SYS-CON

THE WORLD'S LEADING MAGAZINE DEDICATED TO WEB SERVICES TECHNOLOGIES

FOR ALL THE LATEST NEWS AND INFORMATION VISIT:

SOAWORLD™ MAGAZINE

SEPTEMBER 2007 / VOLUME: 7 ISSUE 9

Building SOAs That Benefit Business Users

PLEASE DISPLAY UNTIL OCT. 31, 2007

$6.99US $7.99CAN

09

0 71486 03420 9

_INFRASTRUCTURE LOG

_DAY 79: This is out of control! Our IT environment is rigid and inflexible. Our business needs are changing, but our environment isn't built to change with them. We can't adapt. Oh, no...I was afraid of this. We're so rigid we're stuck in time.

_*Infrastructurus prehistoricus*. I've read about this.

_DAY 80: I'm taking back control with IBM SOA solutions. Now we can align business goals with our IT. We have the hardware, software and services we need to respond to change. Strategy, planning and implementation are in tune with our specific business needs. Now we can deploy and update business processes faster and more efficiently.

_Goodbye, rigid past. Hello, flexible future.

Take the SOA business value assessment at:
IBM.COM/**TAKEBACKCONTROL**/SOA

Come Together

WRITTEN BY **SEAN RHODY**

Back before I began my career in computers, I studied physics. One of the concepts that fascinated me was that of nuclear fusion – bringing two particles together to form a new, heavier particle and at the same time producing energy.

In the business world, I find the concept of a merger between two companies very much akin to the concept of fusion. You have two companies, which are the particles in this analogy, and you bring them together, creating one new particle, the new company, as well as energy. We can debate the actual value of energy in this analogy, but from my perspective when you bring two companies together, you have a duplication of functions (most of the time), and those duplicates that are discarded as a result can be considered the energy.

Service-oriented architecture enters the picture as we begin to consider how to fuse two organizations together. In the world of physics, you often need to accelerate particles at each other at high speeds to overcome the innate positive charge of the particles, which causes a repulsive effect the closer the particles come to each other. Similarly in business, two organizations often have identical functions and duplicate software systems that are the systems of record for each organization. To successfully merge two corporate entities, the repulsive effect of having to operate duplicate systems needs to be overcome. Oftentimes a newly fused corporate entity makes a decision on which systems to use and which systems to abandon. In the end that may in fact be the final path. But often, simultaneous operation must be accommodated for a period of time (which sometimes equates to eternity).

SOA – which provides capabilities for loosely coupling these duplicate systems, federating the data between them and managing the concept of systems of record – is an enabler that eases the pain of bringing two entities together.

Much like a catalyst that enables a reaction, SOA simplifies the integration requirements by lowering the barriers for simultaneous operation. It also provides the ability to provide a common façade over multiple integration points, allowing the indefinite operation of redundant systems, as well as for the eventual retirement of some of the redundant software.

When we create fusion and collide particles, certain particles are more "stable" than others. The more stable a particle, the more difficult it is to achieve fusion. Rather than assign a negative connotation to the word "stable," let's say that some particles accept change more readily than others. These particles are also much more receptive to fusion.

Similarly, an organization that has already embraced service-oriented architecture as a core component of its IT organization becomes more receptive to the concept of merging with another organization. Because concepts such as service catalogs, system facades, business process management, and the like are part and parcel of the organization, the concept of merging a set of redundant systems becomes less of a climactic event and more of business as usual – integration and consolidation. Integrating with another company no longer presents nightmares to an IT organization with SOA in place.

We all know that there are winners and losers in mergers – part of the attraction of merging two organizations is the ability to gain economies of scale by reducing the number of people needed to perform a particular function as volume increases. Often, IT organizations have great fear in a merger, as they are typically candidates for staff reduction. For an organization with SOA in place, the odds of being part of the reduction in staff is greatly diminished compared to staff in an organization that has no integration strategy in place.

Just as in physics, fusing two companies requires a great deal of time, effort, and energy. SOA makes it easier. ■

About the Author

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

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

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

Altova® MissionKit™ 2007 – The secret to savings on top software tools.

ALTOVA® missionkit™ 2007

Gear up for development excellence

Take off with the Altova® MissionKit™, and uncover the secret to savings on top software tools.

Spied in the Altova MissionKit 2007:

- The world's leading XML development tools (XMLSpy®, MapForce®, StyleVision®, etc.)
- Plus application design, data management, and modeling options for software architects

The Altova MissionKit 2007 bundles up to eight Altova products for less than the price of two!* All MissionKits include XMLSpy, MapForce, and StyleVision at substantial savings, plus options are available to add up to five additional application development, data management, and modeling tools effectively free of charge. It's your first-class ticket to the power, speed, and simplicity of Altova's award-winning product line. Save a bundle!

Download the Altova MissionKit 2007 today: www.altova.com

ALTOVA®

*Pricing comparison is edition-specific and based on Altova MissionKit vs. any two individual licenses of the following included products: XMLSpy, MapForce, and StyleVision.

Whither SOA?

Is SOA dead, doomed, or misnamed?

WRITTEN BY **JOHN MICHELSEN**

Many in the media are already calling on the demise of SOA and saying that it's just a passing phase, or it's really just a rebrand of the EAI space, or that it will be segmented only to certain integration-type challenges.

We have a bit of a different take. The term SOA will go away over the next several years but it will go the same route that e-commerce applications went.

If you think back to the mid-'90s, we started talking about e-commerce as an architecture that was distinct from the then traditional enterprise architecture. However, over the past few years, e-commerce as a term has disappeared because it has become the ubiquitous expectation for how we build applications, or what those applications do (some kind of commerce). It consumed the entire enterprise application space.

To pick one SOA platform – we used to buy BEA for the Website part of our applications, and then kept building in a completely disjointed fashion inside the firewall. Today, however, that same BEA server is used both inside and outside the firewall. This is probably an e-commerce implementation as well; we just don't call it that anymore. SOA will become the same kind of default for enterprise applications.

SOA is a new way of looking at a very traditional, classic problem, so I don't think of it as brand new. At the same time, I don't think of it as just a re-named EAI or some passing fancy. In fact, SOA is the way to look at application design so that the business can understand it and it can be better decoupled from all of its interdependencies; as such, the practice of SOA will live on.

Feel free to stop calling it SOA in a few years, when that's just the expected way we build applications. Feel free to not call it SOA at that point, because you will be doing it de facto. Is SOA dead, doomed, or misnamed? It's none of those, but we have to put a name on a new idea until it becomes old hat – and SOA is that name. ■

About the Author

John Michelsen is a founder and chief architect at SOA testing vendor iTKO, Inc. (www.itko.com).

john@itko.com

SOAWORLD MAGAZINE
www.SOA.SYS-CON.com

CORPORATE

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

Senior VP, Editorial & Events
Jeremy Geelan jeremy@sys-con.com

ADVERTISING

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

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

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

SYS-CON EVENTS

Event Manager
Lauren Orsi lauren@sys-con.com

Event Associate
Sharmonique Shade sharmonique@sys-con.com

CUSTOMER RELATIONS

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

SYS-CON.COM

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

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

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

ACCOUNTING

Financial Analyst
Joan LaRose joan@sys-con.com

Accounts Payable
Betty White betty@sys-con.com

SUBSCRIPTIONS

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

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

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

Altova® DatabaseSpy™ 2007 – The unique multi-database query and design tool.

ALTOVA® databasespy™ 2007

Evade constraining query tools

Move up to DatabaseSpy™ 2007, and manage all your databases from one elegant interface.

Spied in Version 2007 Release 3:

- Connectivity with all major databases
- Enhanced support for XML datatypes
- Superior SQL auto-completion with clear presentation of tables, views, and data
- Graphical database structure design and visualization

Altova® DatabaseSpy 2007 is the unique multi-database query and design tool from the creators of XMLSpy. A real steal, DatabaseSpy liberates data management, delivering advanced, coherent capabilities at a fraction of the cost of single-database solutions. Write accurate queries with confidence and get clear, easily negotiable results. Create and edit database structures visually by dropping tables on the design pane and dragging relationship links between them. Even duplicate existing structures in diverse database types. Do data differently! **Download DatabaseSpy 2007 today: www.altova.com**

ALTOVA®

Building SOAs That Benefit Business Users

Matching user needs to the IT benefits of moving to Service-Oriented Architecture

WRITTEN BY **JOE MCGONNELL**

If Service Oriented Architecture (SOA) is all about business agility, then why does the focus continue to be on how services will be built, deployed, and managed by IT instead of how they'll be consumed by business users?

How will SOA services to be rolled out in your organization? Through a Web browser or rich Internet application? Will these Web applications totally replace the applications your business users are currently relying on? Or will they simply add to the growing number of applications that your business users need to access on a regular basis?

Enterprise architects have multiple technology options to choose from as they implement an SOA. However, designing it with the end user in mind can help dictate choices that will speed up application integration, increase overall flexibility, and result in a more productive workforce. By creating a New Enterprise Desktop, architects can transcend the latest SOA messaging, governance or security fads and focus instead on what really drives the successful enterprise — a superior SOA user interface.

Today's Desktop Environment

Let's start by examining the current environment for most of today's business users — the desktop. The reality of the current desktop environment generally consists of the following challenges:

1. Users must interact with multiple applications. A good example of this is the call center industry where, through an informal survey of our customers, the average agent needs to interact with nine-10 applications on a daily basis to resolve customer issues.
2. The more applications used by business users, the more an enterprise's bottom line is affected by reduced productivity and higher training costs.
3. Most applications today are not integrated. Despite all of the investment in integration technologies, the most common form of integration today is still manual copy-and-paste.

4. Most business processes require humans to complete tasks manually.
5. Application functionality is generally fixed and limited to what the original developers created even if it doesn't meet user requirements now in terms of functionality or ease of use. Major change requests queue in your IT backlog or in vendor product roadmap feature request lists.
6. Users are increasingly interacting with applications controlled outside the organization. SAAS applications are good examples of this. More sophisticated partnerships and outsourcing arrangements are also becoming increasingly common.

The promise of user productivity from newer applications generally isn't realized because new applications rarely replace existing applications or streamline processes for users. They simply add to desktop complexity, which is driving the need for the New Enterprise Desktop designed around end-user productivity.

Despite the best intentions, today's business users are required to interact with many different disparate applications to perform daily tasks

SOA in Today's User Environment

The promise of SOA is that applications will be broken down into a set of reusable services creating a more agile computing environment; one in which new services can be created and deployed more quickly and easily. But even the most optimistic of us understand that full-fledged SOA is a long journey, not a near-term reality. In addition to automating services, SOA architects must define processes around those services that secure, govern, and manage their SOAs. In fact, as "service reuse" becomes a mantra, middleware vendors are rolling out vertical industry roadmaps with standardized components that run on top of an SOA platform.

The near-term reality is that services will gradually be rolled out but not as a complete replacement for existing applications. Services will be incremental to existing applications in most environments.

From a user perspective this means that new services will likely be made available via a browser interface adding yet another "application" to learn and use and likely fall into the copy-and-paste category along with other applications.

At our most basic level, business users really want technology to help us do our jobs more quickly and easily. Simplicity and productivity are paramount. We want as few user interfaces to learn and use as possible and yet we need new richer and more integrated functionality including that derived from SOA.

The Frontlines of SOA

How will SOA-based services ultimately be consumed by users in your organization? The most common thinking is via Web applications or "enterprise mashups" but that's only one option and likely not the best solution for most enterprises. The three most likely options are:

1. Services delivered through Web applications or mashups. In this scenario, the browser becomes the new operating environment. But how long will it be before these Web applications or mashups totally replace your existing applications throughout the enterprise? Is that even your plan? The most likely scenario is that Web applications will increasingly deliver business functionality but need to interact and integrate with other enterprise applications, many of which will still reside on the desktop. Web applications will remain one of the application platforms that deliver functionality to users; not "the" application platform. The advantage of such an approach is that there will be plenty of choice in terms of Web application frameworks in which to deploy your services. The downside is that the Web application is only one of the many UIs that a business user must interact with to do his jobs. You could argue that such an approach actually reduces user productivity.

2. Extending existing applications with new Web Services functionality. In this scenario, new services are added to existing applications, so users benefit from new functionality while still interacting with their existing UIs. The main advantage of this approach is that it enables IT organizations to roll out new services with the least impact on business user productivity. Such an approach requires a technology platform, such as the OpenSpan Platform, that can be injected into the memory of running applications to execute the desired automations, such as calling a particular Web Service for a given keystroke or mouse click.

3. Building new thin-client composite applications that reside on user desktops. In this scenario the composite application isn't limited to just Web applications but virtually any legacy or desktop application. Such composite applications mesh well with SOA architectures in that they view supporting applications as nothing more than a set of services, bits of functionality. Composite applications let developers identify only the relevant functionality in a set of applications and deliver that functionality to users in an easy-to-consume format. The Afni example in the next section illustrates both the architecture and benefits to this approach. A major advantage is that composite applications can dramatically increase user productivity and reduce training requirements for new employees because, unlike mashups, they tend to reduce the total number of user interfaces with which a business user must interact. They also provide a mechanism for rolling out new services more quickly. IT organizations can then replace or upgrade business applications at any time without necessarily impacting the user experience on the desktop. They can also gradually roll out new SOA services, replacing older legacy integration points again without negatively impacting the user.

Now, it's likely that your enterprise is already heading down the

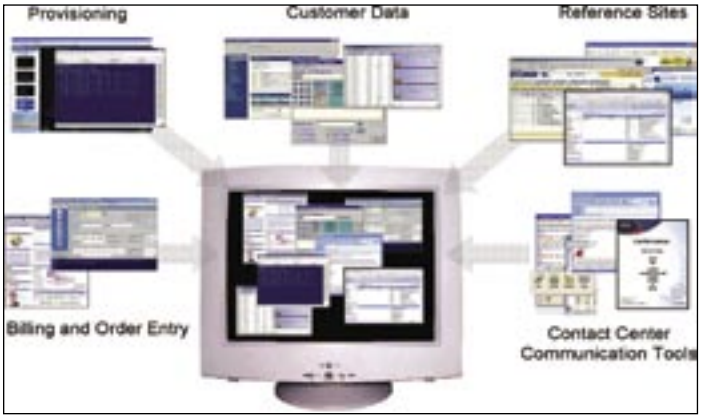


Figure 1: Contact center agents often need to interact with 10 or more applications to resolve even simple customer issues

Figure 2: Afni uses composite applications to improve agent productivity and reduce training requirements

path of option one. The developer side of you is probably thinking that SOA = services; integration = middleware and ESBs; GUI = Web applications. Are you still searching for a framework for building Web applications? Building a user interface shouldn't be a framework since that would imply that you're going to put serious business logic in the wrong place. User interfaces should be built with user interfaces and not a lot of code.

