

The DoD

SoftwareTech

WWW.SOFTWARETECHNEWS.COM

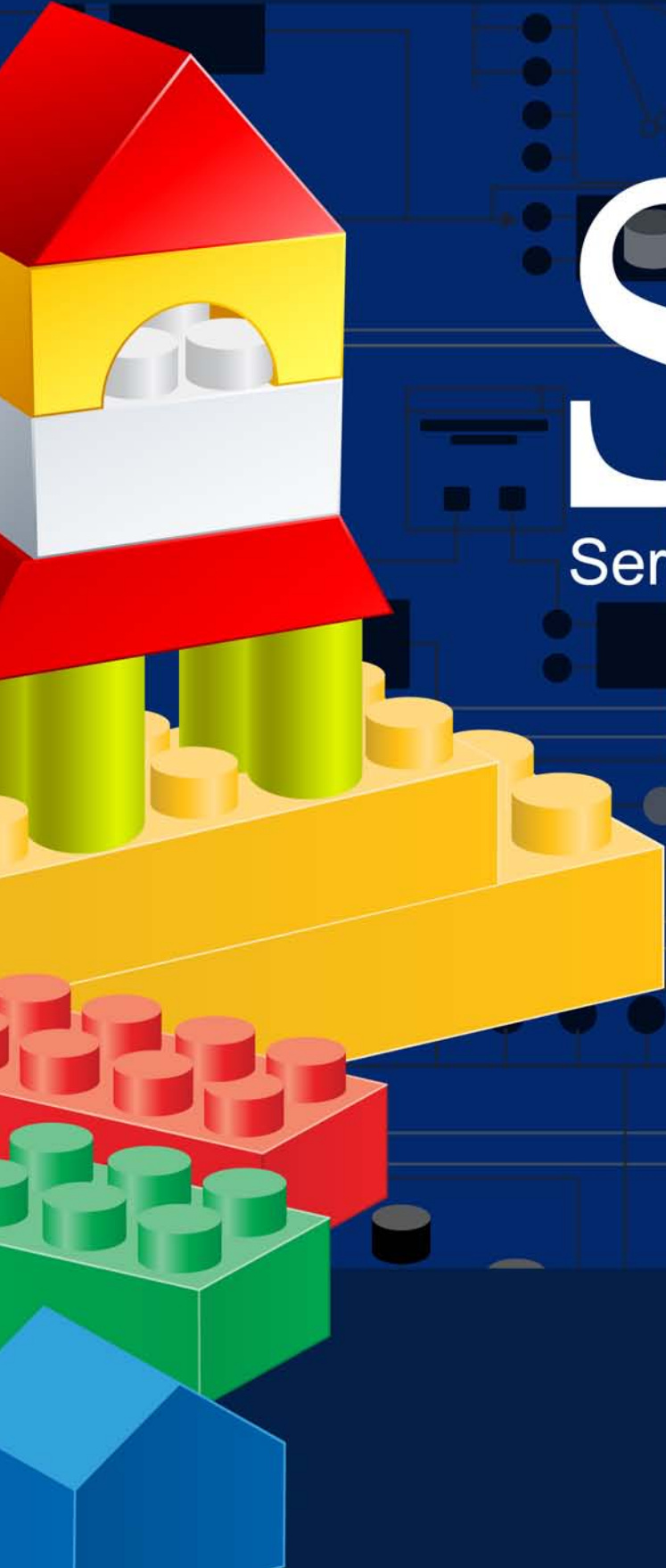
NEWS



JUNE 2008 Vol.11. No.1

SOA

Service-Oriented Architecture



DACS

The Data & Analysis Center for Software

<http://iac.dtic.mil/dacs>

Unclassified and Unlimited Distribution



How long can you wait for CMMI[®] Compliance?

Manage your software and systems engineering projects in guaranteed compliance with the CMMI NOW!

processMax 2i and 3i include all the necessary policies, procedures, guidelines, criteria, templates, and forms in role-based, step-by-step instructions, ready for use—everything you need for compliance with the CMMI-DEV at Level 2 and Level 3 respectively. Integrated with robust document management, workflow, and automated measurement and reporting, **processMax** is the intranet web-based solution for effective and efficient management of your software and systems engineering projects. **processMax** enables you to start operating in compliance with the CMMI immediately! We guarantee that organizations using processMax will not fail a formal appraisal led by an independent SEI-licensed appraiser.



pragma SYSTEMS CORPORATION
www.pragmasystems.com
877.838.PMAX
info@pragmasystems.com

GSA Schedule Contract No. GS-35F-0559S. processMax is a registered trademark of **pragma** SYSTEMS CORPORATION. Although processMax makes use of portions of "CMMI[®] for Development, Version 1.2," CMU/SEI-2006-TR-008, copyright 2006 by Carnegie Mellon University, neither the Software Engineering Institute nor Carnegie Mellon University have reviewed or endorsed this product.

Copyright © 2008 **pragma** SYSTEMS CORPORATION.

The Commercialization of SOA

By Ray Cardillo, ITT Corporation

Why is SOA such a hot buzzword these days and what's the buzz about? Several of the articles in this issue reference various definitions for SOA and concur that the term is getting a lot of attention yet not very well understood by most customers. How can a term that is so commonly misunderstood get so much attention? The most common reason is that enterprises are interested in the promised benefits that are associated with SOA (e.g., simplified enterprise integrations, agile business process adaptation, long term cost savings, etc). However, unlike many other technical buzzwords, SOA is not a technology that can simply be purchased and deployed.

In fact, SOA is not really a technology at all. SOA is an architectural design paradigm which makes use of other technologies that enable the goals of a SOA. In practice, it turns out to be much more complicated than it sounds due to a variety of issues that are encountered at every stage of the software development lifecycle. The promised benefits of SOA are valid, but achieving success has been difficult for many organizations. So it is no surprise that all of the most popular vendors in the enterprise software arena are trying to commercialize the concept of SOA.

The idea of commercializing SOA is for vendors to offer an integrated product suite that helps an enterprise achieve the goals of a SOA "out of the box". However, the notion of commercializing an architectural design paradigm raises some questions. Why are many of these vendors publishing competing SOA methodologies? Are these commercialized SOA suites really adding any value or just rebranding existing products and using the SOA moniker to capitalize on a trend? Does this result in any form of vendor lock-in and invalidate some of the benefits of SOA?

Most Popular Vendors

There are quite a few vendors competing in the SOA arena, but the most well known are: BEA, IBM, Microsoft, Oracle, and Sun. The leaders are BEA, IBM, and Oracle according to Gartner's 2Q07, "Magic Quadrant for Application Infrastructure for New Service-Oriented Business Application Projects". This category is focused specifically on business application projects, so it represents the SOA suites well.

All of these major vendors have web sites that are dedicated to the topic of SOA. They generally help educate potential customers about SOA and the benefits of their particular brand of products or solutions. The SOA product suites typically

include components that are already integrated and provide an easier way of developing and deploying web services, service oriented applications, and orchestrating service invocations to support business processes. References to all of the relevant web sites have been cited in the sidebar that accompanies this article.

Competing Methodologies

Although the goals of SOA are fairly well established by now, the process of designing a successful SOA system is not. Most vendors have evolved their own design methodologies as a result of previous consulting engagements. So it's no surprise that they tend to be tailored to their particular product suite. Tailoring in this context simply means that the terms, capabilities, and practices are well aligned with their solution.

These methodologies are important for several reasons. First of all, if the vendors are to help customers achieve success with their SOA solution, then tools alone are not sufficient. There must be a strategy for pursuing the design that is likely to lead the customer to a successful implementation. Since much about SOA is still in flux, it also helps each vendor stake a claim in the SOA landscape. If they can establish themselves as the SOA experts (even in one particular domain), then they can establish authority. This is why all of the major vendors each have their own methodologies published.

Some of the methodologies are more complete than others, but they all focus on design activities to help identify what services are required, what context they support, and what value they will provide to the enterprise. Various typical design activities are included as well, but most try to pay more careful attention to enterprise architecture concepts such as supporting an orchestrated business process, developing standardized data exchange formats (e.g., XML specifications) where needed to ensure interoperability, etc. In general, the SOA methodologies try to represent best practices that are derived from other successful SOA implementations. So the SOA methodologies end up being just as critical a contribution to a customer's success as the SOA product suite itself.

SOA Moniker Madness

Byproducts of the commercialization of SOA are the deluge of products that have put SOA in the title. It is certainly an attempt to capitalize on the industry buzz about SOA, but

continues on page 4

Tech Views

Continued from page 3.

it would be a shallow illusion unless the new SOA products suites show additional value.

Because SOA is a paradigm and not a technology, most of the products were already aligned with SOA design goals. Various products were used in areas of the enterprise where they were best suited to fulfill the design objectives of the system. However, many of the new SOA suites aim higher than that so that all of the most commonly required features of a SOA system are available in one environment. The most popular features this relates to are web services, service oriented web applications, service orchestration, and business process execution.

Providing these capabilities under the moniker of a SOA product suite is a valuable contribution because it can reduce the risks associated with integrating various offerings. While standards adoption should eliminate integration risk, problems still occur in practice because standards continue to evolve and some new features may be leveraged that other vendors do not yet support. The WS-I standard is a great way to combat this problem, but there is a gap when new features are required before they're in a final WS-I specification that is widely adopted by the major vendors. Although some vendors offer a single suite and others offer a product roadmap, there is some value being added. This is especially true for enterprises looking for one stop shopping. The vendors deliver a methodology to ensure a successful SOA design and a product suite to ensure a successful implementation and deployment.

Avoiding Vendor Lock-In

If we take this discussion up a level to talk about enterprise level integration, specifications become even more important. The ability for one organization to easily integrate with another is an important benefit that any SOA solution should deliver.

Vendor lock-in is typically introduced when unique (non-standard) capabilities of a particular platform are leveraged. So although the feature may be important to the organization, the implementation becomes bound to that platform. To be honest, there is nothing wrong with this if it is done with purpose and is internalized so that it does not proliferate to any external interfaces. Some of the SOA methodologies help identify risks such as this by analyzing the complete flow of service invocations that are required to support each business process. Likewise, the adoption of WS-I standards can help minimize risk further by identifying a set of minimum specifications that are required throughout the enterprise.

With this in mind, vendor lock-in does not have to be viewed from a negative perspective when the products support SOA. Organizations are free to develop components or systems in a way that ties their implementation to a specific vendor as long as they are careful with the definition and requirements of their external interfaces. Methodology and standards are critical to realize this goal, and the commercialization of SOA has pushed these issues to the forefront. If this trend continues and the core standards associated with SOA begin to stabilize, the typical fears of vendor lock-in should gradually fade away.

Evolution or Revolution

Commentary on SOA, past, present and future, varies widely. This is due to the fact that many of the SOA principles evolved from similar distributed computing architectural paradigms and related bodies of work. Some believe that SOA is nothing more than the continued evolution of that path. However, the goals have not been within reach until the development and widespread adoption of various standards such as XML, SOAP, WSDL, UDDI, and BPEL. Yet many organizations are still struggling to understand and adopt SOA. If the commercialization of SOA helps educate customers, results in more successful adoption, and continues to increase interoperability, then the overall result will truly be revolutionary. So let them propose their own methodologies and try to add a commercial spin to a design philosophy. They will be helping customers achieve success in the short term and will be fueling the SOA revolution in the long term.

This issue contains a collection of articles that provide insight into the current SOA landscape from different perspectives. The first article, by Robert Glass, presents the essential elements of SOA and introduces some of the most important standards and technologies that are SOA enablers. The second article, by Andrew Gordon, discusses why SOA transformation is difficult and suggests what factors affect success. The third article, by Emo and Brown, discusses challenges of federation in DoD SOA systems and suggests a solution. The fourth article, by Lance Walker, discusses the IBM SOA journey and includes some of the more important lessons learned. The final article, by Mecheri and Gordon, discusses how Open Architecture (OA) is related to SOA and why they can be synergetic.

About the Author

Ray Cardillo is a Senior Software Engineer in the Advanced

continues on page 5

Tech Views

Continued from page 4.

Engineering & Sciences division of ITT Corporation. He is currently working as a defense contractor focusing upon SOA and Network Centric transformation in the DoD. Prior to the defense industry, he has been both a lead Software Engineer and an Architect of several large systems at companies such as Nortel, Fujitsu, and Oracle. He holds a Master's Degree in Software Development and Management (Software Engineering) from Rochester Institute of Technology.

