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

DECEMBER 2007 / VOLUME: 7 ISSUE 12

A Bright IDEA: The Road Map for Service-Oriented Success

8 BRIAN ERICKSON

4 To Infinity, and Beyond...
SEAN RHODY

6 SOA – A Business Agility Strategy
RICHARD MARK SOLEY

14 What's Next After SOA?
DAVID M. STRAUS

16 Who's Responsible for Quality in a Service-Oriented Architecture?
WAYNE ARIOLA

20 Can Agile Development & Model-Driven Design Solve the Broken Delivery Process?
JACK KENNEDY

26 iTKO LISA 4 SOA Testing
PAUL O'CONNOR

34 SOA as a Corporate Responsibility
DAVID S. LINTHICUM

PLEASE DISPLAY UNTIL FEB. 29, 2008

$6.99US $7.99CAN

12

0 71486 03420 9

IBM, the IBM logo, WebSphere and Take Back Control are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries. ©2007 IBM Corporation. All rights reserved.

_INFRASTRUCTURE LOG

_DAY 84: Feeling really disconnected. We're not getting the most out of our existing assets. Service and application integration is a nightmare. Our connections are restrictive. We've got to stop working on these islands.

_Please rescue me from this lack of connectivity.

_DAY 87: I've taken back control with IBM WebSphere solutions. Now we can service-enable and connect our existing assets for mission-critical goals. We can reuse existing applications and save money by eliminating redundant systems. Now we're ready for any SOA integration project.

_Plus, no more jellyfish stings.

Download the enterprise service bus white paper at:
IBM.COM/**TAKEBACKCONTROL**/CONNECT

To Infinity, and Beyond...

WRITTEN BY **SEAN RHODY**

At the end of the year, it's always interesting to take a moment and try to imagine what's next. In this case, I'm not thinking about whether the Patriots will win the Super Bowl, I'm wondering what comes after service-oriented architecture.

Although SOA has been dramatically hyped as the solution to cure all ills, it has still taken a good deal of time for organizations to adopt SOA in even a limited fashion. It's clear from its adoption by vendors as an integration strategy and by many IT organizations as a design philosophy that SOA will eventually be the mainstream set of architecture and governance principles for IT.

That doesn't mean that the ultimate goals of SOA have been realized. In an ideal world the concept of service would extend completely into the business – creating the service-oriented enterprise. In such a business, composite applications are created that allow the atomic services of the business – whatever they may be – to create powerful processes that are aligned with business goals and priorities.

In many cases to date, IT has selected SOA as an architecture, but we've yet to see line of business acceptance and overall corporate governance changes that mirror the IT work. It seems it's much easier to change code than it is to change organizational structure or political power bases. So the ultimate promise of SOA remains still in the future.

Nevertheless, my thoughts can't help but turn to what's next after SOA has been accomplished. I was having a conversation with a colleague, and our discussion spurred me to consider that the next step might not be new, but rather a revisiting of another previously explored concept – one-to-one marketing.

I'm not talking now about using SOA for marketing purposes, so don't worry. What I am talking about, and where I see the future possibly going is individualized, localized services. Services aimed at you, and only you; services that know where you are and are aware of your preferences, and perhaps a web of other preferences that interact with yours.

We got on this discussion while talking about portable GPS devices. As fellow travelers we both are in favor of those, but what got me thinking was that ultimately we may have some personal device – be it a cell phone, a GPS, or even some evolved form of a tablet computer – that will serve as our "consumer" for services. Not a computer as we currently consider them, but a device that does what we need and want, not what we don't. I'd like to have just one device, instead of the myriad that I carry now (personal cell, work cell, iPod, PSP, TomTom). But combining the devices really isn't the point – I don't want a game station, cell phone, music-playing navigation system. What I want is a device that can offer any of those, and more, as services to which I subscribe. I want to pick my cell carrier. I want to decide if I want movies on my device and which kind. I want the device to be my proxy in the great SOA continuum on the Internet.

Granted this is somewhat aimed at consumer services, but in the end almost all companies provide services, or would be connected to companies that do via the supply chain. The point is all of these "concepts" are services. I want navigation, but sometimes I also want to find a good Italian restaurant along the route from my client to the hotel so I can get a good meal after a day's work. Other times I may not want to take the fast route; I may want to take the route with the most covered bridges so I can see some scenic countryside. Conceivably I might want the game I'm playing to reflect that countryside (this assumes I'm not driving of course). I might want the soundtrack to reflect my musical preferences as well, even if it seems at odds with what I'm playing. In short, I want the ultimate consumer experience, courtesy of SOA. Some of this already exists; for the rest, we'll have to wait and see. ■

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*

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

INTERNATIONAL ADVISORY BOARD
Andrew Astor, David Chappell, Graham Glass, Tyson Hartman, Paul Lipton, Anne Thomas Manes, Norbert Mikula, George Paolini, James Phillips, Simon Phipps, Mark Potts, Martin Wolf

TECHNICAL ADVISORY BOARD
JP Morgenthal, Andy Roberts, Michael A. Sick, Simeon Simeonov

EDITORIAL

Editor-in-Chief
Sean Rhody sean@sys-con.com

XML Editor
Hitesh Seth

Industry Editor
Norbert Mikula norbert@sys-con.com

Product Review Editor
Brian Barbash bbarbash@sys-con.com

.NET Editor
Dave Rader davidr@fusiontech.com

Security Editor
Michael Mosher wsjsecurity@sys-con.com

Research Editor
Bahadir Karuv, Ph.D Bahadir@sys-con.com

Technical Editors
Andrew Astor andy@enterprisedb.com
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
Mike Sick msick@sys-con.com
Michael Wacey mwacey@csc.com

International Technical Editor
Ajit Sagar ajitsagar@sys-con.com

Executive Editor
Nancy Valentine nancy@sys-con.com

Associate Online Editor
Lindsay Hock lindsay@sys-con.com

PRODUCTION

LEAD DESIGNER
Abraham Addo abraham@sys-con.com

ASSOCIATE ART DIRECTORS
Louis F. Cuffari louis@sys-con.com
Tami Beatty tami @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

©COPYRIGHT
Copyright © 2007 by SYS-CON Publications, Inc. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy or any information storage and retrieval system without written permission. For promotional reprints, contact reprint coordinator. SYS-CON Publications, Inc., reserves the right to revise, republish, and authorize its readers to use the articles submitted for publication. All brand and product names used on these pages are trade names, service marks, or trademarks of their respective companies. SYS-CON Publications, Inc., is not affiliated with the companies or products covered in Web Services Journal.

Mindreef®

SOAPscope server™

Whether you're an Architect, Developer or Tester, rely on the Web services and SOA tools that won't leave you stranded.

Mindreef provides an integrated set of tools at the XML software layer for SOA and Web services policy governance, testing, diagnostics and support. Available in server or workstation licensing options, Mindreef SOAPscope® products help professional developers, testers and architects working alone or on teams to understand and address SOA and Web service problems quickly and cost-effectively.

Visit our website and learn why more than 3,000 customers worldwide, including 40 of the Fortune 100, use Mindreef products to successfully build, test and maintain high-quality Web services and SOAs.

SEE IT!

Attend an on-demand webinar on our website to see how Mindreef can improve your SOA quality

TRY IT!

Request an evaluation copy of Mindreef SOAPscope Server or SOAPscope Workstation™

BELIEVE IT!

Start saving time and money, and improving the quality of your Web services and SOA projects

The SOA Quality Company™

www.mindreef.com/not-stranded.html
(603) 465-2204 ext. 581

SOA – A Business Agility Strategy

WRITTEN BY **RICHARD MARK SOLEY**

Service Oriented Architecture (SOA) is a business agility strategy. Repeat after me: Business. Agility. Strategy.

There are so many technology-focused definitions of SOA floating around that it's difficult to remember that simple fact.

SOA is focused on recognizing, precisely defining, storing, retrieving/reusing and optimizing business processes, both automated ones and human ones. If we as an industry allow SOA to mean Yet Another Integration Technology, we doom ourselves to yet another level of integration middleware, and yet more maintenance hassles caused by the brand new legacy systems we are installing today.

SOA is an enterprise strategy certainly. Nevertheless, we recognize that the early champions SOA – those who recognize the enterprise value proposition – are in the IT department. As of such, SOA is wrongly shoved into the "technology box" in the press and in organizations. This mislabeling impedes organizational acceptance of SOA and, more importantly, constrains IT organizations from adopting a strategy that lets them rapidly innovate across the enterprise, positively impact corporate strategy, and drive revenue and profit.

To spur enterprise-wide understanding and acceptance of SOA, the SOA Consortium works with IT leaders and SOA champions to help them communicate to business leaders, business analysts, and project managers the value of designing the business around processes that can be precisely defined, carefully captured, found again, and optimized. Not only must supporting IT infrastructure be organized on SOA architectural principles, the business needs to get value out of better-organized business activities and processes, both across the enterprise and throughout the supply- or value-chain. To accomplish this goal requires recognition, change, and innovation in the IT organization, but also throughout the organization. That requires education.

The SOA Consortium is committed to helping the Global 1000, major government agencies, and mid-market businesses successfully adopt Service Oriented Architecture (SOA) by 2010. Our membership is made up almost entirely of end users who have been on the front lines of this education campaign in their own organizations, and a few visionary sponsors including BEA, Cisco, IBM, SAP, and Sparx Systems. Organizations including government agencies, large retailers, transportation leaders, travel companies, financial services providers, and insurance companies have come together to share their experiences and help get the word out that SOA is a business agility story.

– CONTINUED ON PAGE 12

About the Author

As chairman and CEO of OMG and executive director of the SOA Consortium, Dr. Richard Soley is responsible for the vision and direction of the world's largest consortium of its type. He joined the nascent OMG as technical director in 1989, leading the development of OMG's world-leading standardization process and the original CORBA specification. In 1996, he led the effort to move into vertical market standards and modeling, leading first to the Unified Modeling Language and later the Model-Driven Architecture. Previously, Dr. Soley was a cofounder and former Chairman/CEO of A. I. Architects, Inc., maker of the 386 HummingBoard and other PC and workstation hardware and software. He holds a BS, MS, and PhD in computer science and engineering from MIT.

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

CORPORATE

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

Senior VP, Editorial & Events
Jeremy Geelan jeremy@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

Advertising Events Associate
Alison Fitzgibbons alison@sys-con.com

SYS-CON EVENTS

Event Manager
Lauren Orsi lauren@sys-con.com

Event Planner
Sharmonique Shade sharmonique@sys-con.com

CUSTOMER RELATIONS

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

SYS-CON.COM

VP information systems
Bruno Y. Decaudin bruno@sys-con.com

Consultant 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

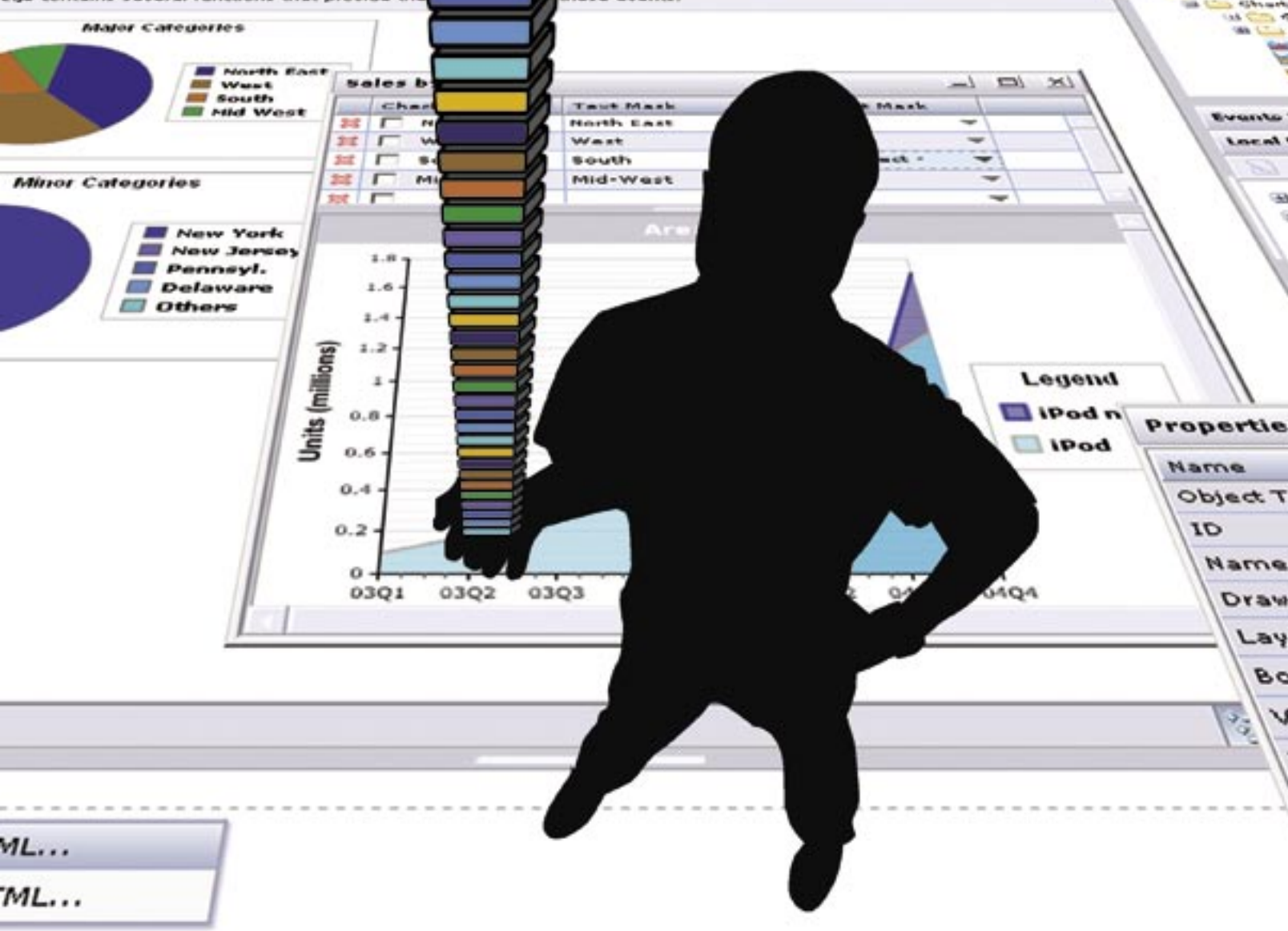
SUBSCRIPTIONS

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

SYS-CON Publications, Inc., reserves the right to revise, republish and authorize its readers to use the articles submitted for publication.