The front-end of SOA should either be your existing UIs or, more likely long-term, a composite application. A composite application will let you pull together documents from Google, customer data from Salesforce.com, partner data from partner Web Services, as well as other enterprise data from your existing legacy applications. The future of SOA requires a composite interface, a New Enterprise Desktop, and not additional composite middleware applications.

Composite Applications in the Real World

A good example of composite applications driving improvements in user productivity can be seen by Afni, Inc., a leading contact center outsourcing firm based in the U.S. Afni has two primary businesses — outsourced call center services and collection services. It provides collections and technical support services to its customers' customers. Outsourcing is a challenging business that requires Afni to provide equal or better service to its customers for less money than its customers can do on their own. In other words, productivity of its contact center agents is critical. Another challenge is that the information about people on the other end of the phone resides in its customer's systems.

> **"The future of SOA requires a composite interface, a New Enterprise Desktop, and not additional composite middleware applications"**

Afni has implemented a series of composite application "buddies" that dramatically simplify life for its call center agents. Instead of an agent having to be trained on eight, 10, 15, or more applications to resolve common support issues, Afni agents interface with a single composite application that delivers all relevant customer information, allows them to change customer status in multiple applications, and add customer notes for audit trail purposes. The composite applications were designed with the business process flow in mind. The application walks an agent through the desired business process providing relevant data along the way. It also requires that an agent complete necessary compliance steps before proceeding.

Afni leverages the OpenSpan Platform for its composite applications. OpenSpan lets Afni integrate virtually any type of application: Windows, Java, host/mainframe, and Web applications. It also lets Afni leverage Web Services made available by its customers. Whenever the customer exposes additional functionality as Web Services, Afni can easily replace legacy integrations with Web Services or simply add the new Web Services without negatively impacting the user experience for its agents.

The results? Afni has been able to reduce average handling time (the amount of time an agent spends on the phone with a customer) by 53 seconds per call for all agents. For new agents, the improvements are even more dramatic since average handling times have been reduced by over 300 seconds per call. New rep training has been reduced by 20% and agent turnover has been reduced by 35%. Afni has also eliminated all performance/compliance penalties with its customers and improved its overall quality scores such that it is now the #1 performing outsourcing center for its clients. Afni is also well positioned for the future. It will be able to leverage new services delivered by its clients almost immediately with little or no impact on the productivity of their agents.

Summary

Although SOA promises to deliver significant value to business and in particular to the IT organizations that support the business, it's critical to develop and execute on a strategy that will enable business users to consume these services without requiring significant training or adding complexity to their existing business processes. Composite applications that leverage both new SOA services and existing legacy applications provide an ideal environment for improving user productivity while also allowing IT organizations to roll out SOA in a controlled environment that won't negatively impact business users. ■

About the Author

Joe McGonnell is senior vice-president of worldwide marketing at Atlanta-based OpenSpan, enablers of the New Enterprise Desktop

Need to integrate CICS applications?

Make sure you know all your options.

September 19 Webcast

Service-Enabling CICS Applications: Best Practices

CICS is evolving and incorporating many abilities that used to be considered the domain of the mid tier. To make sure your CICS assets are put to good use, you might want to ask some critical questions:

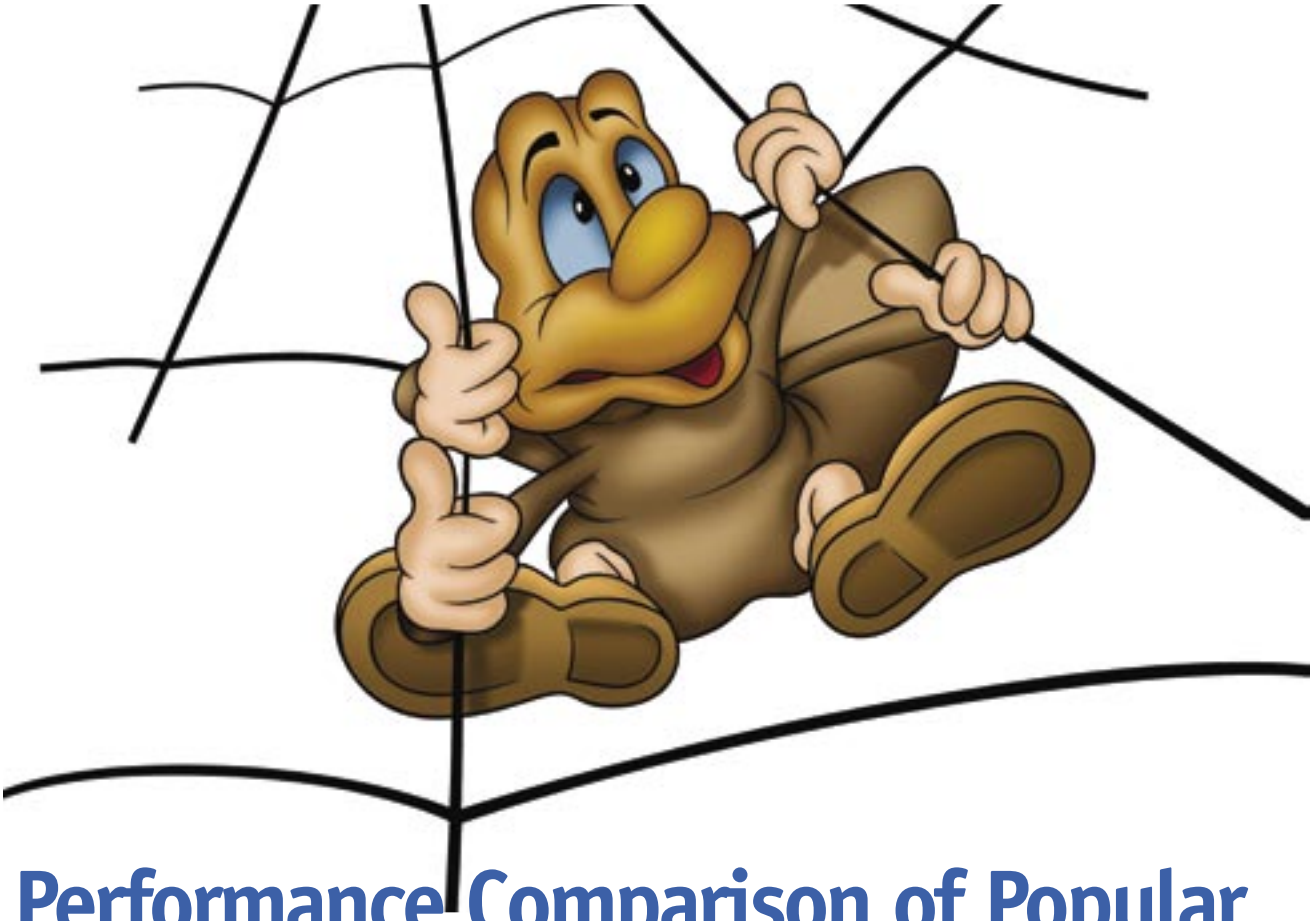
- Will CICS service and integration technologies work with code from your existing applications?
- Will the CICS services be invasive to your mainframe? (If so, is that okay?)
- Is MIPS usage a concern?
- Is the proposed integration approach agile?
- Does your IT staff have the right skill set to deal with mainframe services?

For a look into CICS service-enabling options and how they affect your environment, register for the Sept. 19 webcast.

To attend at 10 a.m. EDT / 3 p.m. GMT, see www.attachmate.com/info/soaworldmag_cicswc1
To attend at 2 p.m. EDT / 11 a.m. PDT, see www.attachmate.com/info/soaworldmag_cicswc2

At Attachmate, we've been demonstrating our mainframe expertise for more than 26 years. So be sure to sign up for the webcast. You'll learn how to get the most from your CICS applications without compromising the best possible outcome.

attachmate®



A Performance Comparison of Popular Open Source Web Services Engines

We really are making progress

WRITTEN BY **KRISHNENDU KUNTI, RAHUL MURALIDHAR, AND NAGARANI BADVEETI**

Service Oriented Architecture (SOA) stands on the twin pillars of reusability and interoperability. Any business function that can be reused is an ideal candidate to be exposed as a service.

A reusable service, until and unless implemented in an interoperable manner, isn't much use. Web Services standards provide standards-based interfaces for service description, discovery, and message definitions to invoke such services. Web Service technology has come a long way to reach enterprise-grade adoption of the core standards. Nonetheless, the process has been facilitated by Web Service engine offerings from both the open source community and product vendors. One of the majors concerns of Web Services adoption is the performance overhead associated with transmitting and processing of XML. Today, Web Services engines and tools provide a means for data compression (XMill), data representation efficiency (fast infoset), and efficient processing of XML. In this article we'll be benchmarking the performance of a few leading open source Web Service runtimes, based on their capability to process XML using different binding frameworks in addition to native XML binding. We aim to bring out a comparative study highlighting the relative merits in using specific platforms for specific scenarios.

The Evolution of Web Service Platforms

Broadly speaking, Web Services engines can be segregated into three generations – the first generation of Web Services engines such as Apache SOAP came to market during the early stages of Web Services core standards definition/adoption. These platforms

used the DOM API and had dismal performance.

The second generation of Web Services platforms such as Apache Axis 1.4 had native support for WSDL, JAX RPC, and SAAJ; however, these engines had limited support for integrating with external XML binding frameworks and solely depended on a native binding framework. Second-generation Web Services platforms boasted far better performance compared to first-generation Web Service platforms; however, the performance degraded significantly for large XML payloads. Third-generation Web Services platforms such as XFire and Axis2 use StAX for XML processing and have a pluggable architecture that allows for integrating with external binding frameworks. These platforms draw from the strength of individual binding frameworks, StAX parsing models, and overall a better and pluggable architecture. Third-generation Web Services platforms deliver better performance both in small XML payloads and large XML payloads and come with support for a number of WS* standards.

Web Service Performance

Web Service performance is a function of several factors, such as the communication protocol, the network latency, the compression mechanism used, the size of the document, and the representation format (encoding) of the information. It's beyond the scope of this article to cover the broad spectrum of factors affecting the overall performance of Web Service.

Here we're concerned with the performance analysis of a SOAP Engine combined with a binding framework, hence we'll limit our performance studies to key related areas such as binding performance, round-trip performance, scalability, speed, and throughput. These are directly affected by factors such as the binding framework used and the parsing model of the SOAP Engine. The study assumes that factors external to the situation of concern remain in a constant state and that reasonable steps were taken to ensure that.

Factors Affecting the Performance of a Web Service Engine

XML data sent over the wire is converted to native objects in a Web Service platform and vice versa; these native objects are passed to and from underlying business methods. The process of converting a native object to XML and vice versa is termed marshalling and unmarshalling, respectively. In J2EE, there are three APIs (SAX, DOM, and StAX) for parsing XML document; each of these APIs has relative strengths and shortcomings. Binding frameworks use one of these APIs to generate custom native objects from XML and vice versa with a little configuration. All Web Services platforms use either a parsing mechanism or binding frameworks for marshalling and un-marshalling data. As a result, the performance of a Web Service platform largely depends on the efficiency of binding frameworks used, the underlying parsing models, and the overall architecture.

The first generation of Web Services engines used the DOM API, which attributed to the degradation of performance for larger payloads and high load conditions.

The second generation of Web Service platforms used native code for marshalling and un-marshalling, which depended on the SAX API for XML parsing. Though SAX-based XML parsing is faster than DOM, it necessitates parsing the whole XML document, which may not be unnecessary at times, and moreover, Web Service platform-specific binding frameworks are often not as efficient as the best-of-breed binding frameworks available. Because of their dependence on custom platform-specific binding frameworks, the second generation of Web Services platforms can't benefit from improvements incorporated in the mainstream best-of-breed available binding frameworks.

The third generation of Web Services platforms come with a pluggable architecture that allows the use of best-of-breed binding frameworks with little configuration, moreover, these platforms use StAX, which combines the best features of both SAX and DOM to deliver consistently better performance for both large payloads and high load conditions.

Methodology for Web Service Engine Performance Evaluation

Overview of the Test Environment

All tests were carried out on a Dell Optiplex GX 260 desktop workstation, with a 1.8GHz Pentium 4 processor, 400MHz front-side bus, and 1GB of DDR SD-RAM, running Windows XP Professional (Service Pack 2).

Table 1 provides the version numbers of the various tools and APIs used in the course of this study.

JRE Configuration

Both Java Runtime Environments were used with their default configuration. No configuration changes were made to optimize the JVM.

Apache Tomcat Server Configuration

The Apache Tomcat Server was used with the default configuration. The maximum permitted heap size of the Tomcat server was 64MB and the maximum number of threads permitted was 150.

Overview of the Test Cases Under Consideration

The test cases were chosen to satisfy three primary goals:

- Compare the performance of second-generation SOAP Engines with their third-generation counterparts.
- Compare the performance of the major binding frameworks in wide use (ADB, JAXB2, and JibX).
- Comment on the level of support/integration offered by different SOAP Engines for the aforesaid binding frameworks.