Author Contact Information

Ray Cardillo

Email: ray.cardillo@itt.com

SIDEBAR : Commercial SOA References

BEA SOA Homepage

<http://www.bea.com/soa>

BEA SOA Methodology

http://www.bea.com/framework.jsp?CNT=soa_methodology.htm&FP=/content/services/consulting/methodology

BEA WebLogic

<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/>

Gartner Magic Quadrant for Application Infrastructure for New Service-Oriented Business Application Projects, 2Q07 - 2007

<http://mediaproducts.gartner.com/reprints/beat/vol2/article3/article3.html>

IBM SOA Foundation

<http://www.ibm.com/software/solutions/soa/offerings.html>

IBM SOA Foundation Creation Scenario

<http://www.redbooks.ibm.com/abstracts/sg247240.html>

IBM SOA Homepage

<http://www.ibm.com/soa>

Microsoft Biztalk Server

<http://www.microsoft.com/soa/products/biztalkserver.aspx>

Microsoft Service Oriented Modeling

<http://msdn.microsoft.com/en-us/library/bb245662.aspx>

Microsoft SOA Homepage

<http://www.microsoft.com/soa>

Oracle SOA Homepage

<http://www.oracle.com/technologies/soa>

Oracle SOA Success Methodology

http://download-uk.oracle.com/docs/cd/B31017_01/core.1013/b28764/designing.htm

Oracle SOA Suite

<http://www.oracle.com/technologies/soa/soa-suite.html>

Sun Java Composite Application Platform Suite

<http://www.sun.com/software/javaenterprisesystem/javacaps/index.jsp>

Sun SOA Homepage

<http://www.sun.com/soa>

Sun SOA RQ Methodology

http://www.sun.com/products/soa/soa_methodology.pdf

SOA Essentials

By Robert Glass, Principal Solution Architect at Oracle's National Security Group

Today, you can't walk around the corner without hearing some mention of Service Oriented Architecture (SOA). Every week, I encounter SOA being misunderstood, misinterpreted, and misrepresented. The Intelligence Community (IC), amongst others, has caught SOA fever. In the last several years the IC has been moving towards SOA with Net-Centric initiatives (i.e. NCES, Horizontal Fusion, and various SOA references), but the problem is that people in the IC have differing definitions of what this means and how this impacts them.

In reality, the DoD and IC today are characterized by stove piped, data dispersed, heterogeneous systems with hard coded, inflexible processes, long lead times to change those processes and uninformed business stakeholders where rip and replace is not an option. Integration technology, which has been around for several years, is attempting to address this problem by shifting towards a standards based, Service-Oriented Architecture.

Based on my experience within the IC and now as a SOA Solution Architect for Oracle, I have put together a list of essential components for building a SOA. These components are as follows:

- **Web Services** - A service that is called in a standard way, so anyone can use it without knowing its internals
- **Enterprise Service Bus** - A way for services to communicate with each other

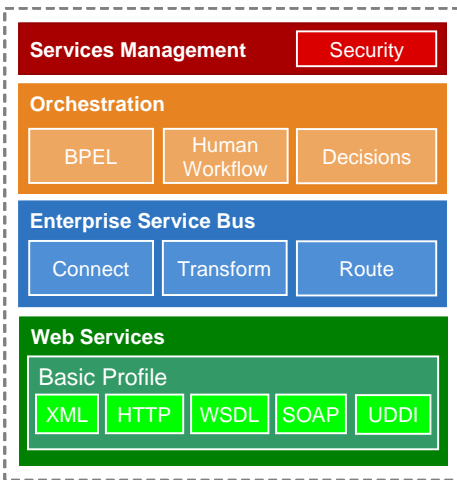


Figure 1: Essential Components of SOA

- **Orchestration** - A means for plugging services together
- **Services Management** - Manage and Secure SOA, via WS-Security & Identity Management (IdM)

This article will define and describe the benefits of SOA and provide the essentials necessary to identify, recommend and implement your own SOA solutions. I hope to dispel what SOA is and isn't and provide you the essential components of a SOA.

First, before we get into these essential components, we need to stop and define some key concepts of SOA. There are several formal definitions for a SOA; the OASIS¹ group has a SOA Reference Model which defines SOA as “a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations”². The Open Group has defined SOA slightly different to be “an architectural style that supports service orientation” which “is a way of thinking in terms of services and service-based development and the outcomes of services.”³ They continue by defining a Service as follows:

- Is a logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit, provide weather data, consolidate drilling reports)
- Is self-contained
- *May be* composed of other services
- Is a “black box” to consumers of the service

In other words, a Service-Oriented Architecture (SOA) is an architectural style with well-defined loosely coupled services that can be published and discovered in a uniform standards-based way using distributed capabilities. These services expose business functionality over the network usually as a Web Service in a standards-based language neutral manner via a Web Services Description Language (WSDL) which is an XML based contract for the Web Service. Since the services are black boxes and are self-contained they can be coupled and de-coupled like legos to create new processes dynamically.

¹ Organization for the Advancement of Structured Information Standards(OASIS)

² <http://www.oasis-open.org/committees/download.php/19697/soa-rm-cs.pdf>

³ <http://opengroup.org/projects/soa/doc.tpl?gdid=10632>

⁴ <http://www.oracle.com/technology/tech/soa/pdf/soa-suite-wp.pdf>

continues on page 7

SOA Essentials

Continued from page 6.

These key elements of the SOA paradigm are illustrated in figure 2.

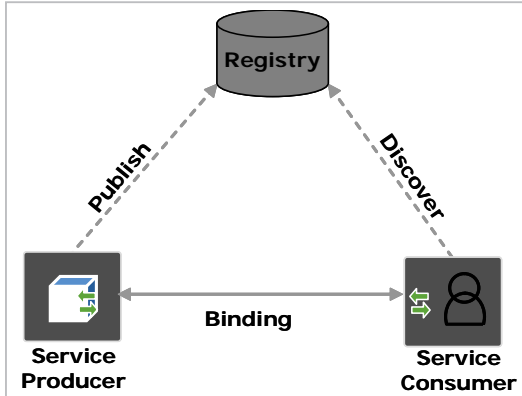


Figure 2: SOA Paradigm

Some of the benefits of SOA include:⁴

- **Interoperability** - Leveraging industry standards enables legacy and stove piped applications to seamlessly interoperate
- **Reuse** - Since these applications are service-enabled, these services can be reused to create higher level business processes
- **Agility** - The flexibility of standards allow implementers to rapidly adapt to changes as their business needs change
- **Visibility** - These processes also improve business visibility by enabling rapid integration into service-enabled enterprise portals and providing in-flight monitoring capability aiding in business decision-making
- **Reduced Costs** - Redundant services can now be eliminated and systems can be consolidated reducing the costs of operation and maintenance (O&M). As systems are upgraded using SOA and legacy systems are phased out, lower integration and O&M costs will provide more funds to enable new services to be built
- **Standardization** - As you standardize operational procedures you begin to gain better insight and control allowing you to have better visibility and governance of security and services management

Now that we have a definition for SOA, you can see why the first core component of a SOA is Web Services. Web Services expose your service via a self-described interface (WSDL) that can be invoked by sending a SOAP message to the service endpoint (the location of the service). SOAP, which was previously an acronym for Simple Object Access Protocol, is now the name of a protocol for exchanging XML based messages over a network usually over HTTP.⁵ See Figure 3.

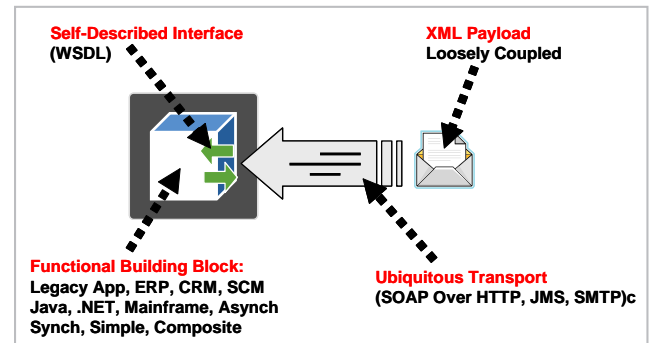


Figure 3: Elements of a Web Service

Web Services provide core capabilities such as integration of heterogeneous applications and data, interoperability across disparate platforms, and reuse of business functionality. Existing technology assets which can't easily be exposed as a Web Service can now be exposed using adapters which talk natively to the asset exposing it as a Web Service.⁶

In the early days of Web Services these specifications were just that and no one knew exactly how they played with each other. Web Service leaders from across the service community came together and formed the Web Services Interoperability Organization (WS-I) to address this problem. WS-I is chartered to develop best practices (profiles) and tools for Web Services and interoperability. Today, there are several profiles, one of which is the Basic Profile that provides interoperability guidance and specifies a set of standards and versions, such as HTTP, SOAP, WSDL, and XML, as well as usage for a basic Web Service.⁷

If you stop with Web Services then you have just created a point-to-point Web Services architecture. To complete the definition of SOA, "...where loosely coupled services are published and discovered", you need a *Service Registry*. A Service Registry is a catalog of services and related metadata

⁵ <http://en.wikipedia.org/wiki/SOAP>

⁶ http://www.oracle.com/technology/oramag/oracle/04-jul/o44web_feature_basics.html

⁷ http://www.oracle.com/technology/oramag/oracle/04-jul/o44web_feature_basics.html

⁸ <http://www.oracle.com/technology/tech/soa/pdf/soa-suite-wp.pdf>

continues on page 8

SOA Essentials

Continued from page 7.

organized by taxonomies.⁸ Taxonomies are basically a way to classify and organize services hierarchically (i.e. such as file directory). The Registry facilitates SOA adoption by enabling users to search for and publish services that meet specific criteria as well as browse offerings available from other providers using the Universal Description, Discovery and Integration (UDDI) standard.⁹ UDDI is an open standard that enables users to publish their services and discover others and define how those services interact over the network.

Next, you need a way to connect, transform, and route these services from place to place; thus, our next component (see figure 4) is an Enterprise Service Bus (ESB). An ESB's main function is to connect to these previously mentioned Web Services whether thru WSDL or an Adapter, then transform the payload to and from data types and route these request across the network.

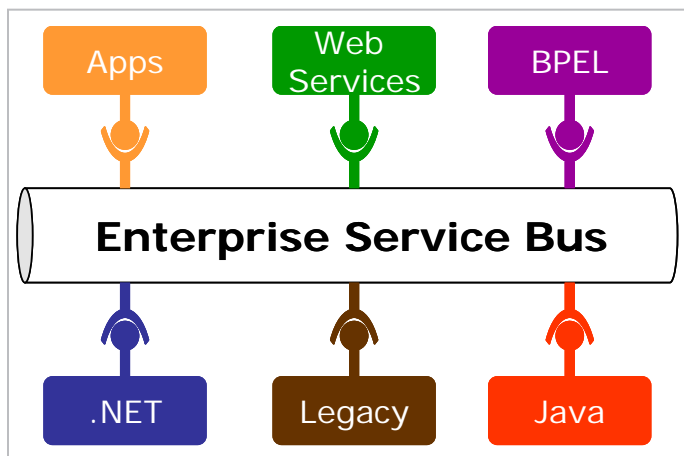


Figure 4: Enterprise Service Bus Component

An ESB provides a virtualization layer between clients and the resources they are trying to reach. As an example, take a database that resides on server A, then, using a database adapter, you can expose this resource as a Web Service through the ESB. The ESB can then provide a service endpoint for this service thus virtualizing the fact that the database resource is running on server A. The benefit here is when the database changes servers to B, the clients don't need to make changes in their code. Instead, the ESB just changes to server B rather than A.

The concept of an ESB has probably been the most confusing entity within the IC. I constantly hear that having an ESB equates to having SOA; that is quite far from the truth. An ESB provides an easy way to get to resources and expose them as Web Services; however it is just a component of SOA. When used properly the ESB can provide a layer of abstraction to provide the connectivity to these legacy systems that cannot easily be "web servicized," to support the necessary transformation and routing across the network.

Another very important capability within a SOA is the ability to mesh services together to create business processes and composite applications, otherwise known as Orchestration, the next component of a SOA. The standard language today for implementing Orchestration is Business Process Execution Language (BPEL). BPEL is a way of weaving together Web Services to form business processes (e.g. an Auto Loan, tasking an asset), which in themselves are Web Services.

With BPEL, we define orchestration logic as the business logic that sequences, coordinates and manages conversations among Web Services. Such orchestration logic can be as simple as a single two-way conversation or as complex as a nonlinear, multi-step business transaction with exception handling and compensation.¹⁰ Taking the Auto Loan example shown in Figure 5, the orchestration logic would include extracting the customer profile from an existing database, requesting the credit rating from an internal service, and then asking the two loan processors in parallel to process the loan application. If an exception occurs it is tasked to a human stakeholder using human workflow.¹¹

Implementing the industry standard for orchestrating business processes and Web services will not only speed the implementation and deployment of new integration projects, but will also reduce the overall cost of management, modification, extension, and redeployment of existing processes.¹² (See Figure 5 on the next page)

As we are all aware security is a big concern and you can't architect a system without thinking about security. Security is a driving force for the last essential component of SOA, *Services Management*, a key to SOA Governance. The main goal of SOA security standards is to provide a basis for interoperability among heterogeneous services across your enterprise.¹³

⁹ <http://www.oracle.com/technology/tech/soa/pdf/soa-suite-wp.pdf>

¹⁰ <http://www.oracle.com/technologies/integration/pdf/bpel-case-whitepaper.pdf>

¹¹ <http://www.oracle.com/technologies/integration/pdf/bpel-case-whitepaper.pdf>

¹² <http://www.oracle.com/technologies/integration/pdf/bpel-case-whitepaper.pdf>

¹³ http://www.oracle.com/technology/products/webservices_manager/pdf/Oracle-SOA-security-whitepaper-Jan08.pdf

continues on page 9

SOA Essentials

Continued from page 8.