Service Oriented Ajax

For Your Service Oriented Architecture

Why are professional developers choosing TIBCO General Interface?

With more than 100 ready-made Ajax components and easy-to-use visual layout, data management, and communication tools, TIBCO General Interface™ enables you to create Ajax applications that look and feel like desktop GUIs with astounding speed.

TIBCO®
The Power of Now®

OpenAjax alliance member

Download and explore TIBCO General Interface, now available through an open source license at http://developer.tibco.com/.

©2007, TIBCO Software Inc. All rights reserved. 1704.0

InfoWorld 2007 BOSSIE AWARDS
Best of Open Source Software

A Bright IDEA: The Road Map for Service-Oriented Success

Implementing SOA without a roadmap and a guide almost certainly means disaster.

WRITTEN BY **BRIAN ERICKSON**

Information always has and always will be the lifeblood of a company. But globalization has profoundly changed how a business collects and exchanges data, and with whom an organization shares that information. In today's sprawling digital enterprise, operations are no longer contained to a single floor, building, or even one campus, country, or continent. Additionally, the outsourcing and offshoring of many basic operations mean the accounts payable office that used to be down the hall may now be in Poland. This complicates transaction processing as more third-parties need to access corporate data stores to support operational functions.

While the pressure to get to market faster with more competitive products is certainly nothing new, globalization and the technology driving it are accelerating the rate of change. The pressure is constantly increasing with greater demand for faster overall product cycle times coupled with even shorter corporate lifecycles.

Now more than ever, companies, whether competing in a new world market or with Joe down the street, are turning to their IT organizations to plot a path to apply innovation to outpace their rivals. The technology organization is tasked with building a fluid, extensible, and productive environment that provides a conduit for faster development, greater collaboration, and more efficient operations. In essence, the pressure for IT to create and sustain technology that drives corporate progress and produces quantifiable financial benefits to the business has increased dramatically.

SOA: The Fast Track to Success?

Companies are increasingly looking to Service Oriented Architecture (SOA) as a way to establish more agile, responsive, efficient, and secure infrastructures while leveraging their existing invest-

ments in technology and process. In an April press release, Gartner, the research house, estimated that 50% of all new mission-critical applications and initiatives in the coming year will be based on a Services Oriented Architecture and that number will jump to 80% by 2010.

What is SOA and what makes the architecture so appealing to so many businesses right now? At the highest level, SOA as described by author Thomas Erl is an approach to software development that aims to help organizations streamline and improve their business processes to create a greater level of operational efficiency and effectiveness. SOA builds autonomous business services that can work together or separately. Each component in a Service Oriented Architecture is responsible for a discrete business function. These components are, by their nature, reusable and so can be combined to build new composite applications. This flexibility facilitates more efficient software design. As a result, developers can create and deploy applications quickly, which in turn can speed time-to-market for new customer-facing services and products.

Aside from simplifying software development, SOA promises a number of other important benefits. The architecture promotes greater process standardization by allowing companies to use IT assets in a common format to solve business problems. SOA also facilitates better integration of applications and data across the entire organization. In addition, SOA allows organizations to leverage existing assets.

Using Service Oriented Architecture, businesses can make process improvements that create greater organizational efficiency. This makes it possible for a company to transfer spending from tactical operations to more strategic initiatives focused on innovation. The end result is much closer alignment between IT and the business.

SOA is getting attention these days for good reason. Organizations are achieving some very real results courtesy of quantifiable process improvements achieved through SOA initiatives. A recent Aberdeen Research study found that 68% of organizations that are realigning business processes around a SOA model are seeing an overall drop in application lifecycle costs.

As appealing as SOA sounds on paper, in reality attaining real, long-term benefits involves a complicated multi-stage journey rather than a quick jaunt with an immediate payoff. So with expectations soaring for the potential dividends SOA can provide, a backlash against the architecture is almost inevitable.

The reality is SOA demands a paradigm shift. Process improvements built around SOA can't be deployed in the same way IT projects have been in the past. Many aspects of the organization will be affected by SOA, and so it's critical that both business and IT personnel be on board with what an implementation entails. This means understanding all the challenges associated with SOA adoption, including the potentially significant start-up costs and the significant long-term IT architecture investments SOA requires. In truth, fully realizing benefits from SOA can take years.

Yet, in spite of all of the possible hurdles, SOA has enough momentum and promise to appeal to organizations willing to endure the challenges to reap great rewards. Those that don't are likely to suffer for their inability to react as quickly to change as competitors who adopt SOA.

One thing is certain: SOA requires enough of a resource commitment that implementing it without a roadmap almost certainly leads toward disaster. But even the best road map is useless without a clear understanding of where the company's starting point is and what its eventual end point should be. With that information, a roadmap can provide important markers to ensure a business takes the most expedient path to reach its objectives.

A Bright IDEA: Going the Distance

Having a well-traveled guide to lead the organization through this journey is essential. Hitachi Consulting has worked with many corporations to implement SOA initiatives, and in doing so, has created a tested roadmap for SOA implementation efforts that provides a realistic and effective take on SOA.

It starts with a maturity model that parallels every phase of SOA adoption. While no two organizations — or their SOA implementations — are the same, all Service Oriented Architecture initiatives go through the same stages and experience comparable challenges. The phases of adoption provide organizations with needed mile markers to measure SOA implementation progress and prepare for critical transitions. There can be significant distances to travel before the jump from one phase of maturity to the next.

The Hitachi Consulting SOA IDEA Maturity Model consists of four levels: Initiation, Deployment, Enablement, and Agility.

Initiation

In the Initiation phase, an organization begins to explore building applications using SOA concepts. During this stage, most businesses take on well-defined, clearly scoped projects and typically realize positive quantifiable results. The key objectives are to determine if continued investment is worthwhile and focus on the best approach for moving forward.

Deployment – Getting Up to Speed

Once a business secures a win with SOA in the Initiation phase, the organization is emboldened to take on more complex and strategic projects. At this point, businesses can move to the next level known as Deployment.

The Deployment stage requires a greater investment as businesses begin to move from narrowly defined point solutions to broader enterprise-level implementations. This requires implementation and adoption of different tools, processes, and policies.

Deployment-phase projects take an organization to the next level with respect to creating more complex and robust services by employing technology such as enterprise service bus and business process execution language. Businesses at this SOA level are also embracing a culture of reuse. At this point, an organization has typically made more than one application available as a set of services, and users are starting to interact with composite applications assembled from those services. Usually, this is where companies begin to realize the true benefits of SOA.

This desire for greater flexibility and effectiveness motivated one of the best companies in the United States to undertake a major SOA initiative. In many ways, the difficulties this defense and technology contractor giant faced with respect to the efficient integration of new acquisitions and their associated systems are typical of the issues that drive businesses to institute SOA initiatives on a broader level. Specifically, the rapidly growing contractor was challenged with human resource benefits document version control. The company wanted to construct a single document repository and a set of common services to synchronize documents.

The result of the SOA effort was a single reliable source for hu-

man resource benefits information. Having one standard data source made it possible for the organization to speed time-to-market for new HR offerings. This initiative also produced new standards across business units, which led to lower litigation costs.

Enablement – Driving Progress

Enablement is the next more ambitious phase. The organizations that have successfully reached this SOA stage can now blend services to ratchet up end-to-end business process improvements and create a far more streamlined and efficient organization. Companies that have achieved Enablement have a solid SOA infrastructure in place, and focus shifts from implementing technology to implementing business and organizational change. SOA provides the foundation for these initiatives, and the SOA development ethic serves as a standard best practice for any design work.

A major Japanese automobile manufacturer's story provides a good example. This company has successfully risen through the ranks to dominate sales worldwide; yet its expansion didn't come without some serious challenges. Like other businesses that have gone through major expansions, the company struggled with growing pains.

One challenge was understanding who its customers were and providing a consistent view of those customers throughout the organization. At a tactical level, this company had 30 different data stores of customer information. Audits found the data was less than 70% accurate. As a result, the company was pouring money into marketing and advertising initiatives based on data that was only 30% accurate.

An application was created that effectively provided a 360-degree view of customer information. The application provided accurate data that helped the company speed time-to-market and create more effective marketing initiatives. As a result, it lowered the costs associated with customer data services. After implementing the project, the company stopped buying customer data from external sources because the quality of its internal data was now better than what they could buy, saving the company millions of dollars. These solutions were delivered by creating an SOA that maintained one view of the customer.

Agility – A Force in Motion

Companies successfully progressing to this point can aim to reach a sort of SOA nirvana – Agility, the highest level in the Service Oriented Architecture maturity model. To reach this phase means an organization has adopted an enterprise-wide service orientation and is likely extending this philosophy beyond its company boundaries. This translates into having services and processes that enable all internal collaboration and work with both their partners and customers' processes.

As appealing as this sounds, most SOA initiatives are still relatively immature, so no company can really claim to have reached this implementation level. However, having these phases as reference points helps businesses gauge where they are to determine if they're on track or simply spinning their wheels by remaining too tactically focused.

So what would a business that reaches the Agility phase look like? For starters, it would rely on reusable services and composite applications. It would also have SOA throughout the enterprise to provide real-time business services, runtime governance, and process coordination. A company that reached the Agility phase would also have virtualized and federated resource management.

But to reach the Agility SOA maturity phase, an organization has to transform itself into a truly flexible business, rapidly responding and reacting to changes in demand and market conditions through technology. And the company would have to be able to extend the developed solutions beyond its four walls. With respect to technology, an organization would have to create effective real-time processing loops.

While no business has yet to progress through all the phases of SOA adoption to reach this level of maturity, once a business does, it will be in a position to compete more effectively against market rivals. At this point, a company might need a consulting partner to provide cross-enterprise SOA solutions to help it create a truly expansive enterprise.

Components of Change

While the Hitachi Consulting SOA maturity model gives business and IT executives a glimpse of the road before them, the view is high level. To begin the practical steps a business and IT organization need to reach a successful end point, both need more detail to guide their efforts. Stakeholders have to understand the roles particular components play at each stage. In effect, companies need to look at SOA adoption as a multi-dimensional exercise.

There are seven components common to every phase of the maturity model. These components, which include strategy, process, governance, data, services, architecture, and people, span the technical, business, and human aspects of SOA adoption. What varies from stage to stage is each component's role and relative importance to the project's outcome based on SOA implementation maturity.

Because SOA adoption impacts so many different aspects of a company, timing is crucial when it comes to focusing on the right issues at the right times to create smooth transitions and guarantee the organization progresses through all phases of the maturity model. Naturally, some components take on more importance in certain phases than in others. For example, during the SOA initiation phase, an organization is probably not going to focus much on governance, but concentrate instead more on data or applications. However, organizations won't be successful with SOA initiatives unless key aspects of each component are addressed at every level of the maturity model.

It's worth noting that as complex as each component can be, perhaps the most volatile — and certainly the most vital — is the human element.

A Map of the New World

Like any architecture promising rich rewards, there's considerable hype around SOA. However, recent research by Aberdeen

"With expectations soaring for the potential dividends SOA can provide, a backlash against the architecture is almost inevitable"

While most IT departments are busy putting out fires, you can start being more liquid.

A Service-Oriented Architecture is the best way to stay fluid in the constantly evolving world of business. BEA's particular approach to SOA allows you to unlock frozen assets and align business with IT.

What makes our SOA solutions so liquid? Our software embraces heterogeneity, working equally well with diverse technologies and multiple vendors. We offer seamless innovation so that you quickly realize tangible benefits, not just suffer empty promises. And our pragmatic, step-by-step approach allows your company to experience a faster time-to-market while mitigating costs and reducing risk.