To achieve this, five test cases were chosen for analysis:

- Axis 1.4 with native data-binding
- Axis2 1.2 with Axis Data Binding (ADB)
- Axis2 1.2 with JibX Data Binding

Component Name	Version and Build
Java 2 Runtime Environment, Standard Edition	Version 1.5, build 1.5.0_04-b05
Java 2 Runtime Environment, Standard Edition	Version 1.4.2, build 1.4.2-b28 (Only for Apache Axis 1.4)
Apache Tomcat Server	Version 5.5.9
XFire SOAP Engine	Version 1.2.6
Apache Axis2 SOAP Engine	Version 1.2
Apache Axis SOAP Engine	Version 1.4
JibX Binding Framework	Version 1.1.5
xsd2JibX Tool	Version 0.2b
JibX Binding	Version 0.2
Generator	Beta
JaxB Binding Framework (RI)	Version 2.0.1

Table 1 Version numbers of the components used

- XFire 1.2.6 with JibX Data Binding
- XFire 1.2.6 with Sun's JaxB 2.0 Data Binding

To make the test more comprehensive, the clients for each Web Service were created using the method recommended by the documentation and the same binding as the server-side service. For example, the Axis2 with JibX service would be invoked by a client generated using WSDL2JAVA with JibX binding.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Employees" type="EmployeesType" />
  <xs:element name="Employee" type="EmployeeType" />
- <xs:complexType name="EmployeeType">
  - <xs:sequence>
      <xs:element name="Title" />
      <xs:element name="FirstName" />
      <xs:element name="LastName" />
      <xs:element name="EmployeeID" />
      <xs:element name="Designation" />
      <xs:element name="Department" />
      <xs:element name="Res" />
      <xs:element name="Off" />
      <xs:element name="Mobile" />
    </xs:sequence>
  </xs:complexType>
- <xs:complexType name="EmployeesType">
  - <xs:sequence>
      <xs:element ref="Employee" minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Figure 1: XML Schema representation of the collection

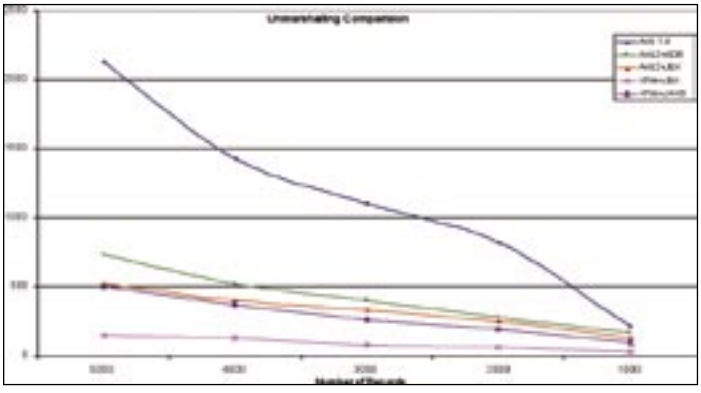


Figure 2: Average un-marshalling times for objects of various sizes

Marshalling Performance:

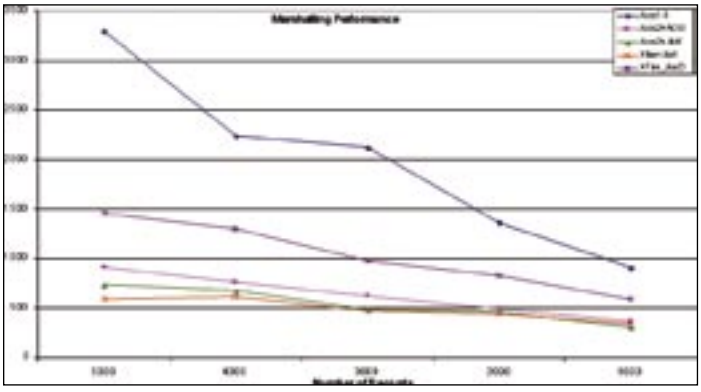


Figure 3: Marshalling performance for objects of different sizes

Round-trip Performance:

Overview of the Test Application

To test the performance of the various flavors of Binding Frameworks and SOAP Engines, a simple standardized test application was first designed. This application consisted of a Web Service that took an array of objects as a parameter and returned the same to the client.

To ensure a fair basis for a comparative study, an XML Schema representing a collection of "Employee" objects was first defined. Each "Employee" object contained nine variables, all of which were of simple types. This schema was used to follow a schema-driven development process for all five test cases explored by this article.

To ensure that our performance figures aren't biased by external environmental factors such as network latency, the clients invoked their corresponding Web Service from the same machine hosting the Web Services.

To quantitatively measure the performance of both the Binding Framework and the SOAP engine, the system time with millisecond accuracy was recorded at the following critical points along the request-response flow:

- When the client invokes the service
- When the SOAP Engine servlet gets a SOAP message from the client
- When the Web Service's business logic begins executing
- When the Web Service's business logic completes execution
- When the SOAP Engine servlet replies to the client with a SOAP message
- When the final object returned is available to the client

From the data above, we were able to calculate the following statistics:

- **Total Round-trip Time:** The time elapsed between the client invoking the service and the client getting the final object returned.
- **Total Marshalling Time:** The time elapsed between the business logic completing execution and the SOAP Engine servlet sending the reply message to the client.
- **Total Un-marshalling Time:** The time elapsed between the SOAP Engine servlet receiving the message from the client and the business logic starting execution.

Benchmarking Procedures

Each test case was put through a two-phased standard procedure to quantitatively measure its performance and give us a fair basis for a comparative analysis. The procedure was designed to measure the performance of each case with different object sizes ranging from a collection with 1,000 records to one with 5,000 records. The objects were created by the client prior to the service invocation and were populated with randomly generated data to avoid the possibility of any pattern-based optimizations.

In the first phase, a single client was used to invoke the Web Service and the time taken to un-marshal the incoming message into objects, marshal the objects into a message, and complete a round-trip were determined. To do this, the service was first invoked three times, off-the-record, to warm up the JVM. This was followed by three more runs for which data was recorded. The average of these three runs was recorded as the result. This procedure was carried out for each test case, with five different object sizes (1,000, 2,000, 3,000, 4,000, and 5,000 records).

In the second phase, the load test simulated 10 users simultaneously and repeatedly invoking the Web Service for a period of one

minute. The recorded results were then used to generate reports and study the scalability, throughput, and speed of the test cases.

Performance Metrics

Using the data collected during the first phase of testing, we calculated and plotted the average values of the un-marshalling, marshalling and round-trip times corresponding to each of the five test cases. The graphs show how the time taken to perform a given operation varies with respect to the size of the object for each test case. Each graph includes data from all five test cases and serves as a powerful visual aid for a comparative study of their performances.

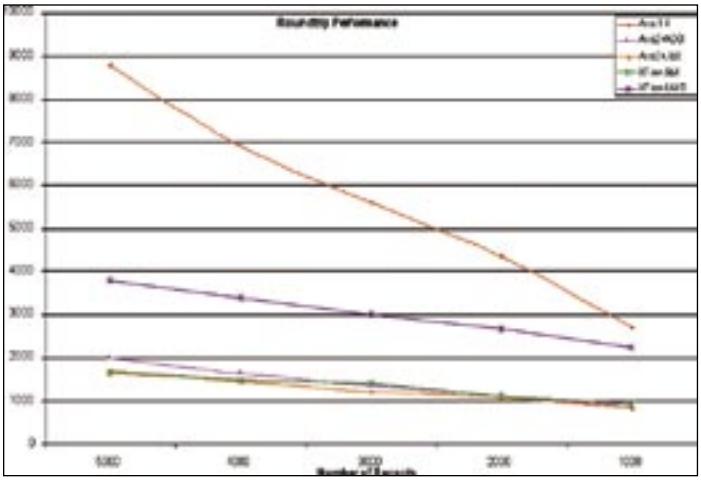


Figure 4 Roundtrip performance of the test cases

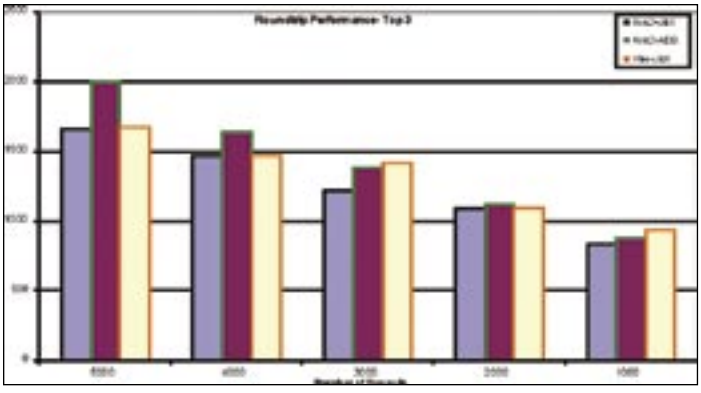


Figure 5 Comparing the top three with respect to dataset size

Load Testing Performance:

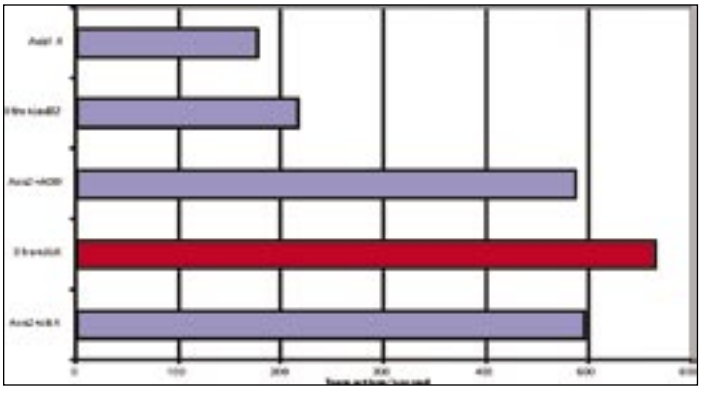


Figure 6 The TPS of the test cases

Un-marshalling Performance

As seen in Figure 2, XFire with JibX data binding offers the best un-marshalling performance. It's also observed that Axis 1.4 fared the worst, doing about 12 times worse than XFire with JibX, on average.

It's interesting to note that XFire with JibX, on average, fared four times better than its nearest competitor, Axis2 with JibX.

Marshalling Performance

From Figure 3, it can be seen that both Axis2-based test cases and XFire with JibX match marshalling performance very closely among themselves for small datasets, however, as the objects increase in size, XFire with JibX seems to gain a marginal lead in marshalling performance.

Axis 1.4 was the worst, doing four times worse than XFire with JibX. XFire with JaxB fared quite badly too – taking about twice as much time as XFire with JibX.

Round-Trip Performance

From Figure 4, you can see that both Axis2 with JibX and XFire with JibX perform almost identically with respect to round-trip time.

Although Axis2+ADB match their performance for small datasets, Figure 5 shows that its performance takes a hit with larger datasets. Axis2 with JibX is, on average, 6% faster than XFire with JibX with respect to round-trip performance, according to the data collected. Since this result wasn't entirely consistent with the findings regarding marshalling and un-marshalling time, where XFire with JibX consistently gave faster performance, we decided to investigate how and why Axis2 with JibX managed to gain a slight edge over XFire with JibX on the round-trip, keeping in mind that the latter outperformed the former in all other critical performance tests. We hypothesized that XFire's client-side marshalling and un-marshalling might be slightly sluggish when compared to that offered by Axis2's client-side API. To confirm this theory, we created an XML-HTTP client to invoke the corresponding Web Services and measured the time elapsed between sending the request and receiving a reply at the client end. Since this HTML client sent and received the actual SOAP messages, we were able to exclude the processing overhead of marshalling and un-marshalling on the client end.

We consistently saw that XFire with JibX was quicker compared to Axis2 with JibX, proving that our hypothesis explained the anomaly.

Axis 1.4 seems to be the slowest (4.37 times slower than Axis2+JibX) followed by XFire with JaxB (2.43 times slower than Axis2+JibX).

Load Testing Performance

In the second phase of testing, a load test was done on all five test cases. These results give us statistics to help compare their scalability, speed, and throughput.

Scalability

The Transactions per Second (TPS) achieved by a Web Service is a good indicator of its scalability. The average TPS of the five test cases are available in Figure 6. From Figure 6, it's inferred that XFire with JibX was able to handle approximately 564 TPS (on average), about 14% more than its nearest competitor, Axis2 with JibX. Axis 1.4 scaled the worst, achieving only about 177 TPS on average.

Figure 7 Average Transaction Response Time in seconds Throughput:

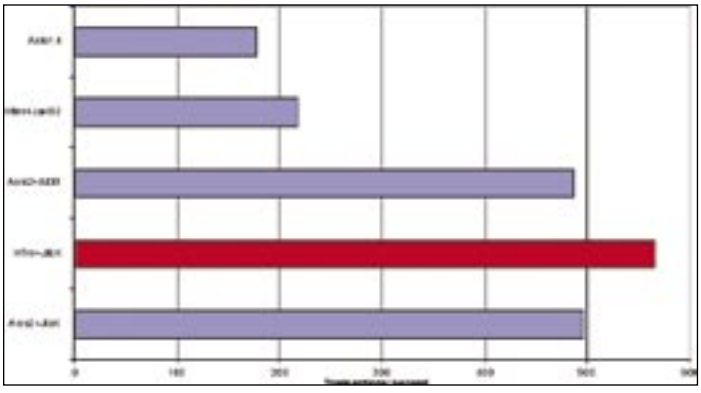


Figure 8 Throughput comparison

Speed

The Average Transaction Response Time is a good indicator of the server-side speed of a particular combination, as it indicates how quickly the server could process the client's request. From the summary graph in Figure 7, it's inferred that XFire with JibX had the smallest average Transaction Response Time, followed by the two Axis2-based applications.

Throughput

Throughput is a measure of performance in terms of how much data the service was able to push to the client in one second. As this figure can be a function of both the number of requests the service can handle per second, as well as the speed with which it can process a client's request, it offers a very comprehensive view of the overall performance of the test cases. From Figure 8, it can be seen that XFire with JibX achieved the greatest throughput, indicating that it offers the best overall performance.

Ease of Use

- ***Axis2 with ADB:*** Axis Data Binding is well integrated with the Axis2 Engine – development using this option is very quick – once the WSDL of the service is available, the WSDL2JAVA tool will generate all classes and stubs required for both the service and a client (except the business logic, of course!). The documentation on Apache's website is detailed and easy to follow.
- ***Axis2 with JibX:*** Axis2 supports JibX, although, by the Axis2 team's own admission, the support seems to be incomplete. Axis2's WSDL2JAVA tool supports generating client- and server- side stubs for JibX binding, however, it doesn't support code-generation for a Web Service from a given schema. Although there are tools available to handle some of the code generation, when this was written, they were only mature enough to handle simple cases; nonetheless, they did provide a good starting point. However, the entire process seemed more cumbersome than using XFire.
- ***XFire with JibX:*** XFire supports JibX binding reasonably well, although the current version of XFire WSGen tool can't generate client- and server-side stubs with JibX binding given a WSDL, it supports code generation for a Web Service from a schema. Moreover, it's fairly straightforward to write a client for such a service, even without a client-side stub. At the time of writing, the documentation wasn't very detailed, but had an easy-to-follow example. Overall, XFire with JibX seemed easier to use and develop than its Axis2 counterpart. However the auto-generated WSDL in XFire JibX combination remains one of the major drawbacks since it didn't confirm to WS-I when this article was written.
- ***Axis 1.4:*** Axis 1.4 is currently one of the most widely used Web Services platforms on the enterprise Web Service landscape. The platform has good support for WSDL first development in terms of both server-side and client-side artifact generation. However server-side code generated using a WDL first development forces a user to populate business logic inside tool-generated code. The platform has limited support for directly exposing existing Java classes as Web Services. If one has to expose existing Java classes as Web Services, developers often generate WSDL from the Java class and subsequently generate server-side artifacts and deployment descriptors using WSDL2JAVA and subsequently map existing business classes to implementation classes generated by the tool.
- ***XFire with JaxB:*** XFire provides good support for Jaxb binding, it comes with both schema first development and code first generation of a Web Service. However Web Services generated using the schema first approach show significantly better performance compared to code first generated Web Services. XFire comes with a WSGen tool that can be used to generate client- and server-side stubs from a WSDL and also supports a WSDL first development.