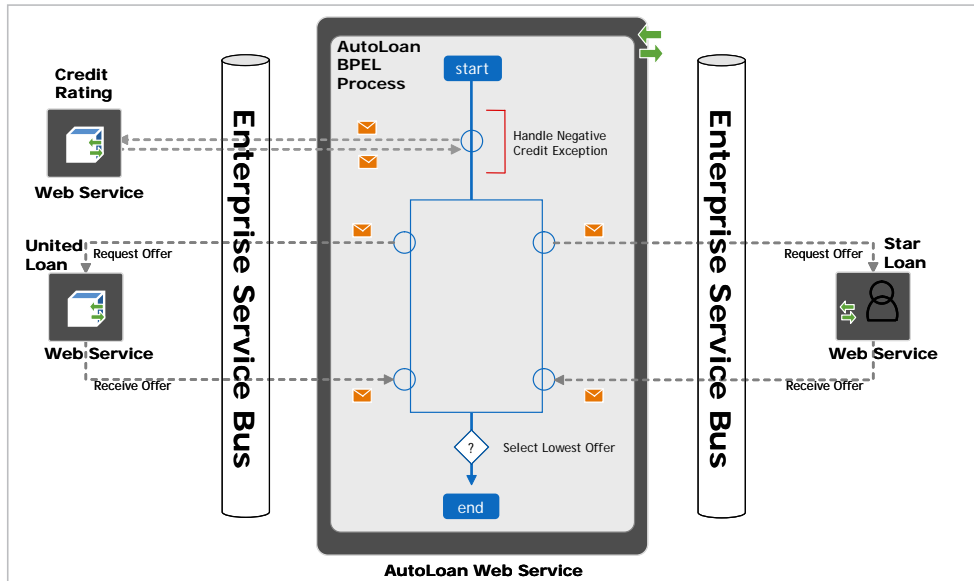


Figure 5: Sample BPEL Process from an Auto Loan

Until recently, the burden of Web Services security rested with the implementer. Developers would code security into their services creating environment “silos” that were difficult to manage and costly to maintain. SOA deployments have become more and more complex, creating additional challenges that developers alone cannot meet anymore, such as cryptographic data protection, identity management, and governance.¹⁴

Policy Decision Point (PDP) for multiple Web Services and the ability to enforce these policies at runtime using Policy Enforcement Points (PEP) as illustrated in Figure 6.

In summary, SOA helps bridge the gap between the stove-piped heterogeneous data systems by leveraging standards. There are several essential components of a SOA which (i) allow you to expose your code or legacy system with Web Services (ii) connect, transform, and route these service with an ESB (iii) orchestrate services using BPEL and (iv) be able to secure and manage these services.

Vendors, such as Oracle , provide complete, integrated standards-based SOA Suites that enable building a SOA and deploying web services to your chosen middleware platform.

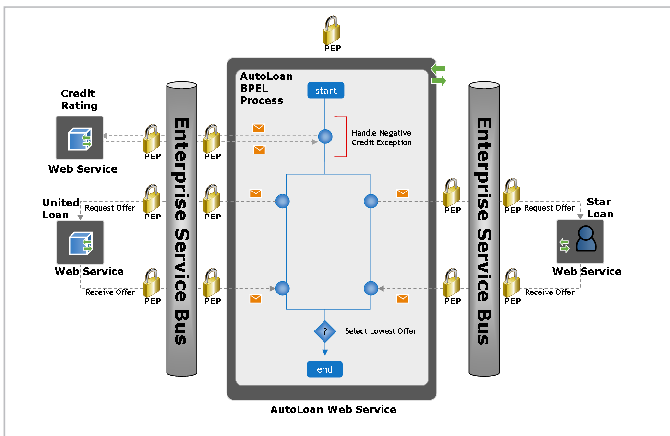


Figure 6: Sample BPEL Process with Policy Enforcement

Declarative security solves this problem and allows a service implementer to worry about the implementation and security personnel to worry about security. Using standards based compliant solutions for security allows for a centrally defined

About the Author

Robert Glass is a Principal Solution Architect at Oracle’s National Security Group. His focus is on Service Oriented Architecture within the Intelligence Community and he has worked within the community for over 15 years. He holds a M.S in Software Engineering from George Mason University.

Author Contact Information

Robert Glass
Email: robert.glass@oracle.com

¹⁴ http://www.oracle.com/technology/products/webservices_manager/pdf/Oracle-SOA-security-whitepaper-Jan08.pdf

¹⁵ <http://www.oracle.com/technology/tech/soa/pdf/soa-suite-wp.pdf>

SATISFY

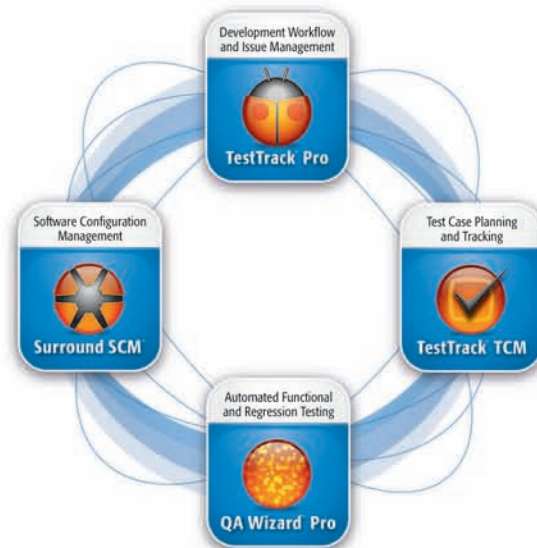
Your Quality Obsession

Software quality and reliability are mission critical. The size, pervasiveness, and complexity of today's software can push your delivery dates and budgets to the edge. That means rigorous quality discipline, and collaboration between development and QA teams has never been more important.

With offices worldwide and over 8,500 customers, Seapine Software is the leading provider of integrated quality-centric application lifecycle management (ALM) solutions. For over 13 years Seapine's easy-to-use, award-winning tools have helped companies achieve success by streamlining communications, improving traceability, and making development and QA teams more productive.

Satisfy Your Quality Obsession.
www.seapine.com/gsa

 **Advantage!** GSA Schedule 70 Contract GS-35F-0168U



Integrated ALM for a Quality-Ready Advantage

SOA: Keys to Sustaining the Transformation to a Service-Oriented Architecture

By Andrew Gordon, Unisys Federal Systems

Today, it's nearly impossible to look into information technology in the federal government without bumping into service-oriented architecture, or SOA. SOA is an IT strategy for delivering business capability. SOA can help you achieve business agility, bringing the government and military the possibility of creating an information sharing environment that can:

- Evolve existing disparate, unconnected, stove-piped systems and processes into re-usable services, creating an environment where services can be rapidly assembled, creating new applications to support changing mission requirements
- Increase the speed at which information and services can be securely shared so others can benefit, including unanticipated users
- Securely interconnect people and systems, independent of time, location, and organizational boundaries improving federal and military situational awareness, and significantly shorten decision-making cycles

Sound Easy? Think Again!

Service-oriented architectures can be implemented enterprise-wide, at a departmental level or even within a single application suite. An enterprise-wide SOA initiative can achieve closer alignment of business and IT throughout the application lifecycle, providing benefits well beyond pure technology. If properly orchestrated, enterprise-wide SOA initiatives can bring formality to processes, enhance communication between departments and organizations, and give all stakeholders an increased awareness of each other's projects and initiatives.

But enterprise-wide SOA initiatives are not easy and typically require both a business and IT transformation – transformation from traditional governance and organization models to an enterprise approach that enables greater agility. This enterprise approach affects business and IT stakeholders causing them to look outside their immediate areas of responsibility and partner across the enterprise to define the strategy and roadmap to achieve their objectives.

Three elements need to be in place for an enterprise-wide SOA to be successful.

1. A prioritization process for requirements that emphasizes enterprise priorities in lieu of departmental priorities

What is SOA?

OASIS (the Organization for the Advancement of Structured Information Standards) defines SOA as the following:

A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

Web services, popular today, is one of many technologies that helps organizations realize the promise of reusability and interoperability if implemented within the constructs of SOA, but in itself is not SOA.

2. A highly transparent, participative governance process comprised of all stakeholders
3. An unbounded commitment of an executive sponsor to steadfastly support this organizational transformation

Setting Priorities

When implementing an enterprise-wide SOA initiative, business and IT alignment occurs in the course of requirements prioritization and budgeting. Requirements for services in an enterprise-wide SOA initiative are determined and funded according to the priorities of the enterprise as a whole, rather than those of departments. This forces alignment of business with IT ensuring the goals of SOA are aligned with organizational objectives.

Achieving concurrence on enterprise-wide SOA priorities requires participatory governance and communication processes, especially greater interaction with the business lines. Active participation from the business lines and IT, with possible input from an IT services vendor is essential. When this happens, information, requirements, and ideas flow more freely. This promotes greater awareness of everyone's projects and initiatives to fully realize the benefits of an enterprise-wide SOA initiative.

continues on page 12

SOA: Keys to Sustaining the Transformation to a Service-Oriented Architecture

Continued from page 11.

To ensure a long-term outcome of an IT environment equipped with sufficient service-oriented capability to deliver on business requirements, it's necessary to properly prioritize and budget between IT and business requirements. It can become quite challenging to ensure the sound service-oriented IT infrastructure needed to support an expanding inventory of shared services with the clear value new end user capability delivers. To meet this challenge, the Enterprise Architecture (EA) function needs to work closely with both IT and the business lines during the requirements phase at project inception.

Since enterprise-wide SOA initiatives are broad and often inter-agency in scope, the EA function must forge relationships with the business lines to ensure that new services align with the business strategy. It is essential for the EA function to incorporate business process groups and business lines. Including the business lines promotes more interactive relationships between the business lines, IT and the IT services vendor. The EA function has the responsibility and opportunity to communicate the business value of a comprehensively designed and delivered shared services approach.

Governance

Formal, transparent and precise application lifecycle processes lay the foundation for a successful enterprise-wide SOA initiative. Sound governance begins with a strategic plan that includes the business goals. These goals, in turn, can be transformed into IT requirements with clear line of sight from business goal to IT requirement, followed by high-level design specifications through testing, deployment maintenance, and application end of life.

The IT organization typically manages the processes in an enterprise-wide SOA initiative, including requirements definition and prioritization with the business lines. Typically, representatives from each stakeholder organization participate in the requirements definition and prioritization process. An IT services vendor should be a managed partner within these processes.

To ensure uninterrupted support of an enterprise-wide SOA initiative, a virtual team convenes as necessary to discuss issues and gain consensus on strategies to eliminate the issues. This virtual team is led by a project director and is composed of representatives from each stakeholder entity. The IT services vendor may need to participate in these meetings, depending

on their organizational role. The project director collaborates with this team to understand issues such as barriers to participation, significant process breakdowns, and works with the team to gain consensus on mitigation strategies, plans, and new enterprise-wide processes. The project director is typically responsible for ensuring that barriers and gaps are identified and remedied in a timely manner.

Empowering this virtual team to develop and implement policy, process and standards is critical. During the early phases of an enterprise-wide SOA initiative, the team will need to meet more frequently. These meetings are essential and independent of the deployed application lifecycle methodology. For instance, imagine the issues that come up when a catastrophic problem occurs with a service consumed by one or more mission-critical business processes. What if it is an inter-agency process? Who do you notify first? What is the process for notifying stakeholders in a timely manner? Who else has to be notified and how quickly? Is this process repeatable by participants in the enterprise-wide SOA initiative? This is just one example of the kinds of issues this team of stakeholders will discuss and solve.

Executive Commitment

Unbounded executive commitment (AKA executive sponsorship) to the creation of an enterprise-wide SOA is a vital ingredient if an organization intends to transform itself from traditional governance and organization models to an enterprise approach that enables greater agility.

A key responsibility of being the executive sponsor (or their delegate) is to be the final decision making authority when the participative governance process reaches a stalemate during a task. The stalemate may occur during requirements prioritization, or there could be a disagreement on the timing for delivering new capability to customers. It is the executive sponsor's responsibility to arbitrate when impasses occur.

Executive commitment must be introduced well ahead of project kickoff, followed closely by establishing formal, precise, and transparent processes to manage the application lifecycle across the enterprise. Creating strong, transparent processes will make not only a strong SOA initiative, but a stronger and more effective organization on a larger scale.

About the Author

Andrew Gordon is Director, SOA and Open Source

continues on page 13

SOA: Keys to Sustaining the Transformation to a Service-Oriented Architecture

Continued from page 12.

