SDLC pertains to the service construction process not to the overall SOA evolution. SOA activities focus on higher-level concerns across a service inventory like granularity, coupling, reuse, and governance. This difference is often missed by teams accustomed to building distributed systems or integrations following an SDLC process.

Building SOA like traditional distributed architecture is identified in Thomas Erl's SOA systems report as the #1 obstacle in SOA adoption. The assumption in traditional projects is that you can change the client and the server simultaneously through development. Design, coding, refactoring, and test occur across the boundary of consumer and provider as needed to make the whole system work. The fact that there are clients and servers separated by a network is more an implementation detail that traditional technologies often try to hide for the sake of the programmer.

Service-oriented solutions, as the name implies, focus on providing services to service consumers and that an exhaustive list of consumers and their requirements are not identifiable upfront. This is the essence of why SOA is interesting in business computing. It's incumbent on service providers to understand and anticipate the problem domain and build services that fit naturally in the larger service inventory to provide a valuable capability to a range of future consumers.

This service-oriented focus results in a separation of the service development and consumer development cycles.

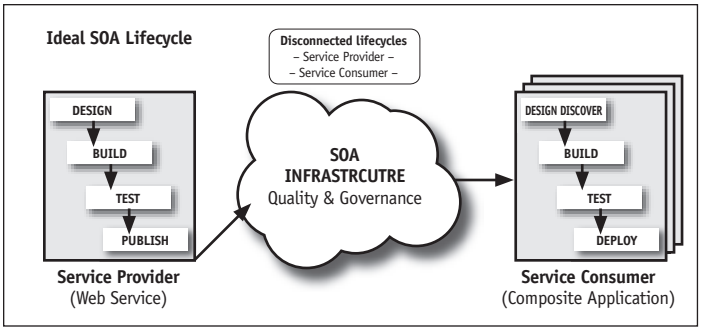


Figure 2: Ideal SOA lifecycles

Consumer developers are separated from service developers in time and space. Since there is a clear separation of lifecycles the terms design time and runtime depend on your point of view.

Often service providers follow a design, build, test, and ultimately publish phases where service metadata is made available in registry/repository.

At some future point service consumers discover services as part of a build vs. reuse design process. Services that fit the need are selected for reuse while missing or mismatched services are built. The service consumer project can then follow the recognizable lifecycle phases of build, test, and deploy.

Consumer Design Time at Service Runtime

Several thorny issues arise, in this common case, when a consuming application's design time occurs during its dependent services runtime. The challenges depend on the capabilities provided by the service provider.

An obvious first question is "What endpoint should consumer development use?" When a consuming application depends on external service managed by other people, how should development and test interactions happen? What techniques are available from all of the dependent services? Will you use live production service instances or is there service stand-ins available? What's the expected level of similarity between stand-ins and production systems if they do exist? These determinations need to be made for each service used by the consumer application. What expectations should the consumer team have about the consistency of approach between all of the services used? Do they all have staging instances or do they use different techniques?

For services that don't have staging instances available what is the external effect of development and test messages? How should load testing the consumer application or service? Are there chargebacks based on service use that apply? There are several techniques to manage the development and test interactions of consumers on dependent services.

1. **Staging Instances:** A service provider can make a separate service instance available with an identical service interface that's not connected to the live back-end. Well-mannered interaction with this staging system isn't important. Consumer development can have its way with the staging system that may even automate refreshing its back-end data periodically. It's expensive to build and maintain but there's not substitute for running the same code in a safe environment.
2. **Simulations:** Instead of building a full-fledged duplicate instance of a service a mock service-style simulation provides many of the same benefits as a service stand-in. The fidelity of the simulation is less likely than a full staging instance but it can be created and maintained by anyone if the service provider doesn't offer an alternative.
3. **Special Accounts:** Using a live service endpoint it's possible to identify your intention by using an explicit test account during authentication. A test account issued by the service provider lets the service provider handle test messages differently or at least correlate transactions with a test account.
4. **Message Header:** A custom SOAP header can assert that a message is intended as a test message. The message can trip the service implementation to operate in an idempotent mode or to correlate transactions for post processing and possible compensation.
5. **Data Partitioning:** A degenerate case of two-four is the use of specific application data to partition development and test requests from "real" requests. Domain-specific options include using a distinguished warehouse name in an inventory service or a particular book title/ISBN in an online store to indicate a test request.
6. **Compensating Transactions:** If none of the above techniques are suitable live transactions may need to be performed and followed up with compensating transactions to return the system to a base-level state. In HR systems hiring a long running test might involve "test employee," giving them a raise then firing them. Keeping the transaction state manually during development and test is probably more trouble than it's worth. It seems plain that sooner rather than later something will fail to get cleaned up properly.

Service Design Time & Consumer Runtime

This is the service provider's chance to break all its consumers and make them wary of ever using that service, or any service, ever again.

Clearly service providers need to keep their consumers in mind when making changes to the service's interface or even its imple-

mentation particularly if the new service version drops in at the existing endpoint. Of course this is easier said than done.

Understanding consumer expectations of service behavior is a complex analysis that begins with analyzing changes to the service interface description but must also include runtime semantics like data coherency and consistency. It may also include understanding how your service is used with other services you've never heard of.

Knowing when you're about to break a consumer is very difficult in practice. Analyzing the service interface changes is only the beginning. Service providers likely don't know how their consumers are using the service in conjunction with other services. The best way for service providers to know is for consumers to tell them very specifically by sharing their tests. When service consumers assert the precise nature of service usage, service providers can include their consumer's perspectives in their regression suites. Consumer-provided shared tests form a folksonomy of semantic expectations from the consumer community.

Once a mechanism for sharing tests exists it's a natural extension to begin testing for behavior that doesn't exist yet as a way to drive enhancements in service behavior. Voila, test driven development for SOA!

Consumer applications make build-versus-reuse decisions of services based on what's available in the service inventory and how well it matches requirements in much the same way we buy software from vendors.

Some reuse decisions will be simple. However; we're all human and have imperfect knowledge of future requirements so consumers will eventually need changes to a service interface, its implementation, runtime characteristics like availability or SLA, or even back-end data access and consistency.

This feedback flows as new requirements to the service provider team and can influence future development the same way new features appear in commercial software applications: when the next version ships. This can tie the delivery schedule of a consumer application to the delivery schedule of its dependent services. Schedule coupling rewards service providers who anticipate the future as much as possible and productize service delivery by following the Software as a Service (SaaS) model where the service is productized.

By treating your services as products delivered as services you can take a more holistic approach including:

1. A mechanism for gathering consumer feedback
2. Notification of upcoming versions
3. Forward and backward compatibility expectations
4. Old version support expectations
5. Impact analysis of change

Metadata Design Time?

Once services and consumers have cycled through their respective lifecycle phases and are both at runtime there's still room for change. New policies and policy changes that manifest in changes to runtime metadata can drastically affect deployed services and applications. Security, transformation location, and routing changes can change the way service availability, semantics, or performance in catastrophic or insidiously subtle ways react.

Overall SOA Quality

When a system exists in a single lifecycle its overall quality is under the direct influence of the development team. Time and resources can be added or changes made to improve system quality. For service consumers application boundaries are vague and no longer in their direct control so assessing and assuring quality becomes a distributed problem. Depending on the actions of each dependent service and changes to intermediary metadata the consumer application's quality is now a function of time and no longer isolated to any identifiable test phase.

This raises the question of "Who is responsible for overall SOA quality?" The individual services in the service inventory certainly have their place but who ensures the consistency and the framework to make SOA quality happen?

There are two sides to the SOA-quality coin. The first is in creating well designed and implemented service and applications in the first place. The second is in providing a resolution mechanism when the inevitable happens. There will be problems between consumers and services no matter what. Plan for the inevitable when consumers and providers need to find runtime problems and avoid figure pointing.

Lifecycle governance is the constraint framework to connect loosely coupled development processes. Extending current notions of lifecycle governance to include the provision of a quality framework lets consumers and providers manage the quality concerns of the other. High-quality service components are necessary but insufficient for healthy and holistic service-oriented architectures. ■

References

From a talk given by Amr Elssamadisy on SOA, Agile, and TDD at SD West:

"Here's the problem in a nutshell. Agile requires that you fix what you break. So if you change the service your team is writing, you must fix whatever that change breaks in the clients using that service. You're responsible for your own messes. SOA, on the other hand, usually assumes that there will be no collective code ownership. So you can't fix someone else's code. What's needed is a way for the team developing the service to run a test to see if their change will break any clients.

For that to work, though, the service team's test suite needs to know what breaks the clients. If they don't own the code then how does this happen? By sharing tests. Everyone shares all their tests with everyone else. That way the client's tests become part of the server team's regression test suite throughout the development process."

- *http://en.wikipedia.org/wiki/Software_development_life_cycle*
- *Ambler's Agile SDLC: http://www.ambysoft.com/essays/agileLifecycle.html*
- *Ambler's XP Lifecycle: http://www.agilemodeling.com/essays/agileModelingXPLifecycle.htm#Figure1XPProjectLifecycle*
- *Ambler's AUP: http://www.ambysoft.com/unifiedprocess/agileUP.html*
- *ErlsTop8SOAPitfalls: http://www.infoq.com/articles/Top-8-SOA-Adoption-Pitfalls*
- *AgileManifesto: http://agilemanifesto.org/*
- *DCOM-COM with a Longer Wire: http://msdn2.microsoft.com/en-us/library/ms809320.aspx*

About the Author

Jim Murphy is vice-president of product management at Mindreef, Inc., and is responsible for overseeing the complete suite of Web Services testing and SOA quality products. He has been with Mindreef for more than four years, most recently as lead software architect. Jim brings more than 12 years experience designing, implementing, testing, and debugging distributed software architectures using Java, .NET, C++, and XML. Before Mindreef, he was an independent consultant and served as a director, software architect, and senior software engineer at several early stage product and consulting organizations.

JavaOne℠

Java™

OPEN POSSIBILITIES

May 8–11, 2007
The Moscone Center, San Francisco, CA
JavaOne Pavilion: May 8–10, 2007

java.sun.com/javaone

> JAVA™ TECHNOLOGY IS NOW OPEN–AND SO ARE THE POSSIBILITIES

The 2007 JavaOne℠ conference has expanded and is definitely one conference you won't want to miss. With the decision to open source Java™ technology, 2007 marks a major milestone for the Java platform. Whether your passion is scripting languages, open source, SOA, Web 2.0, mashups, or the core Java platform, this is a conference that has something for almost all technology developers.

LEARN MORE ABOUT*:

- > Scripting (JavaScript™ Programming Language, PHP, Ruby on Rails, Python, and More)
- > Open Source and Community Development
- > Integration and Service-Oriented Development
- > Web 2.0 Development
- > AJAX
- > Java Technology and the Core Java Platforms (EE/SE/ME)
- > Compatibility and Interoperability
- > Business Management

SAVE $100**
Register Today!
Please use priority code: J7PA3SOA

* Content subject to change.
** Offer not available on-site.

Attend the JavaOne conference, and you will have many opportunities over the course of four days to network with like-minded developers; attend in-depth technical sessions; engage with your peers in Hands-on Labs and BOFs; and experience general sessions featuring speakers from Intel Corporation, Motorola, Oracle, and Sun Microsystems. Meet face-to-face with leading technology companies, and test-drive the latest tools and technologies shaping the industry.

PLATINUM COSPONSORS

intel
MOTODEV The Motorola developer network
ORACLE

GOLD COSPONSORS

bea Think liquid.
NAVTEQ
NOKIA

SILVER COSPONSORS

INTERSYSTEMS
PARASOFT. We make software work.
TERRACOTTA

Exploring the Convergence of Service Management Paradigms in Enterprises

The value proposition of Business Services Management and IT Infrastructure Library for SOA

WRITTEN BY **NAVEEN KULKARNI AND KRISHNENDU KUNTI**

Enterprises have started to realize the value of Service Oriented Architecture (SOA) and have adopted it as a mainstream enterprise IT architecture. However it's important for enterprises to realize that SOA presents a significantly different business and technology adoption model from a management perspective.

Business Service Management (BSM) is an approach to managing and making visible the impact of changes in business and IT. The IT infrastructure library is a framework for managing IT services. In this article we will explore the challenges in managing services in SOA and propose a way for BSM and ITIL to be used to build a platform for managing services in an SOA ecosystem.

Service Management in Enterprises

Broadly put services can be categorized into two types, services offered *by the business* and services offered *to the business*. A service *offered by the business* refers to the way capabilities are organized so they add value to the customers of the business. Services offered *to the business* are those that enable the enterprise to carry on its daily operations. It's important to note that services offered *to a business* diminish in significance if they're unmanaged and can't guarantee optimal performance, in turn greatly affecting the services offered *by the business*.

From the IT perspective , enterprises offer their business services through processes, applications, systems, and networks. All these are lined up to meet specific business needs. Traditionally service management has referred to the services offered by the IT support community that managed the software assets as well as the infrastructure. The most widely accepted framework for this IT service management (ITSM) has been the popular IT Infrastructure Library (ITIL) approach. Some ITSM implementations have begun to look at managing the processes and information part of an enterprise besides the IT support services. Complementarily, Business Service Management (BSM) is a process that provides visibility into the activities performed by the business through measurement and monitoring.

BSMbased approaches for measuring services *offered by the business* are vital for handling dynamically changing businesses, but they're still local in nature and aren't comprehensive enough to handle the management of all parts of the enterprise as required in the face of business changes. The divide between business and IT is the root cause of this so a bridge should be built between business and IT that finds comprehensive ways of addressing the management of the services offered *to the business* alongside the management of the services offered *by the business*.