Conclusion

By analyzing the data collected in the course of this study, several conclusions were reached.

First, it seems obvious that the performance of third-generation SOAP engines is significantly better than that of its first- and second-generation counterparts.

It's concluded that the JibX Binding Framework offers the best performance and a great deal of flexibility through its binding definition file. However, defining bindings for complicated objects can become very cumbersome and time-consuming.

It's also concluded that XFire with JibX offers the best marshalling and un-marshalling performance, and is only marginally slower than Axis2 with JibX in round-trip performance, which can be largely attributed to the slower

client-side processing by the XFire-based JibX client. XFire also seems to offer the best scalability, speed, and throughput among all test cases. Moreover, the ease of development using JibX binding with XFire over Axis2 with JibX makes this an attractive option.

However, it's worth noting that, if ease of use and development is of essence, Axis2 with ADB offers the best balance between ease and performance – Axis2 with ADB is extremely easy to use (however, it's less flexible than JibX) and offers very decent performance.

REFERENCES

- Web Services @ W3C: www.w3.org/2002/ws/
- "XMill - An Efficient Compressor for XML": http://www.liefke.com/hartmut/xmill/xmill.html
- Paul Sandoz, Alessando Triglia & Santiago Pericas-Geertsen. "Fast Infoset": http://java.sun.com/developer/technicalArticles/xml/fastinfoset/
- Dr. Srinivas Padmanabhuni, Bijoy Majumdar, Ujval Mysore, Vikram Sitaram. "XML Binding Frameworks in the Context of Service-Oriented Architecture": http://soa.sys-con.com/read/114130.htm
- WebServices – SOAP: http://xml.apache.org/soap
- "Apache Axis": http://ws.apache.org/axis/
- "Codehaus XFire": http://xfire.codehaus.org
- "Apache Axis2/Java: Next Generation Web Services": http://ws.apache.org/axis2/
- Why StAX? , http://java.sun.com/webservices /docs/1.6/tutorial/doc /SJSXP2.html
- Common DOM API: java.sun.com/j2se/1.4.2/docs/guide/plugin/dom/index.html
- API overview: java.sun.com/webservices/jaxp/dist/1.1/docs/tutorial/overview/3_apis.html
- "Axis2/Java-JibX Integration with Axis2": http://ws.apache.org/axis2/1_2/jibx/jibx-codegen-integration.html#intro
- "Codehaus XFire- Client and Server Stub generation form WSDL": http://xfire.codehaus.org/Client+and+Server+Stub+Generation+from+WSDL

(*All Web sites were accessed on June 26, 2007.*)

About the Authors

Krishnendu Kunti is a senior technical specialist with the Web Services Center of Excellence at SETLabs, Infosys Technologies, Hyderabad. He has contributed to architecting, design, and testing of Syndeo, an in-house shared data services and Web Service management framework. His areas of interest are SOA, application architecture, and business execution languages. He has been involved in architecture definition, design and implementation of a number of SOA-based applications in banking, insurance, and automotive domains.
krishnendu_kunti@infosys.com

Rahul Muralidhar is an undergraduate student studying computer engineering at the University of Waterloo, Ontario, Canada. He is currently pursuing an internship with the Web Services Center of Excellence at SETLabs, Infosys Technologies Ltd., Hyderabad, India. His current areas of interest include optimizing the processing and transfer of large XML documents to improve Web Services performance.

Nagarani Badveeti is a member of the Web Services COE (Center of Excellence) at SETLabs, Infosys Technologies. She has extensively worked on Web Services implementation projects and has been involved in creating a SOA-based integration platform for an auto dealer management systems using the STAR schema. Her fields of interest include SOA-enabling technologies, Web Services, and XML technologies.

—continued from page 34

Loosely Coupled...What?

Now that we know the basic differences between a tightly and loosely coupled architecture, as well as the advantages, perhaps it's a good idea to break down loose coupling into a few basic patterns: location independence, communication independence, security independence, and instance independence.

Location independence refers to the notion that it matters not where the service exists, the other components that need to leverage the service can discovery it within a directory and leverage it through the late binding process. This comes in handy when you're leveraging services that are consistently changing physical and logical locations, especially services outside your organization that you may not own. Your risk calculation service may exist in LA on Monday and in New York on Friday, and it should make no difference to you.

Dynamic discovery is key to this, meaning that calling components can locate service information as needed, and without having to bind tightly to the service. Typically, these services are private, shared, or public service as they exist within the directory. Communications independence means that all components can talk to each other no matter how they communicate at the interface or protocol levels. Thus, we leverage enabling standards, such as Web services, to mediate the protocol and interface difference. Security independence refers to the concept of mediating the difference between security models in and between components. This is a bit difficult to pull off, but necessary to any SOA. To enable this pattern, you'll have to leverage a federated security system that's able to create trust between components, no matter which security model is local to the components. This has been the primary force behind the number of federated security standards that have emerged in support of a loosely coupled model and Web services.

Instance independence means that the architecture should support component-to-component communications, using both a synchronous and asynchronous model, and not require that the other component be in any particular state before receiving the request, or the message. Thus, if done right, all of the services should be able to service any requesting component, asynchronously, as well as retain and manage state no matter what the sequencing is.

The need for loosely coupled architecture within your SOA is really not the question. If you have a SOA, you should have a loosely coupled architecture, if done correctly. However, analysis and planning are also part of the mix...understanding your requirements and how each component of your architecture should leverage the other components of your architecture. With a bit of up-front work, you'll find your coupling loose and your SOA successful.

IT Operations – A Suitable Test Bed for SOA

Orchestrating service-enabled Perl scripts and command line tasks into IT processes

WRITTEN BY MICHAEL RULF, MARKUS ZIRN, AND RAJIV TAORI

You are equipped with a technical understanding of Web Services. You are a strong believer in the power of Service Oriented Architecture (SOA). Now you're eager to bring SOA to your enterprise. You want to get maximum benefit from SOA, so you propose to service-enable the key functions of your company's enterprise resource planning (ERP) and customer relationship management (CRM) applications and automate cross-application processes like order-to-cash.

Quickly, you realize that this means significant change to the organization. Unfortunately, most human beings and organizations resist change. Your CIO is doubtful. He doesn't want to spend any resources on anything that isn't absolutely necessary. He demands 100% proven return on investment. Functional users are limited to their functional silos. Having discussions about cross-application processes requires 20 people to be in one room and still nobody has ownership of the end-to-end process.

Don't be discouraged. There are many ways to get started with SOA. Rather than attacking the most complex cross-application business processes, you can, for example, apply SOA principles to traditional data integration challenges. You can also focus on hot topics such as Web self-service to demonstrate the power of Web Services. In this article, we'll show you another interesting approach to how you can get started with SOA in a less-daunting setting.

We'll show you how you can practice SOA and build up value step-by-step. The first is to service-enable small units of existing custom code. Second, we'll show how you can then reuse these Web Services as part of small process flows. Third, these smaller process flows become useful sub-processes of end-to-end business flows. That was your goal, right?

Let's reveal the secret: we believe IT operations are an interesting test bed for SOA, specifically for SOA-based process management. IT processes are what the CIO understands best. IT processes aren't as organizationally challenging as large end-to-end business processes. Yet, complex IT environments have created IT operations that are ready for process improvement. Today, system administrator salaries are the most prominent IT budget line item, and those highly paid resources are executing mundane, ad hoc, and unscalable tasks. It's no wonder then that ITIL (Information Technology Infrastructure Library), a widely accepted framework for IT Service Management, is today's "hot topic." Hence, it's likely your CIO will listen to your proposal to apply SOA to the IT operations challenge. The good news is that system administrators have written large amounts of custom code in the form of Perl code, Unix shell scripts, or Visual Basic (VB) scripts. These programmatic assets are ideal candidates to apply your skills of service-enabling IT assets.

USinternetworking, Inc. (USi), an AT&T company, made use of

this favorable environment for its endeavor into SOA. Let's learn from USi's success. Concretely, we'll explain how you can quickly create a Web Service from an existing Perl script. Then, we'll reuse the service-enabled Perl script as part of a typical IT process and show how you can incorporate this IT process into a larger end-to-end business process. Finally, we'll dive into the world of IT processes to give you a better feel for the opportunity and how BPEL (Business Process Execution Language) is suited to this domain.

Service-Enabling a PERL Script

The first step in applying SOA in IT operations is to create a set of Web Services. On its quest to automate IT processes, USi made an important discovery: many of the tasks required already existed as Unix shell scripts, Perl code, and VB scripts. System administrators created such code components to simplify their job by automating repetitive tasks and generating the necessary documentation required for audit purposes. In the spirit of SOA, USi decided to reuse these pockets of custom code. By creating some basic "wrappers," these legacy code components were converted into Web Services. By standardizing this wrapper definition, it's possible to take disparate custom code created by a wide range of individuals and expose it as part of a SOA initiative with a common methodology for instantiation, security, and error reporting.

Let's take a look at how you can create such a wrapper for a legacy code component. If you come from the Unix/Linux world, you might be familiar with legacy scripts and code packages written using Perl. USi's Quadir Kareemullah demonstrates how easy it is to convert Perl code, in this case a script to create a Unix account for a given Linux server, into a Web Service. Listing 1 shows UserAdd.pm, the initial Perl script.

Let's create a wrapping package that will expose UserAdd.pm via SOAP (Simple Object Access Protocol). The "use UserAdd" statement tells the new module to use our legacy code in UserAdd.pm. The two SOAP lines that follow are what enable the module to be called as a Web Service. A "sub" section then follows for each function in the original Perl module. A separate wrapping module lets you add additional SOAP error handling to conform to a Web Services methodology as found in the "die" statement near the end of the script. This wrapping package, UserAddWS.pl, is shown in Listing 2.

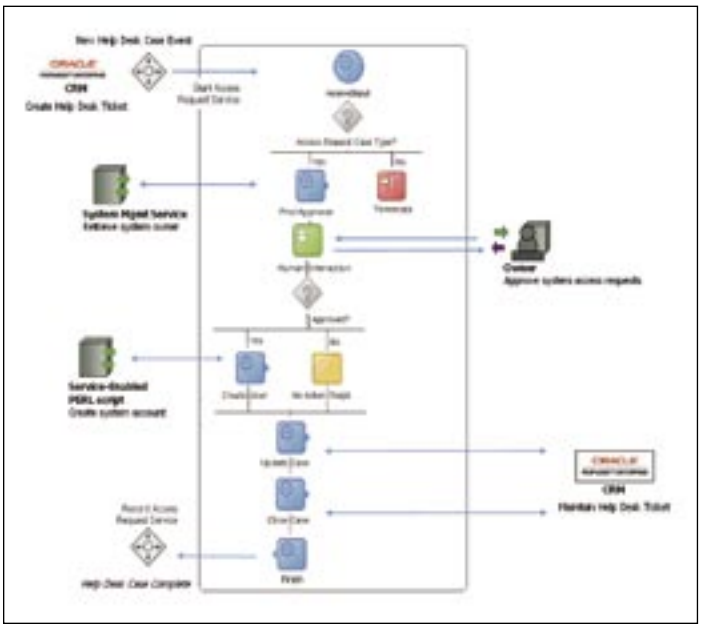


Figure 1 SOA process orchestration

The result is that the original code continues to exist without any change. A significant benefit is that no regression testing is required for adding the wrapper in this non-intrusive manner.

In addition, other features can be introduced to UserAddWS.pl to take care of security. Now that the code is exposed as a Web Service, traditional application security that restricts access to the functionality is no longer in effect. However, support for standards such as WS-Security (Web Services Security) to manage access at the wrapper and WSDL (Web Services Description Language) level can be added without impacting the original application code.

The final step is to create a WSDL definition for the new package so that it can be called by a BPEL process or via statically typed languages such as Java or VB. If you search the Comprehensive Perl Archive Network (www.cpan.org), you'll find Perl modules that help automate the creation of WSDL files. The WSDL for our example code, UserAddWS.wsdl, is shown in Listing 3.

In this WSDL, you can see a number of components relating to the UserAddWS.pl package, including the:

- Namespace definitions
- Request and response message types for each function defined in the package
- Port definition with an operation entry for each function in the package tied to the associated message types
- Binding definition with operation entries to tie all of the definitions together
- Service definition that defines the URL necessary to call the UserAddWS.pl package

As you've seen, the new Perl wrapper "UserAddWS.pl" and the WSDL file "UserAddWS.wsdl" together enable you to convert the legacy Perl package "UserAdd.pm" into a Web Service. Following this methodology, you can rapidly service-enable an entire library of scripts used for IT operations.

Oracle's Enterprise Manager system management offering serves as another example how to make use of this technique to service-enable applications relevant to IT operations. The Enterprise Manager Command Line Interface (EM CLI) lets you access functionality from text-based consoles (shells and command windows) with custom scripts such as Perl, SQL*Plus, OS shell, or Tcl.

Hence, by using EM CLI in a Perl script and then exposing the Perl script as a Web Service as illustrated earlier, you can create Web Services for system management operations like monitoring/managing targets, jobs, groups, blackouts, notifications, and alerts. Concretely, sample Web Services could include:

- IT infrastructure provisioning actions such as adding/deleting targets, submitting/deleting jobs, creating/deleting users
- Monitoring functions such as every day send an e-mail list of backup jobs that were still running after 6am
- Management actions such as every week write pertinent information about failed system management jobs to a file and then purge the system management job history.

Listing 4 illustrates an EM CLI example that can be used to provision a hardware server, using configuration properties from the input file.

Our "emcli provision" submits a job that will provision the image file located under the filepath "Images/myimage" on the hard-

ware server target specified in the variable "mylabel." This system management job will run immediately with a reset timeout of 100 minutes. Image properties will be picked up from the file properties.txt overwriting default image properties. The server "stageserver.us.oracle.com:/private/share" will be used as the staging server and "/private/share" will serve as the staging storage for this task executed with "joe" as the username.

Leveraging the technique we described, you can now execute this hardware server provisioning task via a Web Service by launching this command in a Perl script and wrapping the Perl script as a Web Service.

The main advantage in using Perl or shell scripts as a basis for Web Services is that this technique can be very widely applied. Most platforms provide command-line utilities to interact with a multitude of programs, all of which are available to scripting languages. This technique also transcends the Unix world. For example, Microsoft provides administrative utilities under Windows for Exchange and other products via PowerShell. Using Perl, you can invoke any of these commands programmatically and, in turn, expose them as Web Services as illustrated.