Solutions with Unisys Federal Systems, based in Reston, Va. He is responsible for leading the SOA and open source business vision for Federal Systems, shaping the Federal strategic SOA and open source solutions portfolio and helping federal government clients harness the power of SOA and open source as flexible, reliable, secure and cost effective options for business critical software.

Andrew has 18 years of success at leading and pioneering enterprise wide shared services initiatives and bringing to market service-oriented infrastructure products at Compuware, IBM, Mercator, and Butterfly.net. Andrew built and led

enterprise wide Shared Services (SOA) organizations, exponentially increasing the rate at which organizations could respond to new and changing market requirements while simultaneously delivering a massive reduction in IT cost for itself and its customers.

Author Contact Information

Andrew Gordon

Email: Andrew.Gordon@unisys.com

Phone: 703-439-5491

Jobs at DACS

WANTED SOA DESIGNERS!!!

Experienced software developer specializing in Service Oriented Architecture (SOA). Knowledgeable and experienced with Enterprise Service Bus. Experienced with large scale DoD architectures like Technical Implementation Architecture, Net Centric Service Architectures and IPV6.

DACS is looking for someone with 8+ years of experience, familiar with or knowledgeable in the following areas:

- SOA
- Web Services
- XML
- Pub/Sub Architectures
- DoD Architectures
- TIA (Technical Implementation Architecture)
- Programming in JAVA, C++, C # (C SHARP)

Work Environment:

- R&D Community
- Rome, NY
- Working with the customer
- Leading a small R & D team

Interested candidates should send resume and cover letter to soa-jobs@dacs.dtic.mil





The DACS Gold Practice Initiative:

- Promotes effective selection/use of software acquisition & development practices
- Defines essential activities/benefits of each practice
- Considers the environment in which each practice is used
- Addresses the timeliness of practice benefits
- Recognizes interrelationships between practices that influence success or failure
- Contains quantitative and qualitative information
- Includes discussion forums, case studies and survey opportunities
- A continually evolving resource for the DoD, Government, Industry and Academia
- Free to use/free to join

Current Gold Practices:

- Acquisition Process Improvement
- Architecture-First Approach
- Assess Reuse Risks and Costs
- Binary Quality Gates at the Inch-Pebble Level
- Commercial Specifications and Standards/Open Systems
- Ensure Interoperability
- Formal Inspections
- Formal Risk Management
- Goal-Question-Metric Approach
- Integrated Product and Process Development
- Model-Based Testing
- Plan for Technology Insertion
- Requirements Management
- Requirements Trade-Off/Negotiations
- Statistical Process Control
- Track Earned Value

Learn More About the DACS
Gold Practice Initiative:
<http://www.goldpractices.com>

Become a Registered Member!
http://www.goldpractices.com/registration/register_step1.php

Data & Analysis Center for Software
P.O.Box 1400
Rome, NY 13442-1400

<http://iac.dtic.mil/dacs>



Communication and Trust in a Net-Centric Community Of Communities

By Kelly Emo and Chris Brown, Hewlett - Packard (HP)

Technology is only half the story in building a net-centric service infrastructure in an organization with the Department of Defense (DoD's) unique requirements. Of equal importance is the organizational half of the equation—the communication and trust necessary to ensure that individual communities of interest, each focused on its own critical mission, are willing to contribute to the pool of shared services and take advantage of what other communities contribute.

The DoD's Net-Enabled Command Capability (NECC) is exploring technology for a Federated Development and Certification Environment (FDCE) that exists on the government information grid and addresses the challenges of developing and certifying net-centric services.

The purpose of the FDCE is to “provide the policies, processes, and infrastructure that allow services to be progressively refined, tested, evaluated, and certified in increasingly rigorous situations leading to an operational deployment.” As envisioned, the environment won't be utilized, controlled, or operated by a single organization, but will be a virtual environment made up of service providers, testers, evaluators, certifiers, and operators across the entire DoD. The end result will be to facilitate ongoing interaction and collaboration among organizations.

Communication: The service governance repository

In the commercial sector, where the structure of large enterprises mirrors in many ways the organizational hierarchy of the DoD and its branches and sub-branches, it is commonly agreed that federation of service metadata and associated policies, both run-time and design-time, is the key to service compliance and communication in net-centric infrastructures, which the commercial IT world calls service-oriented architectures (SOAs).

A key communication factor in SOAs is federation of information about services in a service governance repository where potential consumers can “shop” for services. There are two ways of going about federation. One is the federated query, which goes out from a server in the federation to multiple peers when an organization requests a new service, polling resources on local servers and finding the right service to fill the request. The other is replication, which involves service metadata to be either pushed to a master registry from one or more slave registries—or pulled from them—making it the repository

of all sharable services. In industry, some businesses employ one model and some use the other, depending on the unique requirements of their businesses.

Federation—but what kind?

One question, then, facing the DoD as it undertakes its Net-Centric Enterprise Services (NCES) effort, is which federation model—federated query or replication—best fits the business of the DoD. How can the DoD best federate service information across its distributed environment to foster compliance with service policies, help communities of interest accomplish mission objectives, and cost-effectively increase the pool of resources available across the organization?

The answer to that question lies in the unique requirements of DoD communities of interest (COIs), which in turn are driven by the nature of the DoD's primary COIs—the United States Army, Air Force, and Navy. Tasked with national security and war-fighting, these service branches and their constituent agencies are naturally inclined to distrust services provided by other branches and equally inclined to want to maintain tight control over their own resources. To further complicate the picture, capabilities within a COI are commonly created by competing systems integrators, who are looking to gain the largest work share of a program. This leads to further distrust of services provided by someone else. So every user community wants tight control over which services are public and which are private.

This is understandable, given the importance of their missions and given that control of information is an inherent part of those missions. Speed is another key factor. Most of the missions performed by service branches require rapid response and execution, and users will need to be able to search service information by both metadata and content.

Cost is also a consideration, as all DoD agencies operate under strict budget oversight. And since each COI uses its own tools for testing, development and maintenance of services in the production environment, the ability to integrate heterogeneous, distributed environments is a must, and COIs will need to be able to share vocabularies and taxonomies. This also leads to the need for the DoD to base its solutions on open standards—such as the ones controlled by OASIS (Organization for the Advancement of Structured Information Standards) and W3C (The World Wide Web Consortium)—which allow each of the COIs to choose the

continues on page 16

Communication and Trust in a Net-Centric Community of Communities

Continued from page 15.

tool that works best in their individual environment but yet ensures that others outside of the COI can interoperate with services being provided.

For several reasons, replication is a better fit than federated query for the FDCE. One drawback to the federated query approach is that performing a large query and expecting any valuable data back is unrealistic. Another is that search chaining, the most common technique, requires too much bandwidth and doesn't scale, as illustrated by the fact that Microsoft Active Directory has abandoned chaining and now uses replication. One of the myths associated with service chaining is that there is no need to keep a list of servers to search. On the contrary, it is important to understand the servers that are taking part in the federation so that they can be contacted directly, rather than over multicast protocol, which is blocked across firewalls, and to prevent an infinite search loop, which is created by two servers continually calling each other to find data.

But perhaps the biggest drawback for DoD purposes is the problem of opening COI systems to accept bots to search for data. Defense security requirements prevent any kind of widespread or indiscriminant access to systems, and developing a security access system to operate on top of the FDCE would be prohibitively time-consuming and costly. Security is certainly one of the largest roadblocks to a federated query within the DoD. When a query is started by a user, how is that user authenticated and authorized across COI boundaries? Currently in the DoD there is no central user directory to authenticate and authorize users, and the NCES Security working group is still hammering out the way in which an attribute-based authorization mechanism will be implemented across the DoD. To get around this security constraint, service providers should provide a "public" UDDI registry for users outside the COI to discover services. This allows the provider to share information about a service without having to worry about the authentication and authorization of user access. Once potential consumers discover the service, the consumers are able to contact the provider to create contracts and ensure that the service can be correctly provisioned for use by the consumer in the operational environment. Lifting the authentication and authorization requirements allows consumers to more easily discover services and lessens the security burden on the providers.

Replication, on the other hand, offers as many advantages

as federated query does disadvantages for sharing data across COIs. It allows COI service providers to decide which services they will make public and which they will keep private. This is important because it allows the providers to maintain control of the data and services that they provide, which helps them retain the feeling of being in control of their own information. It is also important to understand that not every service that is created will be a sharable service. COI Service Providers may create services that are simply infrastructural services specific to the net-centric environment of the service provider, or services that are relevant only to the COI and not candidates for sharing with other COIs. They can also tag data with their own vocabularies and share those vocabularies with other COIs. By maintaining a central COI repository, each COI can control the lifecycle of services and the policies surrounding them. And control is kept in the COI while public services are exported to a "DMZ" registry—which eliminates the need to have holes in the firewall—to be pulled by other communities, including NCES.

Simplicity is another benefit of using replication. If information is published to NCES and then replicated down to the Army registry, any COI within the Army can replicate data from the Army registry to get both Army enterprise and DoD-wide (NCES) services. This means the lower COIs only have to worry about maintaining relationships with a minimal number of master servers to gain information about services across the entire DoD. Using federated query, a list of servers that participate in the federation would have to be maintained.

Another critical requirement of any federation mechanism is that it be based on open standards to make sure that organizations outside of the DoD, such as the civilian government and commercial entities, can communicate with the DoD without having to conform to a proprietary DoD standard. The replication mechanism outlined in this paper meets that requirement. It is based on the UDDI (Universal Description Discovery and Integration) version 3 specification, using the Inquiry API and the Subscription and Notification API. Since UDDI is an open standard and both the DoD and industry have agreed that it is the interoperable standard for sharing web service information, it is a natural fit in any federation scheme. Federated query mechanisms, on the other hand—such as peer-to-peer or a federated search service like NCES's Federated Search capability—locks consumers of

continues on page 17

Communication and Trust in a Net-Centric Community of Communities

Continued from page 16.

the service into a non-standard interface. For example, in a peer-to-peer scenario, everyone has to agree to use a particular protocol, such as Gnutella, JXTA or Limewire, and nothing else because those protocols are not-interoperable. They also use non-standard IP ports and usually create firewall issues with searches from outside of a community. Even if a SOAP (Simple Object Access Protocol)-based web service specification were created in the DoD, that specification would need to be implemented by everyone who wanted to do business with the DoD—making the specification proprietary and not within the spirit of a net-centric environment.

One final advantage of replication over federated search is within the tactical environment, where bandwidth availability is limited at best and not reliable. In situations such as this it is important to be able to search the local network for service information rather than having to depend on directories outside the local network. Replication allows service information to be stored within the local network, making it easy for tactical users to discover services they need at a moment's notice.

So a combination of local repositories within COIs, public repositories outside the firewalls, and a central repository at the DoD level, all based on the UDDI interoperable service discovery standard, meets the communication needs of NCEC and the DoD as a whole. What is important to note here is that even though replication appears to be advantageous over a federated query, that does not necessarily prevent COIs from implementing a federated query. For example, a federated search capability could be created to search across registries within a given COI. This would allow different stakeholders within the COI to remain in control of what is published within their own registries, but also allow consumers within the COI, which are easily authenticated and authorized, to search within the registry.

Trust: Two complementary systems of record

Once services are able to be discovered by consumers outside of a community of interest, the question becomes why would potential consumers trust the service? Trust is quickly becoming the biggest road block in the Net-Centric vision of the DoD. One reason for the lack of trust is that no longer do developers know each other on a face-to-face basis. Since developers do not know each other, a common phrase emerges that is echoed in "I can create a better service". To get around this type of attitude and others to facilitate trust

across communities, it is important to create a collaborative environment so that developers, quality assurance testers, architects and other stakeholders have all the information about a service they need. This means that providers need to not only provide information about what the service does, but to "air their dirty laundry" about what happened in the quality assurance testing of the service, both on a functional and performance level.

The requirement for trust can be met by combining the consolidation of test management information with the federation of services data. This will require two repositories, but to achieve the stated goals of the FDCE, the two will have to be integrated to advertise the results of testing to consumers outside each COI and at the same time ensure service compliance across multiple COIs (See Figure 1).

As the system of record for services, the services repository needs to provide a number of tools for the testing community. It has to:

- Notify testers of services to be tested or changes to services
- Allow testers to run policy-compliance reports for a service
- Run impact and dependency analysis reports so that testers can understand if other services need to be regression tested because of a change

As the system of record for testing activities, the testing repository allows COI members to understand the progress of the services through the testing lifecycle. The repository creates a collaborative environment between requirements owners, material providers, and the quality assurance team:

- Requirements are managed directly through the repository or pulled from a requirements management tool
- Material providers publish test cases and results of test cases and associate those test cases to the service and requirements
- It provides visibility into requirements coverage
- It makes it possible for testers to manage defects and make sure that they are assigned and closed
- It enables everyone to understand the impact of change on the testing cycle as requirements or service change

continues on page 18

Communication and Trust in a Net-Centric Community of Communities

Continued from page 17.