The IT Landscape in the Service Ecosystem

SOA stands on the twin pillars of reusability and interoperability. The "reusability" factor derives from provisioning reusable business services that can be consumed by a number of players both within and at the edge of the enterprise. SOA promotes at its roots

a loosely coupled architecture as opposed to the point-to-point-based tightly coupled architectures in traditional enterprises. SOA can be manifest at multiple levels of enterprise IT, namely resource, application/data, middleware, and the processes that result in a layered architecture as shown in Figure 1.

Services created at each of these layers cater to top-level business functions. However, an important consideration even in such a service ecosystem is that the traditional applications that are either custom-built or commercial off-the-shelf (COTS) would still continue to co-exist with new composite applications that are process/service-centric.

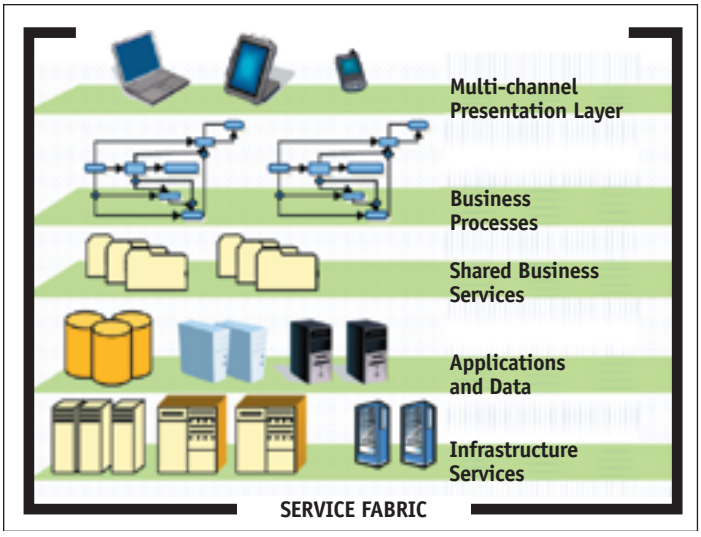


Figure 1: SOA levels of enterprise IT (Source: Infosys Research)

Another aspect of this paradigm shift is that business processes no longer exist in a single application but might be created by assembling services provided by multiple units in the enterprise and stakeholders outside the enterprise. As a result, there are multiple owners in a service ecosystem each with service owner owing a part of the business process. In addition there are other shared infrastructure elements in a SOA, for instance, the enterprise service bus infrastructure owned by corporate IT that provides value-added services such as translations, content-based routing, reliable messaging, message tracking, etc. Hence the shared features of a process like security, reliability, change management; incident management, etc. depend on the service providers of these shared infrastructure/services.

SOA Management in the Enterprise

Traditionally the application-centric or infrastructure-centric view of IT management has been adopted in enterprises. In the existing IT management approach an application usually has a known set of users and the application generally implements a single business function where the application in question is deployed on a single piece of infrastructure.

In such a scenario IT management is simple, involving the monitoring of the related infrastructure and applications, which can be taken care by very high-level SLAs (application or infrastructure level). In the event of a change or incident at both the infrastructure and application level only a certain set of users is affected and the change/incident is limited to the application or within infrastructure boundary. Hence policies to mange such incidents and related changes aren't complex.

However, in a service-based ecosystem processes span services deployed on multiple machines, where diverse and large numbers of users consume these services. In event of an incident or change the effect is no longer limited to a service or piece of infrastructure, for instance, if a service interface changes (say a new release) then all the processes that consume the services need to incorporate the change. Hence the existing approach of application- or infrastructure-centric management of IT isn't adequate for managing services.

Service management needs to take into account the impact of layering in the service ecosystem. For instance, resource tier services can be managed by the enterprise infrastructure team, services created at the business service layer can be managed by business owners whose SLAs are in turn driven by individual application level service SLAs that are managed by the service owners. This clearly identifies the need for properly demarcating service management for each layer and across the different stakeholders. This requires a top-down view of SOA management, linking business processes and the underlying IT resources.

BSM and ITIL for SOA Management

The value proposition of using BSM and ITIL for a service ecosystem is that combined they provide a comprehensive approach to SOA management.

In this IT-enabled business world these two together provide a recipe for Business Technology Optimization (BTO). Inherently both BSM and ITIL are driven from the business perspective and take a business-centric view of IT capabilities in an enterprise. As SOA is adopted to align business and IT, both BSM and ITIL are the right fillers to help enterprises achieve true agility. Figure 2 shows a comprehensive SOA management approach with BSM and ITIL. Together, they provide the ability to:

- Map business objectives the way IT operates, facilitating better visibility
- Create a process-driven ability for management that offers better control
- Create mechanisms to deliver IT operations as services to administer ITSM effectively
- Create a common vocabulary for the management activities for better communication

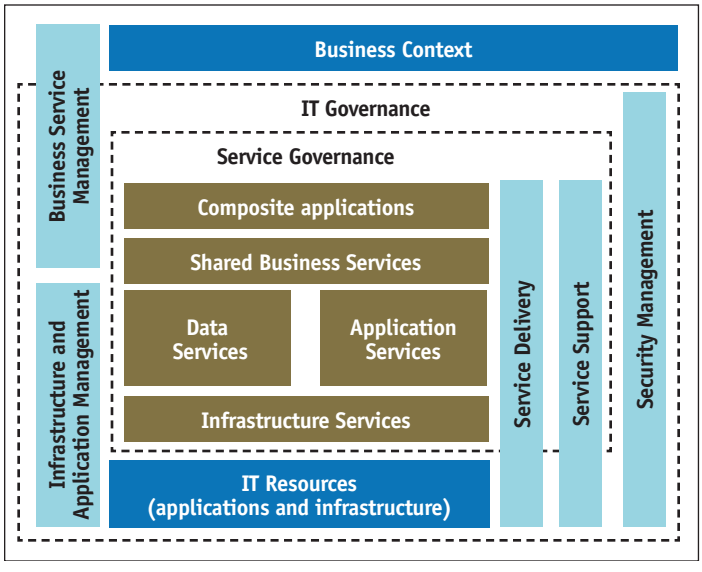


Figure 2: Comprehensive SOA management with BSM and ITIL

BSM provides a method to connect composite applications and business services with the underlying ITSM. It helps to understand and predict how technology changes can impact business and how changing business needs have to be accommodated into IT resources. While the BSM provides mapping between critical business services and IT services it derives business relevance from tying the end-user experiences (in a given business context) as shown in the Figure 2. ITIL is basically a non-proprietary framework for managing IT services proposed by the United Kingdom's Office of Government Commerce (OGC). ITIL provides a set of guiding principles *for aligning business goals* to underlying IT. Figure 2 shows the application of different ITIL modules to services at different levels of enterprise IT for SOA management. The guidelines from ITIL are applied while roadmapping the service adoption in an enterprise.

Governance is considered an important cornerstone for successfully realizing a SOA. A SOA management strategy should ultimately lead to an effective IT governance structure. SOA initiatives in an enterprise would always include people, process, and technology elements. ITIL primarily enforces good governance through its elaborate set of best practices and guidelines around people and processes. This creates a common ground where ITIL and SOA can converge and provide an effective and efficient mechanism to deliver and manage business services. As shown in Figure 2, the recommendations of the service delivery and service support modules of ITIL can establish a good governance structure for managing functional services.

SOA management should also cover existing legacy assets since they're integral to SOA.SOA management should consider improving the operational efficiency and cost-effectiveness of these legacy IT resources. ITIL provides a common vocabulary. As shown in Figure 2, SOA management would adopt ICT infrastructure management and application management modules of ITIL to provide greater visibility.

Areas of Application of BSM and ITIL in SOA Management

Both BSM and ITIL provide at best a framework for implementing a better IT management solution, but stop at suggesting any concrete implementation; it's left to the enterprises that adopt them to build relevant operational solutions or buy BTO solutions. Product companies offer a variety of products with ITIL-based capabilities like IBM's Infrastructure Resource Management (IRM), Tivoli Business Systems Manager (TBSM), Tivoli Service Level Advisor, or BSM-based solutions from Managed Objects. But most of these products offer specific solutions hence a combination of these would be needed to cover comprehensively all aspects of SOA management like planning, governance, implementation, and operations. Here, we'll explore the areas of application of BSM/ITIL under different aspects of SOA management.

1. Planning

When planning for services management in a service ecosystem, the foundations of BSM could be adopted to provide high-level mapping between event management (events arising out of user interactions) and the service level agreements (attached to individual services) with a view to correlating management-related processes and IT services. Taking BSM into consideration during planning will also aid in driving the performance management implementations to be tightly tied to end-user experiences. Further, ITIL can help business managers or stakeholders who are involved in the planning phases of business services in areas like business continuity planning, change management, and service level management. Based on BSM and ITIL's perspectives business managers can lay the foundations for SOA management by adopting the guidelines and practices advised in the second book of ITIL – Planning To Implement Service Management.

2. Governance

The IT Governance Institute defines governance as "a structure of relationships and process to control enterprise in order to achieve the specific agreedupon goals and adding value while balancing the risk versus return over IT and its processes." It goes without saying that governance would put in a structure to control activities to mitigate risks. Management, especially from ITSM perspective, would play a major role in streamlining control activities. This relationship gives governance an important facet to consider when considering management. Service governance, which would be a subset of the overall IT governance, would ensure the delivery of functional services of high quality and on time. Hence service governance from the perspective of ITSM has to take care of:

- Establishing overall standards for creating an infrastructure and providing control directives for the associated services.
- Managing the delivery of a service-based ecosystem to business units or service consumers in collaboration with the stakeholders or owners
- Providing and defining structures for managing SLAs, quality, vendor, and other control activities.
- Building appropriate decision support systems to improve the use of IT support for functional services.

As shown in Figure 2, the service delivery and service support modules of ITIL are used to establish service governance. Further, BSM and other modules of ITIL would help set up overall IT governance.

3. Implementation

As mentioned above, BSM defines processes, and ITIL provides a set of guidelines towards managing IT services. It's essential for those who implement the system to use the right set of frameworks and tools to build the platforms for ITSM and eventually IT governance in an enterprise. This can be elaborated further as:

- **Methodologies and frameworks for service identification:** The first step in implementing ITSM in any service ecosystem is to identify services that serve to align business and IT at all levels of enterprise IT. At this stage appropriate methodologies and frameworks can be used to identify the service with the right level of granularity.
- **Registry, Repositories, and Configuration Management Database (CMDB):** Service registries-repositories and configuration management databases are used to maintain service definitions, metadata, and documentation pertaining to services. They can also be used for provisioning service lookup, service versioning, and other information like service owners, systems using the service, service metrics locations, etc.
- **Tools for visibility and control:** Tools are essential to provide the ITSM platforms much needed visibility and control. Visibility for processes and service is achieved using process-level and service-level monitoring tools. Control over changes, incidents, and problems are managed through appropriate practices. Process-level monitoring tools map business process to applications, services, and infrastructure on which these applications or services are deployed. These tools provide a cockpit view of

deployed processes; the user can also monitor process-level metrics and track failures and performance bottlenecks in individual services, applications, or infrastructure. Tools used for service metrics monitoring are implemented as a part of service implementation code or they can run as an external agent to monitor service metrics. Service metrics can be compared against SLAs to generate compliance reports. These tools also generate alert notifications and send them to the right stakeholders after accessing data from the registry or configuration database. Tools for change management and incident management could be used to define workflows that capture actions that need to be taken to manage a change or an incident. These tools should also take care of approvals and coordination among the multiple stakeholders involved in the change process. Tools for problem resolution are used to acquire information leading to problem diagnosis and resolution. These tools provide maintenance of system knowledge and problem diagnosis. They should do automated andmanual diagnosis and integrate with other tools like service metrics, process monitoring, and change management to get live information updates on the status of any problems.

- **Compliance and Security:** These are the most important of all the ITSM activities to be implemented with all inclusive resources and assets, spanning all levels of enterprise IT. Appropriate measures have to be built into the ITSM platform and tools to reflect the conformance and security measures necessitated by the business. For example, regulatory compliance with Sarbanes-Oxley (SOX) demands things of information that strain ITSM to build capabilities related to access control, detection, evaluation, and reporting. Security policies have to be woven into every operational capability of ITSM. Tools that provide visibility have to highlight the incidents related to compliance and security and work with CMDB and repositories to provide and apply the related policies to enforce control.

4. Operations

Based on the ITIL guidelines of service delivery and support, to provision in SOA in accordance with the SLAs between the service providers and consumers and to support the service lifecycle as applied to functional services in an enterprise and fine-grained policies for services, the following points should be evolved:

- **Service-level management:** Policies around the business practices to ensure that agreed upon services are delivered according to timeline and business needs. The terms agreed are represented through well-defined SLAs that can be attached to templates available with ITIL. This step should also draw on principles of BSM to identify the IT process and corresponding measurable parameters to determine success.
- **Capacity management:** This can be seen in three dimensions, that is, business, service, and resource. First, the business dimension is directly related to service-level management. Defining the appropriate processes will let future business requirements for IT services be identified, forecast, and modeled in a timely fashion. Second, for functional services and enterprise applications (CRM, SCM, etc) policies have to be defined to guide the appropriate provisioning to fulfill the current service SLAs and future demand cost-effectively. Third, procedures and practices have to be developed to acquire and add resources for new applications catering to the services, right-sizing the infrastructure, and monitoring them for optimal consumption.