Automating an IT Process That Uses the Service-Enabled PERL Script

Now, let's create some value for the organization around the Perl script that we've turned into a Web Service. First, we want to reduce manual intervention, especially by a specialist such as the system administrator, who should worry about keeping the servers up and running, not executing mundane tasks such as provisioning accounts. There's a shortage of skilled system administrators and, besides, any manual intervention is error-prone and comes with a time delay (because many system administrators work off account provisioning requests in batches). Second, auditability is key in IT operations. Large data centers need to conform to audit standards such as SAS 70 (Statement of Auditing Standards No. 70). Frankly, for an application service provider such as USi, this was a key driver for applying SOA to IT operations. USi must comply with a number of audit standards, such as SAS 70 Type II, as part of its own corporate policies and procedures. In addition, USi is also subject to audit reviews by its clients in support of their compliance initiatives for Sarbanes-Oxley, HIPAA, and other regulatory legislation.

Figure 1 illustrates how the service-enabled Perl script is embedded into an end-to-end IT operations process orchestrated by BPEL — the SOA process orchestration standard.

Besides the Perl script, two typical IT operations applications participate in this process. The first is an IT help desk system. USi uses PeopleSoft CRM as its help desk system. Your company might use help desk software from Remedy or Peregrine. Any help desk system has interfaces to interact with a BPEL orchestration engine. The second system is a system management application. For USi, this is a custom-developed application, but it could also be a system such as HP OpenView, IBM Tivoli, or Oracle Enterprise Manager, among others. Again, these systems either have ready-made Web Service interfaces or you can Web Service-enable their command-line interfaces.

The process kicks off with a request for a new Unix account on a specific server entered into the help desk system. It then queries the system management application to identify the server's owner. This owner is asked to approve the request in a human workflow step. Oracle BPEL Process Manager provides extensions to easily implement such human workflow steps. Once approved, the Perl script is called to execute the account provisioning request. Then, the help desk ticket is updated and finally closed.

This process automation produces some major benefits. First, the approval step is electronically documented, which meets SAS 70 Type II audit standards. Second, we limited any waiting time to the necessary human approval step. Every other activity, including the automatic lookup of the owner as well as the execution of the Perl script, is fully automated. The process can be initiated as a Web self-service request with the help desk system simply tracking the activity. Importantly, every BPEL process is again a Web Service. This means that this process can be immediately reused as part of larger business processes. In our case, a fitting context would be an HR employee onboarding process with the creation of a Unix account being one of many other steps. The advantages of repeatability, auditability, and the reduced execution times of the IT operations process are now inherited by the higher-level HR onboarding process, which is important to executive management. SOA has made a true difference to the business.

Your Opportunity to Apply SOA to ITIL Processes

The Unix account creation process was just a very simple example of an IT operations process where SOA can add value. We tried to exemplify how the combination of (1) service-enabling custom management scripts, (2) process orchestration including human approval steps, and (3) tying in typical applications used in IT operations, such as help desk and system management systems, can help streamline IT operations and even higher-level business processes.

To fully grasp the opportunity, you need to understand the current environment in IT operations. Over the last decade, IT environments have grown more and more heterogeneous and so complex. Separate subgroups in IT, such as networking, server, database, and applications management, often act as silo'd organizations using different help desk tools. Moreover, technology components, such as servers, networks, storage solutions, and applications, come with their own management point solutions — narrow in focus and simply inadequate for delivering on the automation and control required. This results in IT operations teams implementing highly inefficient processes that involve significant manual effort. It therefore comes as no surprise that 70% of the IT budget is spent on maintenance, with administration headcount being the most prominent budget line item; 60%-80% of outages are due to human error; and highly paid system administrators are executing mundane, ad hoc, unscalable tasks. In brief: IT operations is ready for process improvement.

We are certainly not the first to point out this dilemma. IT managers are under increasing pressure to deliver on demanding service requests and to adapt to changing business needs. Processes in search of optimization include a broad spectrum of different tasks, such as:

- IT incident management (fault/problem remediation)
- IT problem management (root cause analysis)
- IT configuration management
- IT change management
- IT release management

IT organizations are realizing that they need enforceable and repeatable IT processes to improve the efficiency, effectiveness, and quality of their IT services (control); drive compliance with regulatory requirements such as Sarbanes-Oxley (compliance); and reliably support key business needs (alignment). Hence, they're looking to proven best practices as a means of gaining control. In this context, the ITIL IT Service Management process framework has become popular as guidance on how to tame the chaos that often thrives in modern data centers. Started in the mid-1980s by

Visit the *New*

www.SYS-CON.com

Website Today!

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

24/7

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

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

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

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

EDUCATION
The world's leading online i-Technology university

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

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

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

JUMP TO THE LEADING i-TECHNOLOGY WEBSITES:

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

the United Kingdom's Office of Government Commerce, formerly known as the Central Computer and Telecommunications Agency, ITIL is now a mature, non-proprietary IT process framework that is industry- and technology-independent. ITIL has taken off over the last years, with an adoption rate of 45% expected in the United States in 2008. In Europe, where ITIL was originated, adoption is already ubiquitous.

Given the significant need for IT process improvement and broad consensus in the ITIL process framework, focus is then squarely on the technical enablement and automation of ITIL processes. BPEL, the SOA standard for process orchestration, provides a solution to implement ITIL process automation. BPEL was designed to leverage existing software functionality from multiple sources, such as ERP systems, service providers, and custom code via Web Service interfaces. It provides orchestration to enforce consistent execution of a defined process across these assets. Furthermore, in BPEL, processes are modular "building blocks" that can be flexibly assembled into larger end-to-end processes.

When USi embarked on IT process automation, it quickly realized that traditional workflow tools would be too restrictive. Today's IT operating environments require a complex network of supporting system and infrastructure components generating a large volume of "events" to continuously sort through. Traditional workflow tools tend to be very linear in terms of how they execute a process flow, with one step following the other. They're not a good fit for the highly event-driven nature of IT management. Also, for USi, the dynamic nature of the approval process across many internal and client systems required an asynchronous solution. This in turn needed to be coupled with a rules engine for defining business logic to determine the necessary approvers at the time of execution. Oracle BPEL Process Manager provided USi with a platform to meet these requirements, including several added features that provide more value from an audit perspective. Each time a process flow executes, the BPEL orchestration engine provides a graphical view of the process flow. This enables end users to determine where a particular approval request currently stands in the overall business process. The BPEL engine also captures all of the task data, such as the approver and time of approval, for presentation to an auditor. And it provides revision control for the process definition to track when and how the process changed over time — invaluable functionality during audit reviews.

Perhaps, most important, USi learned to plan for growth from the outset when implementing Web Services and SOA technologies. Web Services have a way of quickly proliferating when you start automating discrete tasks. If you don't plan for this growth upfront by investing in a UDDI repository, or using other methods for tracking your toolbox of Web Services, you're destined to have multiple services with overlapping functionality. Reuse and repeatability is key to realizing the value of SOA. Using a UDDI repository promotes this by providing a way to locate existing Web Services in your toolkit so you can capitalize on the fruits of the discovery process you went through to find the scripts and automated tasks you converted to Web Services.

For you as an SOA architect or developer, IT operations are a worthwhile opportunity to shine with your SOA skill set. Most certainly, people will be impressed to see how you can service-enable existing Perl code, Unix shell scripts, or VB scripts, and connect them in conjunction with services provided by help desk and system management applications. More than that, you'll be able to demonstrate business value to your CIO and build momentum for a larger SOA initiative in your enterprise. If you want to go beyond the basics of BPEL, the "BPEL Cookbook" (oracle.com/technology/pub/articles/bpel_cookbook/index.html) is a great advanced resource to learn directly from your peers. ■

About the Authors

Michael Rulf is the vice-president of advanced engineering at USinternetworking, Inc., an AT&T Company. At USi, he provides the strategic direction and management of all product-engineering initiatives with a primary focus on Oracle's ERP products such as Siebel, Oracle E-Business Suite, and PeopleSoft. This includes setting strategic direction and managing the development of various SOA initiatives encompassing Oracle ERP products, Microsoft business productivity products such as SharePoint, and e-commerce solutions such as IBM WebSphere Commerce. Recently, he designed and implemented an Oracle-based identity management system that provisions users with a range of applications using an SOA architecture, resulting in significant industry recognition.
michael.rulf@usi.com

Markus Zirn is a senior director of product management for Oracle Fusion Middleware. He heads the Strategic Customer Program, where he works with Oracle's most innovative middleware customers. Recently, he produced the "SOA Best Practices — The BPEL Cookbook" series on the Oracle Technology Network. He has practical experience designing and optimizing business processes — conducting multiple business process re-engineering projects while a consultant with Booz Allen Hamilton. He holds a master's degree in electrical engineering from the University of Karlsruhe, Germany; the University of Southampton, U.K.; and ESIEE, France.
markus.zirn@oracle.com

Rajiv Taori is the director of product management for Oracle Enterprise Manager, where he's responsible for its application management offerings. He is ITIL-certified and has consulted and worked with companies on ITIL projects for more than five years. He has also worked as a management consultant with McKinsey & Company where he advised clients on their business, product, and technology strategies. He holds an MBA from the Wharton School of the University of Pennsylvania and a bachelor's degree in electrical engineering from the Indian Institute of Technology (IIT), Bombay.
rajiv.taori@oracle.com

Listing 1: UserAdd.pm

```
#!/usr/local/bin/perl
#
# Author: Quadir Kareemullah USi
#
# Purpose: UNIX account creation module
#
package UserAdd;

use strict;
use warnings;

use Data::Dumper;

my $groupadd_cmd = "groupadd";
my $useradd_cmd = "useradd";
my $def_homedir = "/home";
my $def_shell = "/bin/bash";

sub useradd {
  my ($fullname, $username) = shift;

  my $result = { error => 1, status => "Could not run useradd() function" };

  my $cmd = "$groupadd_cmd $username 2>&1";
  my $out = `$cmd`;
  if($out =~ /exists/) {
```

```
     $result->{error} = 1;
     $result->{status} = "Could not add $username user, Group $username
already exists.";
  }
  else {
    $cmd = qq($useradd_cmd -c "$fullname" -d $def_homedir/$username -s
$def_shell -g $username $username  2>&1);
    $out = `$cmd`;
    if($out =~ /exists/) {
      $result->{error} = 2;
      $result->{status} .= "Could not add $username user, User $username
already exists.";
    }
    else {
      $result->{error} = 0;
      $result->{status} = "User $username successfully added.";
    }
  }
}

1;
```

Listing 2: UserAddWS.pl

```
#!/usr/local/bin/perl

package UserAddWS;

use strict;
use warnings;
use diagnostics;

use Data::Dumper;

use lib './modules';
use UserAdd;

use SOAP::Transport::HTTP;

SOAP::Transport::HTTP::CGI
-> dispatch_to('UserAddWS')
-> handle;

sub useradd {
  my $self = shift;

  my $result = UserAdd::useradd(@_);

  if($result->{error}) {
     die SOAP::Fault->faultcode( $result->{error} )
       ->faultstring( $result->{status} );
  }

  return $result->{status};
}

1;
```

Listing 3: UserAddWS.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>

<definitions name="UserAddWS"
  targetNamespace="http://mdsxaaw05.usi.net/wsdl/UserAddWS.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://mdsxaaw05.usi.net/wsdl/UserAddWS.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">


  <message name="UserAddRequest">
    <part name="fullname" type="xsd:string"/>
    <part name="username" type="xsd:string"/>
  </message>

  <message name="UserAddResponse">
    <part name="status" type="xsd:string"/>
  </message>

  <portType name="UserAddWS_PortType">

    <operation name="useradd">
      <input message="tns:UserAddRequest"/>
      <output message="tns:UserAddResponse"/>
    </operation>

  </portType>

  <binding name="UserAddWS_Binding" type="tns:UserAddWS_PortType">

    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http"/>

    <operation name="useradd">

      <soap:operation soapAction=""/>

      <input>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:UserAddWS"
          use="encoded"/>
      </input>

      <output>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:UserAddWS"
          use="encoded"/>
      </output>

    </operation>

  </binding>

  <service name="UserAddWS_Service">

    <documentation>WSDL File for UserAddWS</documentation>

    <port binding="tns:UserAddWS_Binding" name="UserAddWS_Port">
      <soap:address
        location="http://mdsxaaw05.usi.net/soap/UserAddWS.pl"/>
    </port>