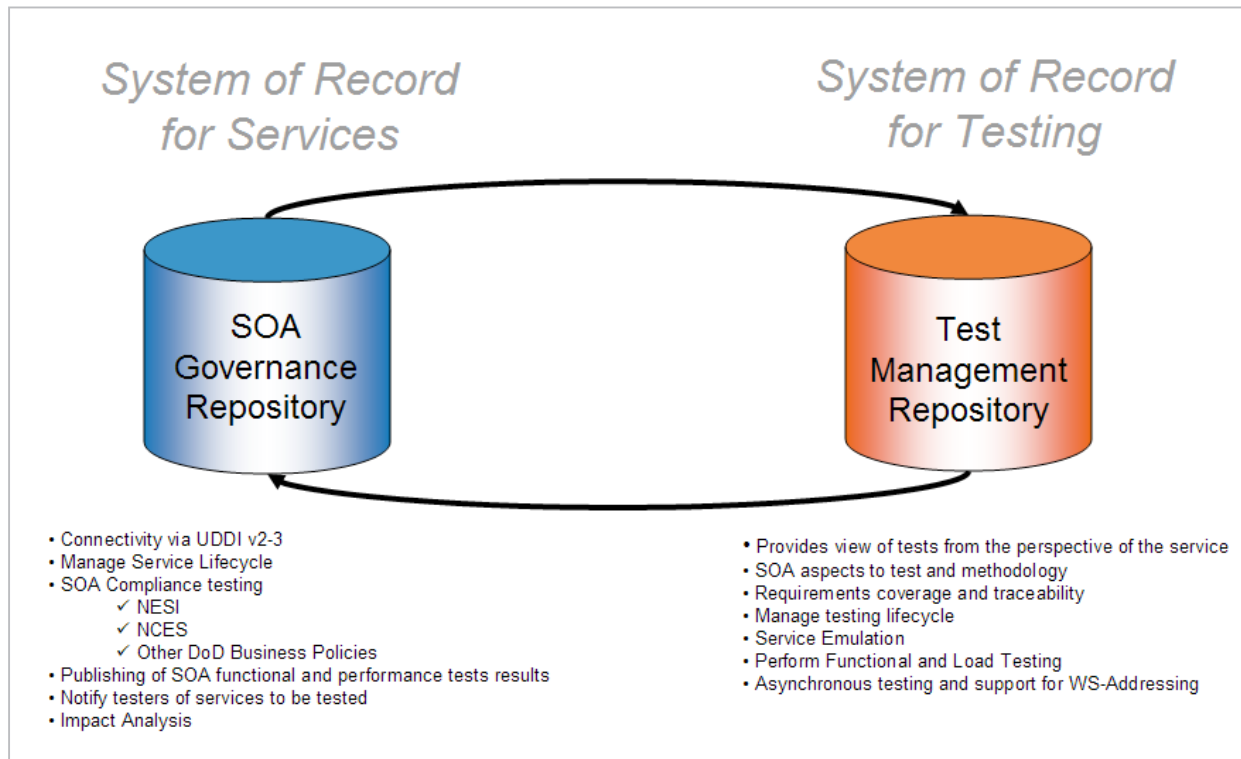


Figure 1: System of Record for Services and Testing

Since the test repository is able to connect with other systems, such as requirements management tools, it allows material providers to maintain their own system while providing the program office visibility to the activities during the testing cycle.

The solution in action

With information on services federated through replication and the governance repository integrated with the test management repository, the two organizational elements of successful net-centricity are in place. Service providers and service users can communicate with one another about the need for services, the availability of services, and the policies that govern services, and they have a foundation for the trust that is essential to make the sharing of services work in the DoD's unique environment.

Potential service consumers can use the governance repository to discover not only the endpoints of the service, but also the metadata around the service. They can pull the test execution reports from the repository to understand how

the service was tested and whether it passed all the functional test cases and run policy-compliance reports to see whether the service conforms to policies, such as WS-I Basic Profile, Net-Centric Enterprise Solutions for Interoperability (NESI) and NCES. Based on those results, consumers can decide to utilize the service or search for another applicable one. Finally, they also have the opportunity to request the use of a service based on one or more service level objectives (SLOs) that the provider has advertised for the service so they know what to expect from the service and whether it meets their mission parameters.

Creating an environment of trust between consumers and providers will be the first step in creating a SOA that is able to cross organizational boundaries within the federal government. The combination of SOA governance, to advertise service compliance and SLO information to potential users, and proper management of the quality assurance cycle is necessary to facilitate trust across the different communities of interest.

continues on page 19

Communication and Trust in a Net-Centric Community of Communities

Continued from page 18.

About the Authors

Chris Brown is currently a Solution Architect focusing on SOA and Net-Centricity in the Federal Government. While at HP Chris has worked with programs across the Department of Defense (DoD) helping them to implement SOA solutions in both the tactical and business environments of the DoD. These solutions include the governance of services, ensuring service quality and conformance to policy, and the management of services in the runtime environment. Prior to working for HP, Chris worked at Booz Allen Hamilton as the technical lead of the NCES Service Discovery capability. Throughout Chris' career he has worked on the development of systems for large telecom and government customers, ranging in technologies from legacy mainframe to the development of artificial intelligence applications.

Kelly Emo, Director of SOA software product marketing at Hewlett-Packard, is a seasoned executive with many years experience in high-tech, mostly in software infrastructure and networking technology areas with a focus on product management, business development and product marketing. Her first several years with Hewlett-Packard were focused on product management and strategy around HP-UX networking

technologies, distributed computing middleware, network and systems management and application development tools.

Kelly helped to launch the HP Enterprise Java program owning product management for Java for HP-UX and also took a break from product management and planning for a few years to manage corporate communications for HP OpenView and HP Telecommunications in the early '90's. Following her initial career with HP, in the "dot-com" days of the year 2000, Kelly joined a newly funded startup in the Software as a Service space as Director of Business Development and later moving into the role of Director of Marketing at Jamcracker, Inc. For the last four years, Kelly has worked at BEA Systems, responsible for launching BEA's Enterprise Service Bus and more recently as Director of Integration Product Marketing, spanning integration technologies including EAI, ESB and BPM.

Author Contact Information

Chris Brown

Email: chris.daniel.brown@hp.com

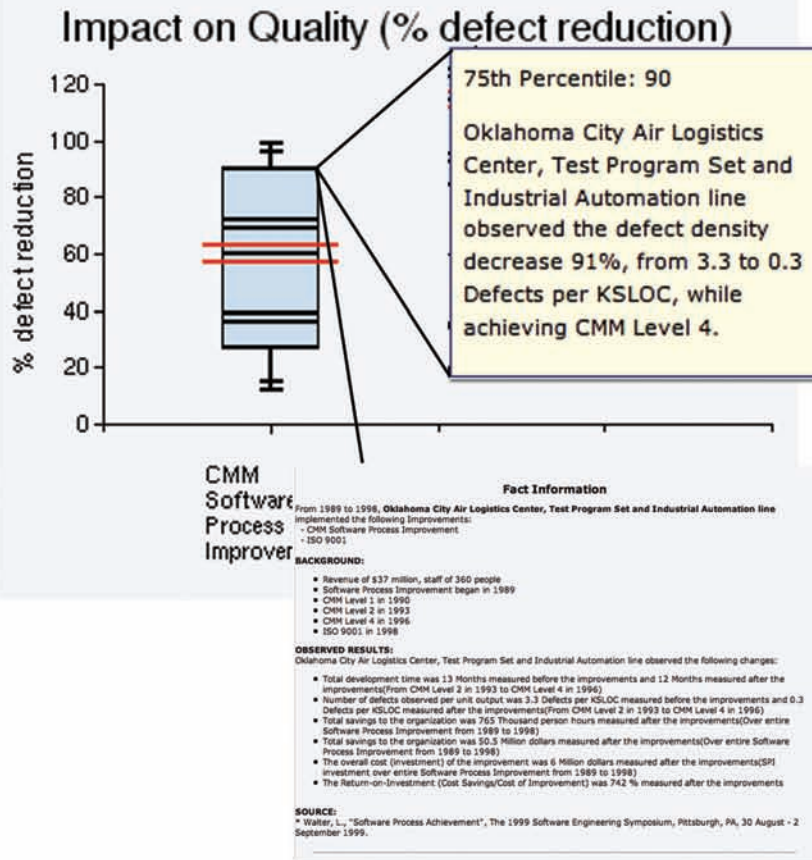
Kelly Emo

Email: kelly.emo@hp.com

Are you getting
All you can from
your software investment?



The DACS ROI Dashboard



Technologies Covered:

- SEI CMM/CMMI
- SEI Team Software Process (TSP)
- SEI Personal Software Process (PSP)
- Inspections
- Reuse
- Cleanroom

And Many More!

Graphs Showing Impact Of Software Technologies On:

- ROI
- Productivity
- Quality

Summarizes Facts From Open Literature

Data & Analysis Center for Software
P.O.Box 1400
Rome, NY 13442-1400

<http://iac.dtic.mil/dacs>



Access the DACS ROI Dashboard!

<http://www.thedacs.com/databases/roi/>

IBM's SOA Journey

By Lance Walker, IBM

Abstract:

With IBM's focus on being the Premier Globally Integrated Enterprise, SOA is ideally positioned to enable a wide range of critical strategic business capabilities. SOA's emphasis on the integration of business and IT naturally supports IBM's IT organization goals to be business driven, and not led by technical solutions. This paper describes the business drivers, business and IT value (to IBM internal), and a brief history of the SOA journey within IBM. It also illustrates the organizational challenges and governance needed to ensure the adoption of SOA as the key enablement technology for global integration. Lessons learned along the way are highlighted as examples of what went right and what could have been avoided - in addition to challenges that remain.

Introduction – the Business Focus for SOA

Several longstanding IBM business drivers support the need for an IT framework based on SOA. Focusing on customer needs has been a key differentiator for IBM since its inception, and responding to evolving market conditions requires constant innovation and change to deliver key business capabilities that can overcome competitive threats. Compliance with government regulation is one of life's constants, and effective cost management in today's market is essential for business survival.

These business drivers compel us to rapidly deliver new business capabilities that are highly responsive and more flexible to adapt to change. These capabilities must also be produced at lower cost and efficiently leverage existing resources. IBM's partners are a major part of its business model, and improved interactions via services must be available that suit partner needs for them to rapidly board onto our processes (for both sales and procurement). Complex operational end-to-end processes, originating from a patchwork of independently delivered worldwide information systems, are being simplified with an SOA integration platform to improve business integrity and coordination.

Legacy applications continue to deliver key business functions, but their funding for modernization and migration to newer technologies often continues to receive a lower priority in comparison to the need for other new business capabilities. SOA is being used to provide incremental approaches to their migration, as well as to leverage and re-factor their assets in a decoupled fashion via services and workflow to extend their ROI. Although some might think that IBM runs on a completely homogeneous IT environment that is based

exclusively on IBM products, it does have several third party products that need to be effectively integrated into the IT landscape by exposing services interfaces.

A Brief History of SOA within IBM

Like many companies, IBM's initiation into SOA began in small steps experimenting with Web Services. Web Services first surfaced as a bottom-up movement among a small number of curious practitioners working on internal projects, in addition to IBM's innovation team that is responsible for exploring new technologies and demonstrating their practical use. In the 2001 – 2002 timeframe, it was not really about SOA yet, but more about services that had not yet found their way into key business applications. During the period of 2003 – 2004, services were being deployed by several mission critical projects, and a Web Services Guidance Council was created to define the start of what is now known as SOA governance (e.g., an internal Web Services standard), to promote services interoperability and delivery consistency. The use of process modeling as a precursor for workflow development was also a key focus in this period, to help ensure that IT was process driven.

A wide range of activities with an increased top-down and enterprise-wide systematic emphasis occurred during the timeframe starting in 2005 and through 2007. SOA was directly incorporated into Enterprise Architecture and Enterprise Architecture governance. Pulse metrics were formalized to measure the rate of SOA adoption, and CIO mandates were first issued requiring a portion of project development funds be assigned to SOA features. The Web Services Guidance Council was transformed into the SOA Guidance Council, with the recognition that most of its newly formed workgroups (to define technology recommendations and guidance) had the broader scope of SOA (e.g., SOA Operations Management and SOA Security). Registries and Repositories were deployed to support SOA governance, and ESBs (Enterprise Service Bus) played a significantly increased role for the use of SOA as an integration platform.

The year of 2008 is the start of making SOA self-sustaining, whereby SOA is automatically being used and readily funded for maximizing its business value to improve IT agility and deliver projects with reduced cost and schedules.

IBM Internal IT Benefits Derived from SOA

IBM internal IT direction to be business and process driven is extremely well complimented with SOA's strengths to

continues on page 22

IBM's SOA Journey

Continued from page 21.

directly couple business intent with technical solutions. SOA methodologies to capture business goals and processes (as is and to-be), as a basis for defining technical activities, have produced solutions that more readily satisfy business objectives. A focus on industry standards for modular processes and IT functions allows a consistent approach for the delivery and consumption of enterprise-wide reusable assets, which are the most obvious and easily derived sources of SOA business value. New projects can take advantage of already deployed services that have been designed with an enterprise appeal (e.g., tax calculation) or for a specific business unit, and avoid the cost of fully developing these functions. When these services based assets are defined in the context of an Enterprise Architecture, a more cohesive picture emerges for the integration of disparate architectures across multiple lines of businesses.