- **Availability management:** ITIL guidelines will ensure the sustained availability of services cost-effectively. In an enterprise this might involve service metrics monitoring, alerts, and acceptable downtimes based on business impact analysis. Adopting measuring techniques in terms of mean time to repair, mean time between failures, and mean time between incidents will offer insights into the nature of the problems faced at each layer that can facilitate the creation of continuity plans specifically targeting the type of functional service.
- **Continuity management, disaster recovery:** With critical business services, ITIL recommendations can be adopted to define plans to respond to incidents during crises or outages including alternative arrangements. For other functional services and IT resources, specific recovery teams (part of IT supply) can be formed with detailed instructions to adhere to mean time for recovery as defined by availability management.
- **Financial management:** The guidelines for the financial management of IT services provided in ITIL can be used in service-based ecosystem for developing costing models for the services being consumed. Some costing policies such as recovery-based, markup value-based, consumption-based, market-based, and organizational unit-based could be adopted for functional services. Apart from charging, practices of budgeting and accounting as applied to IT services can be used for functional services as well. This would help in decisions on investments for creating the service portfolio. A detailed cost analysis on a service will assist in tracking the return on investment (ROI) on each functional service.
- **Configuration management:** ITIL refers to the Configuration Management Database (CMDB) that stores all the configurable items for ITSM along with their relationships to related data. These guidelines can be used to create repositories for a service ecosystem to aid in provisioning functional service with all related information to support better reusability and form the basis for other management activities.
- **Incident and problem management:** Provides guidelines for minimizing the impact of incidents and problems and a means to record the incident and determine the root cause of problems hence avoiding similar issues in future. In an service ecosystem exceptions might be raised in the operation of functional services. Logging them into a centralized monitoring application based on ITIL processes will let the service environment record and classify the exceptions, identify trends and investigate/diagnose, do root cause and impact analysis, issue change requests to fix the problem. In case of an incident the service desk will have access to centralized applications like the services monitoring console, individual service metrics logs, and problem resolution tools for pinpointing individual service-level failures and will the problem or give the user a timeframe for provisioning a solution based on SLA.
- **Change and release management:** ITIL provides guidelines for the change and release management of applications and software. In a service-based ecosystem a change and release management function would involve versioning services, defining policies for a service rollout, its retirement, and upgrading. Changes are introduced through new features or fixes to problems (that come in as change requests from problem management). Such changes might involve creating a new interface to the service or a new version. Configuration update policies of ITIL can be adopted so that registry-repositories are updated with change information that can be discovered later by the service consumer. ITIL practices regarding change control, prioritizing, estimating the impact of change, the approval process, reporting, and communicating could be applied to the functional service portfolio. The change and be introduced into a production or live environment following the release management activities prescribed by ITIL.

Summary

As SOA becomes mainstream in an enterprise, the management of service proliferation at multiple levels of enterprise IT will be a real challenge. If not managed properly it will be a major deterrent in scaling the SOA-based ecosystem. So creating a proper governance structure for the management of IT services is mandatory. ITIL provides a set of guidelines for managing IT services and proves to be an ideal combination along with BSM for managing services at all tiers in an SOA-based ecosystem. However, to implement BSM and ITIL for managing services in SOA, organizations should use the right set of methodologies, tools, and frameworks along with the guidelines provided in BSM and ITIL.

Thus, overall, we can visualize the role BSM and ITIL can play in providing a comprehensive approach to managing the services offered *by an enterprise* alongside the services offered *to an enterprise* in an integrated manner.

References

- *Managing IT business risk. Mercury: http://a330.g.akamai.net/7/330/25828/20070112171555/graphics.eiu.com/ebf/PDFs/Managing_IT_Business_Risk_FINAL.pdf (accessed Mar. 09, 2007)*
- *IT Infrastructure library, http://www.itil.co.uk/ (accessed Mar. 09, 2007)*
- *ITIL and ITSM directory: http://www.itil-itsm-world.com/index.htm (accessed Mar. 09, 2007)*
- *Change Management + Control = Higher Availability. Solidcore Systems: http://research.pcpro.co.uk/detail/RES/1168953930_612.html*
- *SAML. Security Assertion Markup language: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security*
- *Enterprise service bus: http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=landings/esbbenefits (accessed on Mar. 9, 2007).*
- *Jason Bloomberg. SOA-enabled BSM. , Zapthink. January 2006.*
- *A. Shum, A. Dhilion. Achieve IT Agility by integrating SOA with ITIL based BSM,.*

About the Authors

Naveen Kulkarni is a senior technical specialist with the Web Services/SOA Center of Excellence in SETLabs, the R&D arm of Infosys Technologies. His contributions include understanding the intricacies and identifying a definitive approach to solving the problems of services-based computing. His areas of interest and current research include enterprise adoption of services, legacy modernization, and capacity planning as well as QOS in services ecology.
Naveen_kulkarni@infosys.com

Krishnendu Kunti is a senior technical specialist with the Web Services/SOA Center of Excellence in SETLabs, the R&D arm of Infosys Technologies. He has contributed to the architecting, designing and testing of SOA-based data services frameworks for enterprise-grade deployment of data services. His areas of interest include Service Oriented Architectures, data services, the Enterprise Service Bus, and enterprise architectures.
Krishnendu_kunti@infosys.com

> *Businesses that ignore the potential of SOA will find themselves outpaced by rivals who improve their agility and transform themselves into new kinds of enterprises*
>
> — Yafim Natis, Gartner Analyst

3-DAY EVENT!

SOAWorld 2007

Plus

Enterprise OpenSource Conference & Expo 2007

TOPICS INCLUDE:

SOA Web Services

- AJAX and SOA
- Web 2.0
- Universal SOA
- Protecting Web Services
- Troubleshooting SOA
- Governance
- Open-Source SOA
- XBRL
- Service Virtualization

Open Source

- Open Source Business Models
- Open Source ESB
- OpenAjax Alliance
- SaaS and Open Source
- Spring, Hibernate and Eclipse
- Seam
- Open Source Penetration
- Monetizing Open Source
- Open Source Databases
- AMQP
- Open Source Middleware

June 25-27, 2007
Roosevelt Hotel / New York City

Register Online! www.SOAWorld2007.com

SOAEOSCONFERENCE.SYS-CON.COM
REGISTER ONLINE TODAY
SAVE $200!
(HURRY FOR EARLY-BIRD DISCOUNT)

11th International SOAWORLD 2007 CONFERENCE & EXPO

2nd Annual ENTERPRISE OPENSOURCE 2007 CONFERENCE+EXPO

2007 is to many industry insiders shaping up to be a major inflection point in software development and deployment, with SOA, Web Services, Open Source, and AJAX all converging as cross-platform and cross-browser apps become the rule rather than the exception.

Accordingly the 11th International SOA Web Services Edge 2007 again seeks to offer comprehensive coverage and actionable insights to the developers, architects, IT managers, CXOs, analysts, VCs, and journalists who'll be assembling as delegates and VIP guests in The Roosevelt Hotel in downtown Manhattan, June 25-27, 2007

Co-located with the 2nd Annual Enterprise Open Source Conference & Expo, the event will deliver the #1 i-technology educational and networking opportunity of the year. These two conference programs between them will present a comprehensive view of all the development and management aspects of integrating a SOA strategy and an Open Source philosophy into your enterprise. Our organizing principle is that delegates will go away from the intense two-day program replete with why-to and how-to knowledge delivered first-hand by industry experts.

Visit soaeosconference.sys-con.com for the most up-to-the-minute information including... Keynotes, Sessions, Speakers, Sponsors, Exhibitors, Schedule, etc.

BROUGHT TO YOU BY:

SOA World Magazine
focuses on the business and technology of Service-Oriented Architectures and Web Services. It targets enterprise application development and management, in all its aspects.

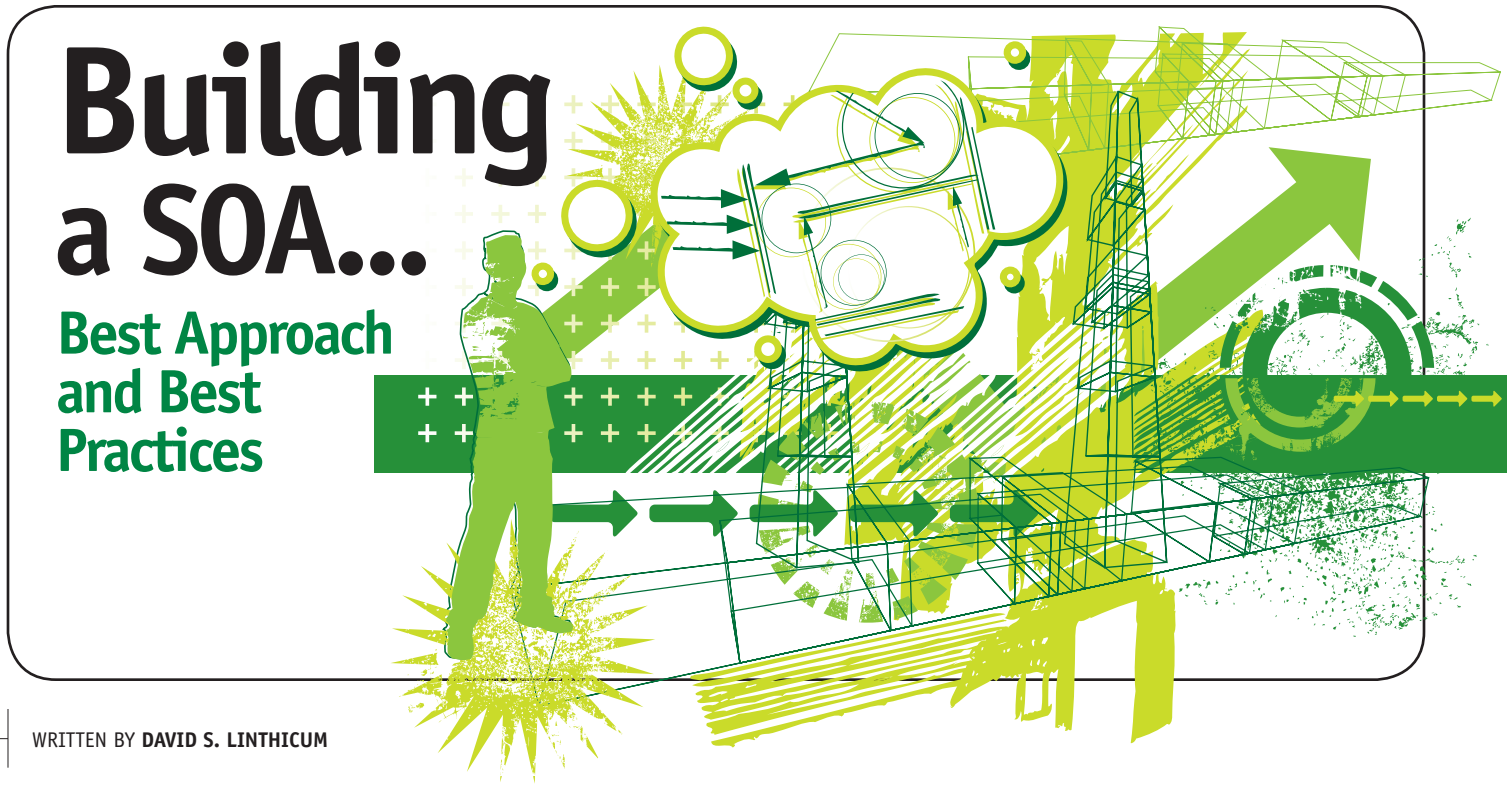
Enterprise Open Source Magazine
EOS is the world's leading publication showcasing every aspect of profitable Open Source solutions in business and consumer contexts.

SYS-CON EVENTS For more great events visit www.EVENTS.SYS-CON.com

Exhibit and Sponsorship Info:
Call 201-802-3020 or email events@sys-con.com

Building a SOA...

Best Approach and Best Practices



WRITTEN BY **DAVID S. LINTHICUM**

While the notion of SOA continues to emerge, those who are implementing SOAs today are faced with a variety of challenges, including the complexities of SOA, and the work involved with understanding their existing problem domain and requirements. Those who want to get SOA right the first time quickly understand the benefits of a sound architecture and a good set of SOA design approaches.

However, the understanding of how you approach your SOA, and best practices around building a SOA, are clearly lacking. Those who are looking to gain the benefits of SOA are perplexed by the wide reach of the technology, its links with traditional enterprise architecture, and how the game has changed since the structured and object-oriented analysis and design days.

Truth be told, SOA is not something you buy, it's something you do. Thus, while the focus has been on the technology in the last few years, as people begin to implement SOA, the focus is now on the process. So, how do you create a SOA? In this article I'll walk you through the notion of SOA, and the unique aspects of the architecture, as well as the design considerations, including the ability to leverage best practices as a guide. In essence, what's working, and what's not.

Understanding the Basics

While there is this notion that SOA is all new and thus requires new approaches, the fundamentals of SOA are firmly based in more traditional requirements, analysis, and design techniques, but with the concept of services and service-based solution instances as the new direction. To that end, you can consider the steps to the construction of a SOA as basic approaches, and with a foundation in traditional enterprise architecture and traditional application analysis, design, and development...processes that should not be all that new or scary for many who are facing SOA today.

Next we should define a "SOA Meta Model" or a reference model to better understand the notions of SOA and the component parts that are required. Figure 1 represents one view of SOA, working up from the data, to the data abstraction layer, to the data services, to the services, to the orchestration layers, and finally monitoring and event management. Note that both governance and security are typically bound to all layers. Also note the way you build this stack to meet your specific needs could be very much like this or very different depending on your requirements.