  </service>

</definitions>
```

Listing 4: emcli provision

```
emcli provision
      -image="Images/myimage"
      -network="Networks/networkprofile"
      -bootserver="booservername.us.oracle.com"
      -stageserver="stageserver.us.oracle.com:/private/share"
      -stgcredentials="joe"
      -schedule="type:immediate"
      -resettimeout="100"
      -target="mylabel"
      -input_file="config_properties:properties.txt"
      -assignment="provision mylabel"
```

Stylus Studio XML IDE

Providing tools to work efficiently with XML

WRITTEN BY **VIJAY PHAGURA**

Extensible Markup Language, more popularly known as XML, has found its uses in many areas of software products. Though XML offers a lot of features and uses, it's hard to develop with and maintain without good tools. This review discusses such a tool that would help XML and Java developers use XML efficiently.

Introduction

Stylus Studio is not only an XML tool but is also an Integrated Development Environment (IDE). It provides the developer with various tools to work efficiently with XML and related applications. It can be downloaded from the site listed in the References section.

Once you download it, it is a simple install. During installation, it will ask the usual questions about the directory location but you can choose the defaults that it provides. Once it's installed and you have launched it, it will ask for the license key and some registration info. Doing all that will bring up the first screen as shown in Figure 1.

Document Creation

There are a variety of documents that can be created using Stylus Studio, some of which are:

1. DTD Schema
2. XML Schema
3. XML Document
4. XQuery File
5. WSDL
6. XSLT Stylesheet

The creation process is very simple and intuitive. Moreover, there are lots of converters and wizards that Stylus Studio provides; for instance, there are many occasions when you have an XML file and want to generate an XSD from it. With Stylus Studio it's just a few clicks away. It provides a converter just for doing that. There is also a custom converter with which a user can convert files to XML format.

There are many more wizards that the Stylus Studio provides such as converting an XSD to XML document, and DTD to XSD. See Figure 2 for a snapshot of the wizard.

XML Editing

Stylus Studio provides a variety of views and tools to edit and create XML files. A user can open an XML file in Stylus Studio and it's ready for editing. The XML can be formatted so that all the tags are aligned to provide more readability just with a click of a button. The elements and attributes are easily identifiable with distinct colors that can also be customized by the user.

Stylus Studio automatically provides multiple views of an XML file in different tabs. Three types of views are provided: text, tree, and grid. These views are helpful during development. Figures 3 and 4 show the tree and grid views. The actual XML file is presented in Listing 1.

An XML schema or a DTD can be easily attached to an XML document and the document can then be validated by clicking the validation button. As soon as a document is attached to an XSD or DTD, the document is validated immediately and the user is made aware of any violations present in the document. This makes it very easy for the user to detect errors and correct them at runtime as they are creating or viewing the XML document.

XML Schema

As mentioned above, it's very easy to create an XML schema from an already existing XML document. As an illustration, I have converted the books.xml document to an XSD as shown in Listing 2.

In Figure 5, the editor shows an accurate graphical view of the XSD, making it very easy to edit and debug. Apart from the presented view of the XSD, Stylus Studio also provides a tree view. A really smooth feature of Stylus Studio is the documentation of the XSD. Any developer would know how tedious and frustrating it is to document an XSD. With Stylus Studio, it's easy and it generates the entire basic XSD document.

The XSD can be attached to any XML document and then validated by clicking on the validation button. Any errors the XML might have are clearly displayed.

XQuery and XPath

XQuery is a programming language that leverages XPath for XML document navigation. As with other languages, having the right tool can greatly simplify the application development process, and this is where Stylus Studio shines. Its intuitive, integrated debugger and XPath expression evaluator allow you to quickly and easily debug XQuery. Debugging is supported for XSLT and Java as well as

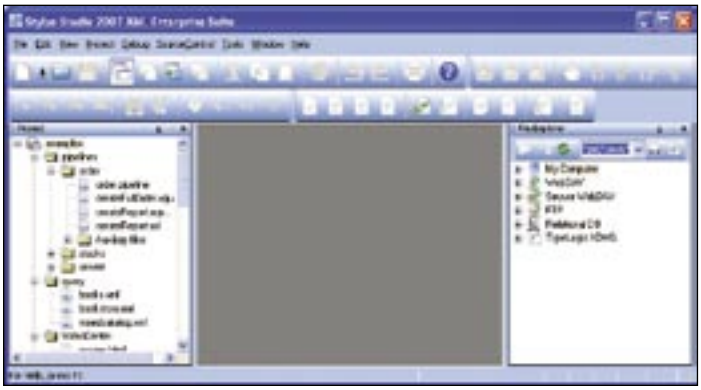


Figure 1

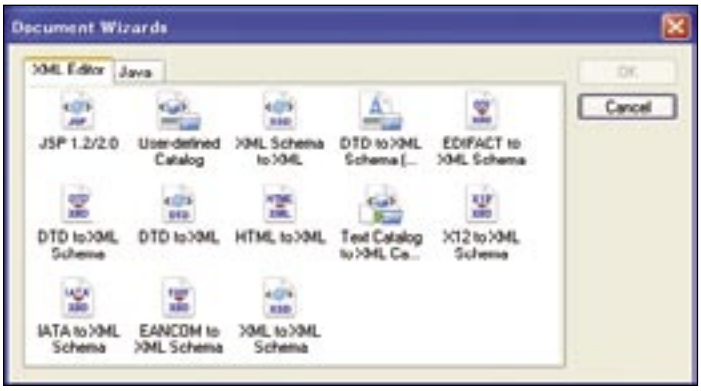


Figure 2

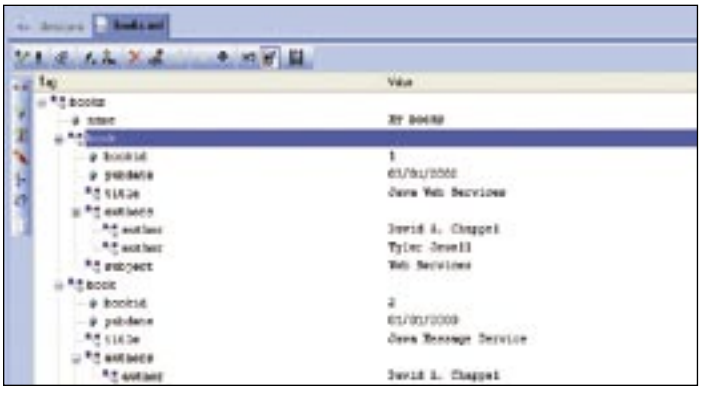


Figure 3

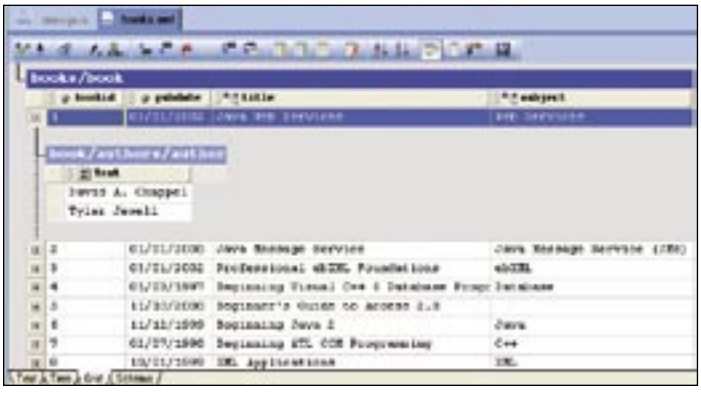


Figure 4

Figure 5

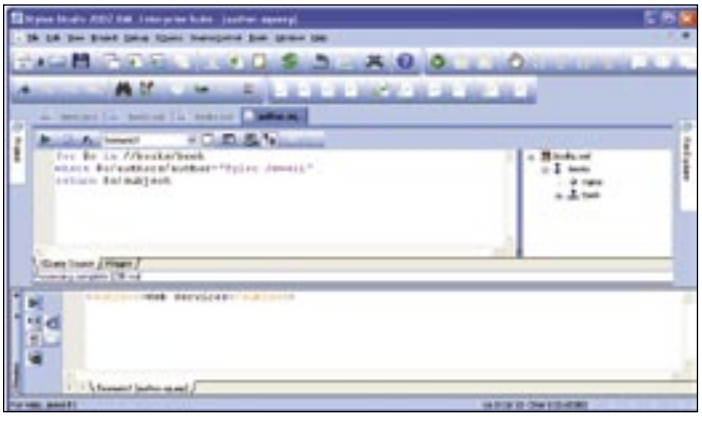


Figure 6

XQuery. This is where Stylus Studio shines. It has a friendly debugger so the XPath/XQuery expressions can be debugged easily.

Here is an example XQuery:

```
for $c in //books/book
where $c/authors/author="Tyler Jewell"
return $c/subject
```

This query is on the books.xml in Listing 1 and returns the subject of the book by author "Tyler Jewell"; the result of this query is shown in Figure 6. Figure 6 also shows Stylus Studio's various panes. As seen in the left-hand side of the Preview pane there are various views and execution buttons. Also, note that the execution of this query is for a particular scenario that is Scenario 1. Stylus Studio allows you to define multiple scenarios so you can evaluate different factors that might influence query performance, such as the source XML file and XQuery processors. Stylus Studio supports numerous XQuery processors, including DataDirect XQuery, which is specially optimized to execute XQuery against relational data. In this example, I used the Saxon 8.9.0.3, another popular XQuery processor supported by Stylus Studio.

The XPath/XQuery debugger is accessible to the user by clicking the "bug" button. Breakpoints can be easily scattered over a query by toggling them and the user can start debugging.

The Java code for the example XQuery can be generated by Stylus

Studio, which can be used as a standalone program or in any other application. Listing 3 shows the Java file generated by Stylus Studio, with appropriate comments in the code about the JARs it needs. As a bonus, Stylus Studio can compile and execute the Java code it generated. Also, users can use the debugger provided by Stylus Studio to debug the Java code.

Summary

Overall I found Stylus Studio very useful and it can save you tons of time and effort while fine-tuning XML applications. This tool can increase developer productivity and the quality of your applications. It performed fairly well compared to similar tools in the market. It was fast to start and restart. Also, it was pretty responsive to view the files and switch between various views.

Stylus Studio is very easy and intuitive to use as the Java IDEs. If you are a Java developer and use Eclipse or other IDEs, using Stylus Studio will come naturally to you

Target Audience	XML/Software Developers
Level	Beginner to Advanced
Pro	• Easy to use • Configurable • Descriptive views • Java source code generator • XPath/XQuery support • DTD and XSD support • Java support
Platforms	• Windows XP • Linux/Unix

Test Environment

Operating System: Win XP Pro
CPU: 2Ghz
Mem: 1G

References

• Stylus Studio: http://www.stylusstudio.com/ !

About the Author

Vijay Phagura is a professional Java/J2EE consultant with over 20 years of experience in software architecture and development. He currently works for a startup; he specializes in designing and developing software using J2EE and other Java technologies.

Listing 1

```
<?xml version="1.0" ?>
- <books name="My books">
- <book bookid="1" pubdate="03/01/2002">
 <title>Java Web Services</title>
- <authors>
 <author>David A. Chappel</author>
 <author>Tyler Jewell</author>
 </authors>
 <subject>Web Services</subject>
 </book>
- <book bookid="2" pubdate="01/01/2000">
 <title>Java Message Service</title>
- <authors>
 <author>David A. Chappel</author>
 </authors>
 <subject>Java Message Service (JMS)</subject>
 </book>
- <book bookid="3" pubdate="03/01/2002">
 <title>Professional ebXML Foundations</title>
- <authors>
 <author>Colleen Evans</author>
 <author>David A. Chappel</author>
 <author>Jean-Jacques Dubray</author>
 <author>Duane Nickull</author>
 <author>Pim van der Eijk</author>
 <author>Vivek Chopra</author>
 <author>Betty Harvey</author>
 <author>Marcel Noordzij</author>
 <author>Jan Vegt</author>
 <author>Tim McGrath</author>
 <author>Bruce Peat</author>
 </authors>
 <subject>ebXML</subject>
 </book>
- <book bookid="4" pubdate="01/03/1997">
 <title>Beginning Visual C++ 6 Database Programming</title>
- <authors>
 <author>John Connell</author>
 <author>Minollo</author>
 </authors>
 <subject>Database</subject>
 </book>
- <book bookid="5" pubdate="11/10/2000">
 <title>Beginner's Guide to Access 2.0</title>
- <authors>
 <author>Wrox Author Team</author>
 <author>minollo@minollo.com</author>
 </authors>
 <subject class="1">Access</subject>
 </book>
- <book bookid="6" pubdate="11/12/1999">
 <title>Beginning Java 2</title>
- <authors>
 <author>Ivor Horton</author>
 </authors>
 <subject>Java</subject>
 </book>
- <book bookid="7" pubdate="02/07/1998">
 <title>Beginning ATL COM Programming</title>
- <authors>
 <author>Richard Grimes</author>
 <author>George Reilly</author>
 <author>Alex Stockton</author>
 <author>Julian Templeman</author>
 </authors>
 <subject>C++</subject>
 </book>
- <book bookid="8" pubdate="10/01/1999">
 <title>XML Applications</title>
- <authors>
 <author>Frank Boumphrey</author>
 <author>Olivia di Renzo</author>
 <author>Jon Duckett</author>
 <author>Joe Graf</author>
 <author>Dave Hollander</author>
 <author>Paul Houle</author>
 <author>Trevor Jenkins</author>
 <author>Peter Jones</author>
 <author>Adrian Kingsley-Hughes</author>
 <author>Kathie Kingsley-Hughes</author>
 <author>Craig McQueen</author>
 <author>Stephen Mohr</author>
 </authors>
 <subject>XML</subject>
 </book>
- <book bookid="9" pubdate="11/12/1999">
 <title>Instant UNIX</title>
- <authors>
 <author>Andrew Evans</author>
 <author>Neil Matthew</author>
 <author>Richard Stones</author>
 </authors>
 <subject>GNU/Linux</subject>
```

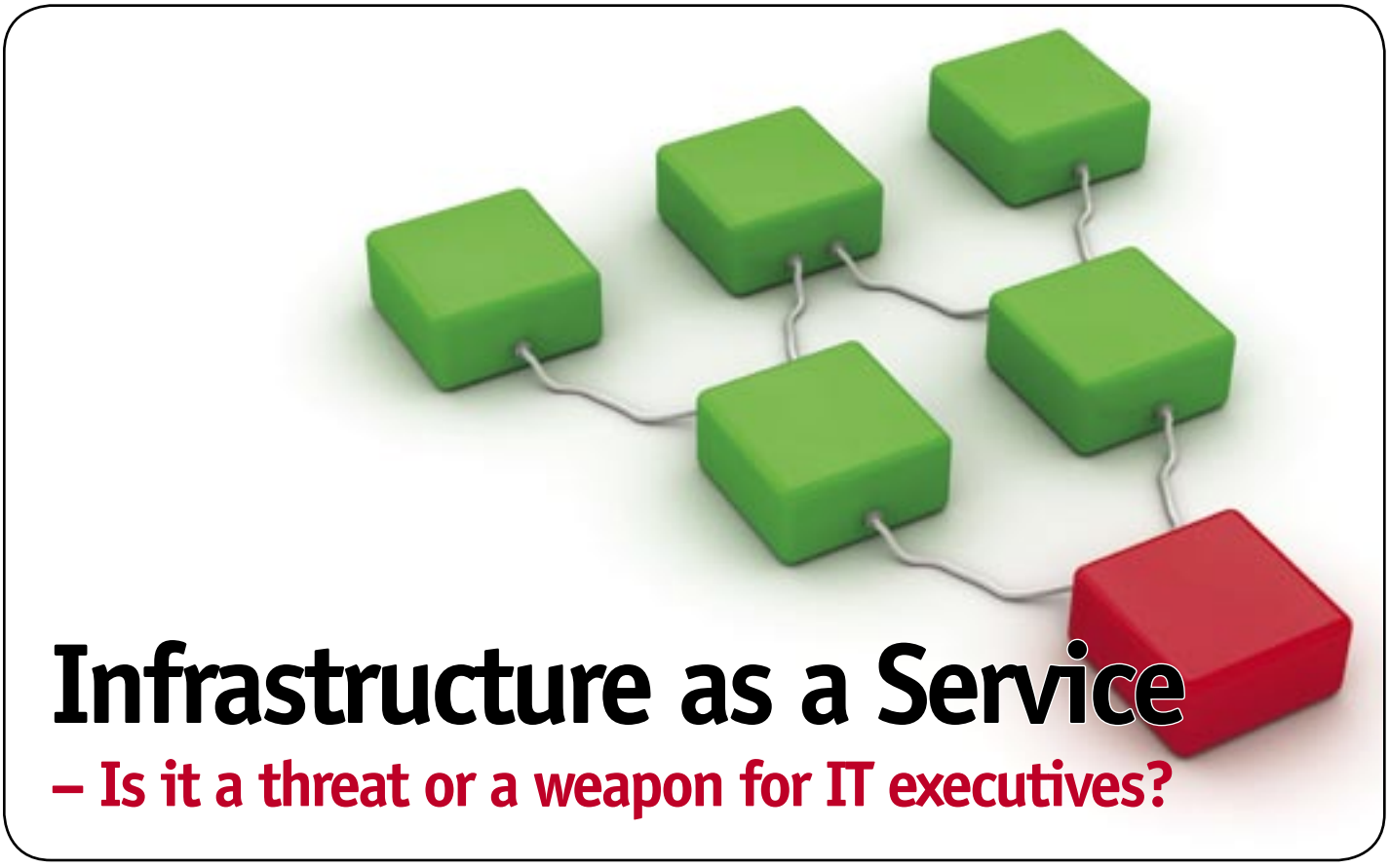
```
 </book>
 </books>
```

Listing 2

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefaul
t="qualified">
  <xs:element name="books">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="book"/>
      </xs:sequence>
      <xs:attribute name="name" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="book">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="title"/>
        <xs:element ref="authors"/>
        <xs:element ref="subject"/>
      </xs:sequence>
      <xs:attribute name="bookid" use="required" type="xs:integer"/>
      <xs:attribute name="pubdate" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="title" type="xs:string"/>
  <xs:element name="authors">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="author"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="author" type="xs:string"/>
  <xs:element name="subject">
    <xs:complexType mixed="true">
      <xs:attribute name="class" type="xs:integer"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 3