SOA has been acknowledged as IBM's fundamental internal IT integration platform to deliver incremental rapid deployments of new and updated business functions. Beginning with a premise of using SOA to integrate multiple systems and information sources has provided an accessible framework for making IT decisions involving complex interactions. Composite Business Service concepts to choreograph existing services assets and human interactions enable the assembly of new 'applications' more efficiently at reduced cost. Visual modeling techniques provided by an integrated tool suite allow development processes to compose IT functions in less time and more accurately. The resultant model's self-describing characteristics promote a simpler understanding of an IT function, which makes it easier for new developers (to a project) to assume the responsibilities of in-progress or already delivered 'code'.

Sense & Respond environments enabled with SOA provide a consolidated view of operational processes that are deployed with combinations of custom code, third party packages, or IBM WebSphere middleware products. Centralized dashboards with views of business events emitted from multiple sources have allowed business analysts to quickly troubleshoot process bottlenecks and perform the right corrective actions. When internal functions are created with SOA concepts that include industry standards and decoupling, they can be more easily outsourced with less disruption to their consuming applications. Several internal business functions, such as those associated with HR, travel, and semi-conductor testing, have been outsourced to external vendors and have successfully leveraged SOA. Services interfaces have facilitated the interactions of employees or internal applications with these

now remotely located capabilities.

SOA Enablement Activities

The CIO's office has funded the Internal Smart SOA Initiative to jumpstart and nurture internal projects to include SOA content. Several activities have been highlighted in 2008 to increase the level of SOA embedding into IBM's IT culture and practices, and to promote a wider use of key SOA technologies and products. Examples of these activities are:

Business Value, Metrics, and Reuse

- **Funding:** For 2008, the CIO's office has provided funding guidance that 10% of each business unit's development funds must be applied to the creation or reuse of SOA capabilities – such as Web services, ESBs, Workflow, and Monitoring. A project office has been established to follow the results and work with each of the units to assist them with their assessments.
- **SOA Business Value:** The qualitative and quantitative values associated with SOA approaches and features are being captured in the 10% SOA development spend management process. Although it has been relatively easy to provide qualitative value statements, such as "easier or more flexible to integrate and update", industry and IBM experiences have shown that it is much more difficult to derive quantitative, monetary values associated these types of statements. Nonetheless, guidance has been provided to help the business units calculate savings based on reuse, reduced development and maintenance hours, application sunsets, decreased operational expenses, and other factors.
- **SOA Pulse Metrics:** The number of in-progress and deployed services, ESBs, automated workflows, SOA enabled monitoring functions, are being tracked as indicators of SOA adoption progress. Measurements for services reuse and the associated quantitative business value (for each reuse) are also collected.

Technology Support

The Internal Smart SOA Initiative has the mission and resources for assisting projects to incorporate SOA features, products, and tools into their IT implementations, which mitigates perceived risks associated to new technology adoption. The following is a sample set of key project support activities in 2008:

continues on page 23

IBM's SOA Journey

Continued from page 22.

Activities to jump-start and nurture SOA projects


- Continue the original focus of starting with process modeling using WebSphere Business Modeler, and integrating the models with workflow tooling, such as either WebSphere Integration developer (to be deployed on WebSphere Process Server) or FileNet.
- **ESBs:** Leverage the WebSphere DataPower product that was deployed into the shared infrastructure at the end of 2007, and publish internal guidelines for a consistent implementation of WebSphere Message Broker within IBM
- **Monitoring:** Continue working with projects to incorporate WebSphere Business Monitor (provide business analysts with dashboards to view and interact with operational processes), and complete pilots for Tivoli Composite Application Manager for SOA (to monitor the IT operations of SOA functions is critical to manage SLAs (Service Level Agreements) and performance)
- **Alternate Service Formats:** Increase the use of REST (Representational State Transfer) style services, and standardize description metadata for their internal deployments
- **Service maturity and discovery workshops:** Continue SIMM (Service Integration Maturity Model) and SOMA (Service Oriented Modeling and Architecture) workshops, both well-established IBM Global Business Services practices, to facilitate an understanding of SOA maturity levels and to identify needed services for projects respectively.

SOA and SOA Governance in the Enterprise Architecture:

Enterprise Architecture:

IBM's internal Enterprise Architecture consists of a coordinated view of Business Process, Enterprise Information, Applications, and the Infrastructure – driven by a global business strategy that encompasses all lines of businesses. The Enterprise Architecture governance includes Guiding Principles and Metrics, Implementation Criteria, and Standards that are published by the CIO's office, but enforced through the individual line of business Architecture Review Councils.

SOA is directly incorporated into the Enterprise Architecture, which includes the Application and Information Blueprint for a view of IBM's internal strategic applications and information sources, consisting of a mixture of standard and SOA enabled applications and information sources – known as Service Components. The following diagram (See Figure 1) is the high level Enterprise Architecture Blueprint (similar diagrams are available for each of IBM's business areas). This is available on IBM's intranet, and is highly interactive – not a static drawing. Each of the information systems can be selected to provide more detailed information, such as worldwide coverage, business owners, supported business processes, and interactions

with other systems. The boxes with the icon, , in the upper left hand corner indicate information systems that expose their capabilities with services interfaces, and these are defined as Service Components. Service Components are designated to be shared by the entire enterprise or within one business unit. In addition to the standard information already mentioned for basic information systems, the Service Components can be selected to reveal information about what services they have, and which operations are provided by each service. The arrows connecting to each of the services can also be selected to understand which services and associated operations are of interest to the consumers of these Service Components.

One example of an Enterprise Service Component in this drawing is CCMS – the Central Customer Master System owned by the CIO's Enterprise Business Information group; it is designed to enable a single, worldwide view of customers for all business touch points. Another Enterprise Service Component is the Tax Engine, owned by IBM Finance, which provides a common, worldwide system to compute taxes in a standardized fashion that meets legal requirements and improves the accuracy of finance management and billing. The Web Identity Enterprise Service Component is the most successfully reused service within IBM, since it provides a common authentication and profiling method for all IBM's external Web sites. All of these Enterprise Service Components, and many others, are owned by organizations that have the mission and funding to provide functions across the enterprise – spanning multiple lines of businesses. They provide the best opportunity to discover and deploy enterprise-wide reusable services.

continues on page 24

IBM's SOA Journey

Continued from page 23.

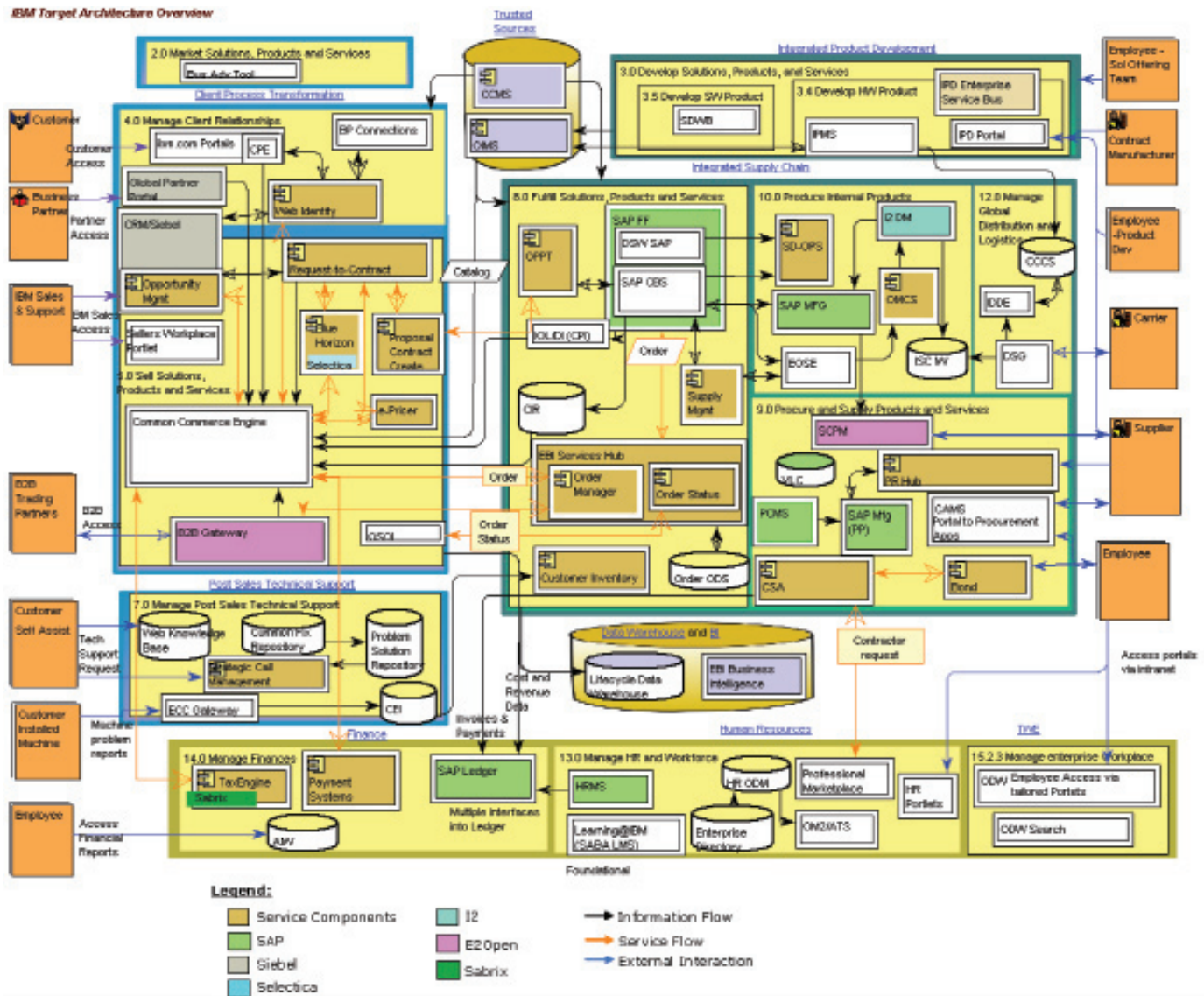


Figure 1: IBM's Target Architecture Overview

SOA Governance

As with the Enterprise Architecture overall, SOA has been directly incorporated into the Enterprise Architecture's governance process, with the objective of maximizing SOA's value for IBM's internal transformation efforts to be the premier Globally Integrated Enterprise. Adding a standalone model for SOA would have unnecessarily created additional work for initiatives and projects, in comparison to a seamless integration into existing governance processes that they already follow. Several Enterprise Architecture governance artifacts have been updated to include SOA related guidance or design

criteria for how and when to use SOA, including a new internal standard for developing Web Services to promote service interoperability and to define when services must be used for inter-application communications. One of the guidance mechanisms includes funding requirements from the CIO's office for each year. As mentioned earlier, the business units must devote 10% of their development funds to incorporate SOA features into their projects during 2008.

An asset repository to manage and encourage reuse is a key facet of SOA governance. SOA assets for internal projects, such

continues on page 25

IBM's SOA Journey

Continued from page 24.

as process models, services, and supporting code are stored in a common repository to maximize visibility for reuse potential. This repository was originally built with custom code, but has migrated to the Internal SOA Asset Community within the Rational Asset Manager instance that is designated as the common repository of choice for all reusable assets within IBM. It is being leveraged as the development-time repository, and is being integrated with the WebSphere Service Registry and Repository, which is positioned as the run-time registry to dynamically route service requests to service providers. The Internal SOA Community Asset Community is also being integrated with SAP's Enterprise Service Repository, to provide non-SAP development environments with views of a selected set of services provided by SAP.

Another important aspect of SOA governance includes the need for a SOA Center of Excellence. IBM has a virtual approach to this that includes collaboration among several entities: the CIO's Internal Smart SOA Initiative, the CIO's SOA Guidance Council, Business Unit Architecture Review Councils, and Global Business Service's Family Architects who manage internal projects for each of the major business units. This provides a combination of a top-down approach from the CIO's office, bottom-up with the Family Architects, and meeting in the middle with the business architects in their respective architecture councils.

SOA Lessons Learned

As a backdrop for this summary of a few of the lessons learned during the journey to imbed SOA into initiatives and projects, a couple of topics provide a level of perspective of the challenges faced by IBM's internal transformation efforts: 1) Composite Application concepts for a system of services that are replacing so-called 'monolithic' applications, and 2) organization impacts.

Composite Applications: As new functions are assembled from services that may be owned by multiple organizations and widely distributed within IBM's network, several factors must be considered that were considerably easier to manage when all functions were confined to a single application. Services owned by different organizations can have varying funding models, interoperability, life cycle management, event and logging systems, and service level agreements. This results in a greater level of inter-dependency among organizations for end-to-end management of operational processes that leverage multiple services.