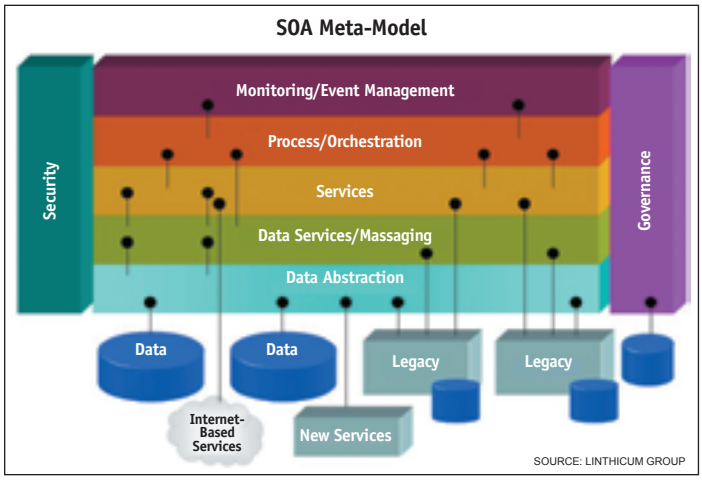


Figure 1: SOA meta model makes it easy to understand SOA before the SOA design and building process begins

The Procedure

Considering all the basics, you can divide the steps to creating a SOA into the following major steps:

1. Definition of the Domain
2. Semantic Level Understanding
3. Service Level Understanding
4. Process Level Understanding
5. New Services Configuration and Design
6. New Processes Configuration and Design
7. Technology Selection
8. Deployment
9. Testing

Note that both governance and security are systemic to the above processes, as is performance engineering. Moreover, you may have special requirements that require additions or deletions to these steps. We're also not defining most of the sub-steps here due to space constraints.

Defining the Domain

You can't boil the ocean, so you must define the scope of your SOA in an enterprise. Most SOAs are best implemented in small steps, such as moving a single division or a portion of a division to SOA, instead of an entire enterprise all at once. Small successes lead to larger, more strategic successes over time, and you need to establish the demarcation lines at the beginning of the project to provide better focus and understanding (see Figure 2)

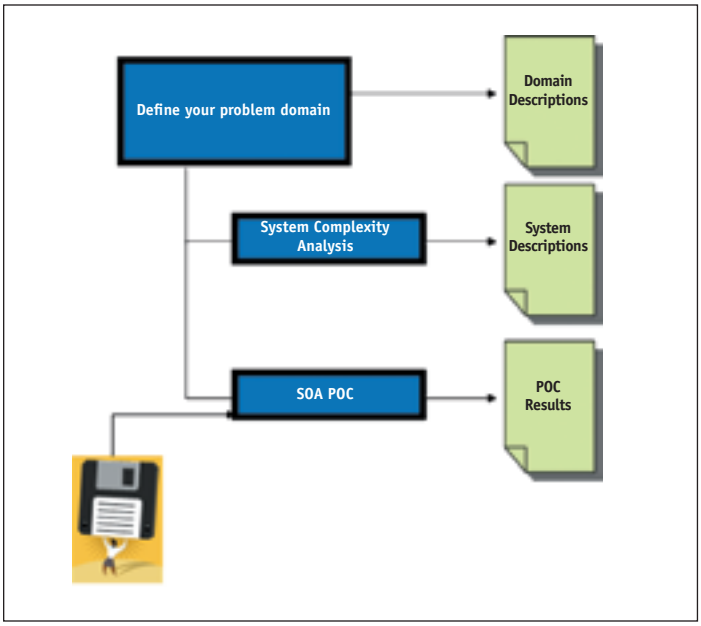


Figure 2: Defining your domain means you must understand system complexity and test the candidate technologies at the same time

Also note that now is a good time to do a proof of concept (POC). This is done at this point so the architect can understand the candidate technology a bit better, which means the right expectations are set as to the final solution. A POC also helps the architect understand the possibilities and limitations for considering service and process design in the target. It's really a point of learning, not a point of selection. Keep that in mind.

Semantic-Level Understanding

You can't deal with information you don't understand, including information bound to behavior (services). So it's extremely important for you to identify all application semantics — metadata, if you will — that exist in your domain, allowing you to deal properly with that data (see Figure 3).

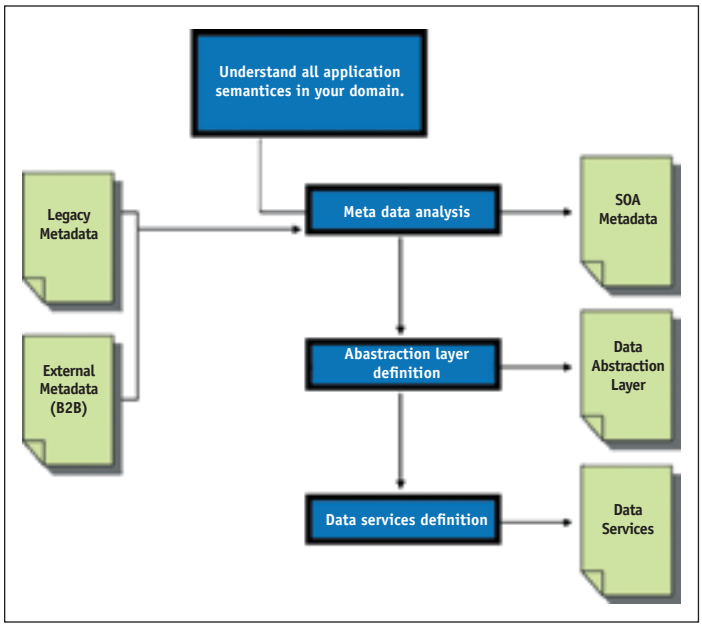


Figure 3: In understanding the application semantics we must understand the metadata and create a data abstraction layer or a logical understanding of complex data for use in the SOA

Understanding application semantics establishes the way and form in which the particular application refers to properties of the business process. For example, the very same customer number for one application may have a completely different value and meaning in another application. Understanding the semantics of an application guarantees that there will be no contradictory information when the application is integrated with other applications at the information or service levels. Achieving consistent application semantics requires an information integration "Rosetta Stone" and, as such, this represents one of the major challenges to creating your SOA.

Defining application semantics is a tough job since many of the existing systems you'll be dealing with are older, proprietary or perhaps both. The first step in identifying and locating semantics is to create a list of candidate systems. This list will make it possible to determine which data repositories exist in support of those candidate systems.

Any technology that can reverse-engineer existing physical and logical database schemas will prove helpful in identifying data within the problem domains. However, while the schema and database model may give insight into the structure of the database or databases, they can't determine how that information is used in the context of the application or service; that's why we need the next steps.

A data abstraction, or a data services layer, provides the loose coupling between the services and the underlying databases/information stores. This provides a point of configuration to both deal with the differences between the data as physically represented and the preferred logical representation to the SOA.

Moreover, as both the data and service requirements change over time to align with the business, the data abstraction layer can be easily reconfigured to account for the differences, and since not coupled, require a minimal amount of work. For instance, the physical databases in many instances doesn't have to be changed, just the abstraction. In essence you've created an agile data layer to go with your agile services and process layers. Without this you'll find that holistic agility with your SOA is difficult to achieve.

Service-Level Understanding

Service interfaces are quirky. They differ greatly from application to application, custom or proprietary. What's more, many interfaces — despite what the application vendors or developers may claim — aren't true service interfaces, and you need to know the difference. Services provide behavior as well as information, so they're service-oriented. There are some services that just produce information; those are information-oriented and shouldn't be included in this step. We are only interested in the former at this point.

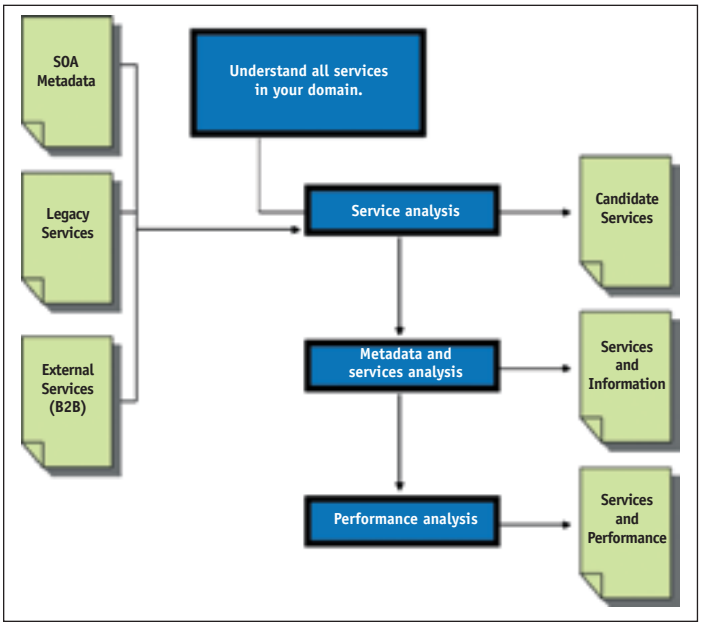


Figure 4: When obtaining a service-level understanding, we need to analyze the existing candidate services and bind the metadata to those services

It's important to devote time to validating assumptions about services, including:

1. Where they exist.
2. The purpose of the service.
3. Information bound to the service.
4. Dependencies (e.g., if it's a composite service).
5. Service access mechanisms.
6. Security issues.

The best place to begin with services is with the creation of a services directory, the artifact of this step. As with other directories, this is a repository for gathered information about available services, along with the documentation for each service, including what it does, information passed to a service, information coming from a service, etc. (see Figure 4). This directory is used — along with the now understood application semantics — to define the points of integration within all systems in the domain.

What is a service? Good question. When using a service, we leverage a remote method or behavior versus simply extracting or publishing information to a remote system. Moreover, we typically abstract this remote service into another application known as a composite application, which is typically made up of more than one service.

A good example of a service is a risk analysis process, which runs in an enterprise to calculate the risk of a financial transaction. This remote application service is of little use by itself, but when abstracted into a larger application — for example, a trading system — then that remote application service has additional value.

Note that we leverage the behavior of this remote service more than the information it produces or consumes. If you're a programmer, you can view application services as subroutines or methods, something you invoke to make something happen.

The basic notion of SOA is to leverage these remote services using some controlled infrastructure that lets applications invoke remote application services as if they were local to the application. The result (or goal) is a composite application or an orchestration (or choreography) made up of many local and remote services.

Process-Level Understanding

You need to define and list all business processes that exist in your domain, either automated or not. This is important because, now that we know which services and information sources and sinks are available, we must define higher-level mechanisms for interaction, including all high-level, mid-level, and low-level processes. In many instance, these processes have yet to become automated or are only partially automated (see Figure 5).

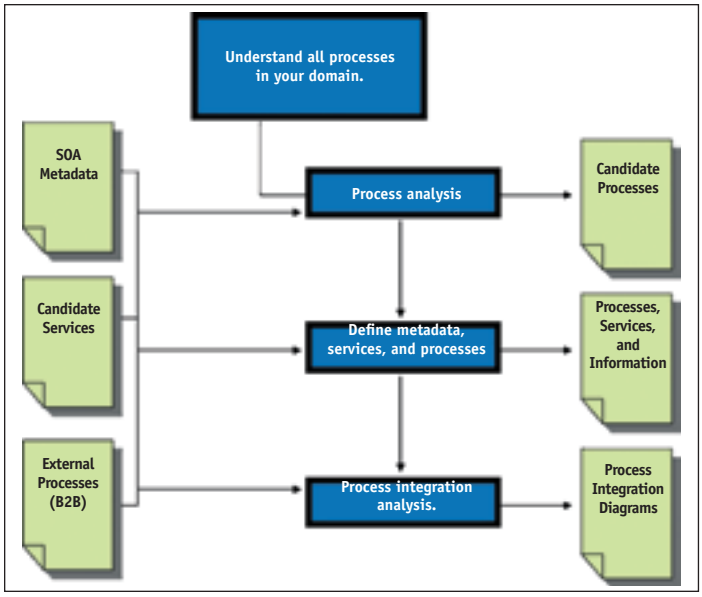


Figure 5: As we define and understand services, we do the same for processes

For example, if a SOA architect needs to understand all the processes that exist in an inventory application, he or she will either read the documentation or the code to determine which processes are present. Then the architect will enter the business processes in the catalog and determine the purpose of the process, who owns it, what exactly it does, and the technology it employs (e.g., Java or C++). These processes are later bound to new processes — or, meta processes — providing orchestration of encapsulated processes or services to meet some business need.

You should also consider the notion of shared versus private processes. Some processes are private, and so not shared with outside entities (or, in some cases, they're not even shared with other parts of

the organization). Other processes are shared, meaning that others leverage the same processes to automate things inter-enterprise. Private and shared processes can exist in the same process space with the process integration technology managing security among the users.

Other information can be maintained in the catalog; information that may include variables used in the processes, object schemas, security requirements, and/or performance characteristics. Each process catalog must maintain its own set of properties, custom-built for each specific application integration problem domain.

New Services Configuration and Design

This is self-explanatory. You must define and design all new services that are to make up your SOA (see Figure 6). These will typically fall into one of three categories.