```
package com.vsp;

import com.ddtek.xmlconverter.ConverterFactory;
import com.ddtek.xmlconverter.ConverterResolver;
import com.saxonica.validate.SchemaAwareConfiguration;
import java.io.File;
import java.io.FileReader;
import java.io.OutputStreamWriter;
import java.io.Reader;
import java.net.URI;
import javax.xml.transform.stream.StreamResult;
import javax.xml.transform.stream.StreamSource;
import net.sf.saxon.Configuration;
import net.sf.saxon.om.Item;
import net.sf.saxon.om.SequenceIterator;
import net.sf.saxon.om.Validation;
import net.sf.saxon.query.DynamicQueryContext;
import net.sf.saxon.query.StaticQueryContext;
import net.sf.saxon.query.XQueryExpression;
import net.sf.saxon.value.AtomicValue;
import net.sf.saxon.value.SequenceExtent;
import net.sf.saxon.value.UntypedAtomicValue;
import net.sf.saxon.value.Value;
import net.sf.saxon.value.Whitespace;

/** Run an XQuery
 * This class requires the following items in the classpath.
 * C:/Program Files/StylusStudio
 * C:/Program Files/Stylus Studio 2007 XML Enterprise Suite Release
2/bin/XMLConverters.jar
 * C:/Program Files/Stylus Studio 2007 XML Enterprise Suite Release
2/bin/saxon8sa.jar
 */

public class Author {

 public static void main(String[] args) throws Exception {

  ConverterResolver resolver = new ConverterFactory().newResolver();

  String    inputUrl     = "file:///c:/books.xml";
  String    xqueryUrl    = "file:///c:/author.xquery";
  Reader    queryReader  = null;


  try {

   System.out.println();
   System.out.println("XQuery starting.");

   queryReader = new FileReader(new File(new URI(xqueryUrl)));

   Configuration config = getConfiguration(resolver);
   StaticQueryContext staticContext = new StaticQueryContext(config);
   staticContext.setBaseURI(xqueryUrl);

   DynamicQueryContext dynamicContext  = new DynamicQueryContext(config);
   dynamicContext.setContextItem(staticContext.buildDocument(new
StreamSource(inputUrl)));

   XQueryExpression expr = staticContext.compileQuery(queryReader);
   expr.run(dynamicContext, new StreamResult(new OutputStreamWriter(Syst
em.out)), null);

   System.out.println("XQuery finished.");
  } finally {
   if (queryReader != null) queryReader.close();
  }
 }

 private static Configuration getConfiguration(ConverterResolver resolver)
{

  SchemaAwareConfiguration config = new SchemaAwareConfiguration();
  config.getSystemURIResolver().setRecognizeQueryParameters(true);
  config.setValidation(false);
  config.setStripsWhiteSpace(Whitespace.IGNORABLE);
  config.setSchemaValidationMode(Validation.LAX);
  config.setValidationWarnings(true);
  config.setURIResolver(resolver);
  return config;
 }


 private static void bindParameter(StaticQueryContext staticContext,
DynamicQueryContext dynamicContext,
                                   String paramName, String paramValue)
 {
  Value xvalue=null;
  try {
   XQueryExpression xqe = staticContext.compileQuery(paramValue);
   SequenceIterator si = xqe.iterator(dynamicContext).getAnother();
   Item item = si.next();
   if (item != null && si.next() == null && item instanceof AtomicValue)
{
    xvalue = (AtomicValue)item;
   } else {
    xvalue = new SequenceExtent(xqe.iterator(dynamicContext));
   }
  } catch (Throwable t) {
   xvalue = new UntypedAtomicValue(paramValue);
  }