But don't take our word for it: IDG Research Services Group recently named BEA the number one choice for SOA solutions and services.*

To learn more visit bea.com.

bea®
Think liquid.™

© 2007 BEA Systems, Inc. All Rights Reserved. BEA is a registered trademark and Think liquid is a trademark of BEA Systems, Inc.
*Source: Conducted by IDG Research Services Group and co-sponsored by BEA Systems, February 2006

demonstrates that those willing to undertake the journey and progress through the different stages will realize significant benefits. Aberdeen surveyed more than 400 companies and categorized their SOA efforts into two groups — companies with "SOA Lite" (deploying Web Services based applications) and companies building full SOA middleware infrastructures. Aberdeen concluded: "Organizations that are focusing on SOA infrastructure are outperforming those that are deploying only Web Services. They are realizing lower application lifecycle costs, better throughput for projects, and higher levels of user satisfaction."

SOA adoption is a complex multi-stage process requiring a good deal of thought and preparation at each phase. No organization should enter on the path without having a clear roadmap that provides critical markers that business and IT executives can use to chart the progress of their SOA implementation. This roadmap will give companies guidance on the best routes they need to take to reach their destination and help avoid the pitfalls and traps that lay ahead.

But before launching a new journey, businesses first need to understand where they are and where they want to go. Only then can they map a clear path that will take them through the process of creating a sturdy foundation for progress and ultimately corporate excellence that will help them stand out in an increasingly competitive world market.

About Hitachi Consulting

As Hitachi, Ltd.'s global consulting company with operations in the United States, Europe, and Asia, Hitachi Consulting is a recognized leader in delivering business and IT strategies and solutions to Global 2000 companies across many industries. With a balanced view of strategy, people, process, and technology, we work with companies to understand their unique business needs, and to develop and implement practical business strategies and technology solutions. From business strategy development through application deployment, our consultants are committed to helping clients quickly realize measurable business value and achieve sustainable ROI. ■

About the Author

Brian Erickson is the managing vice-president of Hitachi Consulting's Strategic Technology Solutions practice. With more than 22 years of management and technology consulting experience in a variety of industries, Brian has worked with some of the largest and most influential corporations on business strategy, business transformation and strategic technology planning. He has significant experience working with clients to develop and execute large, complex, strategic information systems plans and has played a key role in creating Hitachi Consulting's system integration practice, which has helped lay the groundwork for its SOA initiatives. Brian regularly speaks at conferences and is a regular contributor of articles to industry trade media. Prior to joining Hitachi Consulting, he worked for Booz Allen & Hamilton, IBM, and Accenture. He earned a bachelor's degree in information systems and a masters of business administration from Baylor University

SOA – A Business Agility Strategy

– CONTINUED FROM PAGE 6

To help educate the market about SOA, the SOA Consortium has chartered several working groups to target specific issues. For example, the EA2010 working group in the SOA Consortium's community of practice has been actively discussing and defining the next-generation role of enterprise architecture. Specifically, what enterprise architecture looks like – its organization, practices, and people – in a business-driven service-oriented world. They have developed an on-demand webinar that covers a wide range of enterprise architecture concerns, including catalyzing business change, picking up business smarts, shifting focus to business architecture, managing enterprise architecture, participating in strategy and delivery, and winning enterprise constituents.

The Promoting Business-Driven SOA – "Executive Suite SOA" working group has published "lessons learned" from those who have stepped into the great unknown already, completed SOA projects, and lived to tell about it. Organizational Case studies from a broad cross-section of industries representing a variety of project sizes and scopes that have made the transition are critically important to those of us facing the issues today.

The SOA Consortium has also invited experts to discuss hot topics in SOA including Governance and the latest one on the relationship between SOA and Business Process Management (BPM).

One final project to mention: the SOA Consortium Army Adoption Project was just recently launched and is a unique opportunity for members to guide and collect a success story in real-time, as SOA transforms the operations of an enormous enterprise. The U.S. Army is a huge and distributed enterprise with systems and system of systems on every imaginable platform. Driven by the need to develop and deploy solutions against a more flexible architecture that can be more responsive to change, easier to maintain, and less costly, the Army has decided to transition from approaches that promote stovepipe solutions to those based on SOA. And one of the key tactics enabling this transition is the description of a business-centric SOA Lifecycle Management Methodology (LCMM) that focuses on business principles, architecture guidance, governance, funding and acquisition, and change management.

All of this content is readily available on the SOA Consortium Web site (www.soa-consortium.org) and is just a small sample of the work that our members are doing. Through deliverables like the lessons-learned case studies, and by speaking out at industry events and through podcasts and webinars, the SOA Consortium and its members are striking a blow against the old "SOA is technology" myth and redirecting the conversation to the promise of SOA as the key to business agility. Why not join us? ■

Altova® UModel® 2008 – The starting point for successful software development.

ALTOVA® umodel® 2008

Visualize works of software art

Draw on UModel® 2008, and picture better programs based on UML™.

Spied in UModel 2008:

- Support for all 13 UML 2.1.1 diagrams plus a unique XML Schema diagram
- Code engineering for Java, C#, and now VB.NET
- Visual Studio and Eclipse integration with real-time model synchronization

Altova® UModel 2008, the developing force in the software design space, is the simple, cost-effective way to draw on UML. Use it to create or interpret software architectures. Visually design application models and generate Java, C#, or VB.NET code and project documentation. Or, de-code existing programs into clear, accurate UML 2 diagrams. With all 13 diagram types, interoperability via XMI 2.1, an artful user interface, and more, UModel makes visual software design practical for any project. Take the mystery out of UML!

Download UModel® 2008 today:
www.altova.com

UModel is also available as part of the value-packed Altova MissionKit™ software bundle.

UML is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries.

ALTOVA®

What's Next After SOA?

How About What's Next *In* SOA

WRITTEN BY DAVID M. STRAUS

This is the time of year when trend or predication articles start cropping up. Year after year I'm asked if I would be interested in writing about what's to come. You see I have an uncanny ability to pick lottery numbers. Unfortunately my lottery guesses, like most articles that look into the future, aren't right most of the time. So I usually say no, and leave my writing time to topics of here and now.

I was approached about the forward-looking topic, "What's Next after SOA." That one I couldn't turn down. Oh, what an opportunity to write about SOA 2.0 or is it 3.0? Come on, I hear the Web is up to version 4.0. The title "What's Next after SOA" frustrated me. We seem to thrive on moving on before we finish our job. Maybe this is a high- tech/Silicon Valley thing. I think that in the heartland of technology use – like in government and most real companies like banks – getting SOA right is a multi-year, maybe even a multi-decade journey. These are organizations that start with, let's say, 40 years of information technology heritage that can't be thrown out. So what's the point about talking about what's after SOA before we've even conquered effectively implementing SOA. So with a little word change in the title from "after" to "in," I'd like to muse for a minute about "What Next in SOA." Some thoughts on what we must do to get the tremendous benefit SOA can offer our organizations.

First, as you might have gleaned from that last thought, I'm a huge believer in SOA. But I've been a believer in what SOA represents for years. Heck, when I got started it was called structured programming. Everybody talked about it, but not many people did it. Why? It was hard to do. It actually took time to build structured and reusable code. But it paid off. Programs were far easier to maintain. And it was possible, if not easy to implement back then, to reuse pieces of logic. Wonderful pieces of logic that acted like standard business services. Wow, we were building services? But in reality the overall development environment wasn't up to the task. Think of a six-year-old with the intelligence of an adult and the emotional maturity of, let's say, a six-year-old. You can't really put those smarts to full use.

Well, our six-year-old has grown up quite a lot. The environment has changed and now we have a layer of technology that can effectively orchestrate these services. Let's call this BPM. We now have

wonderful standard interfaces that let us deploy these services so they can be used across heterogeneous orchestrated flows — Web Services became the answer. All is well.

Well, not exactly. We still have a long way to go. And I'd like to touch on a few topics of what we still need to do. These are things you should consider in your own service-based architecture. I'll briefly illustrate due to allotted space what needs to be considered.

Explicit and Discoverable Service Contracts

This one has lots of tentacles. First, how do you make a service that other applications or application developers can discover that lets them effectively reuse the service? Seems simple right? Just implement a service registry. But it isn't simple, so you need to consider this topic. There is a tremendous amount of complexity in making a service usable by someone who doesn't understand the inner workings of the service. It's not just the syntax of the interface to the service; it's the semantics. There's work going on in this area; but for services to be truly decoupled and leveraged, developers have to be able to find a service that exists that truly meets their needs. Otherwise, we'll again have multiple groups in a company, basically replicating the same code but this time in a service wrapper.

Atomic Services

In my opinion, this is the hallmark of a well-designed service. No, you can't wrap your entire legacy application with a Web Service and say you're done. Services are supposed to have a purpose that is unique. In an insurance company a service that processes a completely new life insurance policy isn't atomic. It's a process. The activities in the process have to be broken apart. Unique services should be constructed for each component part of this process. Like a service to determine the risk of an applicant (Risk Score), and the value of the applicant to the insurance company (Customer Value Score), and the cost of the policy based on the applicant's risk and value (Policy Cost); now each of these services is most likely atomic.

Why should you care? Because atomic services have attributes that are critical to a well-designed SOA. First, they can most likely be reused without negative effect. This service, that determines the applicant's value to the insurance company (e.g., Customer Value Score), can be reused in making marketing offers, or in real-time call routing applications. Do you really want that code coupled with Policy Cost code? Second is the topic of agility. When made atomic, we have the greatest chance that the specific business organization that owns the service, the organization that determines Customer Value Score in this case, can independently maintain and change the service as needed. These changes can be made without breaking other pieces of the application, injecting agility to the process. If our service isn't atomic and deals with all of these underwriting functions (risk score, customer value score, and policy cost) then we'll feel required to test each of these functions when we change our customer value score function, since we all know that when you change any code, there are unexpected consequences.

Services with a Point

Freud once said that sometimes a service is just a service. Actually that's the problem. Not all services are the same. We should know what kind of service it is. Is it a data service designed to get data needed for the application? Or is it a decision service designed to provide an answer such as Customer Value Score?

Generally the industry hasn't provided a clear definition of the kind of services that should exist. Don't wait for the industry to solve this one for you. Your organization should do this. This would make it easier for the business to think about what happens in its business process and for developers to build services that are explicit and atomic. Consequently it will be easier for consumers of services to understand exactly what the service does.

I could write a paper on this well-defined set of services, but for grins I'll pick on one. Decision Services are near and dear to my heart (that's what my company does). I saw a wonderful quote that said decisions should be the first-class citizen in all of application development. Other than data, which is the history and reality of an organization, what we do as companies is make decisions (which transforms data):

- We review incoming applications to determine if they're correct
- We approve these applications
- We allocate machinery to manufacture product against forecast
- We determine how to ship and store this product for optimal cost against our demand chain

All of these action items are decisions. Other questions to consider are: Has your organization identified decisions as a kind of service? Or routing of work as an explicit service? Does your organization have a clear definition for the types of services you are building? Is each critical decision that exists in your application environment identified as a unique decision service? By defining service types, it becomes much easier to identify services and service boundaries when building a service-oriented application.

So where is this all leading? To get the value out of SOA, we have to continue to invest in clearly defining the architecture and components of the architecture. Building a house without plans usually ends with a pretty messed-up house. For SOA to thrive, we have to be clear about the definition of our services, their interfaces, and how to simplify so that we can reuse them.

In SOA reuse and agility are key. But the combination of all of these new atomic services creates new complexity. The complexity should be dealt with in the definition of the architecture so that we can have simplicity in execution and use. To march down this road effectively and get the value we're seeking, we need to ensure that are services are atomic so that they can evolve independently. We need to make sure they have a purpose that's clear so that the consuming users know what the service does. And we need to make sure that the service interface is explicit and simple so that when the service changes we don't have a negative cascading effect on all of the applications that have come to depend on the service. If we don't deal with this subject, we're doomed to see articles about new architectures (What Next After SOA) because this one (SOA) has failed. To make SOA successful, we need to continue to invest in refining what's next in defining and implementing an effective service-oriented architecture. ■

About the Author

David Straus is responsible for Corticon Technologies' global marketing including product management and marketing, field marketing, and corporate communications. David joined Corticon with over 20 years of software solutions experience in product, marketing, and sales. He has held executive positions at Chordiant Software after it acquired OnDemand, Inc., a company David founded in 1997. Prior to OnDemand, Straus held executive positions at TSW International, OpenVision Technologies, and Hewlett-Packard. David graduated from Indiana University with a BS in business and operations research. For information on Corticon, visit www.corticon.com.

David.Straus@Corticon.com

Who's Responsible for Quality in a Service-Oriented Architecture?

Impact and trust

WRITTEN BY WAYNE ARIOLA

Service-oriented architectures are much different than our traditional siloed applications. SOAs span multiple tiers and multiple applications, their development lifecycles are multi-staged and highly iterative, and there are multiple parties impacted by both internal and perhaps external changes to the organization.

Delivering a secure, reliable, compliant SOA, given these complexities, requires a different approach to quality. In a complex system, quality measures must permeate every aspect to translate into a quality deliverable.