First are services exposed out of existing systems or legacy services, such as ERP, CRM, legacy, etc. These kinds of services really are defined for you, since the services are one-to-one representations of internal application functions or interfaces exposed as Web Services (typically) to facilitate integration. In many instances they are the candidate services you designed in the previous step. You should note that we are calling these "new services," even though many are pre-existing, because they are now accessible by your SOA, exposing true services and not proprietary interfaces. Some of these may be established through a simple upgrade of an enterprise application (ERP, CRM, ERP, etc.) to a service-oriented version (e.g., exposing existing behaviors as Web Services).

The second type of services is composite services, which are services unto themselves that are made up of many different services. In many instances, these services are mere interfaces to many other services and don't add much if any additional functionality. These are complex services, since there are so many dependencies as well as information bound to composite services that you must understand before creating your SOA.

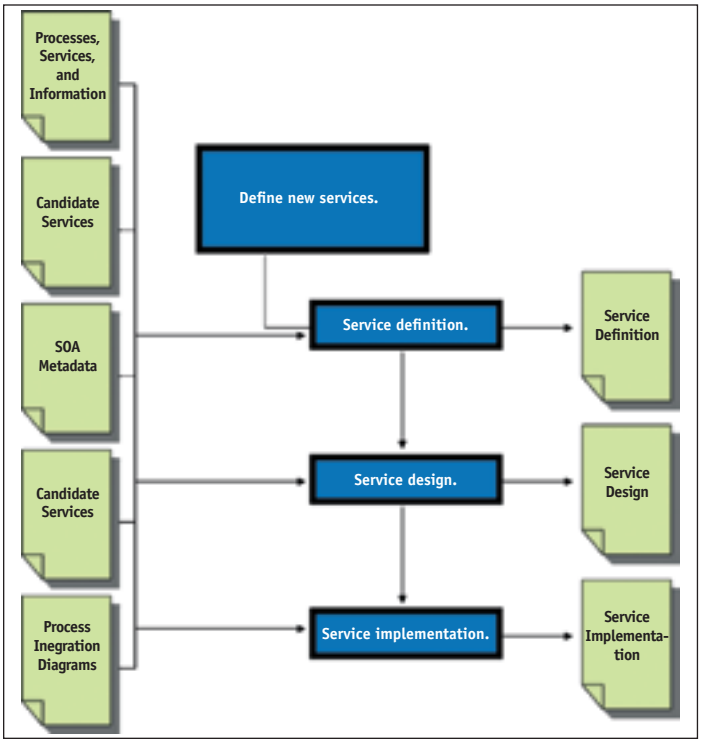


Figure 6: Now we need to take all we know and recast the candidate services as sets of services we'll employ in our SOA

Finally, data services, or services that do little else but represent information either at the data abstraction layer, or directly with the physical database. Most services that make up SOAs are data services.

However, you could have a domain that requires different types of services than those listed above. For instance, some SOAs leverage services that abstract embedded systems, and some SOAs use services that are outside of the enterprise, and so are not under their control. The service design and development process is unique to the requirements of your SOA. Keep that in mind as you design and build your services for your final architecture.

New Processes Configuration & Design

At this point we should understand most of what's needed to define new processes, as well as bind them to existing processes, and automate processes previously not automated. New processes should be defined that automate the interactions of services as well as information flows to automate a particular business event or sets of events (see Figure 7).

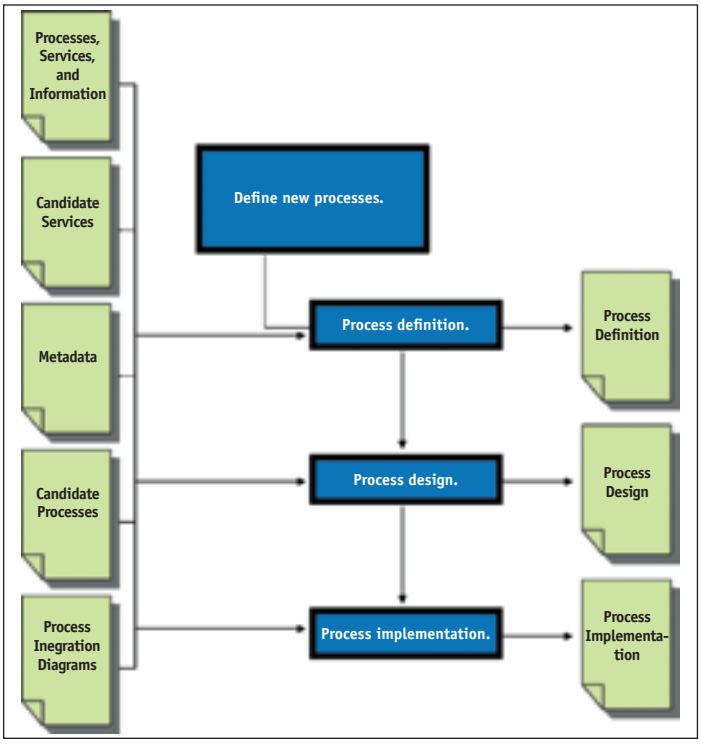


Figure 7: You define new processes much like you designed new services

While you can define some very complex logic in new processes using today's tools, the theme here is to orchestrate existing services and information flows rather than create new functionality. In essence, it's a meta-application that sits on top of many smaller applications, defining interactions from lower to higher levels.

Note that it's a very similar process to service design, and indeed most processes that are defined in orchestration layers, such as BPEL technology, can also double as services. However, processes typically leverage services, organizing them into solutions. That's the difference.

Technology Selection

Many technologies are available, including application servers, ESBs, governance systems, distributed objects, orchestration engines, and integration servers. The choice of technology will likely be a mix of products and vendors that, together, meet the needs of your SOA. It's rare for a single vendor to be able to solve all problems – not that that reality has ever kept vendors from making the claim that they can.

Technology selection is a difficult process that requires a great deal of time and effort. Creating the criteria for technology and products, understanding available solutions, and then matching the criteria to those products is hardly a piece of cake. To be successful, this "marriage" of criteria and products often requires a pilot project to prove that the technology will work. The time it takes to select the right technologies could be as long as the actual development of the SOA. While this might seem daunting, consider the alternative – picking the wrong technology for the problem domain. A bad choice practically ensures the failure of your SOA.

Deployment

This is the "just do it" step, meaning that you've understood everything that needs to be understood, defined new services and processes, selected the proper technology set, and now it's time to build the thing.

Testing

To ensure proper testing, a test plan will have to be put into place. While a detailed discussion of a test plan is beyond the scope of this article, it's really just a step-by-step procedure detailing how the SOA will be tested when completed. A test plan is particularly important because of the difficulty in testing a SOA solution. Most source and target systems are business-critical and so can't be taken offline. As a result, testing these systems can be a bit tricky.

Finding Your Own Path

One thing to understand is that I haven't (and couldn't) define everything you must do to create a successful SOA project. My goal has been to outline most of the activities necessary for your SOA project. In some case you'll need to add some steps to accommodate your particular needs. I'm not sure anyone should ever delete steps. However, I do have clients with simpler domains where one or two of the steps are shortened. These steps reflect best practices as I understand them.

SOAs are as unique as snowflakes. When all is said and done, no two will be alike. However, as time goes on, common patterns emerge that let us share best practices when creating a SOA. Although we still need to travel farther down the road before we can see the entire picture, some best practices are emerging, and that was the purpose of this article; to expose them. Hope you found it helpful! ■

About the Author

David S. Linthicum is an internationally known application integration and Service Oriented Architecture expert. In his career Dave has assisted in the formation of many of the ideas behind modern distributed computing including Enterprise Application Integration, B2B Application Integration, and Service Oriented Architecture, approaches and technologies in wide use today. Currently, he is CEO of the Linthicum Group, LLC, (www.linthicumgroup.com) a consulting organization dedicated to excellence in Service Oriented Architecture planning, implementation, and strategy.

david@linthicumgroup.com

2007 VIRTUALIZATION CONFERENCE + EXPO

www.virtualizationconference.com

Register Now!
EARLY-BIRD SAVINGS!
$200

ROOSEVELT HOTEL
NEWYORK CITY
June 25-27, 2007

Virtualization: Solutions for the Enterprise.

Delivering The #1 i-Technology Educational and Networking Event of the Year!

As today's CIOs and IT managers rise to the challenge of using their enterprise computing infrastructure as cost-effectively as possible while remaining responsive in supporting new business initiatives and flexible in adapting to organizational changes, Virtualization has become more and more important.

A fundamental technological innovation that allows skilled IT managers to deploy creative solutions to such business challenges, Virtualization is a methodology whose time has come. The fast-emerging age of Grid Computing is enabling the virtualization of distributed computing, of IT resources such as storage, bandwidth, and CPU cycles.

But Virtualization can also apply to a range of system layers, including hardware-level virtualization, operating system level virtualization, and high-level language virtual machines.

Register Online!
www.VirtualizationConference.com

THE FIRST MAJOR VIRTUALIZATION EVENT IN THE WORLD!

HURRY, FOR EXHIBITOR AND SPONSORSHIP OPPORTUNITIES
CALL 201-802-3020

— BROUGHT TO YOU BY —

ENTERPRISE OpenSource MAGAZINE
SOAWORLD MAGAZINE

SYS-CON EVENTS For more great events visit www.EVENTS.SYS-CON.com

ITIL: The Blueprint for SOA Success

Delivering repeatable scalable IT functions cost-effectively

PHOTO: JUNKAI CHONG

Citadel at Qaitbey: Fort built on the spot of the great lighthouse of Alexandria

WRITTEN BY **HON WONG**

In its day the colossal Pharos lighthouse of Alexandria was one of the Seven Wonders of the World. When it fell in the fourteen century, the building blocks were reused by the Sultan of Egypt to build the Citadel at Qaitbey that still stands today. Waste not, want not.

Our modern concept of Service Oriented Architectures (SOAs) updates the ancient approach to recycling. The SOA technology framework enables the rapid implementation of business applications using reusable services or functions. It includes a methodology for finding and consuming these pre-fabricated services.

Yet if you want your SOA projects to display the same kind of longevity as the lighthouse at Alexandria, you can't simply start stacking together the blocks willy-nilly. Building a successful SOA can require just as much careful planning and process as any architectural project. Process frameworks such as ITIL can serve as a valuable blueprint to guide construction.

SOA & ITIL

While proven tools and technologies exist to facilitate the effective use of existing loosely coupled and interoperable services to implement composite applications, assuring the overall performance of the final product is another matter.

In building the Citadel at Qaitbey from the ruins of the lighthouse at Alexandria, the Sultan's craftsmen had to select blocks of stone with precisely the right dimensions while taking into account how centuries and earthquakes had affected the strength and durability of the stone.

The same holds true for developers of composite applications. The service level of the application is limited by the strength of the building blocks (e.g., the service level achievable by the worst-performing services available on the network). For maximum flexibility, services can even be accessed across the Web, making it virtually impossible for IT to characterize the performance profile of constituent services or control the numerous moving parts that can impact application delivery or performance.

Since the delivery of the service is abstracted from the business logic, data structure, or underlying computing platform, it is very difficult to:

- Quantitatively determine if the service level is tolerable for the end user or, at minimum, even meets the service level agreement (SLA)
- Quickly determine the existence and source of service delivery problems

These challenges are validated by a survey of 333 U.S. IT decision makers conducted by the global IT consultancy Ovum. The

survey found that 27% of large enterprises and 17% of medium-sized companies have SOA deployed in some areas of their IT infrastructure.

However, satisfaction with the results is patchy at best, with almost one in five of the people surveyed indicating that adopting SOAs had created unexpected complexity. The problem is that traditional IT management processes and tools are not always up to the task of monitoring and managing SOA applications and that SOA deployments require as much support and investment in infrastructure management as they do in development and testing tools.

Ovum research director Gary Barnett even went so far as to claim that most SOA initiatives launched over the next three years will fail. The Ovum survey found a high correlation between a business's level of satisfaction with SOA and its commitment to managing IT as a set of services in accordance with the best practice approaches advocated by the IT Infrastructure Library (ITIL).

ITIL is a set of IT best practices initially published by the British Central Computer and Telecom Agency in 1987. The central theme of this library of books is that IT must be managed from a "service" perspective. (A word of caution. The IT "service" discussed in the seven ITIL books relates to the work IT professionals do while the "services" in SOA describe the computing functions performed by the application module.)

To deliver complex composite applications based on SOA effectively, IT must be process-focused as opposed to technology-focused. To be process-focused, IT should deliver repeatable, scalable, end-to-end functions like service delivery, service support, security management, and application management. IT should not be organized in technology or platform silos like desktop management, server administration, database management, network administration, and Windows development. Taking this holistic view lets IT address the needs of the consumers of IT services better and, more importantly, allow such services to be delivered more cost-effectively. Cost effectiveness is achieved by focusing on business needs or issues faced by the users of the applications. IT spending priorities should not center on technology-oriented silos that form impenetrable towers of Babel. SOA's goal is to effectively deliver business agility that extends beyond the efficient implementation of business applications to the effective deployment and production management of these applications. ITIL, on the other hand, focuses on delivering repeatable and scalable IT functions cost-effectively. The Ovum survey is clear in that, for SOA initiatives to be successful, infrastructure management requires as much investment as development and testing . To implement the ITIL concept, the process of infrastructure management should delivered in a holistic, end-to-end approach instead of the traditional silo approach. It's crucial that sufficient funding is allocated to implementing effective end-to-end infrastructure management of composite applications to assure the success of SOA deployment. With so much riding on the success of SOA and ITIL initiatives, funding for this crucial capability should be part of the corporation's SOA and ITIL initiatives and shouldn't be considered a discretionary spending item.