  dynamicContext.setParameter(paramName, xvalue);
 }
}
```



Infrastructure as a Service

– Is it a threat or a weapon for IT executives?

WRITTEN **BY PAT O'DAY**

With the proliferation of computers and data in the business world, it's time the IT department had a strategic weapon that will enable it to operate with the same flexibility as other departments and with more financial transparency. A large part of the answer is a new concept in hardware and data storage known as Infrastructure as a Service (IaaS).

Every year at budget time, IT executives play the same old guessing game. How much will the company grow in the coming year, and what IT assets will it need to support that growth? A mistake in either direction can be financially disastrous and career ending since companies are increasingly holding managers' feet to the fire about meeting their targets. Guessing too high, based on sanguine business projections that may not pan out, or lack of good planning data burdens a company with expensive hardware that may sit underutilized or IT staff that may need to be downsized.

Guessing too low can mean a scramble midway through the year to squeeze existing inadequate computing and storage capacity to their limits, endangering the business's ability to sustain growth and creating an overwhelmed staff that may find it easier to seek employment elsewhere.

The many crucial jobs IT performs for a company are hard enough — provisioning employees and keeping their workstations up and running; protecting data to meet the stringent requirements imposed by Sarbanes-Oxley, HIPAA, PCI, and other regulations; managing data recovery and business continuity; and so on. The risk of operating with inadequate resources or burning unnecessarily through corporate funds are unwelcome addenda to the IT department's burden.

Now imagine a world where you can scale your IT capacity up or down on command without any capital expenditure. Imagine knowing that all your data is safely backed up and can be restored in hours, even if it's totally wiped out. Imagine being able to free your highly skilled IT staff to work on value-added tasks such as development and planning, instead of chasing bugs and installing patches ad infinitum. And now, as the executive in charge of your IT division, imagine doubling your personal market value (read salary) as you free yourself from tactical concerns and build a path to fulfilling a more strategic corporate role than ever before.

This world exists. It's enabled by a new business concept based

on virtualizing the IT environment and is called Infrastructure as a Service. IaaS extends the remote hosting concept of Software as a Service (SaaS) to hardware.

If IaaS is a fairly new concept to you let me provide a basic understanding. Fundamentally, IaaS provides IT resources — processing power, storage, data center space, services, compliance — on-demand, enabling IT to bill these services as a variable fixed cost.

The interest in IaaS can be attributed to significant increases in IT-enabled business models such as e-commerce, Web 2.0 and SaaS, which drive demand, and by advances in technology that enable it, including virtualization, utility computing, and data center automation.

Just-in-Time power and computing capacity sounds like a dream come true to many people, yet to others it might sound like a nightmare in which they lose control of their IT environment, or worse, become redundant themselves as computing tasks are offloaded to an outside supplier. In my (admittedly biased) opinion, IaaS can be viewed as the newest strategic weapon in the IT arsenal, rather than a threat.

Weapon 1: The IT Administrator Takes a More Strategic Role

According to Forrester Research's 2007 worldwide PC adoption forecast, "There will be more than a billion PCs in use by the end of 2008 and more than two billion by 2015 — a 12.3% compound annual growth rate." With that kind of explosive growth in the computer sector, it's clear that the IT administrator's scope of responsibility is going to change dramatically.

When you hire an IT administrator you don't want someone who just knows how to log into a Windows box, find his way around, and get little things to work. You want someone who understands the big picture, who can grasp the importance of the 200 or 2,000 computers in use at your company and how they all operate together, and can manage them as a fleet. That's the Infrastructure as a Service model.

When administrators can manage five to 10 times the number of devices they're managing today, it might sound like a fast track to downsizing, but the real equation is the availability of trained staff in today's IT market, and how best to utilize their talents. As we know, there's a shortage of IT skills, so what companies need is a weapon that will enable them to recruit and retain good people. Since we can't educate new people fast enough, there's also a need for professionals who understand how to do more with less. IaaS is the enabler.

At some point in their careers, IT people realize that they can only exert their influence or philosophy on things they can touch directly. Without a disruptive change, such as scripting or data center automation, their touch is their limitation. The administrators of tomorrow will need to understand how to manage hundreds or even thousands of devices. IaaS doesn't take away responsibility, but adds a strategic dimension to the job of IT, making managers more marketable, because they're now accustomed to working at a higher level.

The idea of a one-server-to-one-administrator model is gone. The only people likely to be threatened by this concept are those who are comfortable with their limitations.

Weapon 2: It Gets Aligned with the Core Business

Every executive wishes IT would be better aligned with and able to support the core business strategy. Today's businesses realize that IT isn't just a tool to help, but is a critical part of day-to-day operations and frequently instrumental in delivering the end product. With e-commerce, B2B portals, EDI, IP phone systems, and even e-mail, today's applications are fully integrated into the business, so it's critical that they behave the way the business does.

The first step to business alignment is cost control. No additional IT dollars should be spent unless it makes a positive impact on revenue. With IaaS and its variable but predictable monthly costs, you can manage spending on a monthly instead of an annual basis.

IaaS enables a whole new and more transparent way of accounting for IT. Today, IT is one line item. You know what you spend on all of your servers, all of the labor to maintain those servers, all of the power, and so on. You tote it up and allocate it by business unit. In the IaaS model, precise usage and costs are transparent down to the resource level — blade servers, operating systems, storage.

Because these resources are modular and transparent, IT can begin to tie utilization more closely to the bill that each business unit receives. This creates a closer link between what the business unit spends and the "service" it gets. Once that link is established, IT can begin to change business unit behavior away from "all-you-can-eat" to prioritization and cost/benefit.

Weapon 3: IaaS Can Be Implemented In-House

Infrastructure as a Service is both a structural concept and a mindset. As such it can be implemented internally; it doesn't have to come from an outside service provider. If you implement IaaS internally, it means the IT department can charge-back its services proportionally to the parts of the company that are the heaviest users. Suppose you're in manufacturing and the production side never has any sizeable needs in terms of IT, but marketing, HR, and accounting are crushing you. When budget cycle comes around, the accounting people, who are draining half your resource, won't allow you any increase. How do you align with that? With IaaS, you can charge them for IT as a service, so if they want more, they have to pay more.

Alternatively, another new solution enabled by IaaS is to put your IT budget inside other departments' budgets. You give them dollars to spend versus IT having to carry and justify those expenses throughout the year. So if marketing decides in the eleventh month that it needs to double its storage, you don't have to come up with a solution for which you don't have the money. Marketing now has to pay for it out of its budget.

Weapon 4: More Opportunities Captured with Just-In-Time Resources

How often do businesses pass up opportunities because they don't know the cost or aren't in a position to scale? Earlier I alluded to the difficulty of planning IT expenditures on an annual basis. The problem is you're planning blind in many cases.

From a business growth standpoint, you have to be ready to expand, but don't want to spend big bucks on IT resources until you absolutely have to. Let's say you plan to add two more Fortune 1000 customers next year. Why would you spend the capital or even set it aside on January 1, not knowing whether those accounts are going to close in March or October? Even if you put the money in the budget and great things happen and you close the new business, you still need a rapid way to provision that environment. IaaS can

help you ramp up — just outsource the infrastructure until you've built your own capability. You can have entire environments up and running in days, sometimes hours, instead of weeks.

Now factor in serendipity. Instead of adding two large accounts, you add five. Oops! Three accounts to go and your budget is gone. No one can tell you when or if the business will land a third, fourth, or fifth large customer. To handle it, apply the same model. After you land that third big customer, send it to an Infrastructure as a Service provider. Once you get budget approval bring it in-house, pull it back when you're ready. It's that easy.

Weapon 5: Change Is Good When IT Is More Flexible

In business, opportunity implies change, and change can always be a challenge. IaaS enables rapid change because it lets companies add or remove infrastructure and services on-demand. While rapid change can impact stability, with IaaS, you can add horsepower to the 60%-80% of your IT environment that's already stable, while gaining more control over the 20%-40% that's in chaos.

One of the misconceptions about IaaS is that when you decide to use this kind of outsourcing, it's a permanent decision. Imagine if you had to grow your infrastructure 10%-20% in 30 days. If you decided to use a service provider to help, you're not permanently stuck in that mode nor have you set a precedent for the future. One of the ideas behind Infrastructure as a Service is that not only can you scale up quickly, you can also scale down or scale out.

Infrastructure as a Service is generally delivered in addition to a utility computing platform. As long as you have a platform like VMware for virtualization, you look identical to your infrastructure provider. So, if you wanted to push 30 machines or 50 machines out of your data center for 90 days, you could easily bring them back because you're both running the same virtual platform.

Weapon 6: Data Center Automation Kicks IT into High Gear

System administrators in today's data center are typically familiar with only 10-20 specific hosts or devices. These are the units that fall under their area of responsibility and are often their area of expertise. With new technologies such as virtualization and high-density computing, the quantity of actual devices in a data center grows rapidly.

Administrators and engineers have to learn how to manage large groups of like devices as fleets. The days of logging into a single box to run patches, tweak the registry, or change permissions are gone. There simply isn't enough time. Data center automation tools like Opsware (soon to be acquired by HP) and BladeLogic are a large part of the IaaS model because they enable a single administrator to manage potentially hundreds of devices.

These tools provide templates and policies for configuration, patch management, and security compliance. You can configure a single template based on best practices or corporate policy and apply it to several hundred machines. A delta report will show you all of the devices that need attention. Built-in automated remediation lets you select all of the devices and apply a single change or group of changes at once. The tools can group devices (if you have different policies for development versus production, for example) as well as exceptions for patches or changes that haven't been tested yet.

Weapon 7: Achieving Regulatory Compliance Is Easier

If ever two words in the English language were designed to strike fear in an IT director's heart, regulatory compliance are those words. Healthcare has HIPAA, publicly traded companies have SOX and COBIT, the e-commerce world has PCI. Once again, IaaS comes to the rescue. Using the underlying features of IaaS, compliance becomes easier. Pre-compliant virtual machines can be kept in a library, giving you a head start when a new application environment has to be deployed. Instead of installing the server from scratch, you can deploy a copy of a pre-configured (and even pre-compliant) virtual machine. Many organizations maintain a stockpile of pre-built virtual machines in a library for this purpose. It also dramatically improves your provisioning time. After the servers are online, using data center automation templates, you can keep the machines in compliance and even monitor their compliance and patch-level status in a dashboard.

You can even charge more for the cost of a compliant environment as opposed to a standard one to offset the additional management and data center automation software licenses. That way the business can decide and use compliance appropriately. Given the inherent flexibility of an IaaS architecture, it can even change its mind and turn compliance on later.

Weapon 8: Business Continuity Is Assured in the Event of a Disaster

From an earthquake to a virus to a malfunctioning sprinkler system, there are hundreds of ways to lose data and only a few really good ways to recover it. So far, IaaS is the best way I've seen. Data can be backed up automatically in real-time to a strategic network of data centers that serve as mirrored storage and backup sites. Thanks to virtualization you can store multiple backup servers on a single physical server at a backup site and greatly reduce your hardware and operating costs. IaaS providers generally sell these as backup "targets" if you don't want to make a capital spend and put your own equipment in a co-location facility.

If a recovery is declared, you'd initially come up on the consolidated backup hardware, but as performance requirements dictated, your IaaS provider can supply additional hardware on-demand until you're back up to full production status. Because virtual machines are bootable, instead of performing a bare metal restore or reinstall, you simply boot up the VMs, which significantly reduces your Recovery Time Objective (RTO). The virtual machines also contain all of that precious custom configuration information that's so often lost or under-documented.

The use of a virtual approach also reduces the issues of hardware compatibility. As long as the virtual machines will run on VMware and as long as VMware is installed on the recovery hardware, they'll boot. Not only does this help recovery, but it makes testing a lot easier. Most, if not all, of this activity can be done remotely since no tapes are involved. This again saves time because your administrators can avoid the airport and get right down to work.

If you consider the growth of the computer sector, the centrality of data and software to business, and the strain on capital resources that imperfect IT planning can cause, you can see why Infrastructure as a Service is perhaps the ultimate weapon for your success..

About the Author

Pat O'Day is the CTO at BlueLock, LLC, (www.bluelock.com) where he is responsible for the design, engineering and commercialization of their "infrastructure-as-as-service" (IaaS) business model. With more than 15 years of experience as an information technology professional, he defines BlueLock's services, architects its unique hosted IT environments, spearheads its strategic vendor alliances and oversees service deployment and delivery.

Measure Twice, Cut Once

Getting trading-partner integration right the first time

WRITTEN BY **JOHN HARDIN**

SOA! B2B! ASAP! There's a building boom underway in Service Oriented Architectures (SOAs) and business-to-business (B2B) integration, and everybody's rushing to get in on it. After all, integrating application assets and business processes is no longer just an option; it's a necessity. You've got a lot to gain if you do it, and plenty to lose if you don't.

So you better get onboard, and fast. Right?

Well, yes and no. Yes, it's urgent. And no, you shouldn't rush it.

No one could blame you for wanting to plunge right in and get something – anything! – up and running. But when it comes to building a really solid infrastructure for B2B integration in an SOA, planning is everything. You're doing something that's fundamentally different from the way you've always done things before. You're bringing together entities with different systems, cultures, and ways of doing business. And to get it right, you're going to need to proceed carefully.

That's doesn't mean planning it to death, or missing opportunities because you were too busy getting ready for them. But it does mean answering some basic questions before you proceed.

- What's your first B2B project going to be?
- Who's going to be involved?
- How are you going to secure it? Scale it?

Think through everything on the front end, and you'll save time and trouble later. Here are some things you can do to help ensure your success.

Pick the Right Project, Partners, and Protocols

Make your first trading partner integration project the easiest one. Choose partners that already understand B2B as well as the related IT and integration issues; partners who know why the old one-off connections won't work in multiple-partner scenarios, and who've given some serious thought to integration alternatives. Even better, consider partners who already have B2B gateway infrastructures in place.

Once you know who you're working with, you'll need to choose the message-exchange standards and protocols that you'll follow to communicate with each other. (The next section of this article tells you more about how to do that.) Finish laying the groundwork by setting up a simple "request out/response in" message flow for sending and receiving information. Doing these things at the outset will make the rest of the planning process – not to mention the actual project implementation – go smoothly.

Frame It with Standards

If you have dozens, hundreds, or thousands of different trading partners, they will, of course, have dozens, hundreds, or thousands of different ways of doing things. That makes standards extremely important in B2B integration with multiple partners. Standards enable you to have a common set of processes for sharing access to resources that you can apply consistently across B2B projects.

For example, you're going to need a set of standards for exchanging messages and data with trading partners. So you need to choose standards and protocols that cover your specific communication needs, based on factors like the type of business you're in and the degree to which you need to leverage existing standards. Ideally, you'll want to be able to support a variety of protocols and standards that will work with different projects that may come up.

Most B2B transactions today are based on the EDI standard. It's been around for a long time, it's tried and true, and it's widely used. But it's also gradually being eclipsed by standards like OASIS, UBL, and xCBL that are based on the XML message format. XML appears to be on the way to becoming the norm for business communications. It's not there yet, however.

For example, in healthcare, it's hard to justify replacing EDI, because thousands of healthcare organizations have already implemented EDI standards. So what do you do? Split the difference and set up your integration infrastructure to support both EDI and XML. (And be sure the B2B integration software you select is a standards-based product that supports a wide range of standards.)

EDI, OASIS, UBL, and xCBL are all cross-industry standards that are freely available to anyone. But there are also a number XML-based, industry-specific standards around which you can build a standards framework, including:

- AIAG (automotive)
- CIDX (chemical)

- FIXML, iFX, and SWIFT (financial)
- HIPAA (healthcare)
- PIDX (petroleum / energy)
- RosettaNet (electronics)

Today most industries have at least one XML-or EDI-based protocol. If there's no industry-specific standard for your business, help create one by participating in the standards-setting process of a standards body. Okay, that may not sound like the best place to direct IT resources when there's so much else to do – but it's a golden opportunity to help set the direction of the standard that's going to support your industry.

You can divide the standards protocols up into four sections: transport, delivery, security, and business. Transport protocol choices are HTTP and HTTPS (the most popular choices for transport over the Internet), FTP, SMTP, and others. Selecting a delivery protocol such as AS2 or the ebXML Message Service provides you and your trading partners with several necessary items: the ability to assign a digital signature identifying the sender, and the capability of XML Encryption to encrypt either the entire document or parts of the document. Delivery protocols also define the series of acknowledgments and responses for reliability. So adding a delivery protocol to the mix gives both parties non-repudiation.

Business protocols define the actual business document that is the goal of eBusiness messaging. <existing text re: "Most B2B transactions today are based on the EDI protocol.">

Security protocols are closely tied to the delivery protocol. Using SSL (HTTPS) is the most common. Applying a digital signature and using XML Encryption, as mentioned, will lock down the message so that no prying eyes can see its details, and also ensure that the message arrives untampered with.

Secure the Structure

Integrating with trading partners means exposing your internal IT assets to them – and that means identity management has to be a cornerstone of your infrastructure security. To ensure that you get the level of security you need, choose identity-enabled B2B integration software that includes comprehensive, advanced identity management capabilities.

How does identity management support security in SOA-based B2B infrastructures? By enabling you to 1) apply security policies to services and business processes and 2) control access based on policy.

With identity-enabled B2B integration, you'll be able to protect IT assets by centrally managing users' access to applications and services through identity-based authorization and authentication. You'll also be able to audit that access automatically on an ongoing basis through identity-based auditing. Even better, you'll be able to do these things without the hassle of having to roll out and maintain separate solutions and vendors for B2B integration and identity management.

Of course TCP/IP firewalls have been protecting the perimeter of your organization for quite a while by now. But many traditional TCP/IP firewalls allow XML to pass through ports without examining what is actually in the XML data elements. And plenty of nasty stuff can be included in the XML, like denial of service attacks, SQL insertion attacks, and Trojan binaries. If you choose to use XML-based protocols as a delivery or business protocol, it's a prudent (many would say required) strategy to deploy an XML firewall in the DMZ and make it the first stop every message makes on its way into your network. This will provide a higher degree of protection for your organization.

Set Up a Library of Services

You can't have an effective integration infrastructure today without reusable services. That's one of the main reasons that one-off solutions aren't viable: they require you to create a whole new set of services for every partner integration. Instead, you need reusable services to manage many of the processes associated with message flow, security, and business activity monitoring.

Reusable services are ideal for tasks like message validation, processing, tracking, and auditing. They can also be used to deliver mechanisms for secure partner-data storage, access management, and virus protection. Because reusable services are so vital to trading partner integration, be sure to choose a B2B software solution that includes functionality for message processing, security, and other reusable services.

Build It to Scale

You need to get your trading partners connected to your systems fast, so both pf you can benefit from B2B integration as quickly as possible. But the more trading partners you have, the tougher this can be. The following best practices for on-ramping trading partners on a large scale will make it easier:

- Set up a dedicated Web site where partners can go to find general information about integrating with your business, as well as project-specific information about partner agreements, business processes, technical specifications, and so forth
- Engage a third-party service for testing and certification to deliver feedback to partners on where their processes are delivering data successfully and where they're not, so they can take whatever corrective action is needed to keep the on-ramping process going smoothly
- Assemble a dedicated team of IT engineers and project managers responsible for trading partner relations during on-ramping including maintaining the project-specific information on the Web site

In B2B integration, a little planning goes a long way. Make the right choices about people and projects, be informed on issues like standards and services, and keep security and scalability top of mind – and your integration project is far more likely to succeed. ■

About the Author

John C Hardin is product manager, Sun B2B Platforms. He has spent the last 12 years designing strategy and delivering Web applications and B2B ecosystems for both large and small companies. He served as chief architect of eBusiness for General Motors, and has led an OASIS SOA Technical Committee. In his role at Sun Microsystems, his focus is to ensure that the Sun B2B Suite leads the market in ease of use, scalability, and features.

john.hardin@sun.com

Loosely Coupled...What?

For both technical and business advantages

WRITTEN BY **DAVID S. LINTHICUM**

With the advent of Web services and SOA, we've been seeking to create architectures and systems that are more loosely coupled. Loosely coupled systems provide many advantages including support for late or dynamically binding to other components while running, and can mediate the difference in the component's structure, security model, protocols, and semantics, thus abstracting volatility.

This is in contrast to compile-time or runtime binding, which requires that you bind the components at compile time or runtime (synchronous calls), respectively, and also requires that changes be designed into all components at the same time due to the dependencies. As you can imagine, this type of coupling makes testing and component changes much more difficult.

The advantages of loosely coupled architectures, as found within many SOAs, are apparent to many of us who have built architectures and systems in the past, at least from a technical perspective. However, they have business value as well.

First and foremost, a loosely coupled architecture allows you to replace components, or change components, without having to make reflective changes to other components in the architecture/systems. This means businesses can change their business systems as needed, with much more agility than if the architecture/systems were more tightly coupled. Second, developers can pick and choose the right enabling technology for the job without concerning themselves with technical dependencies, such as security models. Thus, you can build new components using J2EE, which will work and play well with other components written in Cobol or perhaps C++. Same goes for persistence layers, middleware, protocols, etc. You can mix and match to exactly meet your needs, even leverage services that may exist outside of your organization without regard to how that service was created, how it communicates, nor where it is running.

Finally, with this degree of independence, components are protected from each other and can better recover from component failure. If the SOA is designed correctly, the failure of a single component should not take down other components in the system. Thus, loose coupling creates architectures that are more resilient. Moreover, this also lends itself better to creating a failover subsystem, moving from one instance of a component to another without affecting the other components in the SOA.

It should be noted, however, that not all tight coupling is bad. Indeed, in some cases it makes sense to more tightly couple components, such as when the dependencies are critical to the design. An example would be two services that can't work apart and must function as one, and thus are better tightly coupled. You have to look at your requirement, and then determine the degree of coupling required in your architecture, and it may not always be loose coupling.

—continued on page 17

About the Author

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

david@linthicumgroup.com

Mindreef

SOAPscope server

Keep Your SOA Quality and Governance on Track With Mindreef SOAPscope Server™

Performance and Quality Matter.
A successful agile SOA requires both governance and quality - especially as services are consumed by more than one composite application - making the requirements for high quality and performance of each service even more critical.

Let Mindreef Keep Your SOA on Track.

Mindreef SOAPscope Server is a team-based platform for assuring SOA quality, with always-on tools for Web services testing, diagnostics, governance and support.

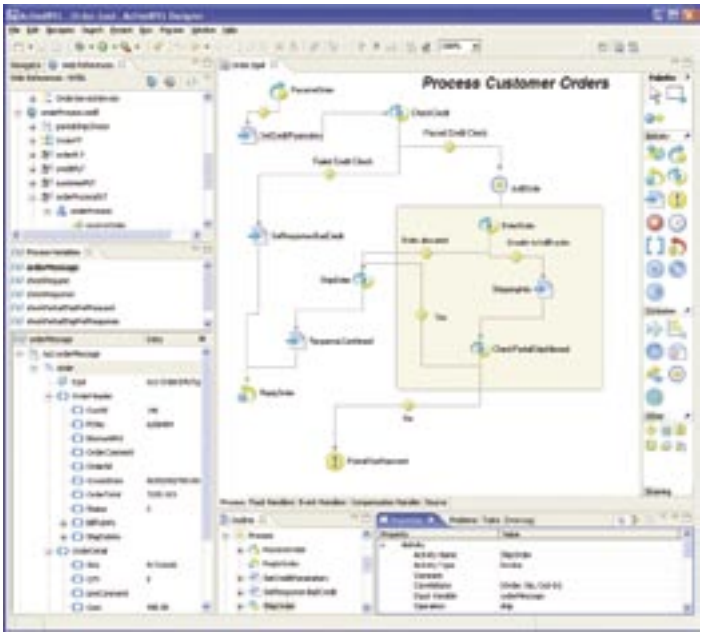
With SOAPscope Server, service teams can leverage robust features while collaborating to improve overall SOA quality, including Web services testing and QA, design-time support, prototyping, change-time support, run-time support, load testing and policy compliance validation.

Learn why more than 3,000 customers at over 1,200 organizations worldwide, including 40 of the Fortune 100, use Mindreef products to successfully build, test, and maintain high-quality Web services and SOAs.

To attend a Webinar or to request a free evaluation copy, visit www.mindreef.com or call (603) 465-2204 ext. 581

www.mindreef.com

Get Started with BPEL 2.0



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

Download BPEL tooling & server software today

active-endpoints.com/soa

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

activeBPEL

active endpoints