Organizational Structure: IBM's brands (e.g., Software

Group, Services, Sales and Distribution, Server – Technology Group, IBM Global Finance) have responsibilities that are focused on their own unique towers, and their transformation strategies and activities are managed by vice presidents called the Business Transformation Executives. Spanning across these towers are business areas that are owned by the Process Transformation Executives, who provide enterprise-wide functions that are required by the Business Transformation Executives. These Process Transformation areas include the Integrated Supply Chain, Client Facing, IBM Finance, Human Resources, and Post Sales Technical Support, and provide the best opportunities to enable enterprise-wide reusable services, since they already have the mission and funding for these types of services.

IBM Internal Executive Considerations and Concerns for SOA Adoption

Technology is often the easier part of adoption, while organizational and funding issues are more difficult to overcome for larger enterprises – they are bound by long standing business and IT practices that preceded SOA. SOA is at times perceived as an "IT only" solution by several parts of the business. The business side of organizations is focused on delivering business capabilities, and less on how IT solutions are created. Business goals tend to be bound to relatively short-term financial cycles, whereas SOA solution benefits often have longer-term expectations. SOA business value is often not sufficiently defined or communicated.

Several difficult questions must be answered to address common services:

- Which organizations will own which common services?
- Who has the mission for needed common services that do not fit within any business unit's scope?
- How to manage the priorities of common services requirements from the business units?
- What is the enterprise-wide funding model for common services?
- What is the right enterprise-wide billing model for commonly shared services?
- How can new SOA infrastructure capabilities be funded without penalizing early adopters?
- Can SOA infrastructure components (e.g., ESBs) be centralized? Several organizations have private infrastructures, with less emphasis on shared infrastructures

continues on page 26

IBM's SOA Journey

Continued from page 25.

The following table profiles business related lessons as well as lessons learned for IT Management.

Business Related Lessons Learned	IT 'Management' Lessons Learned
<ul style="list-style-type: none"> It is critical to start with business needs and pain points, and avoid being caught up with the hype of SOA that leads with SOA solutions looking for a problem to solve 	<ul style="list-style-type: none"> SOA is not a stand-alone processes, it must be part of a comprehensive Enterprise Architecture that collaborates business unit architectures
<ul style="list-style-type: none"> SOA's strength of decoupling to 'hide' systems and changes behind services interfaces and workflow allow business value to be gradually realized through incremental deployments -- as opposed to 'big bang' projects 	<ul style="list-style-type: none"> A single, enterprise view of existing and future strategic SOA assets is required to minimize redundancy and increase sharing
<ul style="list-style-type: none"> Business subject matter experts are often extremely busy managing their operations, and are therefore reluctant to participate with IT activities. However, they are really needed at the beginning of SOA activities, especially during Service discovery workshops. 	<ul style="list-style-type: none"> Through this increased emphasis on reuse, SOA has encouraged IT sharing among business units
<ul style="list-style-type: none"> In addition to the reuse value of enterprise-wide services, it is also important to emphasize the SOA business value of increased flexibility and agility. Business ownership and funding models must be immediately defined for enterprise-wide services. 	<ul style="list-style-type: none"> An executive champion and a key architect for SOA are needed for each business unit
<ul style="list-style-type: none"> Business Unit Executive sponsorship for enterprise objectives is critical to avoid narrowly defined project scopes, which cannot meet enterprise-wide needs 	<ul style="list-style-type: none"> SOA Governance is required fairly early in the picture to adopt SOA
<ul style="list-style-type: none"> New investments or new funding models are required to maximize the development of services for reuse at an enterprise level. Existing silo IT funding makes it difficult to extend services for reuse by more than one business area. 	<ul style="list-style-type: none"> Create a SOA Center of Excellence (single or virtualized collaboration among several entities) as soon as possible, and leverage it to increase the awareness of involved technologies, reduce barriers to adoption, and implement metrics to understand the level of SOA penetration
<ul style="list-style-type: none"> Since one of the primary barriers to common services adoption is the affordability for potential service consumers, consider establishing funding for service consumers to support their migration to new reusable services 	<ul style="list-style-type: none"> Institutionalize SOA corporate guidance for the enterprise, which can include services standards, design criteria, and an ESB Reference Architecture
<ul style="list-style-type: none"> Cost recovery methods to fund new services has been somewhat successful by leveraging existing fund management models (e.g., service consumers paying for new development, boarding, and yearly support), but are not sufficient. An enterprise-wide acceptance of shared business value among Lines Of Business (LOBs) is required to move beyond simple cost recovery models for individual services. 	<ul style="list-style-type: none"> The individual Business Unit Architecture Review Councils are crucial partners to promote SOA and corporate guidance
	<ul style="list-style-type: none"> Implement an SOA repository to manage and to promote reuse of SOA related assets
	<ul style="list-style-type: none"> A highly interactive dialog among corporate groups, business units, and development organizations must be structured for an effective coordination of SOA strategies and activities across an enterprise. This can be facilitated by driving enterprise SOA adoption through collaborative top-down (CIO), middle (business units), and bottom-up (development organizations) efforts.
	<ul style="list-style-type: none"> Having the right skills is extremely important to the SOA's success, so early investments must be made in SOA training to minimize slow SOA startups and SOA avoidance
	<ul style="list-style-type: none"> To maximize SOA's value, a change to its innovation-oriented development culture must occur to accept reuse and consider innovative approaches to reusing existing SOA assets (e.g., services)
	<ul style="list-style-type: none"> Don't forget to enable the infrastructure for SOA, and start this involvement with infrastructure teams near the beginning of project development processes to plan for cost effective deployment options

continues on page 27

IBM's SOA Journey

Continued from page 26.

Summary

IBM's initial focus on simple Web Services has expanded into the full range of SOA capabilities, which also include process model integration, automated workflow to choreograph human and IT functions, ESBs, and business event monitoring. This journey has involved a wide range of internal business-driven SOA activities to improve development productivity and reduce support costs through modularity, reusable assets, composite applications, and model-driven techniques. Organizational and funding challenges have proved to be just as challenging as the technical aspects of SOA.

SOA's incorporation into IBM's Enterprise Architecture has been a key factor for its visibility and management across multiple lines of business. Its strength as an integration platform plays a prominent role in internal IT strategies that need to accommodate a complex landscape of custom, strategic and legacy applications, in addition to third party packages such as SAP.

There will always be a focus on leveraging SOA to increase business value, flexibility, responsiveness, and operational process agility. IBM's use of SOA to be the Premier Globally Integrated Enterprise continues with internal efforts to

completely embed SOA into the consciousness of IT, and to make it autonomically self-sustaining.

About the Author

Lance Walker is a Distinguished Engineer in IBM's CIO Office. He has application architecture responsibilities for several areas within IBM's overall enterprise architecture, and is the Chief Architect for IBM's Enterprise Integration Architecture. He is the Project Lead and lead architect for the corporate initiative that is responsible to accelerate SOA for IBM Internal, and is also the chair for IBM's internal SOA Guidance Council. Prior to this, he led other CIO application architecture initiatives, such as the Isolation Layer Framework, which also had a services framework before SOA became 'popular'. He holds master's degrees in Computer Information Systems (University of Denver), Telecommunications (University of Denver), and Mechanical Engineering (University of Colorado). He is also an IBM Certified Architect.

Author Contact Information

Lance Walker

Email: lancew@us.ibm.com

Phone: 303-562-3159

Technical Training On Demand

Cost effective way for organizations to provide continuous learning opportunities for their employees

- Pay one annual subscription fee (\$475)
- Take as many classes as you want

AFFORDABLE

ACCESSIBLE

Classes available
24/7/52

From:

Home

Office

Travel



FLEXIBLE

- Fit course work into your schedule
- Self paced
- Refresh knowledge
- Study during lunch or after work

COMPREHENSIVE

450+ classes
12,000 topics
Programming + Web Development
Course Catalog

Latest Technologies
Java XML
Oracle Server Technologies
.NET Framework

To view the catalog visit
www.thedacs.com/training

For details call
1.800.214.7921

**ACCREDITATION
CERTIFICATION**

- CEUs granted
- IACET Accreditation Supported



- MCAD
- MCDBA
- MCITP
- MCPD
- MCSD
- MCSA
- MCSE
- MCTS
- MCAS

Open Architecture and Services Oriented Architecture (SOA)- A Compelling Relationship

By Kartik Mecheri and Andrew Gordon, Unisys Federal Systems

1 Introduction

This article exposes how Open Architecture (OA) is critical to Service Oriented Architecture (SOA) as it guarantees complete application agility, and aligns with SOA as an IT strategy for delivering interoperable, reusable, business capability.

1.1 Open Architecture

The definition of the term “Open Architecture”, according to the office of the Navy and OSD, is as follows:

A Navy initiative for: A multi-faceted strategy providing a framework for developing joint interoperable systems that adapt and exploit open-system design principles and architectures. This framework includes a set of principles, processes, and best practices that:

- Provide more opportunities for competition and innovation
- Rapidly field affordable, interoperable systems
- Minimize total ownership cost
- Optimize total system performance
- Yield systems that are easily developed & upgradeable
- Achieve component software reuse

[Office of the Secretary of Defense]

Open architecture specifications are made public and are based on approved or defacto standards. Fundamentally, the principles behind open architecture are focused around:

- (1) component-based
- (2) open-standards based
- (3) interoperability

Usage of these guiding principles of computer and software design has radically increased in both Commercial Off the Shelf (COTS) and open source software. The primary reason for this increase is that this gives the users plug and play capabilities that meet their needs, whether open source or COTS. OA in many organizations is treated as a design constraint, which is taken into account before creating and deploying systems.

Systems that are not built around the fundamental

principles of OA run a huge risk of high maintenance, which leads to high total cost of ownership. These systems may result in tight integration but may not, in most cases, be extensible. The other reason for high cost could be the lack of availability of the labor pool in maintaining such systems.

As mentioned above, one of the principles of OA is openness of the architecture. OA based solutions can potentially result in:

- Complete application agility due to
 - Increased labor pool
 - Increased reusability of the components
 - Plug and play capabilities
 - Easier installation
 - Improved systems management
- Easier adoption of technology
- Improved product quality, including performance
- Ease of communication between domains
- Lower cost of upgrades

It is a common mis-conception that open source and open architecture are the same. While open source software in most cases use the principles of open architecture, open source primarily deals with the source code. It has no licence fee; it generally has a community of developers contributing to its development. On the other hand, OA based products can require the user to pay license fees and are not necessarily, for that matter, open source.

1.2 Service Oriented Architecture (SOA)

From Wikipedia: *“Service Oriented Architecture (SOA) is a computer systems architectural style for creating and using business processes, packaged as services, throughout their lifecycle. SOA also defines and provisions the IT infrastructure to allow different applications to exchange data and participate in business processes. These functions are loosely coupled with the operating systems and programming languages underlying the applications. SOA separates functions into distinct units (services), which can be distributed over a network and can be combined and reused to create business applications. These services communicate with each other by passing data from one service to another, or by coordinating an activity between two or more services. SOA concepts are often seen as built upon and evolving from older concepts of distributed computing and modular programming”*

continues on page 30

Open Architecture and Services Oriented Architecture (SOA)- A Compelling Relationship

Continued from page 29.

Some of the design principles for SOA are:

- Standard Design Contracts
- Loose Coupling
- Service Reusability

There are several other principles such as service visibility and statelessness that drive design decisions to implement SOA based solutions.

Since government processes are generally “common”, appropriately designed SOA implementations can be exposed amongst the Common Process Users of Interest. For example, if a task order process is implemented, it can be made available to government agencies by exposing that process to be consumed by an agency on a properly secured network.

SOA can help you achieve business agility, bringing government and military the possibility of creating an information sharing environment that can:

- Evolve disparate, unconnected, stove-piped systems and processes into re-usable services putting it into a position to mix and match services, rapidly creating new applications to support changing mission requirements
- Increase the speed at which information and services can be rapidly and securely shared so others can benefit, including unanticipated users
- Securely interconnect people and systems, independent of time or location, improving federal and military situational awareness and significantly shorten decision-making cycles

2 OA and SOA together

OA provides transparency by adopting open standards. Transparency can be a measure of efficiency for an enterprise. Open standards based services permit easier integration into existing applications as well as easier development of new services. As a consequence, cross department, cross division and cross agency integration of services and information becomes realistic and achievable for an agency, military service branch, and its constituency. The principles of SOA make inter-agency collaboration possible, while open architecture based SOA

The principles of SOA make inter-agency collaboration possible, while open architecture based SOA makes it simpler.

makes it simpler.

The requirements for SOA also apply to the requirements of Open Architecture. While SOA focuses on multiple systems, Open Architecture focuses on a specific system. The goals for both SOA and OA are complimentary to each other. However, there are some fundamental differences that have to be recognized.