A common thread between SOA and ITIL is the assurance that business functions are delivered to the user at the appropriate service level. While SOA-based applications, especially those delivered via the Web, is complex by nature, ITIL guidelines offer a process-oriented way to ***monitor***, ***report***, and ***respond*** to service-level issues. A clearly defined process as stipulated by ITIL is needed to deal with SOA's complexity so that its benefits aren't neutralized by user dissatisfaction, lost business, and ineffective IT. Let's drill down into the process of assuring a successful SOA implementation:

- **Monitor:** There has to be a quantitative way to determine whether the SOA application is functioning correctly. In other words, "Is the right information delivered to the right user in the right time?" There are numerous QA techniques to assure that the right information is delivered. Furthermore, most organizations by now have the necessary security to assure that the right person is getting the information. Ensuring that the information is delivered at the right time is another matter especially for Web-based applications. Having the tools to monitor the application performance experienced by real users unobtrusively is an absolutely necessity because it is the (i) only way to accurately measure the user's real experience of composite applications for service-level restoration and reporting, and (ii) the key driver for making process or application improvements. For Web-based SOA applications, the starting point for such monitoring should be the end user's browser where the composite application truly "comes together" and transverses the entire database infrastructure . It's only at the browser that IT can identify whether

Ovum says that... Most SOA initiatives launched over the next three years will fail

an *incident* (as defined in ITIL's Service Support Module) has occurred that impacts user satisfaction. Other legacy techniques like network sniffing aren't sufficient for SOA application performance monitoring. Even though a sniffer can monitor end-to-end a packet's round-trip time for a particular network segment, it can't monitor all the critical infrastructure and application components of a composite application. The only viable approach is to monitor from the perspective of the real user at the browser for Web-based SOA applications.

- **Report:** There has to be a centralized, correlated view of the performance of all the infrastructure and application components involved in the delivery of the SOA-based applications. Since composite applications are (i) made up of services that are "black boxes" whose performance can't be controlled or tuned by the developers, (ii) run on infrastructure components that aren't entirely under the control of IT ops, and (iii) have different parts of a transaction served by different data centers or servers, it's important that the performance of each transaction is reported and correlated across all infrastructure tiers and application components. The legacy approach of monitoring and reporting on the performance of each technology silo — like network, server, database, etc. as a standalone unit — isn't sufficient to provide a comprehensive view of the performance of the SOA application. ITIL stresses the need for a common vocabulary across all IT processes. Such a centralized view of transactional performance based on real user experience of "consuming" real transactions offers a common view of application performance across different technology silos or IT functions. It's the only viable approach to determine the service level delivered by a SOA application.
- **Respond:** Assuming that IT has access to the centralized, correlated view of transactional performance, it's also critical that the performance data is presented in an actionable form so that ad hoc or trial-and-error approaches aren't needed to identify and respond to performance *problems*. In ITIL's Service Support Module , a problem is the cause of an *incident* that the browser-based monitoring capability discussed above identifies. Without actionable information, IT incident response teams will have to debate the likely cause of the incident using apples-to-oranges silo-based performance data and attempt to reproduce the problem instead of dealing with the problem and restoring the business function in the shortest amount of time. By analyzing the centralized, correlated transaction performance information, IT can also identify leading indicators of incidents so that they can be monitored and proactively reacted to before an incident impacts user satisfaction or business productivity. This type of information can also be used to improve service support efficiency as well as infrastructure and application performance improvements.

SOA Deployment Example

With the growing complexity of Web applications, having a process for continued performance improvement and problem avoidance is critical. Yes, performance problems will occur and are sometimes unavoidable due to situations beyond the control of developers or operations personnel. The key is to "bake in" performance through a culture of cooperation where developers and operations work together so that performance problems can either be resolved proactively or detected and resolved quickly before they impact user satisfaction.

The ITIL Application Management Volume sets forth the lifecycle of software development projects. The ITIL framework extends the traditional four-step software development lifecycle of *requirements-design-build-deploy* to include two additional steps: operate and optimize (Table 1). These last two phases of the lifecycle are particularly important when dealing with complex Web applications. On the one hand, SOA applications are never fully tested because of (i) complexity, (ii) unknown performance issues with constituent services, (iii) lack of control over third-party infrastructure service providers, and (iv) the ever increasing time-to-market pressure that prevents exhaustive testing of all possible usage scenarios. Hence, the significance of the additional two steps: operate and optimize.

Traditional Software Development Lifecycle	ITIL Application Lifecycle
Requirements	Requirements
Design	Design
Build	Build
Deploy	Deploy
	Operate
	Optimize

Table 1: Traditional vs. ITIL application lifecycle

The addition of the two extra phases to the ITIL application lifecycle means there has to be a link between traditionally discrete development and operations functions. In other words, development and operations must have a common view of the business impact, end-user performance, application infrastructure performance, *incidents* and *problems*. Ideally there's a single tool and set of metadata that can bridge these two functional groups and offer (i) developers opportunities for optimization, and (ii) operations personnel the ability to identify and remedy problems efficiently.

Besides the unknown performance issues inherent in the constituent services, SOA application response time can also be impacted by:

- Adding or changing application functionalities;
- Degrading or enhancing infrastructure capacity; and
- Changing or modifying infrastructure components.

For complex SOA applications, developers are constantly called on to deal with production problems, whether to patch a code-level problem that impacts performance, or an infrastructure problem that requires a workaround. Because of their knowledge of the application, they're also called on to serve on triage teams attempting to recreate or diagnose potential or real performance problems. This activity has a tremendous impact on the development schedule and the developer's productivity.

The automated and continuous monitoring and diagnosis of transactional problems from a real user perspective performs three important functions in a production setting:

- Provide a common case management tool and relevant set of data so that developers and operations people can collaborate to quickly pinpoint the cause (*problems* in ITIL terminology) of performance problems (*incidents* in ITIL terminology), whether inside or outside the firewall. If the problem is located inside the firewall, the team can quickly narrow down which tier of the application infrastructure, which server, which method call, which SQL query, or combination thereof caused the performance problem. The benefit of this approach is to eliminate, in most cases, the need to organize cross-discipline triage teams to debate, reproduce, and diagnose problems. The bottom-line effect is improved IT efficiency and quicker problem resolution.
- Afford a continuous feedback loop whereby developers can gain insight into how the application's performance is being impacted by actual usage, features, and infrastructure issues. For example, if a feature turns out to be a major consumer of computational resources such that it impacts the service level of more critical features then development might consider allocating resources to re-engineer this problematic feature. Having this information will give developers the foresight to proactively make modifications to the application and/or code-level implementation trade-offs so that performance issues aren't just considered in hindsight.
- Assist in regression testing based on real user traffic after code-level and/or infrastructure changes.

To facilitate the cooperation between development and operations in matters of SOA application performance, there has to be a common platform for sharing performance information (metadata) that is relevant to both teams and a defined process for acting on the information. In a way, this is akin to replacing the traditional Chinese wall that separates development from operations with a "glass wall." A metaphorical wall, or predefined and enforceable set of business policies, is important so that developers can't arbitrarily modify released code or the underlying database or infrastructure running the code without following proper release, change, and configuration management protocols. Instead of being opaque, this "wall" should be a "glass wall" so there is informational transparency between the two functional groups.

Development and operations use different tools. Data collected or generated by development tools has no meaning to the system management or DBA tools used by operations personnel and vice versa. This tower of Babel situation, if not remedied through the use of a common tool and metadata, makes the implementation of a ITIL process impractical.

ROI Considerations

Performance issues can easily eliminate the expected ROI from SOA initiatives. Slow performance could lead to excess customer or user complaints and impact the productivity of IT staff. Two statistics clearly highlight these hidden costs:

- Forrester Research reported that 72% of user complaints are not detected by existing IT tools or process.
- And Gartner reported that up to 30% of developers' time is spent troubleshooting applications in production.

Both statistics would be even higher for Web-based SOA applications because of increased complexity and lack of control and visibility into the constituent service "black boxes."

For most organizations, the business impact of customers receiving sub-par application performance is immense and not limited to e-commerce merchants. For example, while frustrated e-banking customers can't click to a competitive bank's Web site to pay bills, performance issues will drive up help desk or customer support costs, and eventually drive customers to move their business to a competing bank. Even for internal or non-customer-facing applications, performance issue will lead to a productivity loss among employees or partners.

Beyond the cost to the business, performance challenges also impact the development cost of SOA applications. The cost to an organization could be in the millions of dollars if a good part of developers' time is spent fixing production problems.

All of these potential "downside" costs of SOA should be considered and minimized prior to embarking on a SOA initiative. Figure 1 is a screen capture of an interactive tool that can help IT management or architects quantitatively determine the hidden cost if browser-to-database monitoring and diagnosis tools aren't used to deploy and manage SOA application in production. This free interactive tool is available at http://www.symphoniq.com/products/roi.asp.

Figure 1: ROI analysis

Conclusion

To maximize the benefits of SOA initiatives, it's imperative that the process-oriented guidelines advocated by ITIL be followed. Only with the insight and knowledge offered by understanding the performance profile of each real transaction from the real user's perspective can IT successfully deploy and manage the complexity of SOA applications. ■

About the Author

Hon Wong enjoys solving the problems that result from increasing system complexity and has founded three successful companies that do so: Ecosystems, NetIQ, and Symphoniq.With EcoSystems he set-out to mange client-server systems. Next he founded NetIQ to keep Wintel systems running and developed the technology underlying early versions of Microsoft MOMs. With Symphoniq he addresses the requirements of the Web, including Web services, SOA and Web 2.0 — the most complex and distributed applications ever created. Mr. Wong has spoken at numerous conferences including Interop 2004, is published in DM Review and Communications News, and is a prolific blogger (symphoniq.typepad.com).

The Flesh and Bone of SOA

BPEL & Role Activity Diagrams

WRITTEN BY **MICHAEL HAVEY**

Over the years business processes have become automated to the point that the BPM community now considers the SOA language BPEL, designed for the orchestration of Web Services, as the best platform for building contemporary processes. But many processes retain some level of human activity, and BPEL's support for human interaction is problematic. Most attempts to integrate human workflow with BPEL, such as BPEL4People (as well as proprietary task subsystems offered by the major BPM vendors), try to fit human activities into BPEL's execution model. Human tasks are simply special steps in the larger process.

But people don't work that way, argues Keith Harrison-Broninski in his book *Human Interactions: The Heart and Soul of Business Process Management.* Their work is complex and ad hoc; they interleave their tasks and adapt how they work as business rules change. Harrison-Broninski proposes Role Activity Diagrams (RAD) as the best way to model human workflow, and dismisses BPEL as an impossible fit.

Harrison-Broninski intentionally poses examples (design, sales, marketing, strategy) that are almost exclusively manual, involving little or no system integration, for which the RAD modeling technique is ideal. But what about processes that involve a mixture of human and automated activity, processes whose automation requirements are well served by BPEL? This article makes the bold claim (an anathema for Harrison-Broninski) that BPEL can model these hybrid processes, that it can accommodate complex RAD-style human workflow in a larger orchestration! Our example is the credit card disputes process of fictional ACMEBank. ACME has been sold on BPEL, Web Services, and business rules technology, but needs a RAD-style of human interaction to support its disputes specialists, who, as we'll see below, use the occasion to employ curious *upsell* techniques.

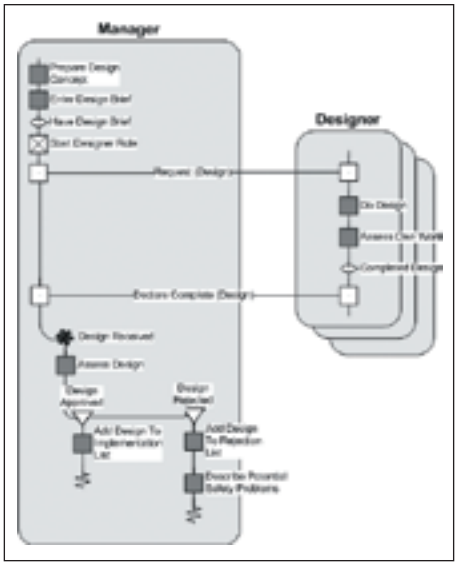


Figure 1: RAD design process
RAD DESIGN PROCESS (FROM [1])

A Radically Different Example

To understand the style of human workflow that RAD offers, consider the design-engineering example shown in Figure 1. In this process, a product manager (for a car maker, let's say) collaborates with designers. The manager documents the basic product design concept (e.g., "I want a car that can transform into a helicopter to escape traffic"), assigns to one or more designers the task of writing a detailed design, and reviews each design, deciding for each whether to accept or reject (e.g., reject the design for the car whose fuel task might explode). The notation is intuitive, even for newcomers to RAD. Manager and Designer, shown as large outer boxes, are *roles*. There is one manager but there can be multiple designers, shown as a stack of outer boxes. A role can instantiate, or allocate work to, another role; the manager role instantiates a designer in the small box labeled Start Designer Role. Interactions between manager and designer are shown as lines that connect small white boxes in each role. Individual tasks are small shaded boxes within a role (Prepare Design Concept in Manager and Do Design in Designer, for example). Ovals represent state; Have Design Brief, for example, means the manager has now reached the state of having completed the design brief. Iteration is shown with a windmill symbol; the logic coming below Design Received is the manager's review of the designs from each designer. Refinement (a form or conditional or parallel branching) is shown as an inverted triangle; there are acceptance and rejection paths to deal with a design that has been accepted or rejected. (The notation is Harrison-Broninski's enhancement of Martyn Ould's classic RAD notation.)