Architecture & Interface Consistency

For example, the architecture and interface must adhere to certain policies and standards to remain consistent across the organization. To achieve this consistency, an organization must drive the standardization and enforcement of such policies and standards during design and development. Once a definable policy has been enforced, consistency and ultimately trust in a SOA asset can be achieved.

The process of promoting trust for a service asset must begin as soon as the asset is created. Early and continuous visibility into the quality of the asset promotes trust and subsequently reuse. To promote trust early, the business must define and enforce policies that govern the different aspects of the services lifecycle. For example, policies that govern the development of services include standards compliance such as schema validity (W3C), semantic, WS-I compliance, and a definition of adopted WS-* standards. Such policies are critical to achieving consistency and ensuring reuse and interoperability.

Policies can also include best practices, both generally accepted best practices and best practices related to the organization's goals for the underlying service artifact. Examples of these domain-general policies include security, maintainability, reliability and performance, and any other policies that are tailored to the domain requirements.

Once the policies are defined, it's critical that they are applied, verified, tested, and measured. We must automate the SDLC infrastructure and process so these policies and standards can be uniformly applied on SOA artifacts by development, test, and QA groups.

Continuous Regression Tests

To implement the policies, contracts, and scenario descriptions for a service asset correctly and robustly, they must be validated via a series of automated, continuous regression suites.

As part of the lifecycle of any given services assets, a continuous regression test must be available (during any release cycle) to provide constant and predictable quality. This continuous regression test must be driven by the business requirements. These tests must include functional requirements, technical requirements, and performance expectations. If these early tests are requirements-driven then QA will have a significant advantage by reusing the test assets.

To ensure secure, reliable, compliant service assets, integration via the asset registry is compulsory. Continuous regression tests should be run nightly during the automated build and the appropriate test data made available via the central asset repository. Tests are then run consistently, without disrupting the existing development process.

This automated testing ensures the automatic validation of requirements and the reuse of test assets between development and QA. Automated regression suites also provide documented proof that the application policies have been enforced, allowing for a more predictable outcome.

Complexity of the SOA Environment

Developing and ensuring functionality in an SOA environment is very complex. The distributed nature of the systems and data make the construction of a staged test environment nearly impossible.

The design part of SOA often requires working with an unfinished

Sound Architecture Requires Proper Planning

WEB AGE SOLUTIONS - YOUR TRAINING PARTNER FOR SOA

In all phases of SOA migration, Web Age Solutions provides training and customized services from awareness to implementation. We support vendor specific or generic SOA training tailored to your organization's needs.

Custom training for complementary SOA technologies

XML • WEB SERVICES • WSDL • SOAP • WEBSPHERE • WEBLOGIC • JBOSS • J2EE • SPRING/HIBERNATE/STRUTS • WID/WBI/WMB

Custom training plans for virtually every job role

BUSINESS ANALYST • ADMINISTRATOR • DEVELOPER • ARCHITECT • QA/TESTER • MANAGER • EXECUTIVE • SECURITY

Consulting services for all phases of SOA migration

IBM Premier Business Partner

Add Web Age Solutions to your plan & stay ahead of the competition

www.webagesolutions.com - 877.517.6540 - soa@webagesolutions.com

WEB AGE solutions inc.

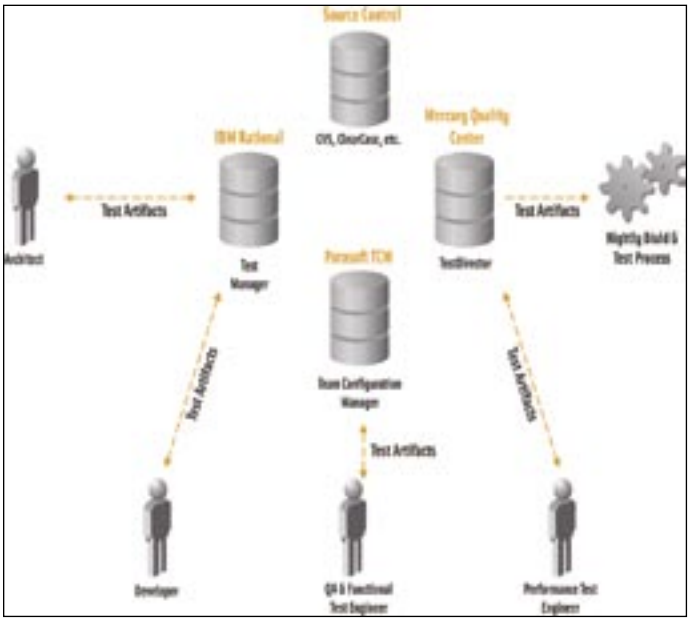


Figure 1

or unavailable service; it would be ideal to emulate services to provide encapsulated services. A developer or QA engineer should emulate services and exercise business scenarios very early in the process to vet problems earlier and to judge the service validity and anticipated performance better.

Process Agility & Speed

Advanced platforms deliver an SOA infrastructure that delivers the speed and agility to meet more complex business demands. Unfortunately, QA processes are traditionally serialized and "over the wall." A collaborative and building-block approach to quality is particularly well suited to handling the complexity that comes with SOAs. If businesses want to achieve agility through rapid incremental deliverables, then organizations can't wait for a serialized quality process.

This approach must change for organizations to truly benefit from the agility that SOA can deliver. As such, an iterative and change-based quality process should be leveraged to deliver true SOA agility.

As seen in Figure 1, reusing the test artifacts created in earlier tasks in the business service evolution process lends itself to increased agility and speed.

In addition, the setup of the continuous regression suites mentioned earlier allows for phenomenal visibility into changes.

Change-based testing allows the team to understand the impact to change and versioning – testing only what matters saves time and delivers a complete and validated solution much faster and more accurately.

Integration

Business Partner Integration is still an "integration" event. Bringing any two business processes together will require some give and take. That said, even a standards-based approach to producing and consuming services can be troublesome. To eliminate finger pointing during integration, a business partner must have access to the service "health report" as well as access to an emulated stub with test data that mocks the service interaction. This will prevent errors early in the cycle with visibility into the service assets regression test performance

Development and Pre-Deployment

In a business process context, architects will have to make choices to deploy certain service assets over others. This decision can't be made in a vacuum. Furthermore, when architects are making service choices, they'll have to be able to test and verify the performance of the service asset. So developers and QA can test their business scenario or business process via intelligent stubs that are updated as a result of the nightly regression tests.

Conclusion

Due to the complexities of the SOA environment, there will be a distinct impact on the roles and responsibilities of the organization. Older methods of application testing can't be relied on to ensure quality for any SOA. In today's SOA world, logic is abstracted at the message layer as well as across systems, making testing more complicated and giving QA less control. Although the roles of development and QA will remain the same, the responsibilities of each group will need to shift and the quality process must start earlier in the development lifecycle to ensure secure, reliable, and compliant Service Oriented Architectures. ■

About the Author

Wayne Ariola, vice president of strategy at Parasoft, oversees the company's business development team. He has the responsibility for growing the company's revenue and customer base through channel sales strategies. Wayne has more than 12 years of strategic consulting experience within the high technology and software development industries. He has a BA from the University of California at Santa Barbara and a MBA from Indiana University.

wayne.ariola@parasoft.com

Advertiser Index

ADVERTISER	URL	PHONE	PG
Active Endpoints	active-endpoints.com/soa		35
AJAXWorld Conf & Expo	www.ajaxworld.com	201-802-3021	32&33
Altova	www.altova.com		13&36
BEA	bea.com		11
IBM	IBM.com/takebackcontrol/connect		2&3
Managed Methods	www.managedmethods.com		27
Mindreef	www.mindreef.com/not-stranded.html	603-465-2204 ext.581	5
Nastel	www.nastel.com		23
Object Management Group	www.omg.org/bpm-soa-wm	781-444-0404	29
SnapLogic	www.snaplogic.com		25
SYS-CON Books	books.sys-con.com	201-802-3020	30
TIBCO	http://developer.tibco.com/		7
Web Age	www.webagesolutions.com	877-517-6540	17
WSO2	http://wso2.com/products.esb		19

General Conditions: The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of ***.Net Developer's Journal.*** All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher. Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in ***.Net Developer's Journal.*** Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

Tired of bloated, complex, expensive ESBs?

Try this ESB for size:

- 40MB download
- up and running in minutes
- industry-leading performance
- simple Web tools and admin
- 100% open source, 100% free
- powered by Apache Synapse
- training and commercial support available

WSO2 Enterprise Service Bus

http://wso2.com/products/esb

WSO2

The open source SOA company

Can Agile Development & Model-Driven Design Solve the Broken Delivery Process?

Eliminating the gap between the business and the technology that drives the business

WRITTEN BY JACK KENNEDY

The software delivery process has long faced many challenges, many of which are exacerbated by the need for organizations to update their internal architectures and the methodologies they use to build and deliver solutions. Over 80% of software projects are delivered late, over 50% don't deliver required features and cost overruns exceed 15% on average.

Traditional waterfall methodologies are especially vulnerable to these types of issues because they generally incorporate flawed perspectives on the ability of a team to fully gather and document requirements in the early project stages. Many demand a significant investment in upfront planning and design, only to find that the volumes of information and documentation generated are rendered increasingly obsolete as requirements change.

Worse, these documents result in draining precious resources in an attempt to keep them current. All too often, software projects conclude unsuccessfully with solutions that no longer match requirements, or documents that no longer match solutions, all with a greater investment than originally predicted. This article describes how iterative and agile development methodologies are being combined with Service-Oriented Architecture (SOA) and model-driven development to solve many of the long-standing challenges in software delivery.

Getting It Right the First Time

To be certain that the software being delivered meets the needs

of the groups who will use it, it's important to consider all of the stakeholders throughout the process.

If the user community is only consulted at the beginning and end phases of product delivery, then it's very likely that the final product won't be well received. Historically this risk was mitigated by attempting to define requirements at a very low level. This approach is time-consuming, brittle, and often the least productive way of integrating software users with technology implementers.

Large software specifications have a small community of readership, many of whom are non-technical and not intimately familiar with foreign concepts like UML diagrams. To get it right the first time, teams are moving towards planning and delivery models that incorporate the end user throughout the process and use "working" software wherever possible to validate and record requirements.

I'll Know It When I See It

Very often, software projects and their ultimate success are dependent on the "I'll know it when I see it" usability test. The user community is often asked to design the solution, or at least describe the set of requirements in such a way as to lead the implementation team to an obvious path of execution. In many cases, the results are several iterations of consider, plan, design, build, review, reconsider, etc. While this set of iterations is important, it is even more crucial that the methodology and technologies that are being used for software delivery correctly plan for and enable this type of delivery process.

Enter the Prototype

At some point, teams got smart and decided that they would attempt to mitigate the problem by creating prototypes. This idea is a great one, giving the end-user community something tangible so that the risks of post-implementation modifications are reduced. While prototyping becomes a great enabler in this methodology, providing useful screenshots for the requirements documents, it's been relegated to project waste.

Historically prototypes have been used for screen mockups, but they're often devoid of several key items such as meaningful data or logical execution flows. It's also rare for prototyping to continue beyond the initial project implementation. Once software delivery has begun, it's more common for requirements changes to be documented and implemented in later cycles rather than reintroduced to the original prototype. For this reason, the value of prototyping is rarely effective beyond the initial delivery phase.

Architecture, Methodology and Models

For years, the software industry has invented and reinvented ways of organizing and managing the process of developing and delivering software. Recent advances have primarily focused on reconsidering the fundamental aspects of the software architecture, with an emphasis on modularizing software into services embodied by the SOA movement. There's also been a focus on processes with general trends driving the industry towards iterative or agile methodologies. Each of these trends represents the cumulative industry response to two of the "most broken" parts of software delivery, and contributes to the root issue, which is a gap between the needs of the business and the technology that drives the business. Prior to these adaptations, the following are generally held to be true.

- *Prior Architecture*
  - Software reusability was low
  - Software implementations were inherently "static"
  - Systems were generally "monolithic"
- *Prior Methodology*
  - The predictability of software projects was low
  - Software delivery plans and requirements were inherently "static"
  - Software projects were generally "monolithic"

Service-Oriented Architecture

The drive towards SOA was in large part an attempt to solve the basic architectural issues described above. Breaking software up into smaller pieces increases the likelihood that any given service could be reused in different ways that may not have been originally considered when the service was first created (see Figure 1). As a result, large systems can be implemented with smaller segments, and the applications move from being inherently static towards being more configurable and dynamic. Another benefit to dividing up software delivery projects into smaller pieces is that it becomes easier for team members to design and implement more specific portions of software that can be designed and tested independently of other pieces.

Agile and Iterative Methodologies

The issues with methodologies required a shift in the rationalization of how software projects are built and how they differ from other kinds of projects where material goods are produced. There's an inherent expectation that software is generally "virtual" and should therefore be fluid in its design, as opposed to a skyscraper, say, which is tangible (see Figure 2).