SOA	OA
Interoperability between systems	Focus is within a system, however, OA helps address the interoperability concerns of SOA because of the open standards and techniques used
Primarily focuses on software architecture	Focuses on software, and hardware architecture
System operations are exposed as services, while the underlying implementations can be open or closed	The primary goal is the entire system must be open

Traditionally, functions within an application have been implemented using proprietary interfaces. The cost of integration and maintenance of proprietary software can be extremely high especially if the application vendor resists implementing required enhancements to your systems, compromising your IT investment. Making OA a design constraint or a non-functional requirement during product procurement, lays a foundation for easier integration and collaboration, and extensible service oriented enterprise.

Using a RFP/RFQ/SOW contracting process example in the government, there is an obvious need to make FAR regulations available as a service in order for agencies to use the functionality

or share the data (sample process is illustrated in Figure 1) in an SOA implementation. For other agencies or partners to be able to use this service, the service must be designed using the OA design principles.

continues on page 31

Open Architecture and Services Oriented Architecture (SOA)- A Compelling Relationship

Continued from page 30.

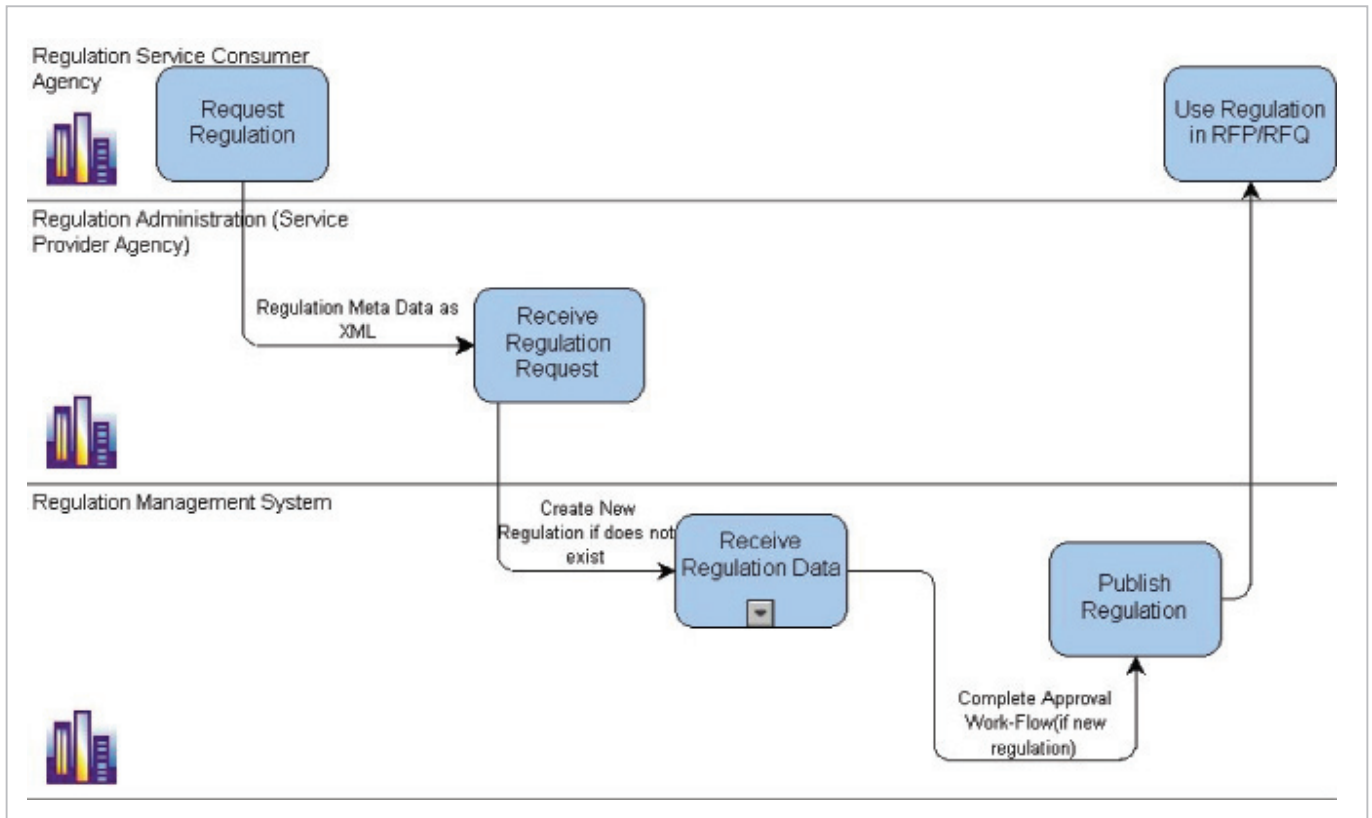


Figure 1: High-level Illustration of multi-agency interaction and usage of the Regulation Service

As you can see, there are multiple advantages to this type of implementation.

- The dependencies between the partners are just the service contracts (note this is an SOA design principle). In the case of web services these could potentially be the Web Service Description Language (WSDL), and xml schema definition (XSD) of the objects and the payloads. There could potentially be a third document part of the contract, the Policy definition for accessing and executing those services and relevant operations.
- Easy integration in a multi-organization environment. It can include software vendors such as Microsoft to be able to play easily in this environment. You know what components are available and what standards are followed, both of which are the design principles of OA.

If the regulation service (please note that, service does not mean just a web service) did not follow open standards (note

this is an OA design principle), the integration would be a lot more difficult and expensive in an inter-agency environment. Some of the difficulties leading to more cost could be:

- New procurement or contract with the vendor implementing the regulation service
- Longer time to implement changes across all agency users as opposed to making changes in one place
- High potential for vendor lock-in
- No reusability of government-wide functionality

3 Conclusion

Open Architecture and Service Oriented Architecture are complimentary architectural patterns. In combination, they provide great value to government (DoD and Civilian) and high potential for inter-department and inter-agency integration and re-use of business processes. Government agencies must take the power of these two patterns seriously and incorporate them as non-functional requirements or design

continues on page 32

Open Architecture and Services Oriented Architecture (SOA)- A Compelling Relationship

Continued from page 31.

Government. The example above is a real-time shared service that is implemented as part of an Integrated Acquisition Suite of applications in a large government agency. All or any sub-processes within the Acquisition process can be leveraged across the entire federal government with minor or no modifications to the existing services.

About the Authors

Kartik Mecheri is Chief Architect for a large government contract and works for Unisys Federal Division. He is a thought leader in the areas of open source and Service Oriented Architecture. He has over 8 years experience working with large federal procurement systems. In addition to his technical responsibilities he is responsible for providing senior government agency officials consultative advice related to the latest technologies to enable them to take advantage of the benefits experienced with emerging technologies. As a member of the Unisys consulting team he has also led large government agencies in implementing the beginning stages of a true “electronic contracting environment” and implementing the latest technologies allowing them to leverage Service Oriented Architecture and enabling an enterprise wide architecture solution. He has over 11 years enterprise software experience and a Masters in Engineering.

Andrew Gordon is Director, SOA and Open Source Solutions with Unisys Federal Systems, based in Reston, Va. He is responsible for leading the SOA and open source business vision for Federal Systems, shaping the Federal

strategic SOA and open source solutions portfolio and helping federal government clients harness the power of SOA and open source as flexible, reliable, secure and cost effective options for business critical software.

Andrew has 18 years of success at leading and pioneering enterprise wide shared services initiatives and bringing to market service oriented infrastructure products at Compuware, IBM, Mercator, and Butterfly.net. Andrew built and led enterprise wide Shared Services (SOA) organizations, exponentially increasing the rate at which organizations could respond to new and changing market requirements while simultaneously delivering a massive reduction in IT cost for itself and its customers.

Author Contact Information

Kartik Mecheri

Email: kartik.mecheri@unisys.com

Phone: 703-605-2136

Andrew Gordon

Email: andrew.gordon@unisys.com

Phone: 703-439-5491

The first 50 people to send in a completed survey will receive a FREE DoD/IT Acronym CD from the DACS.

This valuable CD-ROM contains over 9,000 Department of Defense and Information Technology acronyms. There are hundreds of acronym lists available but none are as well done as this CD AND specifically targeted towards DoD and Information Technology. This unique-shaped CD-ROM plays in your computer's regular, hub-mounted, CD drive. You'll use this great resource over and over again. It's FREE, just for filling out our brief survey on the next page!



Fold Here

Data & Analysis Center for Software (DACs)



<http://iac.dtic.mil/dacs/>

Fold Here



Software Tech News Feedback

1. Which volume of the Software Tech News did you receive? **STN 11:2 June 2008 SOA**

2. When did you receive the newsletter? (month/year)

3. How satisfied were you with the **CONTENT** of the newsletter? (Article Quality)

Very Satisfied Satisfied Neither Satisfied nor Dissatisfied Dissatisfied Very Dissatisfied

4. How satisfied were you with the **APPEARANCE** of the newsletter?

Very Satisfied Satisfied Neither Satisfied nor Dissatisfied Dissatisfied Very Dissatisfied

5. How satisfied were you with the **OVERALL QUALITY** of the newsletter?

Very Satisfied Satisfied Neither Satisfied nor Dissatisfied Dissatisfied Very Dissatisfied

6. How satisfied were you with the **ACCURACY** of the address on the newsletter?

Very Satisfied Satisfied Neither Satisfied nor Dissatisfied Dissatisfied Very Dissatisfied

7. Approximately how much of the newsletter do you read?

The entire issue Most of the content About half the content Briefly Skimmed Didn't Read

8. Would you read this newsletter in an E-mail newsletter format?

Definitely Probably Not Sure Probably Not Definitely Not

9. How did you request the product or service?

Phone Call E-mail DACS Website Subscription Form Other _____

10. Would you recommend the Software Tech News to a colleague?

Definitely Probably Not Sure Probably Not Definitely Not

11. What topics would you like to see this newsletter devoted to?

Comments (Optional)

Register for the first time

Update current subscription

Name: _____ Position/Title: _____

Organization: _____ Office Symbol: _____

Address: _____

City: _____ State: _____ Zip Code: _____

Country: _____ E-mail: _____

Telephone: _____ Fax: _____

Functional Role: _____

Organization Type: Air Force Army Navy Other DoD _____
 Commercial Non-Profit Non-US US Government FFR&D Other _____

*Note: You must give us your address to receive the CD.

About the Software Tech News

STN Editorial Board

Ellen Walker

Managing Editor

ITT Corporation, DACS

Philip King

Production Editor

ITT Corporation, DACS

Paul Engelhart

DACS COTR

Air Force Research Lab

Morton A. Hirschberg

Editorial Board Chairman

Army Research Lab (Retired)

Thomas McGibbon

DACS Director

ITT Corporation, DACS

Dr. Kenneth E. Nidiffer

Software Engineering Institute

Dennis Goldenson

Software Engineering Institute

John Scott

Mercury Federal Systems

Advertisements

The Software Tech News is now accepting advertisements for future newsletters. In addition to being seen by the thousands of people who subscribe to a paper copy, an electronic version is available at the DACS website, exposing your product, organization, or service to hundreds of thousands of additional eyes every month.

For rates and layout information contact:

news-editor@dacs.dtic.mil

Cover Design

Erin Jones

Graphic Designer

ITT Corporation, DACS

Ideas by Author,

Robert Glass



Article Reproduction

Images and information presented in these articles may be reproduced as long as the following message is noted:

“This article was originally published in the Software Tech News, Vol.11, No.1 June 2008. Requests for copies of the referenced newsletter may be submitted to the following address:

Philip King, Production Editor
Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400

Phone: 800-214-7921
Fax: 315-838-7130
E-mail: news-editor@dacs.dtic.mil

An archive of past newsletters is available at www.SoftwareTechNews.com. In addition to this print message, we ask that you notify DACS regarding any document that references any article appearing in the *Software Tech News*.



Distribution Statement:
Unclassified and Unlimited

DACS

P.O. Box 1400

Rome, NY 13442-1400

Phone: 800-214-7921

Fax: 315-838-7130

E-mail: dacs@dtic.mil

URL: <http://www.thedacs.com/>

About This Publication:

The Software Tech News is published quarterly by the Data & Analysis Center for Software (DACs). The DACs is a DoD sponsored Information Analysis Center (IAC), administratively managed by the Defense Technical Information Center (DTIC). The DACs is technically managed by Air Force Research Laboratory, Rome, NY and operated by ITT, Advanced Engineering and Sciences Division.

In This Issue

Tech Views --- The Commercialization of SOA

By Ray Cardillo, ITT Corporation..... 3

SOA Essentials

By Robert Glass, Oracle National Security Group..... 6

SOA: Keys to Sustaining the Transformation to a Service-Oriented Architecture

By Andrew Gordon, Unisys Federal Systems.....11

Communication and Trust in a Net-Centric Community of Communities

By Kelly Emo and Chris Brown, Hewlett-Packard.....15

IBM's SOA Journey

By Lance Walker, IBM.....15

Open Architecture and Services Oriented Architecture --- A Compelling Relationship

By Kartik Mecheri and Andrew Gordon, Unisys Federal Systems.....29

Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400

Return Service Requested

PRSR STD
U.S. Postage
PAID
Permit #566
UTICA, NY