An impatient BPEL developer who saw the diagram but did not understand its nuances might quickly whip up the process shown in Figure 2, a screen capture from the Oracle JDeveloper graphical BPEL editor for Oracle's BPEL Process Manager engine. The process uses a while activity to model iteration (the cycle symbol just below designComplete), switch for refinement (the question mark symbol below the cycle symbol), sequence for activities that in RAD appear to follow each other in succession (e.g., prepareDesignConcept is followed in sequence by enterDesignBrief), and partner links to show interactions between roles (this process has the perspective of the manager role, and DesignerRole is its partner).

If RAD were really that simple, there wouldn't be much point in talking about it.

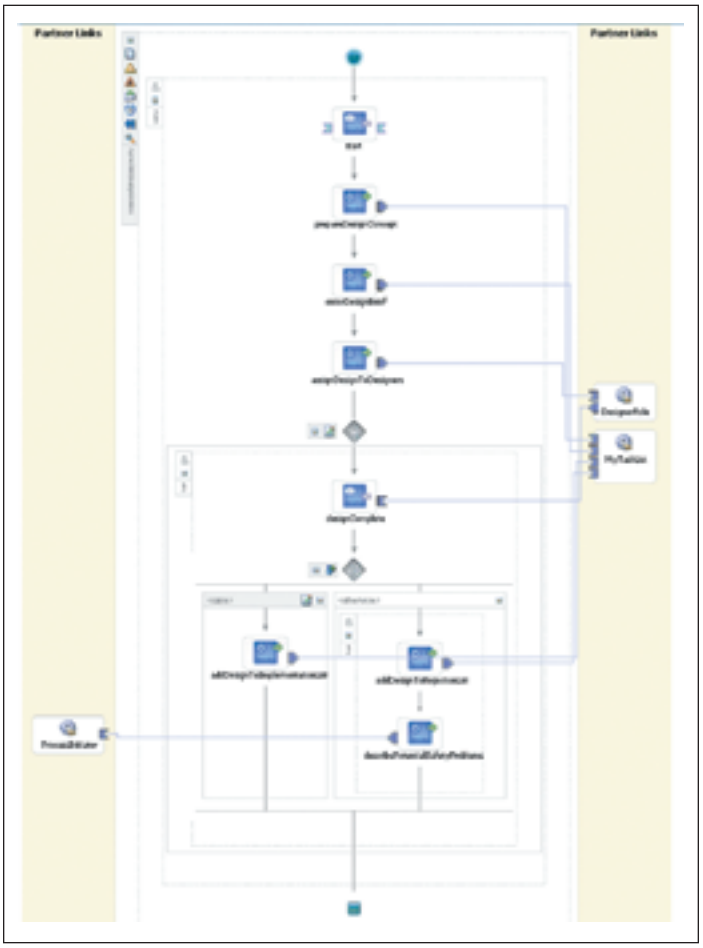


Figure 2: Naive BPEL implementation of the RAD design process

As it turns out, the naive BPEL process from Figure 2 is completely wrong! Recognizing this is the first step in understanding RAD.

Although the RAD diagram appears to have rigid control structures like loops, branches, and sequences, it's fundamentally adaptive. The person performing a role isn't bound to execute tasks in order, to stay on a particular branch, or to perform iterations of a loop in succession. The control flow implied by the diagram is merely a guide, an archetype. The manager might, for example, approve a design, then change his mind and add it to the rejection list, or prepare the design concept and enter the design brief simultaneously. Furthermore, the manager isn't bound to assess the designs one at a time, but may, for example, examine two similar designs at once, or defer the more complex designs and work on the easier ones first. The initial BPEL process has none of this flexibility.

As a contrast to the rigidity of the BPEL example, consider the ineffectual approach of Business Process Modeling Notation (BPMN), the leading standard process modeling notation, which advertises support for the modeling of ad hoc processes and, moreover, defines an overall mapping to BPEL. Figure 3 depicts how we might model in BPMN the manager role of the design process. The process (whose outer box is marked with a tilde, designating

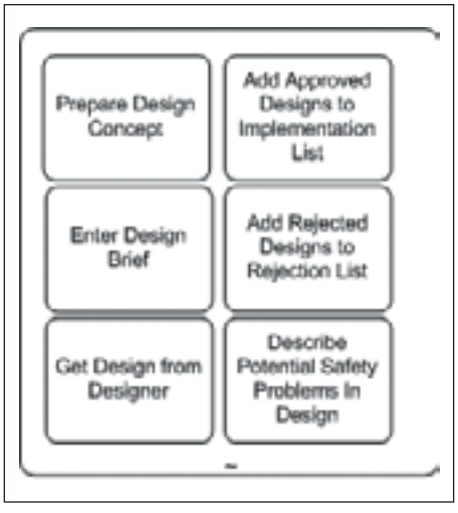


Figure 3: RAD design as BPMN ad hoc

Figure 4: RAD design in BPMN "enhanced" ad hoc

Ad Hoc) has six tasks, but there's no flow connecting them. Execution is up to the manager. The diagram has two shortcomings. First, unlike RAD, it provides no archetypal control flow. There's no explicit notion of iteration or refinement; it doesn't make explicit that there are multiple designs to assess, and that approved designs are treated differently than rejected designs. The second shortcoming is a showstopper: the diagram has no obvious mapping to BPEL; the BPMN specification admits that there can be no general mapping of ad hoc processes. In other words, we can't model the process very effectively, and we can't map it to BPEL anyway.

The BPMN diagram in Error! Reference source not found (see Figure 4) is much more expressive: it conveys archetypal control flow, but, being designated as ad hoc, allows the manager the flexibility to stray from the archetype. Unfortunately, this diagram is not valid BPMN (an ad hoc process may not have explicit sequence flow), and even if it were, the BPMN specification wouldn't provide a BPEL mapping for it. But let's do something audacious and adventurous: let's cheat and allow this enhanced BPMN notation; and let's demonstrate that this sort of notation REALLY CAN be mapped to BPEL!

Radish BPEL

Actually, our goal isn't to build pure conformant RAD in BPEL, but to embrace the spirit of RAD and build a RAD-ish BPEL, as if BPEL were a salad and we decided to toss in a pungent ingredient. There's nothing preposterous about this mixture. Every bank has jumped on the SOA bandwagon; BPEL or some similar BPM technology is in every bank's technology stack. But you'll find many other kinds of components too. Don't assume the architecture is accidental. Complex requirements necessitate curious technology choices. If you see RAD on BPEL, take the architect at face value when she explains: We're an SOA shop, but we need RAD for our most complex human work.

ACMEBank is no exception. Recently ACME built a straightforward disputes process on BPEL similar to the disputes processes of its competitors. But they're not done! ACME has discovered through analysis of past disputes cases that when "high-value" customers call in to report a dispute and are treated favorably, they are much more likely to accept sales offers. The psychological explanation is that disputable charges make customers feel anxious, and they see their bank as an ally in the battle against the merchant. If the customer has deep pockets, why not seize the opportunity to sell products? But the manner in which the disputes specialist should present offers is extremely subtle, and is difficult to model in vanilla BPEL. ACME gets enough disputes from high-value customers to think there's a business case to build complex RAD-style sales decision logic in BPEL.

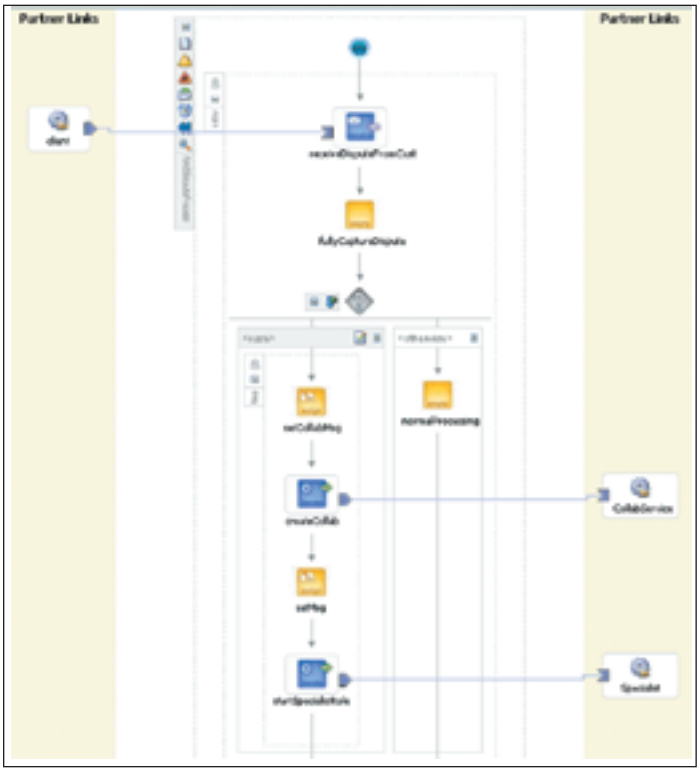


Figure 5: Overall BPEL disputes process

The overall disputes process that ACME intends to build is shown in Figure 5. (This figure shows the process in Oracle's graphical editor. The actual XML BPEL code for this example, as well as the Design example described above, will be available at http://webservices.sys-con.com/read/issue/archives/ under this issue/article .) The process begins when ACME receives a dispute from a customer (receiveDisputeFromCust). ACME then ensures that the dispute is fully captured (which might involve contacting the customer to get supporting information such as receipts) and then splits in two directions, one of which is normal processing, which ACME already supports (the normal chargeback cycle that follows rigid business rules). The other path is the RAD path, in which a sales-savvy disputes specialist makes the right choices to ensure that the customer wins the dispute and buys something along the way.

The RAD path is actually a collaboration involving three roles, In Figure 5, the main process sets this collaboration in motion: the invoke activity createCollab calls a Web Service known as CollabService to initialize the collaboration; the invoke activity startSpecialistRole launches a second BPEL process that manages the primary role in the collaboration, the Dispute Specialist role. The top-level process is now complete, and the RAD collaboration, implemented as three interacting BPEL processes representing the three RAD roles, now commences.

Before turning to those processes, let's examine the nature of the required RAD collaboration by studying its RAD diagram, shown in Figure 6. The dispute specialist begins by assessing the dispute (Assess Dispute) and deciding on an upsell strategy. If he thinks the dispute is invalid and would typically be rejected (the Upsell Reject refinement path), he instead writes it off (because rejections don't sit well with high-value customers) and makes a "level 3" (or "L3") sales offer (e.g., offer the gold card and waive the first-year membership fee). If he thinks the dispute would typically be written off (Upsell Writeoff), he writes it off and makes a slightly better, "level 2" ("L2") offer (e.g., a line of credit at a favorable rate). In the final two cases (Upsell Possible Loss and Upsell Likely Win), the specialist decides to chargeback the item to the merchant to try to recover the charges. If the dispute seems likely to win (i.e., the merchant's bank will probably concede), the dispute specialist assigns a sales specialist (Assign to Sales Specialist) to find a good "level 1," or "L1," offer (e.g., a retirement catch-up loan with the no payments for the first 24 months), makes the offer to the customer, and assigns to a Dispute Lackey (the colloquial term at ACME for a dispute specialist who knows the ins-out-outs of disputes management but doesn't interface with customers) the task of managing the chargeback; if the dispute is lost, the specialist unhappily writes it off. In the case where the dispute specialist thinks the dispute might lose but is worth charging back anyway, he makes an L2 offer and has the lackey follow through with the chargeback. If it loses, the specialist writes it off; if it wins, the dispute specialist asks sales to prepare an L1 offer and extends it to the customer. (Notice that the customer in this case is made two offers: an L1 and an L2. Because a dispute can drag on for weeks or months, ACME prefers to have the specialist make a modest L2 offer immediately, then the more lucrative L1 when the dispute is won.)

The need for RAD arises because of the adaptive nature of this collaboration. In the first place, many activities may be interleaved; in Upsell, Likely Win, for example, the specialist may assign tasks to sales and the lackey in parallel. Furthermore, if the specialist doesn't like the L1 offer prepared by sales, he may decide to make an L2 offer instead. And that's just the tip of the iceberg. In the future, more refinement paths could be added, more activities added, more roles added. The typical BPEL control structure can't easily support this.