Figure 1:

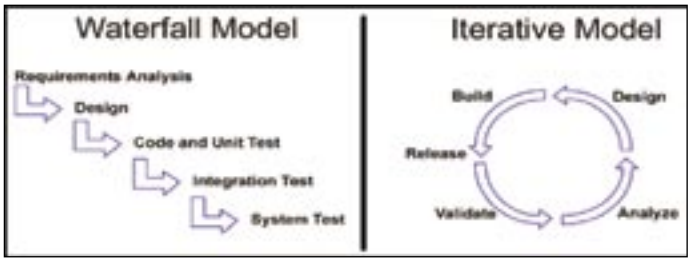


Figure 2:

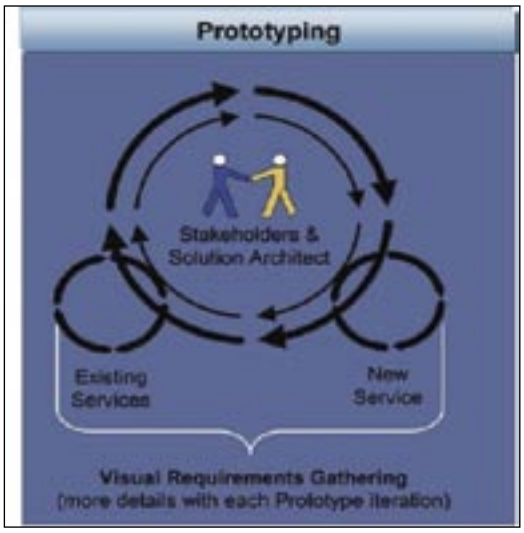


Figure 3:

The drawback was that the same methodologies used for material goods were being applied to software. This issue was compounded by the fact that the underlying technology infrastructures that software is built on are continuously changing and therefore introduce unpredictability into the software delivery practice.

Agile methodologies take an approach similar to SOA in solving these basic issues. First, the methodology is shifted that it doesn't attempt to describe every nuance and detail of the system prior to planning and implementing it. It then introduced the fundamental tenant of iteration over linear development practices. Previous methodologies with linear paths had a devastating impact on projects when non-linear events occurred like changed requirements or downstream technology modifications. Fundamentally, the shift was similar to SOA in the way the project was broken into smaller pieces and prioritized so the pieces of highest priority were addressed first before creating full designs for other pieces. This minimized the risk of wasted investment in requirements gathering and focused teams on implementing software modules that worked and were demonstrable as early in the process as possible.

The Missing Link

And so at first glance, it would appear that as the industry adopts the SOA approach and incorporates the tenants of agile methodologies into its delivery environments, the resolution to the gap between the business and the technology would be filled. What's becoming increasingly apparent, however, is that while the gap may be closing, there are still significant missing pieces that will have to be incorporated.

Both SOA and agile are clearly steps in the right direction, but there is a set of factors that are gating their ability to achieve their proposed goals. The basic approach to requirements documentation and software implementation is still broken. The reason is that while the methodologies and architecture have improved, the linkage between the two has not. Fundamentally, the third major movement in the software industry is underway and it's set to fill the last gap and enable SOA and agile to reach their full potential.

Declarative Software Delivery

Historically, there have been many attempts to create a solid link between business requirements and technology implementations. The original focus of UML was to create this link in the form of documentation that followed a consistent set of constructs with the goal of fully documenting system requirements. Early software projects were plagued not only by changing requirements but also by completely undocumented requirements. It stood to reason at the time that to solve the software delivery conundrum, you should create a set of technologies and documentation practices that would allow teams to consider fully all of the requirements of the project before starting. While this advance was an important first step in the process, it was indeed a first step.

Fundamental problems with software modeling and requirements documentation and management have persisted and have gotten far less attention and support than the architecture and methodology as evidenced by the shift to both SOA and agile. As the industry becomes comfortable with these two shifts, it's refocused its efforts on software modeling and declarative software delivery as the third necessary ingredient in bridging the gap.

Why Is Software Modeling Lagging?

Just as the software industry has found ways to solve some of the fundamental issues with software architectures and delivery methodologies, so too is it actively pursuing the goal of updating software modeling so that it can be reincorporated into modern SOA and agile development practices. Below are some of the primary issues being addressed:

Fundamental Issues

- UML is rooted in a set of object-oriented design techniques that hinged on a classic waterfall delivery process.
- UML was originally designed as a way of documenting requirements, not of generating software
- The predominant architecture has shifted from object-oriented to service-oriented
- The predominant methodology has shifted from waterfall to agile
- A new requirement has been added to utilize models to generate full software solutions
- The core concepts of the Unified Modeling Language (UML) haven't been updated to enable these shifts.

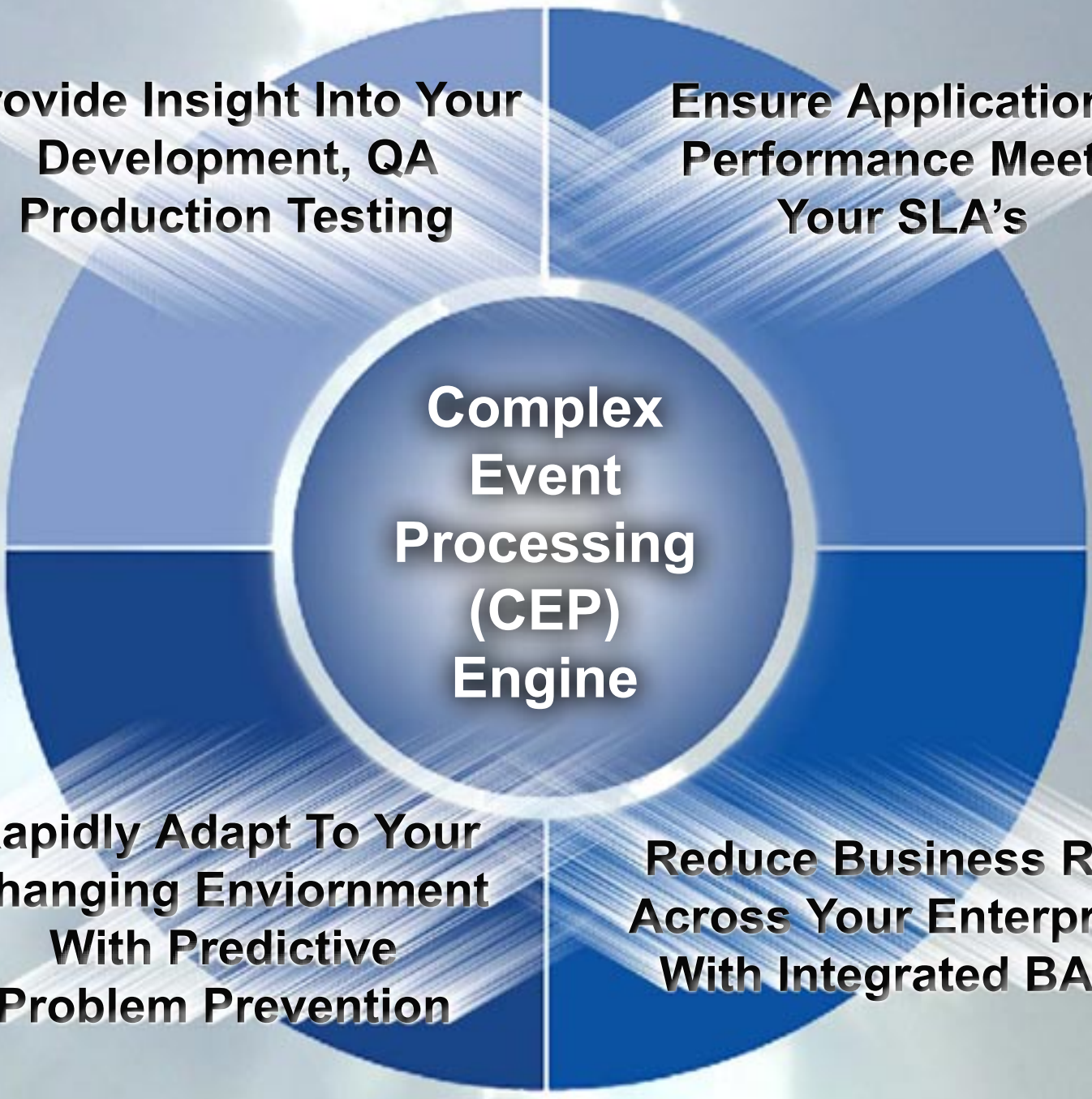
Specific Criticisms

The following criticisms of UML have been leveled for many years:

- **Language bloat**. UML is often criticized for being gratuitously large and complex. It contains many diagrams and constructs that are redundant or used infrequently. This criticism is directed more at UML 2.0 than UML 1.0, since newer revisions include more design-by-committee compromises.
- **Problems in learning and adopting**. The problems cited above make learning and adopting UML problematic, especially when management forces UML on engineers lacking the prerequisite skills.
- **Only the code is in sync with the code**. Another perspective holds that it's working systems that are important, not beautiful models.
- **Cumulative impedance/impedance mismatch**. As with any notational system, UML can represent some systems more concisely or efficiently than others. So a developer is pushed towards solutions that most comfortably inhabit the intersection of the strengths between UML and the coding language. This problem is particularly pronounced if the coding implementation language doesn't adhere to orthodox object-oriented doctrine.
- **Tries to be all things to all people**. UML is a general-purpose modeling language that tries to achieve compatibility with every possible implementation language. In the context of a specific project, the most applicable features of UML must be delim-

We Shed the Light On SOA

Solutions For Complex Application Performance Management



NASTEL™

We work with some of the world's largest banks, financial services, retail chains and insurance companies. IT managers in these companies have to comply with stringent service level agreements (SLAs) and government-mandated regulations and reporting requirements. We provide the most flexible, powerful mechanisms to ensure the performance of the complex applications that drive business by quickly finding and fixing problems to meet business requirements. For more information, visit http://www.nastel.com.

ited for use by the design team to accomplish the specific goal. Additionally the way to restrict the scope of UML to a particular domain is through a formalism that's not completely formed and is itself the subject of criticism.

Where Modeling Is Going

The agile doctrine substitutes working and demonstrable software for bloated requirements processes, but it does so with the implication that working software can become a more fluid contract between the user community and technology implementers. To achieve this, it's important to be able to deliver working software as quickly as possible. There are several key requirements for software modeling that must be achieved to incorporate it successfully with SOA and agile techniques:

- Incorporates stakeholders in the modeling process
- Creates working prototypes quickly
- Prototypes are durable and aren't thrown away
- Self describing and documenting
- Includes service-oriented concepts
- Technology-neutral and XML-based
- Inclusive of integration and distribution concepts
- Easily understood by a wide audience

By updating software modeling to meet these standards, the industry is beginning to create the direct link that enables service-oriented software delivered through modern methodologies. The models are taking the dual role of describing the software requirements while also generating the working solutions creating quicker iterations and higher-quality working software. The resolution to the modeling challenges of historical technologies and approaches yields additional benefits:

- Synchronized requirements and implementation
- Faster and more frequent iterations
- Tighter integration between stakeholders
- Greater predictability
- Alignment of requirements and architecture
- Increased productivity
- Decreased waste
- Broader adoption

Iterative Prototyping

The value of being able to iterate software solutions quickly throughout the process is that the prototyping stage is never lost and can be incorporated throughout the lifecycle as an empowering force to the communication between the various stakeholders (see Figure 3).

The tenants of creating working software on short iterations, which are communicative of both the software's requirements and its implementation, are achieved by using the software model for both purposes. This ensures that there's never a gap between the requirements and the code, and reinforces the active participation of the user community in the delivery of software and services.

This approach blurs the line between requirements, code, and prototype. In addition, it minimizes many of the traditional difficulties with keeping the software project synchronized over time. Prototypes generated from software models generally contain a large portion of the software logic, business rules, and final form user experience that becomes increasingly complete as the project progresses. This advance not only accelerates the pace of the iteration, it also improves the quality of evaluation resulting in fewer requirements-related miscommunications.

"The use of XML is becoming increasingly popular as a way of describing software"

The Role of XML

The use of XML is becoming increasingly popular as a way of describing software. While this trend isn't new, some of the outcomes are. For example, XML standards have become prevalent as a way to describe various Web Services standards representing both the definition of service endpoints, as well as the data that is communicated during the service execution. Business Process and Execution Language (BPEL) and other process languages express the orchestration of services as XML. Most development platforms use XML documents to drive pieces of the software's development, including user interface flow control, persistence rules, and deployment concepts. Service data models are also generally described as XML. Many tool vendors are adding visual composition capabilities to these types of XML documents creating something that resembles a visual development language. Software modeling itself is generally being driven by a combination of XML documents and visual XML designers. These are tailored to incorporate standard programming concepts that are represented visually and persisted as XML.

Modernized Software Delivery

Modern software delivery practices are incorporating a richer, more mature set of methodologies, technologies and tools to maximize the quality and usability of their projects. The benefits of iterative development and SOA are being amplified by revolutions in the software modeling space, which incorporate service concepts directly into models while also providing an actionable record of the changing software requirements. These additions are increasing software predictability and promoting software reuse through componentization and generative software techniques. They also ensure consistency of software delivery standards. XML-based software declarations continue to gain prominence moving beyond data definitions and process orchestration directly into service and user interface modeling and implementations.

These trends are all derived from the common purpose of simplifying software delivery and continuing the process of eliminating the gap between the business and the technology that drives the business.

References

UML. http://en.wikipedia.org/wiki/Unified_Modeling_Language

About the Authors

Jack Kennedy, vice president of product delivery at Skyway Software, cofounded the company along with Daniel Aegerter and Jared Rodriguez. He leads all phases of software development at Skyway Software. Before joining Skyway Software, Jack was a consulting architect for MarketMile, a leading e-procurement service provider. He served as chief architect of the Marketplace product for Ariba, following Ariba's acquisition of TRADEX. Jack earned a BA in management of information systems from the University of South Florida.

jkennedy@skywaysoftware.com

Need data integration now?

You drag and drop.
We'll do the REST.

Data Integration for the Last Mile

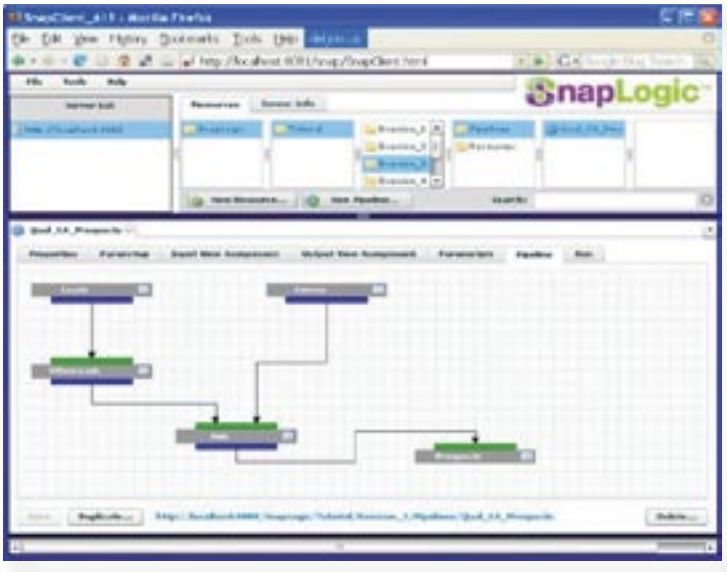
SnapLogic applies the power of the Web and uses a **RESTful** architecture to transform data into data services. Using a simple drag-and-drop interface, SnapLogic dramatically accelerates your access to integrated data.

SnapLogic is free, Open Source software that you can begin using today to address projects such as:

- **Data integration between Edge data sources and Core SOA infrastrcture**
- **Data integration for reports and Business Intelligence**
- **Enterprise mashups combining disparate data sources**
- **Rich Internet Applications**

Download the SnapLogic Community Edition today at **www.snaplogic.com**

Read the latest SnapLogic White Paper, *"Making Sense of SOA, SOAP, and REST"* **http://www.snaplogic.com/makingsense/**



SnapLogic Designer makes it easy to design and execute data integration Pipelines in a Web browser

SnapLogic™

+1 (650) 655-7200 www.snaplogic.com

(c) 2007 SnapLogic, Inc. All rights reserved. SnapLogic and the SnapLogic logo are trademarks of SnapLogic, Inc.

iTKO LISA 4 SOA Testing

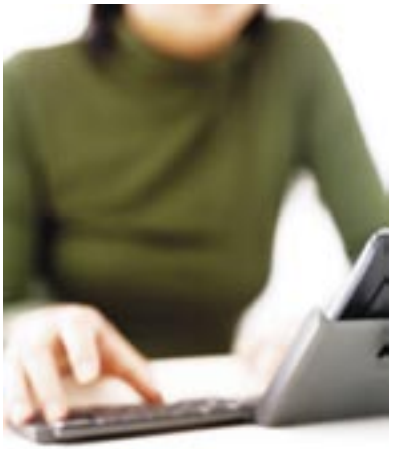
A key enabler of broad SOA design and implementation

WRITTEN **BY PAUL O'CONNOR**

Testing is a first-order constituent of SOA governance. Assuring that services and service infrastructure components meet functional and technical requirements across lifecycle stages and environments, including production, is an architectural precept in SOA.

Gone are the days when testing was an isolated hop in the old disjointed SDLC. You know the one I'm talking about – an architect would design a component, hand it to developers to develop and they would hand it to testers to test, and testers, in turn, who would hand it to system administrators to operate. The components were often large, complex, and dependent on other complex components and systems, and so it was hard to write enough test cases to adequately assure quality.

The result was a lot of uncaught errors that led to a vicious cycle of patches and more testing, consuming valuable enterprise resources for no appreciable gain. Thankfully, SOA has served to break those monolithic components into manageable services, service infrastructure pieces, and integration components, and has brought to fruition collaborative service lifecycle governance.



Test cases can be written against smaller components, shared among staff members with varying roles, and ultimately composed to test business processes across every level of the architecture.

The result can mean deep integration of your testing regime and your solutions, and the automated collaboration of all contributors. But you're going to need some specialized SOA test tooling to get there.

Introducing LISA 4

I recently had the opportunity to take a close look at the LISA 4 SOA testing suite from iTKO. I was immediately struck by the completeness of the tool's vision as a key enabler of not only service governance but of broad SOA design and implementation itself.

iTKO describes the vision for LISA as "the 3 Cs":

- **Complete** – testing of business processes from the system level all the way down through to the lowest-level component. You'll see shortly how this vision is actually executed in the tool.
- **Collaborative** – sharing of test processes and assets between all relevant parties in the solution delivery model.
- **Continuous** – testing in SOA never stops and LISA supports this across all phases of service and solution lifecycle, including runtime.

Version 4 of the tool drives further towards this vision with some interesting new features:

- Virtual service environment
- Dynamic load testing
- BPM and workflow testing
- Enhanced support for governance platforms
- Native ESB integration

The LISA test suite comprises three components:

- **LISA Enterprise** – a desktop test client for building and staging test cases – unit, functional, regression, and process tests.
- **LISA Server** – a server-side engine that schedules and orchestrates tests for the purpose of continuous validation, driving load tests, and managing the virtual service environment.
- **LISA Extension Kit** – an API for developing instrumentation for legacy applications and platforms, which can then be included as steps in a test case

Creating Test Cases with LISA

LISA offers a unified UI that comprises LISA Enterprise functionality for creating and running test cases, as well as deploying and managing tests on a LISA Server. Figure 1 is a screen capture of an example Test Manager test case with the test steps expanded in the left explorer pane. Test steps are created to exercise the system under test, or execute utility functions, with a wide variety of transports, protocols, messaging styles, and data formats available. Response data can be filtered ("Filters" in the figure) and is tested against a set of assertions ("Assertions" in the view). Data sets are configured against a number of backing stores to drive the test data, and can be randomized. An interactive test run utility lets testers step though test cases and visualize the responses.

LISA ships with the following test steps:

- Web Services (including Web 2.0)

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

Copyright 2007 Managed Methods Inc. All rights reserved

- Web – HTTP/HTML
- Swing
- Java – EJB, RMI, POJOs
- CORBA
- SQL/JDBC
- External processes (including external JUnit tests)
- Messaging (native ESB, SonicMQ, Web Methods, Websphere MQ, and Tibco Rendezvous)
- Invoking subordinate test cases as sub-processes

Each category includes test steps that enable the test case to interrogate a component of the system under test, e.g., a Web page, Web Service, or database. Also included are utility steps for setting up the interrogation steps, e.g., collecting a SAML token for use in a Web Service invocation. And the ability to invoke subordinate test cases enables composing existing cases into a larger test of a business process or solution. Just as service composition in SOA is a key functional enabler, composability of test cases in SOA is a key governance enabler.

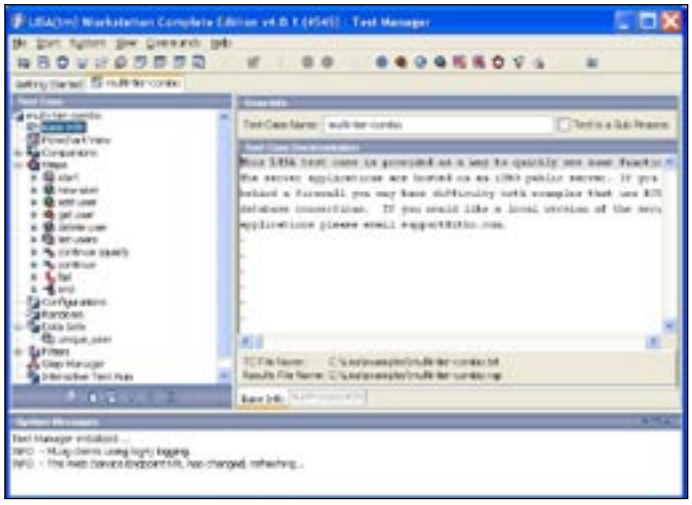


Figure 1: LISA's view of no-code test creation and management

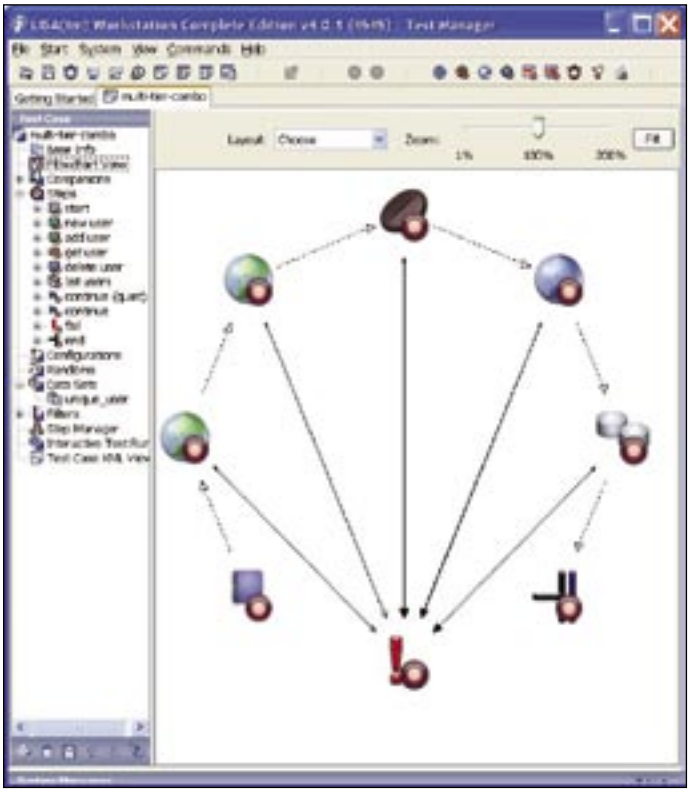


Figure 2: LISA's workflow view showing multiple SOA components

The sequence of test steps provide a complete test case and form a workflow that can be visualized in the flowchart view, as illustrated in Figure 2. And like any workflow, LISA supports parallel flow of test steps as well as looping. You could, for example, build a test case composed of ESB component service tests, a BPEL process test, a business rule service test, and validated results against a database system of record.

Each type of test step has its own icon. The dotted line around the outside of the "Ferris Wheel" view constitutes the happy path. The lines extending to the middle stop node represents assertions that must be met for the test to continue. The flowchart view is useful for diagnosing workflow problems like infinite loops. As you should be able to tell by now, LISA operates as a holistic system test tool, able to test every layer of a solution architecture, from the data layer all the way through to the Web layer and all interconnects along the way. Figure 3 is a screen capture of Test Manager's test monitor for the example test case.

Taking a deeper look at a Web Service test case, LISA includes steps for introducing WS-Security elements to service requests. You can generate a SAML token and put it into the SOAP Header, as well as encrypt fields and add digital signatures. You can also add custom transport headers and add client certificates for use in an SSL/TLS certificate exchange. LISA will also procure the Web Service access point from a UDDI inquiry service with a binding key, and even validate the response for WS-I compliance. You can even add attachments to your service requests.

Other LISA 4 Functions and Services

The Continuous Testing Service (CTS) functionality of the LISA Server allows test cases to be scheduled against running systems. Test cases in CTS are called monitors, and the CTS dashboard visualizes running monitors, pass/fail ratio, pass/fail for each monitor displayed as a histogram, and a monitor average response time graph. It's easy to envision how this functionality could come

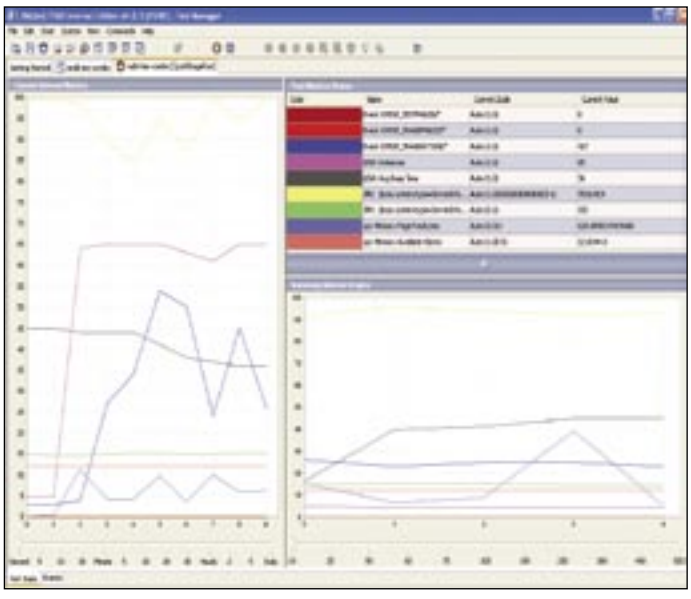


Figure 3: Continues loud and performance test metrics in LISA

OBJECT MANAGEMENT GROUP

ANNUAL SPONSOR:

Sparx Systems

SILVER SPONSORS:

Mega

Telelogic

Register before December 17, 2007 for early-bird savings and hotel discounts!