Our strategy is to throw a sensible combination of workflow patterns and business rules at the problem. Figure 7 shows the disputes specialist BPEL process in the Oracle editor. The process begins by receiving its start event from the top-level disputes process (startMeUp), and then it calls CollabService to add itself to the collaboration (addRoleToCollab). Next (in the scope configureMyRole shown collapsed in the figure), the process calls the CollabService to configure its tasks, associating each task with its role in the collaboration; these tasks are AssessDispute, WriteOff, MakeL1Offer, MakeL2Offer, MakeL3Offer, AssignLackey, and AssignSales. The process then starts a while loop that runs for as long as the role is in the enabled state. In the first step in the loop, the process calls CollabService to run business rules (runRules) to determine the enablement status of the role and each of its tasks. This leads into a pick, in which the process waits for one of three events to occur. The first is a signal from the end user that one of the enabled tasks has now been completed. If that task is AssignLackey or AssignSales, the process sends a message to the BPEL process representing that role to join the collaboration. The other two events in the pick are callbacks from the Lackey and Sales processes indicating that their work is done. In each case, the process enriches the role's data with information

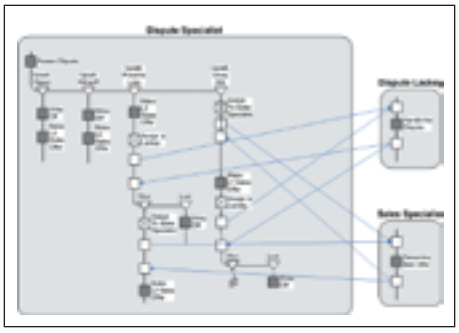


Figure 6: The RAD part of disputes

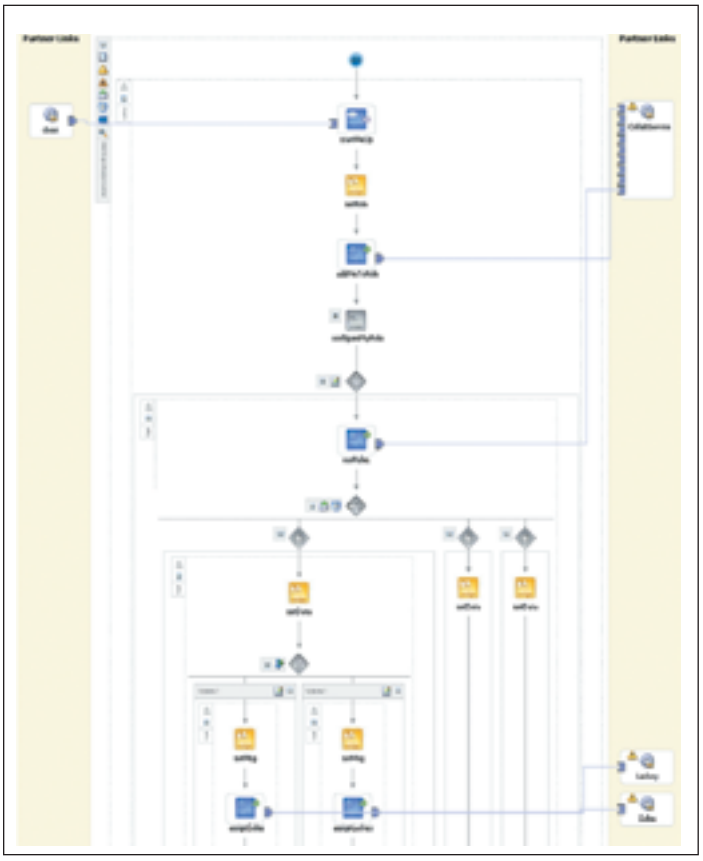


Figure 7: Disputes specialist BPEL process

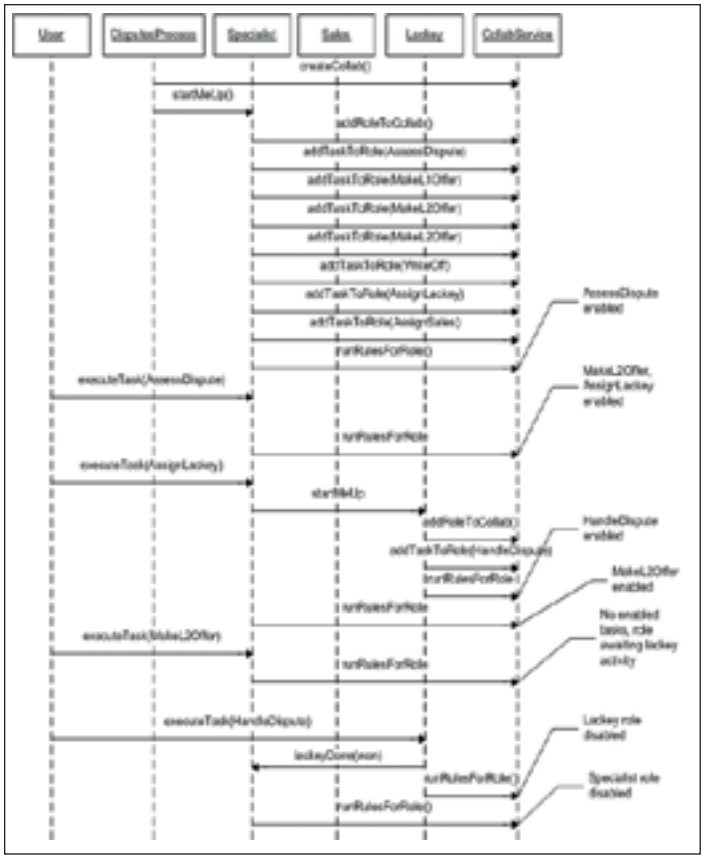


Figure 8: Interactions in UpsellPossibleLoss Path

about the event. That data is input to the business rules in the next iteration of the loop. Next time around, the set of enabled tasks might change, or the role itself might be considered complete, and thus disabled.

The processes for Lackey and Sales are similar, but simpler. Each role has only one task. The process adds its role to the collaboration, adds its task to the role, and waits for that task to be executed, on which it calls back the Dispute Specialist process to indicate that the work has completed.

Figure 8 shows the end-to-end sequence of interactions for the Upsell, PossibleLoss path. At this point, we can make the following observations on this design of RAD in BPEL:

- The primary purpose of the BPEL processes is to manage role interactions. Roles run in separate processes and communicate by Web Services on partner links in a natural SOA manner. Using this mechanism, to start a role is to call the startMeUp entry point of the role's process. To execute a role's task is to call the process's executeTask entry point. Roles interact with each other by calling each other's entry points, as in the case where the Lackey process invokes the callback lackeyDone to interact with the Dispute Specialist process.
- The processes make heavy use of the CollabService. This service, as we'll see shortly, manages the state of the collaboration and oversees the execution of enablement rules. Significantly, the BPEL processes are not designed in the usual naïve way in which tasks are laid out as invoke activities. Rather, the processes tell CollabService what their tasks are and use CollabService to apply rules to determine their execution path.

The while loop in the Dispute Specialist process exhibits two common workflow patterns: Multiple Instances and Milestone.

- **Multiple Instances:** In a sense, the goal of a role is to get through its tasks, and the while loop's job is to iterate until all required tasks have been executed. Each iteration waits for a task completion event, which gets the role one step closer to this goal. In the Disputes example, all tasks are known at design time. In the Design example, an initial set of tasks is established upfront, but additional tasks may be added at runtime; on a given iteration, the role might get one step closer or one step further from completion. In other words, tasks are *instances*, and the challenge is to complete those instances.
- **Milestone:** In contrast to the naïve BPEL approach in which a task is enabled when its predecessor has completed, in RADish BPEL, a task is enabled whenever its enabling, or *milestone*, condition is met. The process does a full sweep through the tasks on each iteration of the loop, enabling or disabling tasks according to rules. (The actual execution of an enabled task occurs outside of the process, aided by a UI tool. That tool sends an event to the BPEL process when the execution is complete.)

(For more information on workflow patterns, see www.workflowpatterns.com.)

Finally, CollabService is a Web Service that uses a set of database tables, shown in Figure 9, to maintain the state of the collaboration. As the figure shows, a collaboration, identified by a collabID, has one or more roles. A role that participates in a collaboration is identified by its name (roleName) and an instance number (in case multiple instances of the same role are required, as with the Designer role in the Design example considered above). A role also has data (consisting of several pieces of structured, unstructured,

and control information meaningful to the role), as well as a rule class and an enablement status. The rule class is a pointer to a class (e.g., Java, C#) or rule-set containing rules that decide, *based on the role's current data*, the enablement status of the role and each of its tasks. The tasks themselves are identified by name and instance (multiple instances of the same task occur when a role has iteration, as in the Manager role of the Design example above).

The collaboration service provides four methods to update this data: three — createCollaboration, addRoleToCollaboration, and addTaskToRole — to add new records to it, one — runRulesForRole — to update, based on business rules, role data and role and task enablement status. The service doesn't provide queries, although they could be added. A RAD UI tool should have read-only access to the tables to perform desired queries.

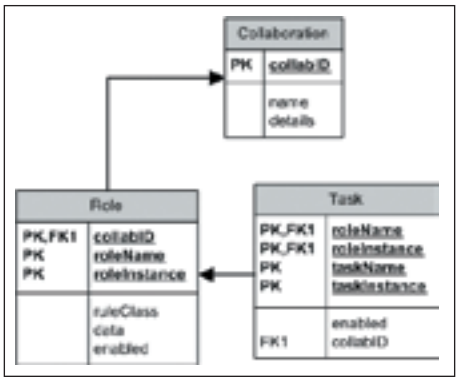


Figure 9: RAD data model

The behavior of the service method runRulesForRole is best understood by example. Consider the rules for enablement in the dispute specialist role specified in Table 1. When runRulesForRole is run, it tests the enablement condition for each of that role's tasks, as well as the role itself. On its first run, runRulesForRole will enable the task AssessDispute, because that task satisfies its condition that it hasn't been executed yet. No other tasks can be enabled initially.

When the specialist completes AssessDispute, he also specifies an assessment decision (i.e., one of the upsell strategies examined above). Assuming that decision is UpsellReject, next time the rules are run the following enablement changes are made:

- AssessDispute is disabled because it no longer satisfies the condition "task not done."
- WriteOff is enabled because it hasn't been completed yet and the assessement is UpsellReject
- MakeL3Offer is enabled because it hasn't been completed yet and the assessment is UpsellReject.

The dispute specialist can work WriteOff and UpsellReject in any order. When these are completed, the role itself is disabled, because its enablement condition of having at least one enabled task or pending work from sales or the lackey fails.

Considering a more complicated scenario (shown in Figure 8), assume the specialist's decision in AssessDispute is UpsellPossibleLoss. The next enabled tasks are AssignLackey and MakeL2Offer. The specialist might then execute both tasks right away, which leaves no enabled tasks for the specialist role, but does trigger the lackey role. The dispute specialist role is still enabled, because it's waiting for information from the lackey. When the lackey completes his work, if the dispute was won the dispute specialist role is disabled, because no tasks are enabled and no activity is pending. On the other hand, if the dispute was lost, the WriteOff task is enabled, and the dispute specialist role waits for its completion.

Task/Role	Enable Condition	On Execution
AssessDispute Task	Task not done	Set task done and set assessment to UpsellReject, UpsellWriteOff, UpsellPossibleLoss, or UpsellLikelyWin
WriteOff Task	Task not done and assessment is UpsellReject or UpsellWriteOff or dispute was lost	Set task done
MakeL1Offer Task	Task not done and sales has specified an offer and dispute specialist hasn't suppressed the offer	Set task done and, optionally, suppress offer
MakeL2Offer Task	Task not done and either assessment is UpsellPossibleLoss or assessment is UpsellLikelyWin but L1 offer is suppressed	Set task done
MakeL3Offer Task	Task not done and assessment is UpsellReject	Set task done
AssignLackey Task	Task not done and assesement is UpsellPossibleLoss or UpsellLikelyWin	Set task done and start Lackey process. When Lackey responds, set dispute status to won or lost.
AssignSales Task	Task not done, L1 offer not suppressed, and either assessment is UpsellLikelyWin or assessment is UpsellPossible and dispute is won	Set task done
Dispute Specialist Role	No tasks enabled, no pending sales or lackey activities	

Table 1: Business rules for disputes specialist

Conclusion

RAD-like human interaction can be modeled in BPEL using business rules and common workflow patterns, and the implementation isn't as onerous as critics claim. The approach is ideal for hybrid processes, whose automation requirements demand an SOA stack built on BPEL, but which infuse ad hoc human work. The people who participate in hybrid processes, like the opportunistic dispute specialists at ACMEBank, are the "flesh and bone" of SOA. Their contribution is significant, but their effort is not quite the "heart and soul" of the BPM envisaged by Harrison-Broninski. ■

References

1. *Keith Harrison-Broninski. Human Interactions: The Heart and Soul of Business Process Management. Meghan-Kiffer. 2005.*
2. *"Business Process Modeling Notation Specification." Object Management Group. http://www.bpmn.org/Documents/BPMN%20V1-0%20May%203%202004.pdf*

About the Author

Michael Havey is a Chordiant consultant with a decade of industry experience, mostly with application integration. Michael's book Essential Business Process Modeling was published by O'Reilly in August 2005.

michael.havey@chordiant.com

SOA for SMB's?

small budget
BIG QUESTIONS
mission-critical services
MAJOR PRESSURE
short timeline
reduced staff

JaxView

A flexible Web Services management solution for small and medium-sized businesses that need the following capabilities:

- Monitoring of Web services
- XML gateway and authentication
- Runtime SOA policy enforcement
- Automated Web service discovery
- Runtime message modification and routing

Download a free trial at www.managedmethods.com

JaxView
versatile, scalable, affordable

Managed Methods

Altova® XMLSpy® 2007 – The industry standard XML development environment.

ALTOVA® xmlspy® 2007

Bring your development plans to light

Sneak a peek at XMLSpy® 2007, and see how vital it is to master XML.

Revealed in XMLSpy 2007:

- Completely redesigned graphical WSDL editor
- New XPath Analyzer window that stores multiple expressions and results
- Advanced find-in-files search and replace capability

Altova® XMLSpy, the industry standard XML development environment, is indispensable for modeling, editing, transforming, and debugging XML-related technologies. Illuminate your strategy with the world's leading XML editor, the original graphical schema designer, a code generator, file converters, debuggers, profilers, support for XSLT, XQuery, WSDL, SOAP, and a wealth of brilliant XML utilities and enlightened usability aides. Become a markup mastermind!

Download XMLSpy® 2007 today:
www.altova.com

Join Altova at Microsoft TechEd, Orlando

ALTOVA®

XMLSpy is also available as part of the value-packed Altova MissionKit™.