SAVE THE DATE!

OMG's Maximizing BPM Investments with SOA Workshop

January 14–17, 2008 • Orlando, FL

Maximize BPM investments while meeting real-world needs.

- Need to get your business folks talking to your IT folks?
- Need to know about how SOA can help you to maximize your BPM investments?
- Want to hear how other folks have succeeded?

This in-depth, three day workshop provides guidance and evidence on how business leaders and technologists can apply an SOA approach in developing IT support of their business needs and processes. Drawing upon OMG's experience in developing software standards, attendees will gain valuable insight through tutorials, keynotes, presentations and demonstrations on how industry leaders have worked through the hype and are developing BPM solutions that maximize investment while meeting real-world needs.

KEYNOTE JUST ANNOUNCED!

Maximizing BPM Investments with SOA

David Linthicum
Managing Partner
ZapThink LLC

Highlights include:

- In-depth tutorials detailing how SOA defines an architecture for business processes; modeling techniques for achieving business solutions; what standards mean to the business person; and the role of Event-driven Architecture (EDA).
- Informative sessions covering vertical market case studies and best practices; technical issues; strategic planning; and the "human factor."
- Expert panels exploring a model-driven approach to BPM and SOA and cooperation among standards organizations.
- A unique hands-on exercise—a team of pros working with the audience to solve a sample business problem using available technology.
- State of the art technology exhibits and cutting edge demonstrations from a wide variety of solution providers.

Presenters will represent leading organizations such as:

Adobe Systems	IBM	PricewaterhouseCooper
BEA Systems	IDS Scheer Global	Relativity Technologies
Burton Group	Interfacing Technologies	SOA Consortium
Collaborative Consulting	Kohl's Department Stores	The RuleML Initiative
CSC	Lombardi Software	TIBCO
Deutsche Bank	ModelDriven.org	VISA
Elemental Links	OASIS	W3C
Harvard Computing Group	OCTL Consulting	*and many more!*

For more information or to register:
ONLINE: www.omg.org/bpm-soa-wm
CALL: +1-781-444 0404
EMAIL: loughry@omg.org

Don't miss this unique event! Register before December 17, 2007 for early-bird savings and hotel discounts. Visit www.omg.org/bpm-soa-wm for complete details.

in handy. You could set functional test cases to run periodically against test data in production to ensure that functional requirements are being met. And you could schedule load tests for off-hours to catch any performance problems. Coupled with LISA's built-in reporting engine, which records specified test metrics and performance, a rich historical view of operation and performance of solutions is a snap.

One of LISA's coolest features is its comprehensive Virtual Service Environment or VSE. If you can give LISA a WSDL for a Web Service, or even point it to a UDDI registry inquiry service with a service key, it will create a virtualized version of the service suitable for use in your test case. The virtual service's responses can be fulfilled by multiple backing stores, including JDBC and XLS sources, and even a pre-recorded session of data from the real service. And you can configure your test case to switch over to the virtualized service, and dynamically recorded response data, when and if the real service becomes unavailable. Think of how empowering this is – a BPM developer can thoroughly test their business processes in the absence of underlying services and systems, thereby streamlining promotion to integration testing. And by eliminating a redundant test environment in most large enterprises LISA's cost can be returned tenfold. In this respect, LISA becomes a key enabler of implementing an SOA itself – removing dependencies between elements of the architecture means that they can be implemented in parallel and integrated later.

Another LISA innovation is the Pathfinder, a code instrumentation tool that tracks requests as they are processed through the container. For some application services, e.g., JBoss, enabling Pathfinder is a configuration step. In other cases, it may be required to add an annotation to the code itself. And you must add a LISA code library to the deployment. Pathfinder can be used to probe Web pages, Web Services, ESB process flows, and Java classes, among others. After having enabled Pathfinder at the code execution point, leveraging the data in LISA's test client is a simple matter of adding a technology-specific filter to the test case. Then the interactive test run facility will display deep container metrics that can be extremely helpful in performing root cause analysis.

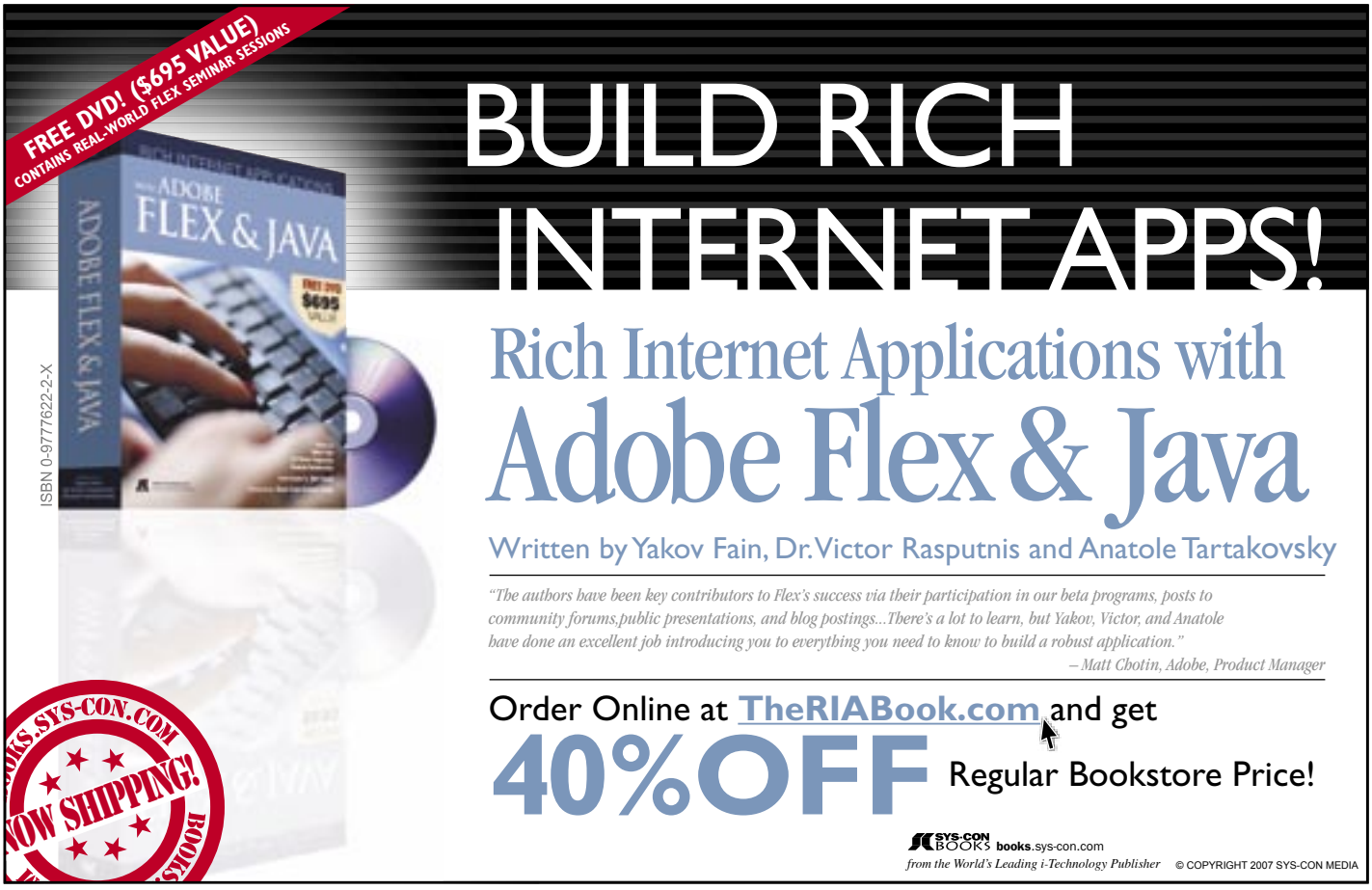
Conclusion

Testing is a key enabler of governance in SOA. Done properly, testing is collaborative, continuous, and done across all layers of the architecture in the context of a solution test case. LISA 4 from iTKO represents a mature tool with some compelling features to support implementing this vision. Its intuitive user interface, extensive suite of test steps, extensibility, continuous testing service, deep component instrumentation, and virtualization service should get you nicely towards implementing a great SOA testing program. ■

About the Author

Paul O'Connor is SOA Practice Director and Chief SOA Architect for e-brilliance LLC (a leading NE SOA consultancy), and is currently doing major SOA architecture and implementations for Fortune 100 clients across the US. Previously he was chief architect for Damascus Road Systems, specializing in security architecture.

poconnor@e-brilliance.com



Visit the *New*

www.SYS-CON.com

Website Today!

The World's Leading *i*-Technology News and Information Source

24/7

FREE NEWSLETTERS
Stay ahead of the i-Technology curve with
E-mail updates on what's happening in your industry

SYS-CON.TV
Watch video of breaking news, interviews with industry leaders, and how-to tutorials

BLOG-N-PLAY!
Read web logs from the movers and shakers or create your own blog to be read by millions

WEBCAST
Streaming video on today's i-Technology news, events, and webinars

EDUCATION
The world's leading online i-Technology university

RESEARCH
i-Technology data "and" analysis for business decision-makers

MAGAZINES
View the current issue and past archives of your favorite i-Technology journal

INTERNATIONAL SITES
Get all the news and information happening in other countries worldwide

JUMP TO THE LEADING i-TECHNOLOGY WEBSITES:

IT Solutions Guide
Information Storage+Security Journal
JDJ
Web Services Journal
.NET Developer's Journal
LinuxWorld Magazine
Linux Business News
Eclipse Developer's Journal
MX Developer's Journal
ColdFusion Developer's Journal
XML Journal
Wireless Business & Technology
Symbian Developer's Journal
WebSphere Journal
WLDJ
PowerBuilder Developer's Journal

Rich Web Technologies for Enterprise Web 2.0 & Social Applications

6TH INTERNATIONAL

AJAXWORLD™ CONFERENCE & EXPO

March 18-19-20, 2008
New York City
www.ajaxworld.com

Join More than 10,000 Early Adopters Who've Already Attended!

Come & Learn How to Achieve the Optimum Balance Between Business Needs, Technology Potential, & Enhanced User Experience!

Let AJAXWorld Provide Your Business Blueprint for Future Web Success. Doing business on the new Web is 99% about innovation and optimizing the user experience, and AJAXWorld Conference & Expo is 100% about the tools that encourage and support a superior user experience - the tools for business survival and Web success in 2008 and beyond.

Reasons to be there...

- You will leave AJAXWorld Conference & Expo 2008 East with resources, ideas and examples that you can apply immediately.
- Whether you're going to be using the technology to enhance an existing Web application or to migrate a client/server application to the Web, AJAXWorld will help you get up to speed more quickly.
- Attendance will improve your company's ability to adapt and prosper in the Web 2.0, user-centric environment.
- Whether you need to work with a designer or do the work yourself, front-end engineering makes more sense after you return from AJAXWorld!

The Real-World Questions That Will Be Answered by This Yearts Faculty!

- AJAX development IDEs: what are the alternatives?
- When it comes to accessibility, how do I know what's required?
- How can I make AJAX applications that easily go offline?
- How do you make a secure AJAX application?
- Are mashups really contributing to the experience of the Web?
- What's the difference between a mashup and a composite application?
- Am I better off using an AJAX framework, a toolkit or just coding?
- Is AJAX commoditized or will it be soon?
- Will AJAX be standardized in the form of widget APIs or declarative markup?
- What AJAX development tools are available for visual development today?
- How should you handle Web analytics and metrics for Rich Web sites?
- Do we really need JavaScript 2.0?
- How do you scale an AJAX-based design to a very large application?
- How can I harness the energy of my customer base online?
- On what timeline will AJAX skills become commoditized?

Technical sessions will cover challengers to AJAX such as new innovations like Silverlight and JavaFX. We will be dealing with performance and browser compatability issues, showcasing viable strategies for performing unit/stress testing on an AJAX application, and how to apply user interface patterns and user experience designs to your AJAX project.

Every aspect of AJAX and rich-Web alternatives will be covered, and the emphasis throughout will be on actionable insights, real-world case studies, and up-to-date technical information and advice.

Who Should Attend...

- CTOs & VPs of Engineering
- Directors of Technology
- Sr. User Interface Architects
- Front-End Engineers
- VCs & Industry Analysts
- Directors of Business Development
- Software Engineers
- Senior Architects
- Application Programmers & Software Developers
- Project Managers
- Web Programmers & Designers
- Companies & Organizations that need to stay in front of the latest Web technology

2008 Call For Papers Now Open!

AJAXWorld offers the undisputed best platform to position your company as a leading vendor in the fast-emerging marketplace for AJAX and Enterprise Web 2.0.

REGISTER TODAY & SAVE

Must-See Speakers

BOLOKER, IBM · WEI, Nexaweb · FERRAIOLO, OpenAjax · CROCKFORD, Yahoo! · PELED, Gizmox · PELLER, IBM

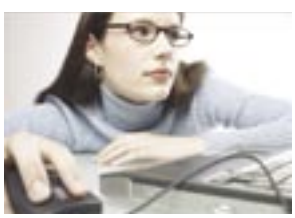
SCHALK, Google · GUPTA, Sun · MIRAGLIA, Yahoo! · STAGNER, Microsoft · POLLACK, Ruby Users · REGAN, Sonoa

AJAXWorld 2008 East - Tracks

- **Track 01:** Enterprise AJAX
- **Track 02:** Rich-Web Case Studies
- **Track 03:** Enterprise Web 2.0 & Social Applications
- **Track 04:** Enterprise Web 2.0 Mashups
- **Track 06:** iPhone Developer Summit
- **Track 06:** Diamond Track

Introducing in 2008

AJAXWORLD™ SECURITY BOOTCAMP



ajaxsecuritybootcamp.sys-con.com

March 18, 2008 New York
October 20, 2008 CA

iPhone Developer Summit

iphonedevsummit.com

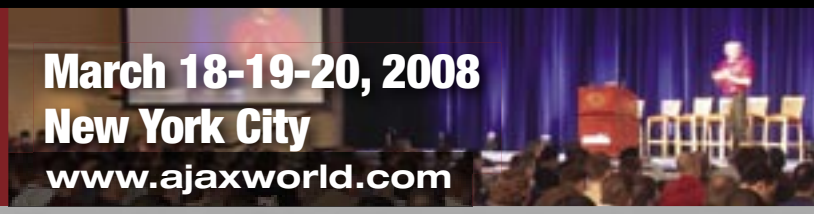
March 19-20, 2008 New York
October 20-22, 2008 CA

Call Now for Sponsor & Exhibitor Opportunities
Space Is Limited 201-802-3021 or carmen@sys-con.com

CALL 201-802-3020 OR E-MAIL EVENTS@SYS-CON.COM

Rich Web Technologies for Enterprise Web 2.0 & Social Applications

6TH INTERNATIONAL
AJAXWORLD™ CONFERENCE & EXPO



March 18-19-20, 2008
New York City
www.ajaxworld.com

Call 201-802-3020 to Register Today. Register online at events.sys-con.com

2008 EVENTS CALENDAR

3rd International Virtualization Conference & Expo

2008 VIRTUALIZATION CONFERENCE + EXPO
www.virtualizationconference.com

The concept of virtualization is rapidly evolving from taking one computer and making it appear to be many computers, to virtualizing almost any information technology resource. At Virtualization Conference & Expo 2008 East, the two-day technical program will show your business how the cost of virtualization is going down and the efficiencies to be gained by it are increasing.

virtualizationconference.com

June 23-24, 2008 New York | November 2008 CA

13th International SOAWORLD™ 2008 CONFERENCE & EXPO

SOA World Conference & Expo 2008 East brings together the best minds in the business to New York for a two-day conference that offers comprehensive coverage of SOA and will help you rethink how your own business can better leverage its IT investments today. Learn from more than 60 speakers about how SOA is transforming business – and altering forever the way IT and business managers think about their businesses, processes, and technology.

soaworld2008.com

June 23-24, 2008 NY | November 2008 CA

AJAXWORLD™ SECURITY BOOTCAMP

Introducing the world's first-ever full, one-day immersive "AJAX Security Bootcamp" - led by one of the world's foremost AJAX security experts and teachers, Billy Hoffman.

ajaxsecuritybootcamp.sys-con.com

March 18, 2008 NY | October 20, 2008 CA

iPhone Developer Summit

In the two-day iPhone Developer Summit, developers will hear from industry experts about the impact the iPhone is having on the delivery of rich content to mobile users. Technical sessions will explore a world of Web development opportunities on the iPhone including building applications using the new ASP.NET MVC framework, building social applications, and developing high-quality, iPhone-style Web-based GUIs for your applications.

www.iphonedevsummit.com

March 19-20, 2008 NY | October 20-22, 2008 CA

OPENWEB DEVELOPER SUMMIT

Are you up to speed on Google Gears? What is Google Mashup Editor? Have you started building mini-apps with Google Gadgets? Or OpenSocial, the set of common APIs for building social applications across many Websites? Do you use GWT to build AJAX apps in Java? How about Android, the software stack for mobile devices including an operating system, middleware and key applications? Will the Android SDK change the wireless world?

events.sys-con.com

April 21-22, 2008 NY | October 20-22, 2008 CA

AJAXWORLD 2007 SPONSORS & EXHIBITORS INCLUDE

IT MILL, kapow TECHNOLOGIES, ORACLE, REARDEN, lightstreamer, THOMSON, krugle, Methods & Tools, PARASOFT, SERVOY, Tele Atlas, TIBCO, JACKBE, Laszlo, 3tera, PASSPORT, SmartClient, ThinWire, OpenSpot, Apress, CYNERGY, froglogic, ICESOFT, JetBRAINS, MANNING, OpSource, helmi, FARATA SYSTEMS, BACKBASE, Gongtomet.com Conference Directory, ADDISON-WESLEY, ILOG, KAAZING, tilestack, UNIVERSAL MIND, webtide, BZ Media, Extension MEDIA, Conference Guru, SnapLogic, TECHTRACKER MEDIA, VertexLogic, WEB SPIDERS, Adobe, Dynamic Toolbar, QUASAR, Microsoft, PushToTest, IBM, Google, Sun, Etelos, nexaweb

SOA as a Corporate Responsibility

The new motivator

WRITTEN BY **DAVID S. LINTHICUM**

While few will disagree that the inefficiencies of existing enterprise architectures have reached a critical level, many count on "flying under the radar" of those who look at a company's efficiencies. Let's face it; enterprise architecture is very technical and difficult to understand by the layman. The well-publicized corporate scandals have focused on shady accounting practices and corporate mismanagement while IT has gotten a pass in years past. This will no longer be the case. Today, those who look at value and invest in a company poke around all aspects of a corporation, especially any areas of inefficiencies, including IT. In other words, dysfunctional enterprise architectures devalue the corporation overall and can put the business at risk.

Enterprise architectures, however, can't be fixed simply by bolting some new technology on or building connections between systems. Architecture is, well, architecture, and it requires a great deal of planning and analysis to create a strategy that rejuvenates the enterprise architectures into something that lives up to corporate expectations of business agility and efficiency. So, fixing your architecture could take years, and now is the time to take the first steps toward creating an architecture that serves the needs of the business and not the other way around.

Reacting to this dilemma, enterprises have created positions called enterprise architects. They are single individuals, or groups, in an organization who are responsible for driving the enterprise architecture strategy going forward. While a good idea in theory, the reality is that many of these enterprise architects simply don't have the political or budgetary clout in their companies or government agencies to make much of a difference. Many times they have been relegated to creating reports and presentations that nobody reads and providing direction and guidance that's easily ignored.

So, without good architectural governance and ongoing corporate management pressure to redirect resources to tactical IT projects, enterprise architectures continue to become more unnecessarily complex, static, and fragile. What was a mere annoyance only a few years ago today is a clearly limiting factor in the ability of business to create shareholder value. The company can't easily shift into new and emerging markets, acquire companies, and adjust major business processes without a great deal of latency. In some cases, they're completely unable to change. In other words, things are bad and getting worse.

SOA isn't a miracle cure for bad architectures. However, it's a step in the right direction for those looking to move their existing enterprise architectures into something much more efficient and valuable to the business. Those who embrace SOA as a practical architectural pattern in the context of a long-term strategic architectural plan, and can execute architectural rejuvenation without tactical interruption, will find that they're quickly ahead of the game. SOA provides two primary values. First is the ability to save development dollars by reusing services. Second is the ability to change the IT infrastructure faster to adapt to the changing needs of the business. Agility is SOA's primary value proposition, and enterprise architecture's for that matter.

Those looking to leverage SOA are tempted to sign up for the SOA-in-a-box type of solution...perhaps an ESB, a BPEL engine, or a governance tools, or all of them. Unfortunately, "buying-and-bolting-on" technology solves very little, and can actually make things worse. As ZapThink has said many times, SOA is something you do, not something you buy. But the buying patterns of those in the planning stages of SOA are still very much influenced by "hype-driven" "manage-by-magazine" solutions that can cause many people to find SOA distasteful if the technology doesn't live up to the hype. There are no quick fixes; real work must be done.

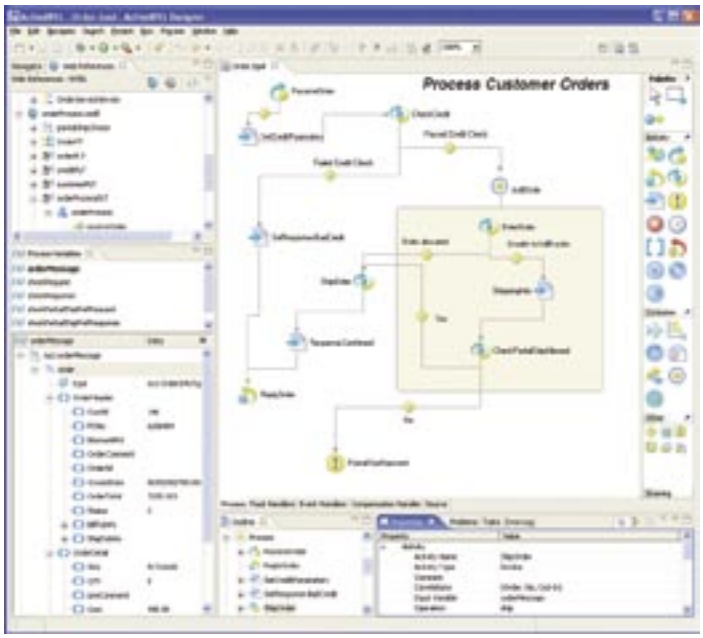
Indeed, doing SOA is a complex undertaking, and you'll have to learn a great deal to become efficient with the emerging approaches, techniques, technologies, and methods. Those who are successful at SOA plan and design long before they develop and implement. The path to a truly strategic and valuable SOA is something that only comes to those who understand the importance of the work leading up to the technology. They also have corporate sponsorship, appropriate funding, and proper amounts of mentoring and training.

The movement toward a SOA should be something that has key strategic significance in a company or agency, much like a new product or line of business. In fact, a well-planned well-implemented SOA will be far more valuable by comparison, considering its long-term ROI. In essence, SOA should have boardroom visibility. Sometimes it seems that architecture must have a lousy PR agency. The value is clear to anyone who analyzes the real cost of the limits that bad enterprise architecture put on the business. However, the negative effects on the business are still widely accepted and thought of as something that really can't be fixed. Nothing could be further from the truth. If those who manage a company aren't motivated to fix their architectural issues themselves, perhaps shareholder lawsuits or missed strategic opportunities that cancel yearly bonuses will become the new motivator. ■

About the Author

David S. Linthicum (Dave) is a managing partner with Zapthink, LLC (www.zapthink.com), a consulting and advisory organization dedicated to excellence in SOA planning, implementation, training, mentoring, and strategy. He is an internationally known application integration and SOA 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 SOA, approaches and technologies in wide use today. He keynotes at most major SOA and Enterprise Architecture conferences, maintains one of the most read SOA blogs, is the host of the weekly SOA Report Podcast, and is the author of 10 books, three on integration and SOA topics.
david@zapthink.com

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

Copyright 2006 Active Endpoints, Inc. All Rights Reserved.
All product names are trademarks or service marks of their respective companies.

Altova® MapForce® 2008 – The premier data conversion, transformation, and integration tool.

ALTOVA® mapforce® 2008

Give your data direction

Link up with MapForce®, and exchange data with ease.

Spied in MapForce 2008:

- ValueMap function for mapping numeric values to names, etc.
- Support for aggregate functions: SUM, MIN, MAX, AVG, STRING-JOIN
- Enhanced options for XML output

Altova® MapForce 2008, the award-winning data conversion, transformation, and integration tool, makes it easy to exchange data between XML, databases, flat files, EDI, and/or Web services. Simply drag connecting lines from data sources to targets and drop in data-processing functions. MapForce instantly converts the data or auto-generates a data integration application for royalty-free use. Languages for code generation include: XSLT 1.0, XSLT 2.0, XQuery, Java, C++, and C#. Get connected!

Download MapForce® 2008 today: www.altova.com

MapForce is also available as part of the value-packed Altova MissionKit™ software bundle.

ALTOVA®